Creation and Deployment of Web Services and Web Services Flows

Overview

- **Web Services Basics**
  - Definitions
  - SOAP
  - WSDL
  - UDDI

- **Web Services Flows**
  - WSFL
  - XLANG
  - BPEL4WS
  - BPML

- **Examples**
  - Web Services development and deployment
  - BPEL Process developments and deployment

© 2003 Dimka Karastoyanova, TUD
Web Services - Definitions

- A Web service represents a unit of business, application, or system functionality that can be accessed over the Web.

- Web services are a new breed of Web applications. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple request to complicated business processes... Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service. [IBM]

- A Web-service is an interface that describes a collection of operations that are network accessible through standardized XML messaging [Microsoft]

- ...

The Model (Participants)

- Service Provider
  - Offers a service

- Service Requestor
  - Looks for and
  - Uses a service

- Service Broker:
  - Provides service registry
  - Allows services to be registered (by the services provider)
  - Allows service look-up (for the service requestor)
General Characteristics

- Web Services are fully XML based
- Expose interface
  - Clients access the service functionality through interfaces
  - Communication between applications as opposed to communication between users
- Self-describing modular units
- Accessible over the Web
- Interaction
  - XML based messages over standard Web protocols
  - Loose coupling
- Registered and discoverable at the Web service registry
- Platform, Language and Protocol independence
- Compositions of Web Services are possible

What is a Web Service?

Operations

Port Type

Interface

Business Functionality

Internet

Port Type

Message

Message

Operation
**The Platform**

- **Basic platform: HTTP and XML**
  - Access & Invocation
  - Description

- **The basic Web Service Platform: WSDL + SOAP + UDDI**
  - Augmenting the web with Web Service specific support
    - SOAP - XML Communication
    - WSDL - Description
    - UDDI – Registration and Discovery
  - Full functionality

- **Enhanced concepts – enterprise strength**
  - Composition (Web Service Flows, Processes)
  - Web Service Transactions
  - Security (authentication, encryption, etc.)

---

**The Basic Web Service Platform**

- **SOAP**
  - Communication protocol
- **WSDL**
  - Web Service (interface) description
- **UDDI**
  - Web Service discovery

---
SOAP

SOAP - The Protocol

- Provides:
  - a modular **packaging model** and
  - **encoding mechanism** for encoding data within modules using XML and XML namespaces

- The "package" is called message
  - SOAP messages are one-way transmissions, but can be combined in request/response pattern
  - Data to be transported is called payload

- The protocol has 3 parts
  - Envelope – contains the message components
  - Set of encoding rules
  - Conventions for representing remote procedure calls
SOAP Message

- **SOAP Envelope** defines:
  - what is in a message
  - who should process it and
  - is it optional or mandatory to process the message content

- **SOAP Header** – optional:
  - Used to extend the exchanged information without prior knowledge of the communicating parties
  - Possible extensions
    - Security - Authentication, Encryption
    - Transactions – Transaction Contexts, Transactions [IDs]
    - Other possibilities – tokens for correlation, brokered communications etc.

- **SOAP Body**:
  - Provides a mechanism for exchanging mandatory information intended for the ultimate recipient of the message
WSDL

WSDL = Web Services Description Language
- A standard language for defining Web Service interfaces
  - Therefore expect the constructs for defining interfaces
    - Interfaces (port types)
    - Methods (operations)
    - Parameters and their data types (type, messages, parts)
  - WSDL helps to generate automatically stubs/skeletons for Web Services, regardless of the platform

A WSDL interface:
- Describes a service in terms of the messages it accepts and generates
- Represents a service as a collection of ports at which certain functionality is made available
WSDL Document Structure

- **Abstract part/definition**
  - Also called "Service Interface"
  - Contains (abstract) definition of Web Service interface
  - It is protocol neutral - abstract

- **Concrete part/definition**
  - Called also "Service Implementation"
  - Binds a concrete transport protocol to the abstract interface definitions
  - Protocol mapping, access points, encodings

- Several concrete definitions may reference one and the same abstract definition

- An abstract part may be in a separate file
  - Imported into the concrete definitions document

WSDL Abstract Part

- Each Web Service has a unique interface for all users
  - A Web Service Interface has one or more Port Types
    - A PortType describes the service in terms of the (types of) operations it provides

- **Operation = Method**
  - Is an abstract description of an action (function, method) implemented by the service
  - Is a sequence of messages (parameters)
  - Defines a specific input output sequence

- A WSDL message
  - The data to be communicated
  - The single piece of information exchanged between the client and the service
  - Contains a list of message parameters – parts
  - A Part name is unique within a message

- Each WSDL Message Part is of certain Data Type
  - Type definitions are optional
  - Data Types are protocol and language independent
  - Complex types can be defined
Concrete Part

- The Binding element
  - Specified for each port type
  - Determined by the transport and communication protocol - HTTP, SMTP/POP3, SOAP etc.
  - The operations of the port type are described within the binding
  - How to serialize operations messages into e.g. HTTP+SOAP?
  - Several bindings are possible for one and the same port type

- The Service element
  - Represents a Web Service as a collection of Ports
  - Groups ports of the same Port Type
  - But referencing different Bindings

- A Port is associated with a concrete Binding
  - Port ≠ PortType
  - A Port represents the availability of a service with a specific binding at a specific end-point
    - End-point - Server URL

UDDI
**UDDI Service Registry**

- **UDDI = Universal Description, Discovery and Integration Specification**
- Defines a way to publish and discover information about businesses and their Web services

- The name UDDI is used to refer to:
  - The specification (UDDI Service Registry Specification)
  - The registry itself as a service (UDDI Registry Service)

- **UDDI Registry is logically centralized, physically distributed service with multiple nodes replicating data with each other**
  - Once a service is registered at a node, the data is automatically shared with other UDDI root nodes
  - Technical discovery, business discovery

---

**UDDI Specification**

- The UDDI specification consists of
  - An XML schema
  - A description of the UDDI APIs

- **The UDDI XML schema defines four information types**
  - White, Yellow, Green pages + tModel

- **The UDDI APIs**
  - Define a set of functions for a programmatic access to the UDDI Registry
    - Publish a Web Service
    - Discover a Web Service
WSDL – UDDI Relation

UDDI

- `<businessEntity>
  - name, contacts
  - description, identifiers, categories
  - `<BusinessService>
    - `<bindingTemplate`
    - `<bindingTemplate`

WSDL

- Service Implementation
  - `<import`
  - `<service`
  - `<port> + `<binding`
  - `<port> + `<binding`

Abstract Part (Service Interface)

- `<types`
- `<messages`
- `<portType`

Web Services and Flows

© 2003 Dimka Karastoyanova, TUD
Workflows and Web Services

A business process specifies:
- The potential execution order of operations (from a collection of applications)
- The data to be shared among tasks
- The partners’ involvement in the business process
- Joint exception handling
- How multiple parties and services participate
- Might reflect organizational structures
- Might specify the exact application programs involved in the workflow
- Enforce constraints, business rules ...
- …
BPEL4WS

Business Process Execution Language for Web Services

- BPEL4WS = Business Process Execution Language for Web Services
  - BPEL4WS is a flow language
  - Microsoft and IBM
  - A combination of WSFL and XLang, replacing them
    - WSFL – support for graph oriented processes
    - XLang – structural constructs of processes

- BPEL4WS describes business processes
  - XML based Web Service Flow definition language
  - A process is a combination of activities, involving multiple WS
  - The process itself is a Web Service, too
BPEL4WS Processes

- BPEL4WS Process – a composition of Web services
  - Execution order of activities
  - Process’s and partners’ roles within an interaction
  - Data flow
  - Control flow
  - Error Handling
  - Compensation activities

- Process interface description – Web service interface
  - A WSDL document
  - portTypes and operations the process exposes
  - References to partners’ Web services taking part in the interaction
  - Types of relationships to partners (ServiceLinkTypes)

- Executable and abstract BPEL4WS processes

BPEL4WS Processes Parts

- Any BPEL4WS Process consists of two+ parts
  - The (+) part
    - WSDL interface descriptions of all partners
    - Imported in the definitions
  - Definitions part (1)
    - Interface of the process
      - As a web service (WSDL)
    - Types of Relationships between the process and the Web Services it uses
  - Process description part (2)
    - BPEL document
    - Partners and containers
    - Definition of the process itself
Definitions (1) – Process Interface

- BPEL4WS Processes are exposed as Web Services
  - Definitions = abstract definitions
    - Partner Interfaces
      - Namespaces of partners’ services are imported
      - Location of the WSDL file locally is specified
    - Types (optional)
      - Refer to types defined by partners’ Web services
      - Define new process types (complex types)
    - Messages (optional)
      - Reference Types (default, complex or imported)
      - Used to define: Operation, Containers, Correlations ...
    - PortTypes, Operations
      - Describe the BPEL process as a Web service
      - Used to define: Partners, Service Link Types, Roles, Process activities
    - Types of relationships to partners
      - Service link types (see next)
    - ServiceLinkTypes
      - Define types of links/relationship between partners and process
- Implementation Specific
  - Concrete definitions are generated by the BPEL process engine

(2) Process Definition

- Variables / Containers
  - Store messages
  - Typed
- Partners
  - Represent the WSs advertised by partner organisations and
  - Used by the process
- Activities – units of work
  - Simple – cannot be decomposed
  - Complex – control flow, data manipulation, alternatives
- Advanced Issues
  - Failure handling
  - Transactional support
  - Compensation
Scopes (1)

- Scopes
  - Mechanism that allows for the definition of a set of activities with a common execution context

- The collection of activities a scope represents is treated like a unit of work
  - Associated with the notion of transaction
  - And referred to as long-running transactions
  - Can be undone collectively in the case of erroneous situations

Scopes (2)

- A <scope> has an associated
  - Fault handler
    - Provides a way to define fault-handling activities – defined as <catch> activities
    - BPEL4WS has defined several standard faults with their names and data
  - Compensation handler
    - A wrapper for an activity executed to compensate for another activity
    - Invoked by the <compensate> activity

- However, long-running transactions of BPEL4WS
  - Are part of a single business process
  - Reside on a single BPEL4WS engine
  - Typically long-lived units of work - locking of resources is not desirable during the whole process execution
Business Protocols

- BPEL can be used for specifying business protocols as well, apart from the executable processes
  - A business protocol specifies the potential sequence of message exchange by one particular partner with its other partners to achieve a business goal.
  - Business protocols/views are not executable - they hide internal details and complexity
  - Referred to as
    - The public interface
    - Abstract process

WS-Coordination, WS-Transaction

- BPEL4WS - a language defining business processes combining Web Services
  - Transactional behaviour enforced only for a single instance of a non-distributed business process
- WS-Coordination and WS-Transaction - complementary specifications
  - Business transactions for distributed business processes
  - WS-Coordination Framework
    - Coordinates activities of business partners
    - Coordinates the communication between business partners according to a coordination protocol
    - The goal: achieve an agreed upon result between participants in a business process
  - WS-Transaction specification
    - Defines two types of coordination protocols for Web Services
    - Atomic transactions and business transactions
WS-Coordination

- A Coordination Framework
  - Defines a Coordination Service – Coordinator
  - Coordinator is a Web Service
  - Each participant has its own coordinator
  - Allows context propagation among participants in a business process
  - Allows participants to register for certain coordination protocols
  - Coordination protocols are plugged into the framework

Further Issues

- **WSDL:**
  - Describes Web Services/operations that take place in a BPEL4WS process
  - and is used to describe the Web Services business process provides

- **WS-Transaction:**
  - Specifies coordination protocols for distributed atomic transactions involving Web Services described by WSDL
  - The agreement protocol between a scope and its nested scopes in BPEL4WS for determining the outcome of long-running transaction
Examples

- Creation of a Web Service
- Creation of a Web Service Flow

How to Develop, Deploy and Run a Web Service?

1. Design the Service Interface
   - 1.1 Register
     - 1.1.1 Register Business Entity
     - 1.1.2 Register tModel
     - 1.1.3 Register the service using the WSDL file from
   - 1.2 Program the Web Service Functionality
     - 1.3 Write additional files e.g. deployment descriptor etc.
   - 1.4 Deploy
   - 1.5 Generate and publish the WSDL file

2. Program the Web Service
   - 2.1 Register
     - 2.1.1 Register Business Entity
     - 2.1.2 Register tModel
     - 2.1.3 Register the service using the WSDL file from
   - 2.2 Test – perform a dummy search

3. Write additional files e.g. deployment descriptor etc.
   - 3.1 Search the UDDI Registry
     - 3.1.1 Retrieve the WSDL file for the service
   - 3.2 Generate the stub
   - 3.3 Program the client
   - 3.4 Run
BPEL4WS Process Implementation

How to Develop, Deploy and Run a BPEL4WS Process?

1. Create WSDL description of the process
2. Process description
3. Deploy the process
4. Create Client
   - Run
Literature and References

- **Web Services Essentials**

- **Simple Object Access Protocol (SOAP)**
  - [http://www.w3.org/TR/2002/WD-soap12-part0-20020626/](http://www.w3.org/TR/2002/WD-soap12-part0-20020626/)

- **Programming Web Services with SOAP**

- **Web Services Description Language - specification**
  - [http://www.w3.org/TR/wsdl12/](http://www.w3.org/TR/wsdl12/)

- **Web Services Description Language (WSDL)**
  - Erik Christensen and Greg Meredith, Microsoft, Francisco Curbera and Sanjiva Weerawarana, IBM Research
Literature (2)

- Understanding Web Services
- Understanding WSDL in a UDDI registry
  - Peter Brittenham, Francisco Cubera, Dave Ehnevuske, Steve Graham, IBM Developer Works, September
- UDDI Executive White Paper
- UDDI Technical White Paper

Literature (3)

- BPEL4WS Specification
- Business Process with BPEL4WS
  - Sanjiva Weerawarana, Francisco Curbera
- Business processes in a Web services world
  - Frank Leymann, Dieter Roller
- Business Process Execution Language for Java Run Time
- Reference guide for creating BPEL4WS documents
- ...