On Specifying Web Services Using UDDI Improvements

Sven Overhage, Peter Thomas
Dept. of Application Engineering and Business Information Systems
{overhage, thomas}@bwl.tu-darmstadt.de

Agenda

Introduction

Requirements for Specification Frameworks

UDDI: State and Challenges

WS-Specification: Improved Web Service Specification

Introduction – Web Services...

- ❑ are interoperable pieces of application based on a syntactic wiring standard (XML, SOAP)
- ❑ offer services using Web based interfaces (WSDL)
- ❑ are invoked using remote procedure calls (physical reuse)
- ❑ hide their implementation (black-box)
- ❑ are catalogued using standardized registries (UDDI)

→ Web services are software components

Critical Success Factors

- ❑ Find and choose suitable (“the right”) Web Services
- ❑ Correctly configure Web services to applications

→ Requires both conceptual and technical information:

  - ❑ Conditions of acquisition (e.g. price, service quality)
  - ❑ Technological heterogeneity (e.g. technology or platform)
  - ❑ Syntactic heterogeneity (e.g. data formats)
  - ❑ Semantic heterogeneity (e.g. implemented concepts)
  - ❑ Pragmatic heterogeneity (e.g. implemented processes)

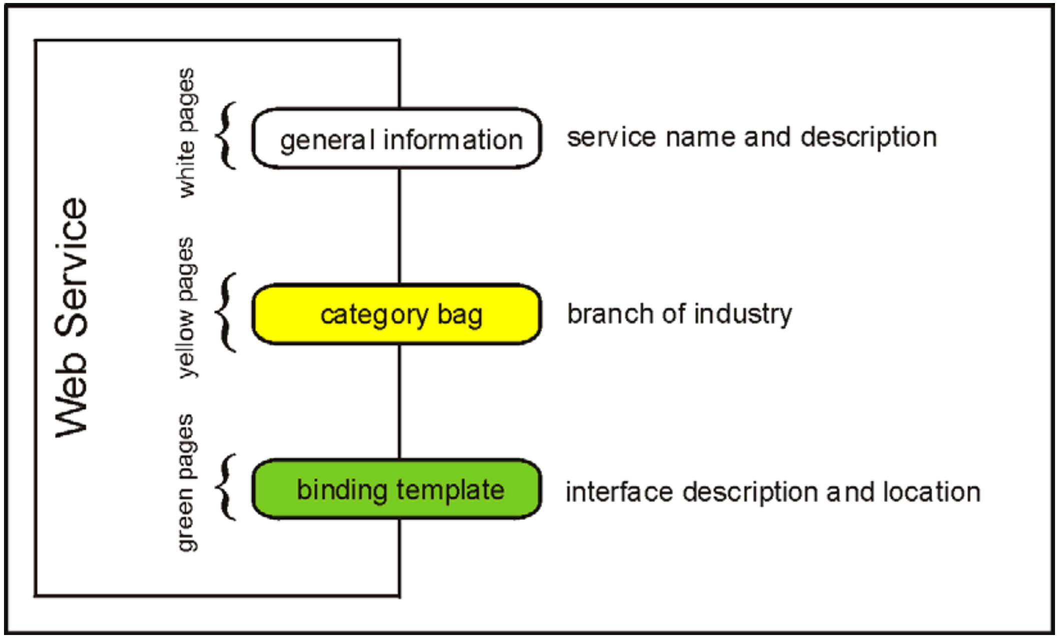
General Requirements for Specification Frameworks

- ❑ Contain human- and machine-readable specifications
- ❑ Determine what (extent) and how (notations) to specify
- ❑ Provide ordered perspectives (separation of concern)
- ❑ Satisfactorily specify the external view (“provide all the information necessary”)

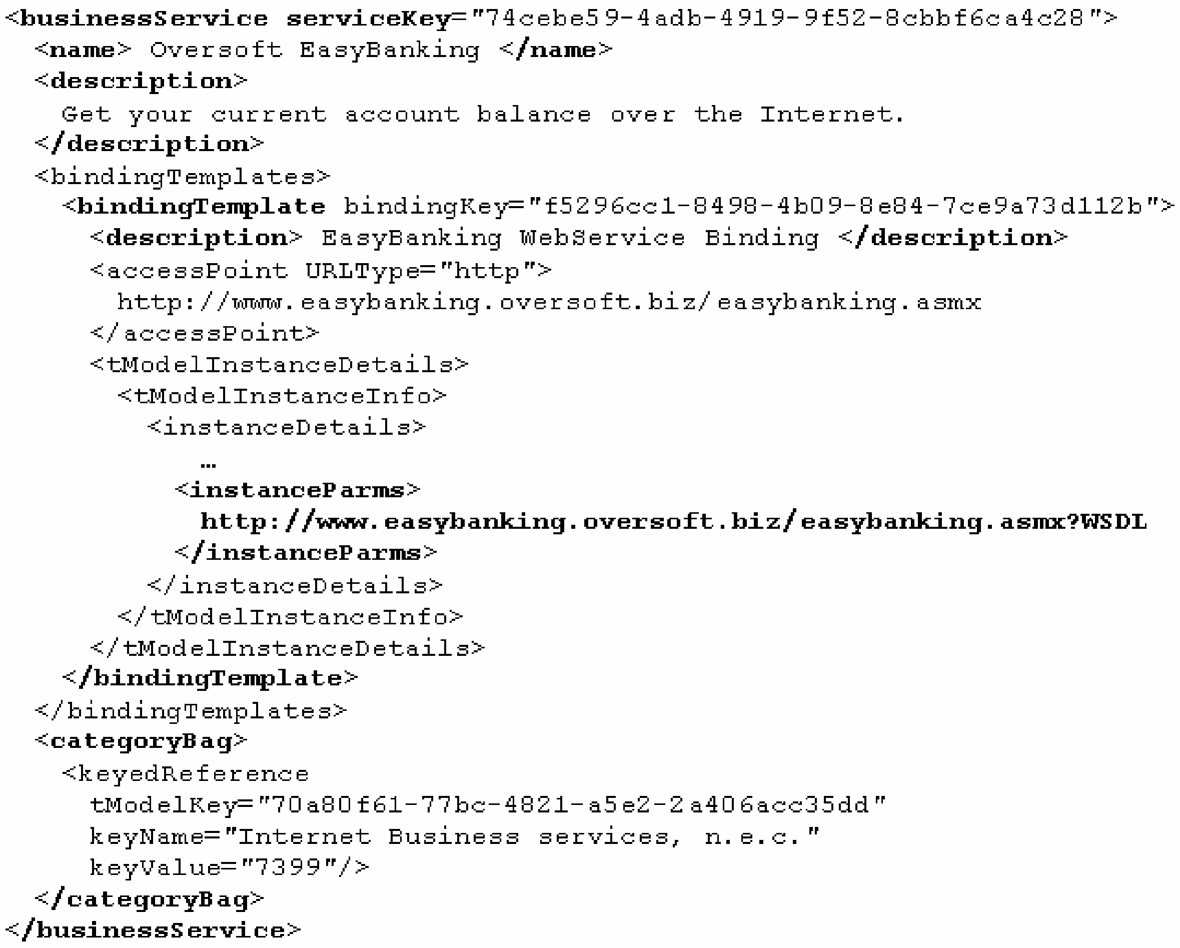
UDDI: State and Challenges

- Mainly provides information referring to businesses rather than Web services
  - White pages: general information (business name, contacts)
  - Yellow pages: classifications (branch of industry, location)
  - Green pages: information on provided services
- Information referring to Web services is rather sparse
  - Few general specifications and classifications
  - Location of the Web service interface description (WSDL)

UDDI Web Service Specification



UDDI Sample



```
<businessService serviceKey="74cebe59-4adb-4919-9f52-8cbbf6ca4c28">
  <name> Oversoft EasyBanking </name>
  <description>
    Get your current account balance over the Internet.
  </description>
  <bindingTemplates>
    <bindingTemplate bindingKey="f5296cc1-8498-4b09-8e84-7ce9a73d112b">
      <description> EasyBanking WebService Binding </description>
      <accessPoint URLType="http">
        http://www.easybanking.oversoft.biz/easybanking.asmx
      </accessPoint>
      <tModelInstanceDetails>
        <tModelInstanceInfo>
          <instanceDetails>
              …
            <instanceParms>
              http://www.easybanking.oversoft.biz/easybanking.asmx?WSDL
            </instanceParms>
          </instanceDetails>
        </tModelInstanceInfo>
      </tModelInstanceDetails>
    </bindingTemplate>
  </bindingTemplates>
  <categoryBag>
    <keyedReference
      tModelKey="70a80f61-77bc-4821-a5e2-2a406acc35dd"
      keyName="Internet Business services, n.e.c."
      keyValue="7399"/>
  </categoryBag>
</businessService>
```

WS-Specification: Targets

- ❑ Provide a more sophisticated specification of Web Services
- ❑ Consider the pre-mentioned general requirements

→ Technical and conceptual specifications

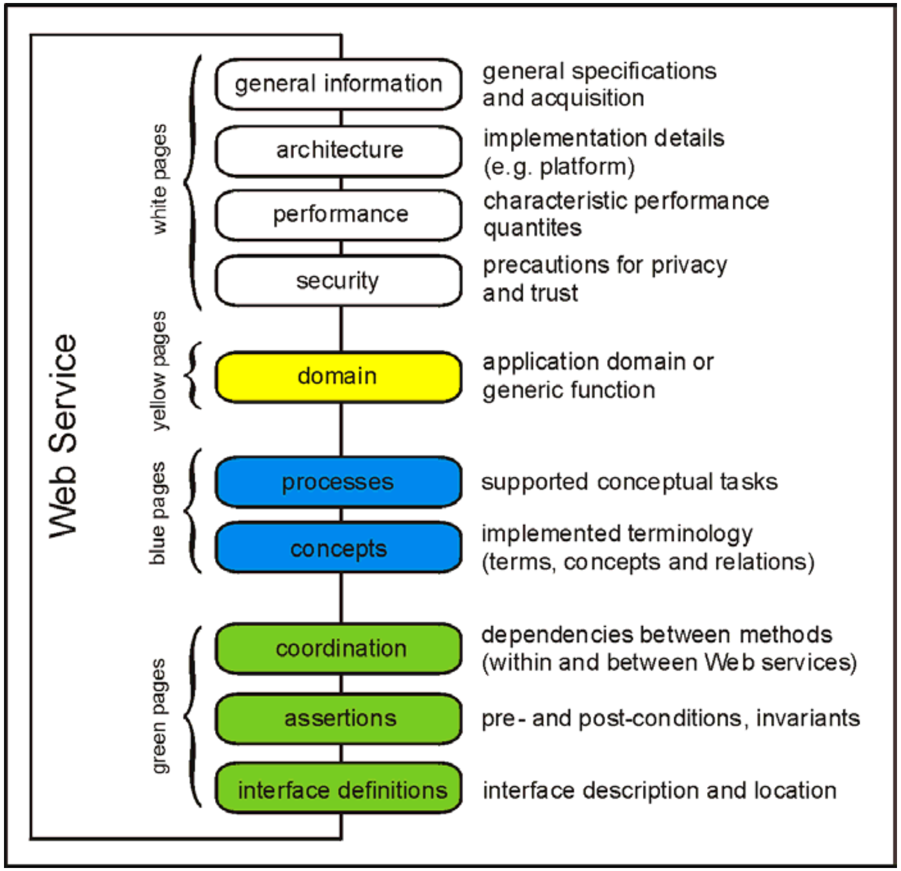
  - ❑ Technological, syntactic, semantic, and pragmatic perspectives

- ❑ Maintain backward-compatibility to UDDI

→ Augment the UDDI data model by:

  - ❑ Adding new specifications to existing perspectives
  - ❑ Introducing new perspectives

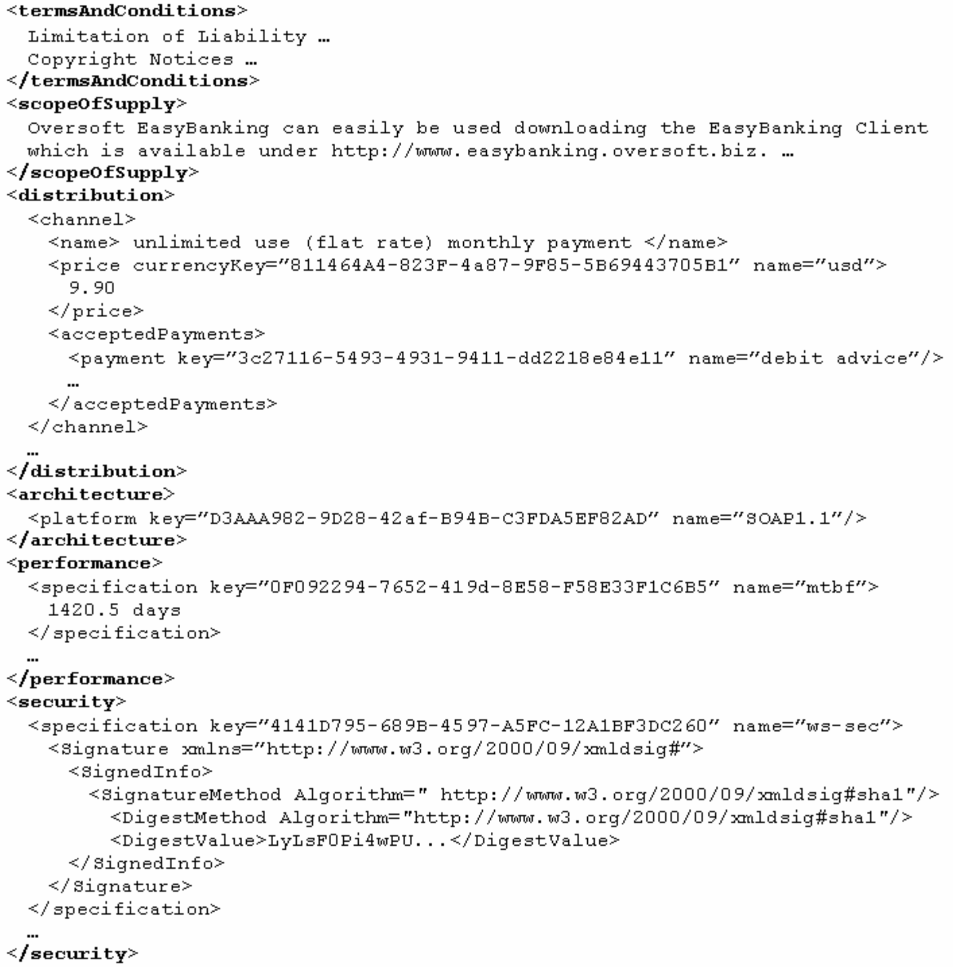
WS-Specification: Framework


Web Service
white pages
yellow pages
blue pages
green pages
general information
architecture
performance
security
domain
processes
concepts
coordination
assertions
interface definitions
general specifications and acquisition
implementation details (e.g. platform)
characteristic performance quantites
precautions for privacy and trust
application domain or generic function
supported conceptual tasks
implemented terminology (terms, concepts and relations)
dependencies between methods (within and between Web services)
pre- and post-conditions, invariants
interface description and location

White and Yellow Pages

- **Acquisition:** license agreement, scope of supply, distribution channels, prices, and payment
- **Platform:** used version of the Web Service Architecture, dependencies to other Web services (e.g. MS Passport)
- **Performance:** quality of service, data throughput
- **Security:** message integrity, confidentiality, and authentication (denoted using WS-Security)
- **Classifications:** application domain taxonomies

White Pages: Sample



```
<termsAndConditions>
  Limitation of Liability …
  Copyright Notices …
</termsAndConditions>
<scopeOfSupply>
  Oversoft EasyBanking can easily be used downloading the EasyBanking Client
  which is available under http://www.easybanking.oversoft.biz. …
</scopeOfSupply>
<distribution>
  <channel>
    <name> unlimited use (flat rate) monthly payment </name>
    <price currencyKey=”811464A4-823F-4a87-9F85-5B69443705B1” name=”usd”>
      9.90
    </price>
    <acceptedPayments>
      <payment key=”3c27116-5493-4931-9411-dd2218e84e11” name=”debit advice”/>
      …
    </acceptedPayments>
  </channel>
  …
</distribution>
<architecture>
  <platform key=”D3AAA982-9D28-42af-B94B-C3FDA5EF82AD” name=”SOAP1.1”/>
</architecture>
<performance>
  <specification key=”0F092294-7652-419d-8E58-F58E33F1C6B5” name=”mtbf”>
    1420.5 days
  </specification>
  …
</performance>
<security>
  <specification key=”4141D795-689B-4597-A5FC-12A1BF3DC260” name=”ws-sec”>
    <Signature xmlns=”http://www.w3.org/2000/09/xmldsig#”>
      <SignedInfo>
        <SignatureMethod Algorithm=" http://www.w3.org/2000/09/xmldsig#sha1"/>
          <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <DigestValue>LyLsF0Pi4wPU...</DigestValue>
      </SignedInfo>
    </Signature>
  </specification>
  …
</security>
```

Blue Pages: Conceptual Information

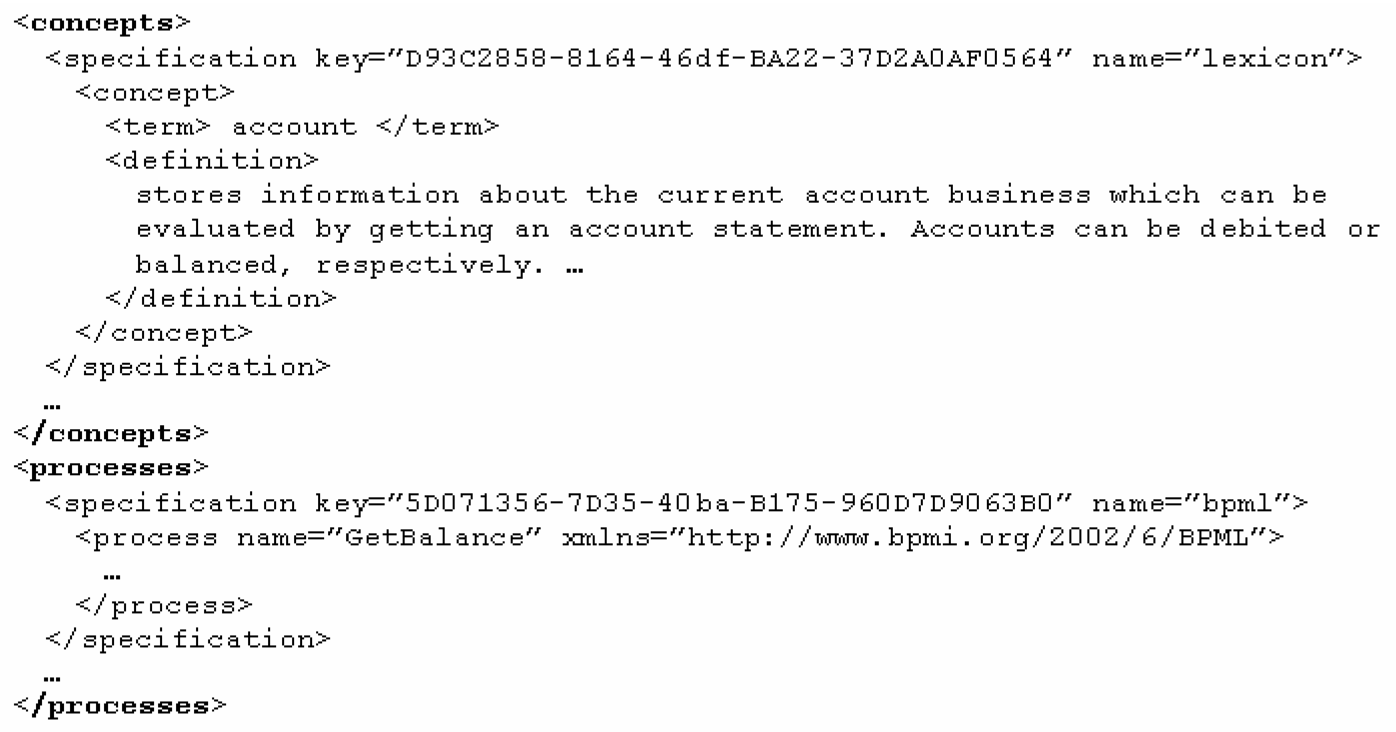
- ❑ **Conceptual semantics:** implemented concepts with definitions (denoted using ontologies, or normative language)

  Sample: “Price” includes value added tax (VAT).

- ❑ **Conceptual pragmatics:** implemented (business) processes (denoted using workflow languages, e.g. BPML)

  Sample: “Buying” consists of contract negotiation, contractual conclusion, good delivery, and compensation (payment).

Blue Pages: Sample



```
<concepts>
  <specification key="D93C2858-8164-46df-BA22-37D2A0AF0564" name="lexicon">
    <concept>
      <term> account </term>
      <definition>
        stores information about the current account business which can be
        evaluated by getting an account statement. Accounts can be debited or
        balanced, respectively. …
      </definition>
    </concept>
  </specification>
  …
</concepts>
<processes>
  <specification key="5D071356-7D35-40ba-B175-960D7D9063B0" name="bpml">
    <process name="GetBalance" xmlns="http://www.bpmi.org/2002/6/BPML">
      …
    </process>
  </specification>
  …
</processes>
```

Green Pages: Technical Information

- ❑ **Syntax:** Interface Definitions (denoted using WSDL)

  methods, datatypes, exceptions (faults), properties

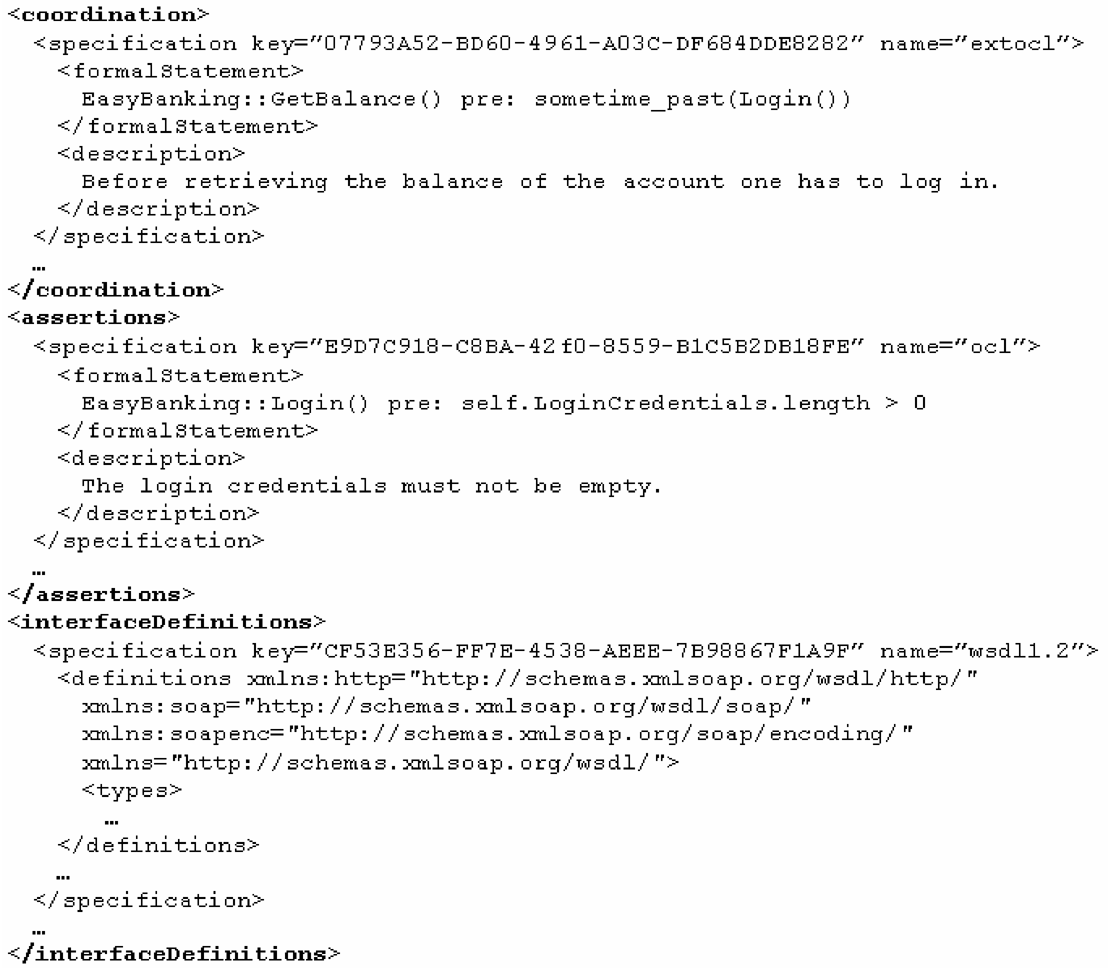
- ❑ **Semantics:** Assertions (denoted using UML OCL)

  pre- and postconditions, invariants

- ❑ **Pragmatics:** Method coordination (denoted using WSFL, WS-Coordination, or temporalOCL)

  order constraints, concurrency constraints

Green Pages: Sample



```
<coordination>
  <specification key=”07793A52-BD60-4961-A03C-DF684DDE8282” name=”extocl”>
    <formalStatement>
      EasyBanking::GetBalance() pre: sometime_past(Login())
    </formalStatement>
    <description>
      Before retrieving the balance of the account one has to log in.
    </description>
  </specification>
  …
</coordination>
<assertions>
  <specification key=”E9D7C918-C8BA-42f0-8559-B1C5B2DB18FE” name=”ocl”>
    <formalStatement>
      EasyBanking::Login() pre: self.LoginCredentials.length > 0
    </formalStatement>
    <description>
      The login credentials must not be empty.
    </description>
  </specification>
  …
</assertions>
<interfaceDefinitions>
  <specification key=”CF53E356-FF7E-4538-AEEE-7B98867F1A9F” name=”wsdl1.2”>
    <definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns="http://schemas.xmlsoap.org/wsdl/">
      <types>
        …
    </definitions>
    …
  </specification>
  …
</interfaceDefinitions>
```

Thank You…

… for your attention. Questions?

- ❑ Today
- ❑ Later: overhage@bwl.tu-darmstadt.de