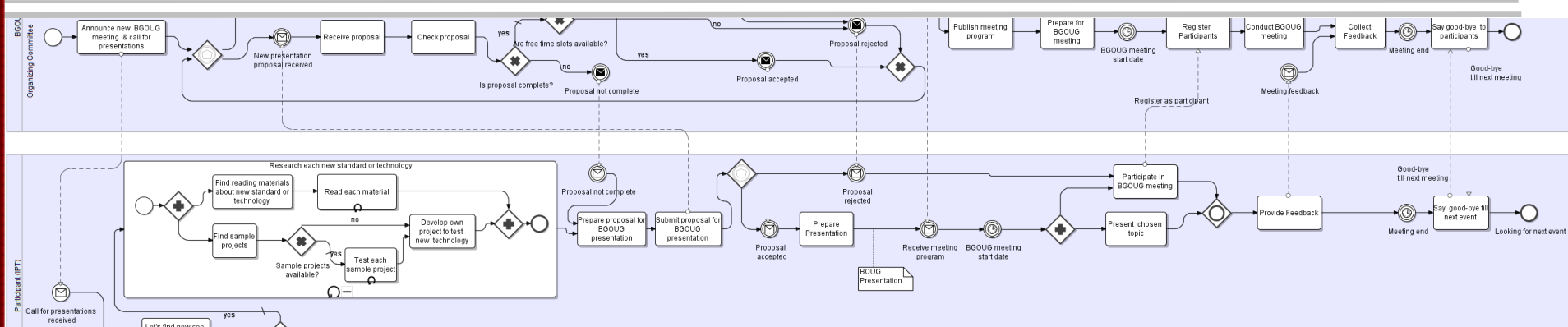


Java API for XML Web Services (JAX-WS). AXIS 2: JAXB, ADB, XMLBeans, JiBX



Траян Илиев

IPT – Intellectual Products & Technologies
e-mail: tiliev@iproduct.org
web: <http://www.iproduct.org>

Oracle®, Java™ and EJB™ are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Oracle®, Java™ и EJB™ са търговски марки на Oracle и/или неговите подразделения. Всички други търговски марки са собственост на техните притежатели.

Съдържание

1. Сравнение между JAXB, ADB, XMLBeans и JiBX
2. WS-* стандарти
3. Сравнение между различни платформи за реализация на уеб услуги Metro, CXF, Axis2 и JBoss
4. Java API for XML Web Services (JAX-WS)
5. AXIS Web Clients from WSDL
6. AXIS 2 Web Clients from WSDL

Java Data Binding технологии

- Позволяват свързване на **XML данни** със свойства на **Java обекти (Data Binding)** и дават възможност да се автоматизира процесът на запазване/извличане на данните от/в XML формат (**marshalling/unmarshalling**)
- Използвани технологии (AXIS 2):
 - Java Architecture for XML Binding (JAXB)
 - Axis2 Databinding Framework (ADB)
 - XMLBeans
 - JiBX

Java Architecture for XML Binding (JAXB)

- Java Architecture for XML Binding (JAXB) е стандартна технология в Java SE/EE 5 и нагоре, при която съответствието на XML структури и Java обекти се описва декларативно – с помощта на **анотации** върху **POJO**
- Позволява автоматично генериране на класове от/към **XML Schema** описание
- Използва се като базова инфраструктура за **Java API for XML Web Services (JAX-WS)** заедно със **StAX**
- Лесна за използване и платформено независима
- Позволява генериране на различни от XML сериализации – например **JavaScript Object Notation (JSON)**

Axis2 Databinding Framework (ADB)

- **ADB** е сравнително прост и лесен за използване метод за достъп до данните в съобщенията при генериране на **Axis2** клиенти
- В повечето случаи класовете които съответстват на WSDL типовете съобщения и са реализирани като вътрешни класове на един основен Stub клас
- Има сравнително добра производителност (performance) и генерира малко на брой класове
- Има ограничения – например при използване със XML Schema описания от тип **extension** и **restriction**
- Платформено специфична е за **AXIS 2** веб услугите и клиентите за тях – използва **Axis Object Model (AxiOM)**

XMLBeans (1)

- XMLBeans – по сложна за използване технология, която обаче предоставя възможност за обектно представяне и навигиране на произволни по сложност XML Schema структури
- За разлика от ADB, XMLBeans е пълнофункционален XML Schema компилатор – поддържа wild cards, substitution groups и complex type restriction
- Поддържа пълния XML InfoSet – включително document order, mixed content, white space, comments и processing instructions
- Генерира голямо количество файлове/класове, които съответстват на XML Schema типовете и това може да бъде объркващо

XMLBeans (2)

- Основна цел на **XMLBeans** е да предостави универсален начин за не-поточна (in memory) XML обработка, която може да се ползва за абсолютно всички възможни схеми (универсалност) и позволява достъп до данните на произволно ниво без нужда от други инструменти
- Основни интерфейси:
 - **XmlObject** – базов за всички генерирани от XML Schema Java класове, предлага строго типизирани get и set методи за под-елементите дефинирани в схемата – сложните типове в наследници на XmlObject, простите схема типове водят до get/set методи от примитивни Java типове и String

XMLBeans (3)

- **XmlCursor** – можем да го получим от произволен XmlObject, осигурява ефикасен начин за достъп до XML Infoset на ниско ниво, представя позицията в XML инстанцията
- **SchemaType** – XMLBeans предоставя пълен XML Schema обектен модел, който може да бъде използван за отразяване на подлежащата схема информация – например от него можем да генерираме инстанция на схемата или да открием валидните стойности за даден enumeration

XMLBeans

- **Simple API for XML (SAX)** – събитийно ориентиран интерфейс, при който документът се чете последователно и неговото съдържание се рапортува като извиквания на отделни **callback** методи на обработващия обект (**handler**)
- **Document Object Model (DOM)** – зарежда и манипулира в паметта целия документ под формата на дърво
- **Pull Parsing: Streaming API for XML (StAX)** – третира документа като серия от възли, които се обхождат последователно с помощта на шаблона **Iterator**
- **Свързване на данни (Data Binding): Java Architecture for XML Binding (JAXB)** – дава възможност да се автоматизира процесът на запазване/извличане на данните от/в XML формат (marshalling/unmarshalling)

JiBX (1)

- Според някои сравнения е най-бързият от data binding подходите
- При него **POJO** клас файловете след компилация се разширяват на ниво байт-код (**bytecode enhancement**) с възможности за връзка към **JiBX binding runtime**, който стъпва върху **StAX / XMLPull API**
- Използват се отделни мапинг файлове, които дефинират съответствието между XML елементи свойства на Java класовете
- Уникалното в случая е че генерирането на допълнителни класове се поддържа, но не е задължително

JiBX (2)

- Инструмент за генериране на binding описания от XML Schema:

```
java -cp ../lib/jibx-tools.jar org.jibx.schema.codegen.CodeGen -t ../src StockQuoteService.xsd
```

- Допълва инструмента **WSDL2Java** на Apache AXIS2:

Сървър: WSDL2Java -uri StockQuoteService.wsdl -s -ss -uw -sd -ssi -ap -ns2p

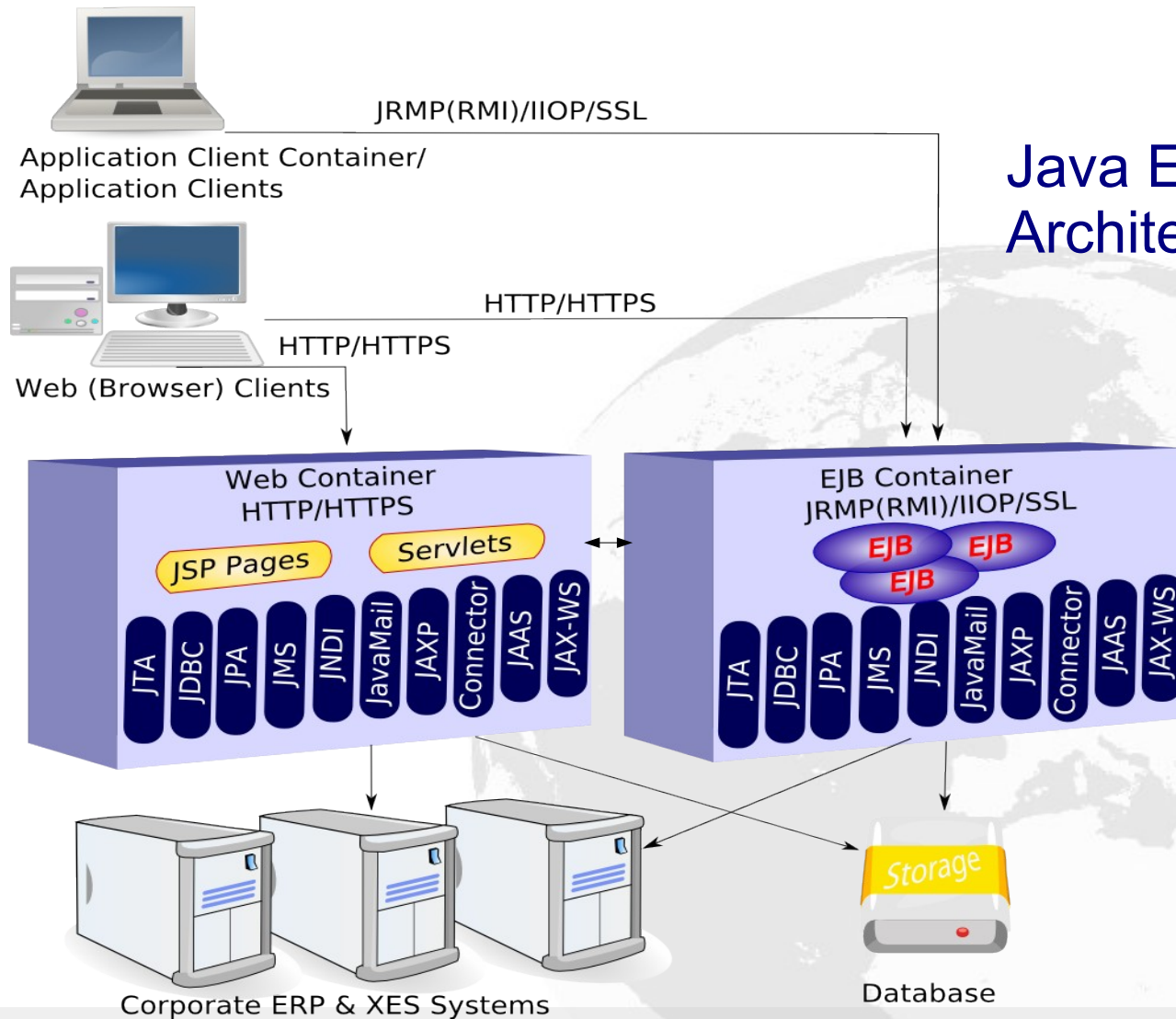
```
http://quickstart.samples/xsd=samples.quickstart.service.jibx.xsd -l java  
-p samples.quickstart.service.jibx -d jibx -Ebindingfile binding.xml -o ..
```

Клиент: WSDL2Java -s -uw -ns2p

```
http://quickstart.samples/xsd=samples.quickstart.service.jibx.xsd -l java  
-p samples.quickstart.service.jibx -d jibx -Ebindingfile binding.xml -o ..
```

Пример за JiBX binding.xml

```
<binding name="binding" package="samples.quickstart.xsd" trim-  
whitespace="true">  
  <namespace uri="http://quickstart.samples.xsd" default="elements"/>  
  <mapping class="samples.quickstart.xsd.GetPrice" name="getPrice">  
    <value style="element" name="symbol" get-method="getSymbol" set-  
method="setSymbol" usage="optional"/>  
  </mapping>  
  <mapping class="samples.quickstart.xsd.GetPriceResponse" name="getPriceResponse">  
    <value style="element" name="return" get-method="getReturn" set-  
method="setReturn" usage="optional"/>  
  </mapping>  
  <mapping class="samples.quickstart.xsd.Update" name="update">  
    <value style="element" name="symbol" get-method="getSymbol" set-  
method="setSymbol" usage="optional"/>  
    <value style="element" name="price" get-method="getPrice" set-  
method="setPrice" usage="optional"/>  
  </mapping>  
</binding>
```



Java EE 6 / Java SE Standards APIs

Съгласно Java EE спецификацията:

- Web Services
 - ~~Java API for XML-based RPC (JAX-RPC)~~
 - Java API for XML Web Services (JAX-WS)
 - Java Architecture for XML Binding (JAXB)
 - SOAP with Attachments API for Java (SAAJ)
 - Java API for XML Registries (JAXR)
- RESTful Web Services
 - Jersey – RESTful Web Services - JAX-RS

Java API for XML Web Services (JAX-WS)

- Пакети: javax.jws, javax.jws.soap
- Основни анотации:
 - **@WebService** - маркира Java клас като имплементиращ веб услуга или Java интерфейс като дефиниращ интерфейс на веб услуга
 - **@WebMethod** – маркира методи които ще бъдат публикувани
 - **@WebParam** – мапинг на параметрите
 - **@WebResult** – мапинг на резултата
 - **@Oneway** – веб метод, който не връща резултат
 - **@SOAPBinding** – специфицира мапинг между Web Service --> SOAP message protocol

Tools

- Admin Console
- asadmin
- asant
- appclient
- capture-schema
- package-appclient
- JavaDB
- verifier
- xjc
- schemagen
- **wsimport**
- **wsgen**

Спецификации за качество на уеб услугите (QoS): WS-* (1)

- Infrastructure & Configuration Services
 - Web Services Policy
 - Web Services Meta-data Exchange
- Binary and Transport Optimization of Messages
 - SOAP MTOM & XOP
 - Web Services Addressing
- Reliable Messaging & Transactions
 - Web Services Reliable Messaging
 - Web Services Coordination
 - Web Services Atomic Transactions

Спецификации за качество на уеб услугите (QoS): WS-*

- Web Services Security & Trust
 - Web Services Security
 - Web Services Security Policy (extends Web Services Policy)
 - Web Services Trust
 - Web Services Secure Conversation

Платформи за реализация на уеб услуги с отворен код (WS Implementation Stacks)

- Glassfish Metro – <http://metro.java.net/>
- Apache CXF – <http://cxf.apache.org/>
- Apache Axis – <https://axis.apache.org/axis/>
- Apache Axis 2 –
<http://axis.apache.org/axis2/java/core/>
- JBoss WS – <http://www.jboss.org/jbossws>

Glassfish Metro и Jersey

- Java™ EE 6 Reference Implementation – поддържа дефинираните от Java™ EE стандарта JAX-WS (SOAP + WSDL) и JAX-RS (REST) APIs
- Поддържа още:
 - WS-AtomicTransaction v1.0,
 - WS-MetadataExchange v1.1,
 - WS-Policy v1.5,
 - WS-PolicyAttachment v1.5,
 - WS-ReliableMessaging v1.2,
 - WS-ReliableMessaging Policy v1.2,
 - WS-MakeConnection v1.1,
 - WS-Coordination v1.0,
 - WS-Security v1.1,
 - WS-SecurityPolicy v1.2,
 - WS-Trust v1.4,
 - WS-SecureConversation v1.4,
 - OASIS Security Profiles

Apache CXF (Celtix + XFire)

- Поддържа дефинираните от Java™ EE стандарта JAX-WS (SOAP + WSDL) и JAX-RS (REST), JavaScript и Simple APIs
- Поддържа още:
- WS-Addressing & WS-Policy
- WS-ReliableMessaging (1.0 complete, 1.1 ongoing)
- WS-Security, WS-SecurityPolicy, WS-SecureConversation, WS-Trust (client-side)
- Транспортни модули – HTTP-Servlet, HTTP-Jetty, HTTP-OSGI, JBI integration, JMS, Local (In-VM)
- Все още не са имплементирани – WSDL 2.0, WS-Atomic Transactions, WS-Coordination, S-MetaDataExchange, WS-Eventing, WS-BusinessActivity, WS-Transfer

Apache Axis 2

- Поддържа Java™ EE стандарта **JAX-WS** (SOAP + WSDL) и собствено REST API, WSDL 2.0, добра интеграция с Spring
- Поддържа като модули:
 - **WS-Addressing** (2005-08 version), **WS Policy**
 - **Sandesha2**: WS-Reliable Messaging (v 1.0, February 2005)
 - **Rampart**: WS-Security & WS-SecureConversation, WS-SecurityPolicy
 - **Kandula**: WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity
 - **Mex Module**: WS-MetadataExchange version 1.1
 - **Transfer Module**: WS-Transfer (September 2004)
 - **Apache Muse**: WS-Notifications

JBoss WS

- JBoss официално поддържа поне три стека за уеб услуги: JBossWS – Native, JBossWS – CXF, JBossWS - Metro
- За всеки от тях JBoss осигурява възможности като **endpoint metrics, record management, endpoint address rewrite** и др.
- Поддържа:
 - JAX-RPC and JAX-WS (2.2)
 - EJB 2.1, EJB3 and JSE endpoints
 - Attachments Profile Version 1.0
 - Support for MTOM/XOP and SwA-Ref
 - WS-Security 1.0 for XML Encryption/Signature of SOAP message
 - WS-Addressing (W3C candidate release) and JSR-261
 - WS-ReliableMessaging, WS-Eventing, WS-Policy

Практически демонстрации и упражнения

Реализация на уеб услуги и клиенти
за Glassfish Metro, Apache AXIS и Apache AXIS 2
в Eclipse IDE + допълнителни плъгини

...

Референции

- Java EE 5 Tutorial – <http://java.sun.com/javaee/5/docs/tutorial/doc/>
- Java EE 6 Tutorial – <http://java.sun.com/javaee/5/docs/tutorial/doc/>
- Apache CXF – <http://cxf.apache.org/>
- Apache Axis – <https://axis.apache.org/axis/>
- Apache Axis 2 – <http://axis.apache.org/axis2/java/core/>
- JiBX – <http://jibx.sourceforge.net/>

Благодаря Ви за Вниманието!
Въпроси?