THILO FROTSCHER
Software Architect & Trainer

Enterprise Java, Web Services and XML

Best practices for interoperable web service applications

Java User Group Wellington

http://www.frotscher.com thilo@frotscher.com

THILO FROTSCHER
Software Architect & Trainer

Introduction

- Freelancing software architect and trainer
- Technologies
  - Enterprise Java
  - Web Services and XML
  - Interoperability
  - Patterns and software architectures
- Numerous articles for computer magazines
- Book: „Java Web Services mit Apache Axis“
- Specialist expert for XML-/Web Service books
- Always interested in new, exciting projects ☺

THILO FROTSCHER
Software Architect&Trainer

Agenda

- Reasons for interoperability issues
- Code First vs. Contract First
- Contract First in detail
- Tool support
- General interoperability tips
- Summary

- Optional: Live-Demo

THILO FROTSCHER
Software Architect&Trainer

Introduction

- Web services and SOA have made the step from hype to reality – really!
- Development of interoperable web services apps still seems to be a big challenge
  - Type conflicts, unsupported message formats, poor specifications,...
- WS-I and the Basic Profile...
  - Guideline for using Web service technology
  - Supported by most current tools and frameworks (sometimes incomplete)
  - Analysis tools for web service artifacts
  - Includes hints for developers

Reasons for the difficulties...

- How are web most web service apps developed?
  1. Write implementation code for a web service or use existing code
  2. Create WSDL document from this code (somehow)
  3. Generate client proxies from generated WSDL
- Frequently demonstrated approach (and also the most documented one)
- Step 2 causes many interoperability problems
- "Code First" or "Implementation First"
  - Interface derived from implementation

What‘s the trouble with this approach?

- Code First is simple and (sometimes) works great... but holds problems in heterogeneous systems
  - tempts to use platform-specific types in web services, like `Vector` or `DataSet`
  - Problem: there‘s no defined mapping between those types and XML data types
- XML Schema defines a set of data types – each platform/language has its own set of types
- „Specifications“ for web service runtimes define mappings between XML schema types and platform-specific types

Golden Interoperability Rule

XML is the least common denominator!

When starting with XML Schema data types,
it's simple to map these to platform-specific types
because according mappings are already defined.

However, when starting with platform-specific types
it's possible that there's no mapping to XML Schema.

Sandpit projects...

- All green field development
- No predefined schemas
- "Implementation First" approach works great:
  - Just start coding...
  - Create WSDLs and schemas "on the fly"
  - Tools support this approach
    (and especially older ones even work best this way)
- But things change when starting with real projects
  - It's always harder to learn the more difficult way last
  - „But the easy way worked well for the test project..."

THILO FROTSCHER
Software Architect&Trainer

...and real-world projects

- In big integration projects there are typically numerous, pre-existing schemas
  - e.g. insurances: Origo and Accord standards
  - e.g. tourism: OTA (Open Travel Alliance)
- WSDL must defined services based on the pre-defined types from these schemas
- Involved development teams can then code against this "contract" simultaneously (Java, .NET, Perl, ...)
- Vice Versa?
  - Generate WSDL automatically from Java, C#, etc
  - Hope that generated WSDL fits pre-existing schema...

THILO FROTSCHER
Software Architect&Trainer

"WSDL-First" / "Contract-First"

- Use solely primitive data types that are defined by the XML Schema specification
  - `int`, `long`, `float`, `double`, `string`, `boolean`, `dateTime`,...
- Create application-specific schema defining complex types and/or arrays based on primitive types
- Use WSDL to create an *interface contract* based on the complex types defined in your schema
- Generate service skeletons and client proxies from your interface contract
- What you get: heavily reduced chances for interoperability issues caused by incompatible data types

THILO FROTSCHER
Software Architect&Trainer

Thinking in interfaces

- Starting with WSDL means starting with the definition of the service interface
- A change in the way many developers are thinking about web services is necessary anyway
  - Don‘t think in objects and methods... (RPC message format encouraged this way of looking at services)
  - Use the terms „message" and „service" instead
- Message exchange pattern is defined in „contract“
- Loosely-coupled apps, better interoperability, Service-Oriented Architectures (SOA)

THILO FROTSCHER
Software Architect&Trainer

Interoperability = Contract First!

- What‘s new about that? Nothing!
  - see CORBA, DCE, COM...
  - We already learned that Contract First improves interop
- So why did we suddenly start using a Code First approach?
- Problem: available tools and documentation focused mostly on *Code First*
- Interesting fact: discussion about Contract First development of web services initially started in the Microsoft world ☺

THILO FROTSCHER
Software Architect&Trainer

Interoperability = WS-I Basic Profile

- Current tools usually generate WSDL according to the rules of the Basic Profile
  - However, don‘t rely on that!!!
- Developers should make use of the WS-I Analyzer tool when designing their service interfaces
- Important Rule: Use `Literal`, not `Encoded`
  - Implies XML Schema for specifying the data types
  - Unfortunately `Encoded` is still the default setting of some tools and frameworks...

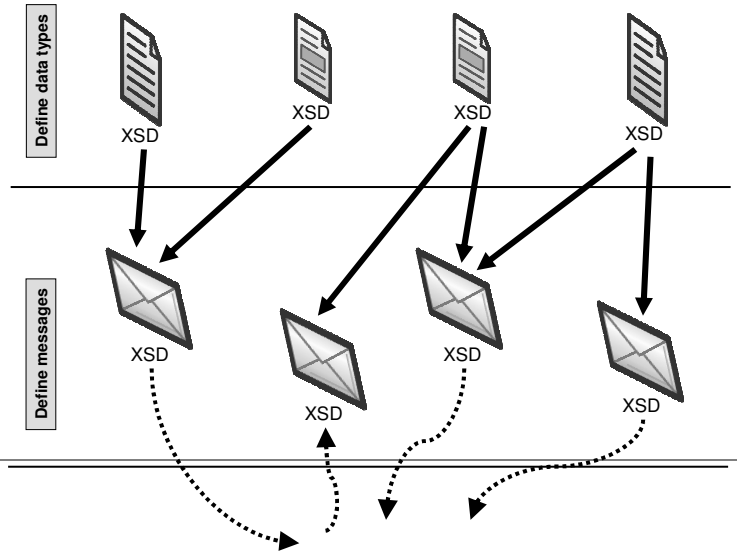
THILO FROTSCHER
Software Architect&Trainer

Schema-based Contract First

- Four simple but effective steps (or five…)
  - Define your data types
  - Define your messages
    - Steps 1 & 2 can sometimes be merged to one single step
  - Create your service contract
  - (Check if service contract complies with Basic Profile)
  - Create code skeletons and proxies
  - Iteration…

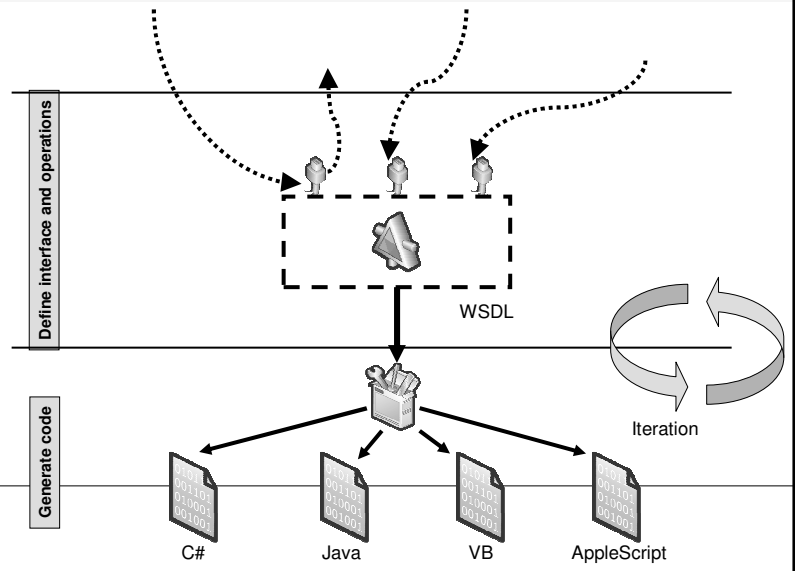
THILO FROTSCHER
Software Architect&Trainer

Contract-First - Steps 1 & 2



THILO FROTSCHER
Software Architect&Trainer

Contract-First - Steps 3 & 4



THILO FROTSCHER
Software Architect&Trainer

Fair enough, but...

- Contract First requires deep understanding of XML Schema and WSDL design
- Developer must be able to understand and follow WS-I Basic Profile
- That doesn‘t exactly sound like fun...
  - Who really knows all the details of WSDL?
  - Who wants to know all the details of WSDL? ☺
  - WSDL 1.1 is partly mistakable and contradictory
  - Did you ever have a look into the XML Schema spec?
- Aren‘t there any tools to help us out?

THILO FROTSCHER
Software Architect&Trainer

Insufficient IDE support

- Visual Studio.NET
  - schema editor, XML editor, no WSDL editor
- Eclipse
  - No built-in WSDL or schema editor
  - WTP project has good support
- JBuilder 2006
  - no schema editor, limited WSDL support
- NetBeans
  - no schema / WSDL editor
  - no plug-ins available (?)

THILO FROTSCHER
Software Architect&Trainer

Contract First approach hardly supported

"...it has been proven that contract-first development improves interop. While there are some people within MS that believe this to be true, clearly not everyone does. When we ask why there isn't more of a focus on contract-first, the standard answer is: everyone just wants to write classes.

I find this hard to believe. If anything, it's the lack of tool support that makes people feel this way (do they even know of another way?)." Aaron Skonnard

- Tool vendors want to sell RAD:
  little or no coding + much generation = service

THILO FROTSCHER
Software Architect&Trainer

Clever developers know how to help themselves

- Creating a WSDL from scratch is hard, so...
- Start implementing your application using the "Implementation-First" approach
- Use tool to generate WSDL document
  - Caution: potential interop problems through this step!
- Make WSDL interoperable manually
  - Yeah right! ☺
- Carry on with "Contract-First" from here

- Quote: "...you have to fight against today‘s tools if you want to use the Contract First approach."
  - Most recent tools are actually not that bad anymore

What do we need?

- WSDL editor
  - Generally the majority would agree that creating a WSDL document manually is hard
  - Ideally we would have a graphical or dialog-based editor
- Schema editor
  - XML Schema specification is even more complex than WSDL's
  - Again, a visual editor would be nice
- Integration into IDEs
  - Generation of service skeletons and proxies from WSDL & schemas
- Hide details of WSDL and XSD from developer
- An ideal solution would be a Contract Editor

Available Tools

- XMLSpy 2006 (Altova)
  - XML Schema editor, WSDL editor
- Eclipse WTP
  - XML Schema editor, WSDL editor
- Stylus Studio 2006 (Progress Software)
  - XML Schema editor, WSDL Editor (dialog-based)
- <Oxygen/> Xml Editor 7.1
  - XML Schema editor, WSDL Editor (text-based only)
- CapeClear Stylus Studio
- No real Contract Editor available ☹

THILO FROTSCHER
Software Architect&Trainer

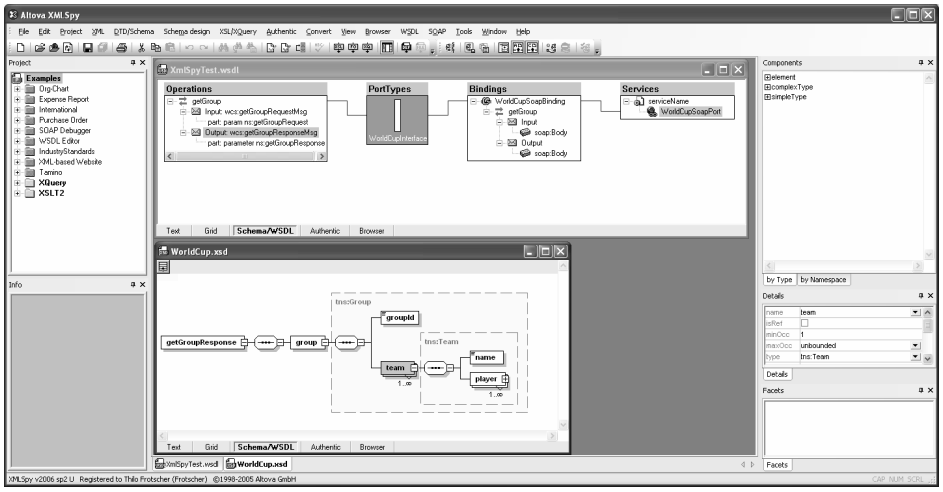
Altova XMLSpy Enterprise Editition 2006

- Graphical schema editor
- Schema agent: Client/Server-based collaboration tool for editing schemas in a team
- Edit embedded schemas in WSDL
- Graphical WSDL editor
- WSDL validation
- Plug-ins for Eclipse and Visual Studio.NET

- Home Edition is free (no WSDL editor)
- License for Enterprise Edition: 999 EUR

THILO FROTSCHER
Software Architect&Trainer

XML Spy‘s WSDL editor



THILO FROTSCHER
Software Architect&Trainer

XMLSpy 2006 - Conclusion

- Good XML Schema editor
  - Requires some knowledge of XML Schema though
- Average WSDL-Editor
  - Import of schemas works well, but not very intuitive...
  - Graphical editing not either
    - you need to know WSDL well to get where you want to go
  - Changing SOAP message style from RPC to Document doesn‘t work without manual extra work
  - Developer must know URI for HTTP transport himself
  - Binding elements are not generated automatically
    → again good knowledge of WSDL is necessary
- XMLSpy is not a "Contract Editor" either...

THILO FROTSCHER
Software Architect&Trainer

Annotations and web services

- Define service operations, message contents, encodings etc. through extra info in your code
- WSDL is generated from this information
- Leads to Code First approach
  - ...and possible interoperability issues
- Works partially quite good anyway
  - Platform needs to make sure that Basic Profile is strictly followed when interpreting / processing annotations
- Examples
  - Microsoft .NET
  - JAX-WS 2.0

THILO FROTSCHER
Software Architect&Trainer

General interoperability tips

- Encoded vs. Document
- Think twice when creating your schema
  - XML Schema allows definition of very complicated data structures
  - Some of these can‘t be realized with today‘s programming languages → code generators fail
- Schema imports in WSDL documents
- .NET code generator causes heaps of headaches
  - Problems with `attributeGroup` und `choice`
  - Problems with chained schema imports
- SOAP Faults and Exceptions...
- SOAP Action
- WS-Security, certificates, encryption algorithms

THILO FROTSCHER
Software Architect&Trainer

Summary

- WSDL First / Contract First / Schema First vs. Code First / Implementation First
- Contract First is best approach for green field dev, sometimes even in case of existing code
  - Approach is robust and reliable
  - Best preservation of semantics
  - Leads to best interoperability of your services
  - But you still need to follow some rules for XSD/WSDL
- No silver bullet!
- Quite often a mixture of approaches makes sense
  - In case of pre-existing code check out Axis2 & JiBX

Summary

- Tool support sometimes still not sufficient, but becoming better
- Visual schema and WSDL editors becoming available
  - Caution: tools tend to generate schemas and/or WSDLs that are invalid or not interoperable
  - Deep knowledge of specs still necessary, especially in case of WSDL
- When will we get a Contract Editor???
  - Good approach: WSCF from thinktecture

Interoperable web service apps

Thank you!

thilo@frotscher.com
http://www.frotscher.com