Apache Axis :

History, Architecture, and Why Open-source SOAP Implementations Are So Cool

Glen Daniels
Macromedia / Apache
October 10, 2002

Outline

- Axis History
- Architecture and Usage : Some Gory Details
- Extensibility with Axis
- Spec Compliance
- Why Open Source?
- Futures
- Q&A

Axis History

- In the beginning, there was SOAP4J...
- Submitted to Apache, became SOAP 2.0
- Issues:
  - Extensibility / Modularity
  - Support for headers
  - Doc/lit support
  - Performance

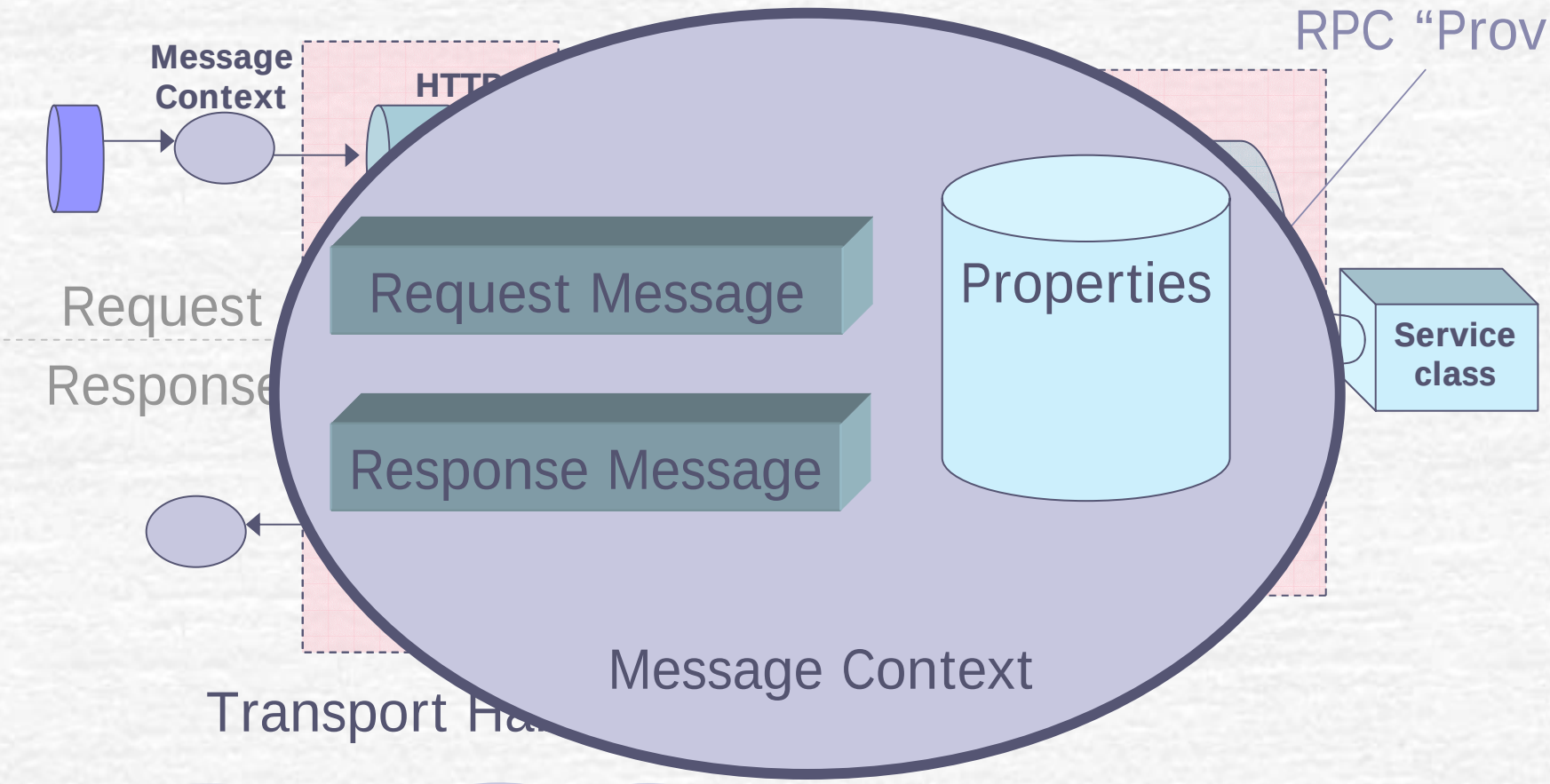
History continued...

- And then there was Axis...
- F2F @ XML2000 – about 15 people
- Real work began in Spring 2001
- Things have changed a lot, but the basic idea/framework is the same
- Integrated into JRun, WebSphere, etc.
- Several companies building products on top of Axis

Architectural Guidelines

- Extensibility / Flexibility
  - “Handlers” as core concept
  - Messages flow through a configurable pipeline of components
- SOAP header processing
- SAX based XML handling
- Simplicity / ease-of-use
- WSDL - Stubs, Skeletons

Message Processing (Server)



Mighty Morphin' Messages

- A Message in Axis exists in a "form"
  - byte []
  - String
  - SOAPEnvelope
- Messages are transformed when a particular form is requested
- Intermediaries might pass messages through with no parsing

XML / Java Databinding

- Readin' : SAX-based deserializers
- Writin' : XML SerializationContext
- Handles basic types, beans, collections, multirefs natively
- Can register custom serializers / deserializers

Client APIs

- Dynamic

```
Object [] arguments = new Object [] { "test Arg" };
String ret = (String)call.invoke("method", arguments);
```

- Generated stubs

```
FlightList flights =
        airlineStub.checkFlights("BOS", "SFO", today);
```

- Handlers work on the client side too

Service Deployment in Axis

- JWS - "Instant" Deployment
  - Write a Java source file, save as .jws, drop in your web hierarchy
  - Engine compiles + deploys on the fly
- WSDD - Using Deployment Descriptors
  - Custom type mappings
  - Handlers
  - More configurability

Example of WSDD

```
<wsdd:deployment>
  <service name="Test" style="RPC">
    <parameter name="className" value="TestClass"/>
    <parameter name="allowedMethods" value="*"/>
    <parameter name="customManagement" value="yes"/>
    <beanMapping qname="myNS:type"
                 type="org.wsdevcon.Type"/>
  </service>
</wsdd:deployment>
```

Service Scopes

- Axis supports three scopes
  - request - new service object on each SOAP request
  - singleton - service objects shared by all requests
  - session - service object per Session
- "Session" is an abstract interface
  - We include HTTP + SOAP-header versions

Extensibility : Handlers

- Handlers can communicate to each other via MessageContext properties
- Handlers can be used to implement:
  - Security
  - Management
  - QoS
  - New transports
  - Routing
  - Debugging

Using Handlers : An Example

1. SecurityHandler pulls <sec:encrypt> header out of SOAP message
2. Writes "security.principal" property into MessageContext, based on encryption cert.
3. Decrypts SOAP envelope and updates the Message
4. Later, another handler can directly access the "security.principal" property in the MessageContext

Spec Compliance

- SOAP 1.1
- WSDL 1.1
- XML Schema 1.0 (partial)
- JAX-RPC 1.0
- SAAJ 1.0
- Some SOAP 1.2

Sidebar : WSDL 1.1 Issues

- <soap:header>
  - Can't specify related or dynamic headers
- Fault model
  - No place to spec faultcode/string
- doc/lit/rpc/encoded - too many options
- This stuff should be fixed in 1.2

Why Open Source?

- SOAP is plumbing; commoditizing the engine is a good thing
- Team is composed of more than one company's resources
- Fixes (interop, etc) can happen **fast**
- Smart people can jump in and participate

How Well Does It Work?

- IRC helps a lot (high-bandwidth chat)
- Some concerns are indeed addressed much quicker than in traditional dev...
- ...but some aren't
- Team needs critical mass of committers with shared vision
- An active user community is fabulous!

Future Directions

- Track specs, esp. SOAP + WSDL 1.2
- Factor out XML/Java databinding
  - Fully support XML Schema spec
- Packaging for extensions/services as components (jars)
- Start building "Module" library
- Funnel experience into JAX-RPC "dotnext"

Summary

- Axis...
  - is powerful, flexible and extensible
  - is open-source
  - has good buy-in from the Java community
- Axis 1.0 is out, more to come
- Want to get involved?
  http://xml.apache.org/axis

Q & A

- Questions?