Woden Project Proposal

Table of contents

1. Project Proposal

March 2005.

The Woden project is proposed as a subproject of the Apache Web Services Project to develop a Java class library for reading, manipulating, creating and writing WSDL documents, initially to support WSDL 2.0 but with the longer term aim of supporting past, present and future versions of WSDL.

There are two main deliverables: an API and an implementation. The Woden API will consist of a set of Java interfaces. The WSDL 2.0-specific portion of the Woden API will conform to the W3C WSDL 2.0 specification. The implementation will be a high performance implementation directly usable in other Apache projects such as Axis2.

Members of the WSDL4J team, Apache Axis, Eclipse and W3C will collaborate on this project. The Axis2 community have already put some effort into a WSDL 2.0 Architecture. The Eclipse Web Tools project has experience in developing WSDL validators and one of its members is also a member of the W3C WSDL 2.0 Working Group.

2. Potential applications

- Axis2, build time emitters and runtime parsing
- Eclipse tools, WSDL editors
- vendor products
- user applications

3. About the name "Woden"

- Woden is a mnemonic for Web service Description (i.e. WoDen).
- Woden is a Norse god and naming the project after a god is in keeping with other WS-Apache project names like Apollo, Hermes and Muse.
- Woden invented a form of writing called Runes (i.e. a form of Description).
- The symbol CEN in Runes looks like "<", a left angle bracket, which may suggest we can trace the origins of XML back to Woden!

4. Rationale

There are current requirements for a WSDL 2.0 processor (e.g. Axis2, Eclipse tools, the WSDL4J community). This demand is expected to increase when the W3C issues a Recommendation for WSDL 2.0.

The W3C WSDL 2.0 specification is currently at the "Call for Implementations" stage, which expects two independent implementations before progressing to the implementation review stage and then on to a W3C Recommendation. A WSDL 2.0 Java implementation from Apache in response to this call could help verify the quality of the specification and improve its implementation experience.

Previous WSDL processors have been version specific, such as WSDL4J which is tightly bound to WSDL 1.1. In the longer term the Woden project could take the experience gained developing previous WSDL processors and deliver a WSDL processing "framework" that could accommodate future WSDL specifications without the need to completely rewrite the processor.

We anticipate that implementations of the Woden API will also be developed by other projects for specific applications (e.g. the Eclipse Web Tools project may develop an EMF based implementation to support WSDL 2.0 editors, as it has done with JWSDL for WSDL 1.1). A common API will be a good candidate for future standardization on a WSDL processing API.

5. Scope

In the short term, the scope of the Woden project is:

- A WSDL 2.0 API that will significantly cover the W3C WSDL 2.0 specification, including the WSDL 2.0 Component Model, Soap bindings and message exchange patterns and XSD schema typing.
- A high performance implementation of the API with read-write capability, but optimized for run time applications rather than build time tooling.
- Fast, efficient XML parsing (e.g. StAX-based).
- A validator component that will provide optional WSDL validation, both schema validation and semantic validation.
- An Ant task for command line usage based on Ant XMLValidate.
- Support usability and flexibility across different usage scenarios through the handling of behaviour such as I/O, error reporting, progress monitoring, memory efficiency and thread safety.

Longer term, Woden's scope will include:

- Full support for remaining parts of the W3C WSDL 2.0 specification, such as other protocol and type extensions.
- A Woden framework based on an extendable architecture that will allow for known or future versions of WSDL to be plugged in, rather than hard coded in the core of the WSDL processor (e.g. plugging in WSDL 1.1 support).
- A pluggable choice of XML parsing and object representation to support particular usage

scenarios (e.g. a StAX-based, DOM-free XML model for a runtime application versus a DOM-based model for build time tooling).

6. Initial Source

IBM will donate the WSDL4J source code. Although WSDL4J is the reference implementation of JSR110 (Java APIs for WSDL), Woden will not be an implementation of JSR110. WSDL4J is specific to WSDL 1.1 but some of the code and the intellectual property may be reused to develop a WSDL 2.0 processor.

7. Resources to be created

Mailing Lists: woden-dev, woden-user

SVN Repository: ws-woden

Issue Tracking: JIRA

8. Apache Sponsor

The Champion of the Woden incubator project is Davanum Srinivas (dims at apache.org).

The Sponsor of Woden is the Apache Web Services PMC.

9. Initial Committers

Chathura Herath (chathura at opensource.lk, WS-Apache Axis)
Jeremy Hughes (hughesj at apache.org, WSDL4J)
John Kaputin (jakaputin at gmail.com, WSDL4J)
Lawrence Mandel (lmandel at ca.ibm.com, Eclipse Web Tools (WTP) / Eclipse Web Service Validation Tools (WSVT))
Arthur Ryman (arthur.ryman at gmail.com, Eclipse Web Tools (WTP) / W3C WSDL 2.0 Working Group)

10. Incubator Mentors

Paul Fremantle (pzf at hursley.ibm.com)
Sanjiva Weerawarana (sanjiva at opensource.lk)

11. Relevant links

- Eclipse Web Service Validation Tools (WSVT) - http://www.eclipse.org/wsvt

- Eclipse Web Tools (WTP) - http://www.eclipse.org/webtools
- W3C WSDL 2.0 Working Group - http://www.w3.org/2002/ws/desc/
- WSDL4J - http://sourceforge.net/projects/wsdl4j/