Mathematical Software as Web Services

O. Caprotti* W. Schreiner
RISC-Linz
Johannes Kepler University
Linz, Austria
{ocaprott,schreine}@risc.uni-linz.ac.at

May 13, 2002

1 Introduction

This poster is a description of the usage of mathematical computational software to setup the web services available at `http://perseus.risc.uni-linz.ac.at:8080/openmath`. We report our experience and explain the components of the architecture we have realized while working on the design of a description language for the specification of mathematical web services [3]. The general principles of the architecture are not limited to mathematical services but apply to any web service, however in this study we deal with certain technologies that are specific to the area of on-line mathematics, in particular OpenMath [4].

2 Architecture

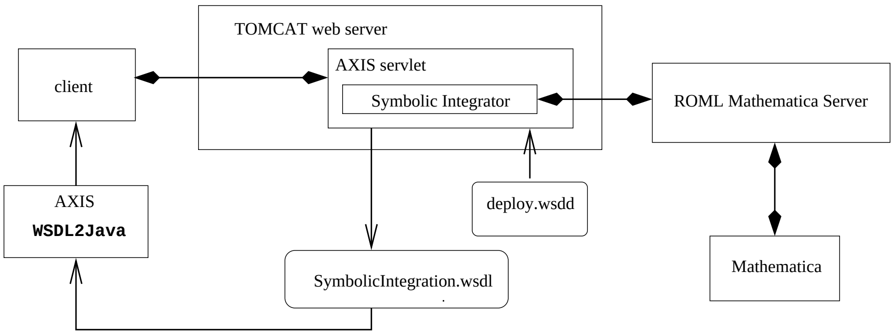

Figure 1: General Architecture

Figure 1 depicts the various components of the architecture used to provide web services on-line. The central node is a web server enhanced by the capability of handling the Simple Object Access Protocol (SOAP) used by the services it manages. Another node houses the computational engine(s), in this specific example it is the computer algebra system Mathematica. A third node represents the client that accesses the on-line services either by using a web browser of directly by a command line application. Notice the two types of arrows: the ones with a full arrowhead represent the flow of information between the client and the final computational engine, the other represent input and output flow between the service or client developer. In the next sections we will explain the details concerning each node.

*Funded by FWF Project P15183 "Brokering of Distributed Mathematical Services"

2.1 Web Server

As Figure 1 shows, the web server is the core of the architecture. Such a server provides a container which administers the services that one wants to make available. Administration in this case is very simple, it amounts to being able to list the services, provide a description of their interface and the ability for the administrator of dynamically adding/removing a service. To the final client of a service, the web server provides a URL, which points to somewhere within its domain, where the service access point is located.

We have installed the Apache Tomcat web server, a highly configurable server that supports Java servlets. The web container application that handles web services and SOAP is also an Apache package called AXIS (Apache eXtensible Interaction System) [1].

The integration service is implemented by a Java class `SymbolicIntegrator` that calls the Riaca OpenMath phrasebook for Mathematica. In order to deploy the service on the server the AXIS `AdminServlet` needs to be informed of the newly available service. This is achieved using a `wsdd` (web service deployment descriptor) file which specifies certain properties of the service such as the name of the Java class which implements it. After deployment, the service is on-line at a certain URL, in this case `http://localhost:8080/axis/services/SymbolicIntegration`.

2.2 Computational Engines

The computational engine in this architecture is the computer algebra system Mathematica accessed through the Java service wrapper provided by the Riaca OpenMath library [2]. The wrapper takes as input an OpenMath object encoded in XML and returns OpenMath output. Within this wrapper, the OpenMath Mathematica phrasebook is the program that, given an OpenMath object and a command directive (in this example it is a request for "evaluation"), knows how the query should be translated to a query that Mathematica understands. Vice versa, getting a response from Mathematica, the phrasebook converts it to OpenMath. The choice of OpenMath as the language in which to convey the data manipulated by the service ensures transparency with respect to the input/output syntax. If we would decide to use another software package for supplying the computation, we would only need to replace the phrasebook.

Figure 2: Screenshot of the Symbolic Integration service

2.3 Client

A client is a piece of software that connects to the service. In particular for services that expect data in XML, it is common that the client produces the XML from a more natural, and user-friendly interface. This is especially true for mathematical input which can be asked in a structured way and then internally organized as required by the service.

The client we have implemented as a web application takes as input an expression, parses it and generates the OpenMath XML encoded object representing it. This object is then the argument sent to the `SymbolicIntegration` service which is in charge of decoding it to the proper syntax, depending on the computational engine used for the evaluation. There are three main components of the client:

Parsing A small Java parser has been generated by the tool ANTLR from a grammar for simple algebraic expression in the variable x.

GUI A JSP (Java server page) page acts as graphical user interface for accessing the service allowing the user to input the algebraic expression, converting it to OpenMath, shipping it to the service and displaying the result for instance in MathML.

Service Connection The code for the classes that allow to connect to the service have been generated automatically by the AXIS application `WSDL2Java` from the WSDL description of the service. Such a description is obtained easily by adding `?wsdl` to the URL of the service, namely `http://perseus.risc.uni-linz.ac.at:8080/axis/services/SymbolicIntegration?wsdl`.

3 Conclusion

We have documented the implementation of a mathematical web service providing symbolic integration. Although quite simple, yet this example can be considered the basis on which to construct more involved/specific services. We have also mentioned several technologies that were heavily used to carry out parts of the development: web server technologies, such as JSP, mathematical communication technologies, such as OpenMath and OpenMath phrasebooks, and web service technologies such as AXIS, SOAP, and WSDL. The combination of all of these made the overall job possible.

Our main interest and future work is in the design of a specification language for mathematical services which can be suitable for registration and service discovery as well as formal documentation of the service semantics [3]. In particul, it should support "intelligent" discovery of mathematical services.

References

[1] Axis SOAP library. `http://xml.apache.org/axis`.

[2] RIACA OpenMath Library. `http://www.riaca.win.tue.nl`.

[3] O. Caprotti and Wolfgang Schreiner. Towards a Mathematical Services Description Language. In *Proceedings of the International Congress of Mathematical Sofwtare 2002*, Beijing, China, August 2002.

[4] OpenMath Society. `http://www.openmath.org`.