Book Review

Programming Web Services with XML-RPC
by Simon St. Laurent, Joe Johnston, and Edd Dumbill

O'Reilly & Associates, 2001
Cover Price US$34.95
ISBN 0-596-00119-3
(230 Pages)

Before I opened this book, I knew I was interested in the Web Services and XML (Extensible Markup Language) part of the title, but I was not so sure about the RPC part. After all, the Web Services technology everyone's been talking about lately is SOAP (Simple Object Access Protocol). Now that I have read the book, however, I understand:

- The main differences between XML-RPC (Remote Procedure Call) and SOAP (SOAP is object-oriented and allows other transport layers besides HTTP; RPC is simpler and stateless).
- When to use XML-RPC and SOAP, respectively.
- How to use XML-RPC.
- More about the origins of SOAP within RPC technology.
- The key to XML-RPC technology in the HTTP Post request.

For all of these reasons, and for the variety of applications it discusses, the book is worthwhile reading. An advanced book on SOAP would complement this offering very nicely.

This book concentrates on the XML-RPC protocol specification and its uses in Web Services, including implementation in Java, Perl, PHP (hypertext preprocessor) server-side scripting language, Python, and Active Server Pages so that these languages can talk to each other. In XML-RPC, the XML markup language describes Remote Procedure Calls (RPC) transmitted via HTTP. The book's Foreword includes a brief history of this technology, written by one of its inventors. There's also an interesting Appendix on the XML and the HTTP used in the text.

It turns out that Web Services that rely on protocols such as XML-RPC facilitate interoperability between computer programs in different languages running on different operating systems -- and this aids in
building and debugging distributed applications. Basically, it works like this: An XML-RPC client specifies a method name and parameters packaged in an XML file payload and issues a Post request to a target server where the method is called. (The book’s Appendix reminded me that forms can post/upload files by specifying an input of type file and a content type of multipart/form-data; for efficiency, XML-RPC headers specify a content type [MIME type] of text/xml instead.) The method’s output is sent as an XML packaged response, including XML structs converted from the host programming language data type notation. The client need not be a browser. In contrast to Java’s Remote Method Invocation, which supports CORBA interoperability, XML-RPC does not pass objects, and is therefore cheaper to implement.

The authors discuss both the advantages of open programming interfaces and the scalability and security minefields involved in using XML-RPC. In the process they document many examples, using a variety of languages and libraries:

- The helma.xmlrpc library for RPC-XML clients and servers for Java.
- The Frontier::RPC library for Perl.
- The xmlrpc.inc and xmlrpcs.inc library for PHP.
- Pythonware’s xmlrpclib for Python.
- xmlrpc.asp.
- A C++ COM object implementation of XML-RPC for Active Server Pages.

Depending on which of these languages you have available, you may want to experiment with some examples and gain insights from reading the others. One example application used PHP to integrate two Web applications (a discussion server and a news database) into one interface, so that users could annotate technology news articles and read comments posted by others. The Active Server Pages example used XML-RPC to make an MSAccess address book accessible to a Linux Web server on the network and a MySQL address book on Linux available to an IIS server on Windows NT. Before reading this book, I had worked with calling, receiving, and applying XSL (Extensible Style Language) stylesheets to the responses of some webcrawlers and public Web Services allowing Web clients. The examples in the book helped me to better understand the internals and the variety of application types for which XSL may be used.

I recommend this book for Web developers and program integrators who want to use the proven, and relatively simple, XML-RPC standard to integrate systems or to provide services, or who want an excellent foundation for moving on to more complex protocols such as SOAP.

- Evelyn Roman
Rational Software Internet Services
Build Leaders at Every Level by Noel M. Tichy.