Web Services

By G. Alonso, F. Casati, H. Kuno, V. Machiraju,
Springer-Verlag, 2004, 354 pages,
ISBN 30540-44008-9

Review by:
Dirk Wodtke
dirk.wodtke@sap.com

Yet Another Book on Web Services? If you type in the term "Web Services" in the search field of Amazon.com you will get more than a thousand entries in the result set. One of the entries is for "Web Services: Concepts, Architectures and Applications" written by Alonso, Casati, Kuno und Machiraju. The question you may have is: Is this "yet another book on web services"? Clearly, the answer is "No".

This book differs from many other books on the topic of web services in that it focuses on the concepts, origin, and the future of web services, as opposed to tutorial like books on the topic. It is a must read for software developers, architects, technical decision makers in R&D positions and students who want to get an overview of the state of the art of web services. If you want to find out if web services are the killer technology for solving all existing software integration problems or if there are already better solutions on the horizon then this is definitely the right source of information for you. However, if you are looking for API documentation or programming samples and expect to immediately start implementing web services after reading through the 354 pages you're better off with other books on the web service topic.

The book is organized in two major parts. The first part describes the principles and the architecture of "traditional" (pre-web service era) middleware. Examples of traditional middleware are RPC based communication protocols, TP monitors, Object Request Brokers, Message oriented middleware, transactional messages queues, to name a few. The authors describe how middleware can be used for Enterprise Application Integration (EAI) which has strong links with the area of Workflow Management. Since the authors are well regarded Workflow researchers it is no surprise that they position web services as the key technology for Workflow. Workflow complements EAI in that it provides means for defining and controlling business process whereas EAI hides the heterogeneity of the applications which are plugged into business processes. EAI based on traditional middleware has some shortcomings: In order to run business processes heterogeneous middleware components which can be heterogeneous themselves have to learn how to talk to each other first. Also EAI is intended for short-lived interactions, however, the execution of a business process is usually long-lived.

The second part of the book focuses on web services. The authors describe the three important building blocks which comprise a web service based solution:

- A language for describing the interfaces of web services – the de facto standard is Web Service Definition Language (WSDL)
- A transport protocol - the de facto standard is Simple Object Access Protocol (SOAP)
- A registry for the discovery of web services - the de facto Standard Universal Data Discovery Integration (UDDI)

These three components just provide the pure plumbing for using web services. In order to provide an EAI solution things like a process context which is kept over a sequence of web service calls, the ability to nest web service calls or to combine web service calls into transactions are required.

These problems are tackled by some emerging standards: WS-Coordination provides a context which can span several web service calls, WS-Transaction is a 2PC based transaction protocol for web services. According to the authors theses standards still need to mature, are quite complex and even contradictory. In addition to these standards there are "vertical" protocols like Rosetta Net. This is a standard for exchanging business document in supplier-customer scenarios. Furthermore, there is a brand new standard, Service Composition, which allows for composing web services into complex web services. A well done example for web service composition is the Business Process Execution Language (BPEL). By employing a graphical tool service composition can be greatly simplified so that in the end business processes can be defined without having to implement any program code. According to the authors this goal is easier to achieve with web services than with traditional middleware.

Where will Web Services go from here? Web services play the role of application wrappers which allow for invoking applications in a homogenous manner. For integrating existing applications the main effort will be on the implementation of web service based application wrappers. The good news is that there are more and more applications which already come with web service wrappers provided by the application vendors. According to the authors both SOAP and WSDL have a good chance to survive as EAI enablers. This does not apply to UDDI, though. According to the authors it is quite unlikely that during the execution of business processes software agents will ever dynamically query UDDI repositories for finding the services they are looking for.

The book might not answer all questions readers might have. For example, the authors don't quantify the performance impact of adding one more software layer, i.e., a web service layer, on top of software that in many cases already consists of a considerable number of layers. Another aspect which raises a question the book does not answer is how web services can be used for integrating application which come with a graphical user interface into business processes. Some passages of the book appear to be lengthy discussions of background topics. Do readers of the book really need to understand what Enterprise Java Beans are in full detail? The introduction of the book could be shorter: On the first 120 pages of the book web services are virtually not mentioned. Nevertheless the first part of the book is very informative. Would "Web Services and Middleware" be a more appropriate title of the book?

Despite these minor shortcomings the book is definitely worth buying. Unlike other textbooks on web services there are no lengthy sample XML snippets in it. The assessment of the potential of web services is very much down to earth. The book avoids all web service "hype". Consequently, the authors judge web services to be a natural evolutionary step of software technology which simply wraps traditional middleware. Web services are not the "big bang" standard which immediately makes existing and competing standards obsolete. The authors predict that the old technologies and standards will coexist with web service based solutions for some time to come.