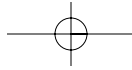


## Foreword by Steve Mills

Web services are about enabling complete business-level interoperability across vendor middleware platforms. Starting with the first baby steps—in the form of SOAP—more than five years ago, I have seen the industry moving steadily toward this integration platform with the introduction of one key component after another. While we are not completely there yet, I am confident that a solid technical foundation has been laid and that now the community building process is under way.

This book is about that technical foundation. Starting with an explanation of some of the motivating factors driving customers toward full, in-depth integration of middleware platforms, the authors take you through the entire Web services platform and give you an understanding of why this platform effectively solves the integration problem. Two case studies, one in a business-to-business setting and the other in an enterprise application integration setting, illustrate how the Web services platform can be effectively applied to meet all the integration requirements. Finally, the authors provide an outlook of what is still coming down the pipe, in terms of both the short-term and long-term evolution of the integration space.



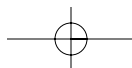
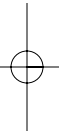
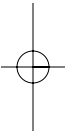
xxii Foreword by Steve Mills

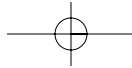
While there are dozens of books out there about Web services, this book is unique for two critical reasons: it is the first book to provide comprehensive coverage of the entire Web services platform; and, rather than providing an unguided explanation of a plethora of “WS-\*” specifications, the authors make subjective judgment calls on which specifications are core to the platform and which are not. Their bold assertions of what is core to the platform makes it clear that the Web services platform has a clear underlying architecture and that it is not just a collection of specifications, as is commonly held.

The second reason above hints at a key reason for why this book is extremely unique: the authors are in fact the very same people who have been involved with defining *all* of the specifications that comprise the Web services platform. This is as “from the horse’s mouth” as it can ever get!

Given the importance of true interoperability across vendor platforms, no one can afford to not fully understand Web services. I suspect this book will be a key milestone in your travels down the Web services roads.

Steve Mills  
Senior VP and Group Executive  
IBM Software Group  
December 2004

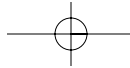




## Foreword by Ronald Schmelzer

One of the constant struggles for enterprises of all types, sizes, and industries is the ability for their Information Technology (IT) systems to meet their evolving business needs. Because most enterprises' IT systems are a hodgepodge of systems of different types, ages, architectures, and technologies, companies must continue to invest in their increasingly complex IT infrastructure while seeing gradually diminishing benefits. At times, it seems that the more that companies spend on IT, the less business benefit they receive, because more time is spent on making existing systems talk to each other rather than on making them accomplish new, productive tasks for the enterprise.

Part of the reason for the ongoing IT problem is that companies don't make technology and architecture decisions all at once. Rather, most companies have to make IT decisions as they go using the best information and reasoning they have at the time. At times, these decisions are made due to expediency, and at other times, the decisions are made for strategic reasons that are lost when the company goes through significant change, such as a merger or acquisition, or during a period of significant economic downturn. As a result, many of these IT decisions result in significant expense for the company.



xxiv Foreword by Ronald Schmelzer

Furthermore, as a company accumulates IT assets, new technologies emerge that seem to make older ones implemented in the enterprise obsolete. However, companies are loathe to throw out their previous IT investments (legacy systems) and replace them with new systems, and as a result, these companies are faced with trying to make their old systems work in new environments. So, rather than simplifying their IT ecosystems over time, the problem only grows more complicated, more expensive, and more inflexible as new systems are introduced. This problem of heterogeneity is at the core of why integration challenges continue to plague so many organizations.

To solve these problems, companies need two sorts of solutions: technology solutions that aim to simplify integration problems through a combination of standards-based interoperability and enable reuse of legacy environments, and architectural solutions that aim to change the way in which companies build, deploy, manage, secure, and scale their IT systems. While the latter solution requires changes in IT management, structure, and governance, the former solution requires a standards-based, comprehensive technology platform for working in a heterogeneous IT environment. Web services-based Service-Oriented Architectures (SOA) hope to provide both of the above sorts of solutions to the enduring problem of IT inflexibility.

Smart enterprises are increasingly realizing that the real value in Web services is in using loosely coupled, standards-based technologies to build SOAs. Many enterprises have achieved success implementing Web services to solve point-to-point integration problems and are now looking to leverage the power and flexibility of Web services strategically across the enterprise, which means building loosely coupled, standards-based SOAs.

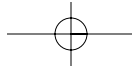
The power and flexibility that SOAs can offer the enterprise are substantial. If an organization abstracts its IT infrastructure so that it presents its functionality in the form of coarse-grained services that offer clear business value, then the consumers of those services (whether they are with the same company or one of that company's business partners) can access those services independent of the underlying technology that supports them. Furthermore, if service consumers can dynamically discover and bind to available services in an agile manner—building applications out of composed services—then the IT infrastructure behind those services can offer extraordinary flexibility to the businesses that invoke them.

The difference between the practice of SOA and other approaches to enterprise architecture is in the business agility that SOA offers. Companies have become used to the fact that IT decision making and implementations impede their organization and that technology and its limitations often drive business decisions. Service orientation, however, has the potential to change this equation and enable business decisions to finally drive their technology decisions. Business agility is the ability of a company to respond quickly and efficiently to change and to leverage change for competitive advantage. For the architect, building an architecture that provides business agility means creating an IT infrastructure that meets as-yet unknown business requirements—a situation that throws traditional IT planning and design out the window.

Today's distributed computing transition, while every bit as significant as the ones that came before, has an entirely different economic model. Instead of massive IT investment, today's IT executive is concerned with thrift. Thrift depends on one of the holy grails of software development: code reuse. In an SOA, developers should construct the services to be as simple as possible, where they continually refactor them so that they are as broadly applicable as is practical. The resulting services are then reusable at runtime—both fine- and coarse-grained nuggets of software functionality that can be used in a variety of situations, as contrasted to typical code reuse, which is a design time principle.

Once the SOA is in place, new business requirements will continue to generate the need for new and updated services. The IT staff can then make the required changes on an ongoing basis. In addition, taking an ad hoc upgrade approach to services, which are composed into business applications, reduces the need to “rip and replace” large portions of the IT infrastructure. Companies thus only consider a rip and replace strategy as a last resort, and then only within a service orientation context.

There is an additional, related concept to broad applicability that goes even further: the concept of consumability. It's not enough for a service to have the potential to be used in a variety of situations; it must actually be usable. Not only must the service's functionality be technically applicable to various situations, but people must know about the service, understand its use, and be able to find it when they need it. As such, technologies such as repositories and metadata management tools are rapidly becoming as important as the runtime and design-time infrastructure for the services themselves.



xxvi Foreword by Ronald Schmelzer

While the concepts of SOAs sound compelling to most enterprises, building service-oriented infrastructures is not easy. That is why this book, *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*, is such an important and critical reading. In this book, Sanjiva Weerawarana, Francisco Curbera, Frank Leymann, Tony Storey, and Donald Ferguson make the effective argument that Web services and SOA are required technologies that help businesses meet their continually evolving requirements.

The book offers not only implementation-level coverage of the technologies and specs as required by technical developers, it also provides the right technical context for business readers. An emerging audience of “enterprise architects,” which are more business focused than developers, yet tasked with more specific technical requirements than purely line-of-business users, will find significant value in the book. This book gives these users the ammunition to discuss the topics with their more technical developers, and the nuts-and-bolts to implement higher-level concepts decided by their more business-level superiors.

In the first chapter, the authors make the correct observation that SOA is a new paradigm shift that requires not only a change of technology, but also a cultural mind shift in how to create, manage, secure, and deploy service assets in the network. The concepts are on target—correctly mapping the business driver of agility, or flexibility, to the technical capabilities of an SOA to enable rapid change with low economic penalty. The book then continues to discuss business flexibility in the abstract while tying the concepts to specific notions of the business process.

As the book progresses, the reader will learn incrementally more about the basic, core Web services specifications—SOAP, WSDL, and UDDI, as well as the XML-based underpinnings of those formats. What makes this book stand out are its detailed discussion on emerging specifications, including WS-Policy, WS-Addressing, WS-ReliableMessaging, WS-AtomicTransaction, WS-BusinessActivity, as well as the BPEL specification, which—while still emerging specifications as of the time of writing this foreword—will surely be the formative specifications for SOA in the future.

In addition, the authors spend a considerable amount of time discussing the practical implementations of SOA, especially in a messaging-centric context. Their discussions can immediately be applied to a wide range of message-oriented middleware approaches as well as emerging Enterprise Service Bus implementations.

In fact, what makes this book credible is the experience of the authors. The authors, hailing from the pioneering software group at IBM, have not only rich experience with Web services and SOA, but have also been involved in the creation of the specs themselves. As such, the authors not only bring their experience in implementing Web services and SOA at IBM, but also their experience in crafting the very specifications they outline in the book—this book is an “insider’s” guide to Web services and SOA, if you will.

In conclusion, service orientation represents the next major trend in enterprise computing, and as a result, requires a new perspective, new techniques, and new tools for implementing technology solutions that meet the needs of business. At this point in time, the IT industry stands at a cusp—a tipping point where sporadic applications of Web services become a movement toward agile, thrifty computing based on SOAs. When people stand at such a threshold, they often have a difficult time planning for the future because many of the business patterns that have applied in the past may no longer apply. It is our hope that through learning the basic precepts of SOA and Web services and the detail required to implement them in your most important applications, you can become informed and educated enough to participate and lead the revolution that SOA means to the enterprise.

Ronald Schmelzer  
Senior Analyst  
ZapThink, LLC  
December 2004