Service-Oriented Architecture

A Field Guide to Integrating XML and Web Services

Thomas Erl

PRENTICE HALL
PROFESSIONAL TECHNICAL REFERENCE
UPPER SADDLE RIVER, NJ 07458
WWW.PHPTR.COM

Contents

Chapter 11

CHAPTER
1

Introduction

1.1 Why this guide is important

"Hype," by definition, is an exaggeration of fact. Both XML and Web services have had their share. When looking back at the rise of these technologies and the excitement that surrounded them, many who bought into the hype are beginning to feel disappointed. Organizations are realizing that their technical environments are not magically transformed simply by adding XML or Web services to the mix.

I actually believe that much of this publicity has been warranted. This platform's potential is real, and its importance cannot be understated. When properly applied, it not only improves the technology of an organization, but also the manner in which business automation is delivered. *When properly applied.*

What's been exaggerated isn't as much the potential, but its perceived simplicity. A well-designed, service-oriented environment can simplify and streamline many aspects of information technology, but achieving this state is no simple matter. The technology set introduced by XML and Web services is diversely complex. In order to truly leverage its benefits, you first need to appreciate the implications of this complexity. Then, you need to strategize.

1.1.1 The hammer and XML

It all starts with XML. Like a hammer, XML is a tool. If you pay attention to how you use it, you'll hit the nail every time; if you don't, chances are your thumb will take a beating. On its own, it does not solve or create problems—the results of using XML are directly related to how intelligently it is applied.

For years, industry analysts have theorized and speculated about the benefits XML will introduce to the age of online data sharing. Because this technology platform provides a potential ideal for a universal data format, it will lead the world into a new era of information unity and parity.

The potential is real, but the manner in which XML is being applied in the real world is anything but ideal. XML is a specification, a revolutionary innovation that exists in a document that describes a simple idea, with huge implications. The fact that XML has been adopted into the IT mainstream is good. It establishes a common technology used for a common purpose. Simply using XML, however, in no way guarantees that you

will realize any of its true benefits. You will be staying current, complying with a worldwide platform shift, and you will not feel left out when reading about how others are riding the XML wave, but… you will not see anything revolutionary happening in your world.

1.1.2 XML and Web services

If XML is a hammer, then Web services are… what? The nail? The hand that holds it? Whatever it is you're building, Web services are the building blocks you can use after you've first pounded out a solid foundation with XML. That doesn't mean you can't start using Web services without first properly integrating XML, but then there's nothing stopping you from building that dream house in the swamplands either.

This guide, in fact, is mostly about integration with Web services technologies and service-oriented design principles. This makes it no less of a book about XML, since the Web services platform is a natural continuation of the XML movement.

1.1.3 Web services and Service-Oriented Architecture

As you read through this book, you will notice that the path to building service-oriented architecture is riddled with pitfalls and risks. Too often, organizations investing in Web services discover the errors of their ways once entire solutions have been built and deployed. This is not necessarily a bad or neglectful occurrence. It's simply a tribute to the vastness of this platform.

With its complex and comprehensive feature set, though, comes a load of power. Use this technology the right way, and you truly can build a better enterprise. That statement goes beyond IT, because service-oriented concepts can reach out and change the way you model your business. Grasping the potential is an important first step. Equally as important, though, is understanding what's involved with realizing this potential. That takes us back to integration strategy.

1.1.4 Service-Oriented Architecture and the hammer

Unfortunately, the majority of corporate IT departments do not employ any form of integration or migration strategy. Without a planned integration, standards cannot be positioned, and the resulting ad hoc usage of these technologies only ends up contributing to existing disparity. It's the equivalent of construction workers building a home without direction and without a blueprint. With the absence of a planned and coordinated effort, a group of hammering workers will not only *not* create a quality foundation, they won't be building anything resembling a foundation at all.

1.1.5 The hammer and you

Strategizing with a foreknowledge of how to best incorporate XML, Web services, and service-oriented design principles into the various domains that make up your automated enterprise, however, will put you on a path at the end of which lies a sophisticated and adaptive automation environment. It will allow you to transition toward an integrated enterprise with superior data sharing and unprecedented control of your corporate business models.

This guide is your map. The strategies, recommendations, and best practices provided here collectively form a framework that offers direction and guidance through the twists and turns along the road to building service-oriented architecture and, ultimately, a service-oriented enterprise. So, grab your hammer and enjoy the ride!

1.2 The XML & Web Services Integration Framework (XWIF)

As an independent consultant, I've worked for many companies. More often than not, I've been part of projects that involved highly complex environments, unique problems, and difficult integration issues. Solutions frequently required an element of creativity that ventured beyond traditional mindset boundaries. Many of the ideas expressed in this guide, therefore, may be new to you, and hopefully will provide you with alternative perspectives to common integration problems.

The contents of this book are part of an integration framework that I've been developing for a number of years through my company, XMLTC Consulting Inc. The XML & Web Services Integration Framework (XWIF) consists of an enterprise standardization strategy, supported by a series of best practices, integration strategies, and processes for planning and delivering service-oriented integration projects.

Each piece of this collective intelligence is designed with the common goal of transitioning an organization toward a service-oriented enterprise.

Table 1.1 An Overview of XWIF

XML & Web Services Integration Framework		
Best practices	Standards	Processes
Strategies	Service Models	Roles

Much of what XWIF preaches is a common-sense approach to resolving typical integration issues with XML and Web services. Some of the guidelines provided are in use elsewhere in the industry, whereas others are unique to XWIF. Many best practices and

strategies, and all processes in this guide were developed exclusively as part of this framework.[1]

NOTE
This book isn't about XWIF, nor does it discuss this framework in any detail. Much of the information provided here was borrowed from XWIF and assembled into this generic field guide. To learn more about XWIF, visit `www.xwif.com`.

1.3 How this guide is organized

The Field Guide is different from most IT books; it doesn't prescribe to the traditional tutorial or process-oriented formats. This guide provides a collection of strategies and best practices that have one common theme: the integration of XML, Web services, and service-oriented architecture.

As I put this book together, it became evident that I had to include background information on the many technologies it discussed. Therefore, the three chapters in Part I contain a series of lightweight tutorials for the primary technologies that form contemporary XML and Web services architectures.

Since the guide is intended for a range of IT professionals that no doubt will be using it under different circumstances, there was no perfect way of organizing its many topics. After a number of iterations, I decided to categorize the XWIF strategies and best practices into Parts II, III, and IV, as illustrated in Figure 1.1.

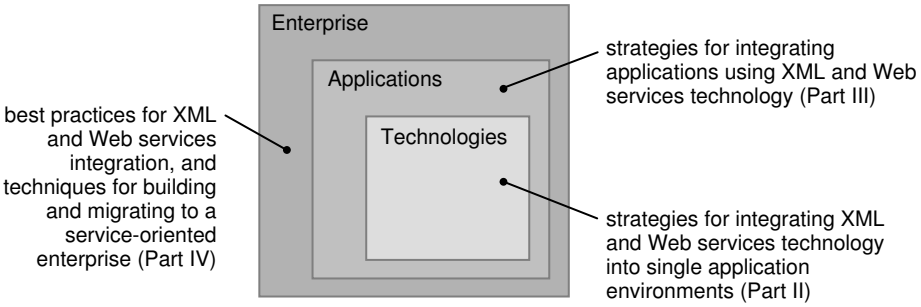

Figure 1.1
Three categories representing Parts II, III, and IV of this guide.

1. Even though the framework is occasionally referenced throughout this book, I've avoided prefixing every single strategy, process, and best practice with "XWIF."

Here is a quick reference overview of what is covered in each part and chapter.

1.3.1 Part I: The technical landscape

Nearly 30 XML and Web services specifications are discussed throughout this part of the book, with a focus on the 17 core standards listed in Table 1.2.

Table 1.2 Reference Matrix of Technology Tutorials

XML technologies (Chapter 2)	Web services technologies (Chapter 3)	Second-generation Web services technologies (Chapter 4)
XML	WSDL	WS-Coordination
DTD	SOAP	WS-Transaction
XSD	UDDI	BPEL4WS
XSLT		WS-Security
XQuery		WS-ReliableMessaging
XPath		WS-Policy
		WS-Attachments
		WS-Addressing

Note that Chapter 3 also covers numerous concepts relating to the Web services framework, including:

- requestor and provider roles
- intermediaries
- initial sender and ultimate receiver roles
- message paths
- message exchange patterns
- correlation
- choreography
- activities

Also note that Chapter 3 introduces service-oriented architecture (SOA) concepts. Later, Chapter 14 continues this discussion with a detailed tutorial on SOA design principles.

1.3.2 Part II: Integrating technology

As shown in Figure 1.2, Part II confines the scope of topics to single application environments in order to focus on strategies for integrating technology within application tiers.

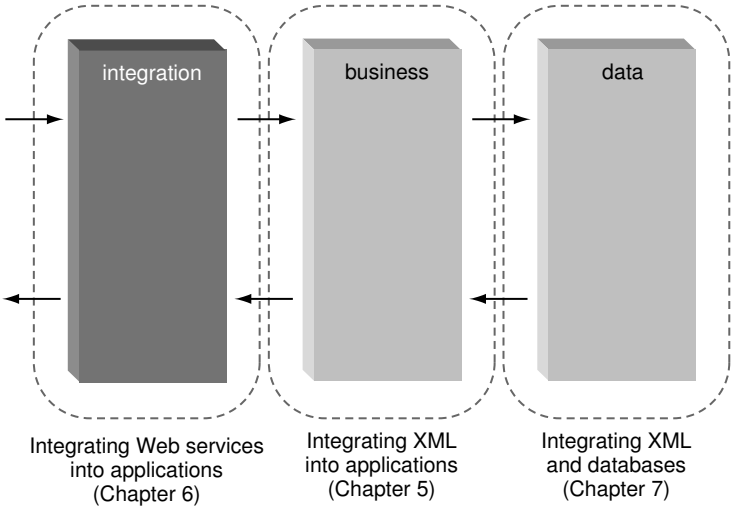

Figure 1.2
The three chapters in Part II roughly correspond to the three backend tiers of a distributed application architecture.

Within Part II we clearly separate integration issues relating to XML and Web services. This allows XML strategies to be used independently of service-oriented environments, if required.

Integrating XML into applications (Chapter 5)
The scope of this chapter is the integration of core XML technologies with the purpose of establishing a fundamental data management architecture. Numerous strategies for addressing common integration issues are provided, organized according to the functional areas most likely to be affected by the integration, as follows:

- XML data representation strategies for conceptually and technically incorporating XML as a data representation format and delivery mechanism
- XML data validation strategies that explore the utilization of schema definitions, with an emphasis on XSD
- an XML schema administration process that highlights the importance of centralizing ownership of XML schema definitions

- XML transformation strategies that cover the integration of XSLT for structural and aesthetic transformation of XML documents
- XML data querying strategies that position XQuery as a technology to centralize and abstract data access logic

These sections assume you have a base knowledge of the discussed technologies. If you don't, you should study the tutorials in Chapter 2 first.

Integrating Web services into applications (Chapter 6)
Here we focus on fundamental design concepts that allow you to establish a foundation for a service-oriented architecture, and prepare an application for future interoperability. The scope of this chapter, therefore, is limited to the integration of Web services technology within application environments.

The following XWIF service models are established:

- utility services
- business services
- controller services

XWIF also supplies us with these two modeling processes:

- modeling service-oriented component classes
- modeling Web service interfaces

A collection of integration and optimization strategies are provided next, addressing the use of service assemblies, Web services performance, and interface design.

Integrating XML and databases (Chapter 7)
XML opens up a whole new world of data modeling that contrasts traditional approaches to structuring and defining schemas. This chapter is dedicated to exploring techniques for integrating the hierarchical structure of XML documents with traditional relational repositories. Since this is a common area for which knowledge and resources typically are limited, we explore issues with more syntactical detail than in other chapters.

First, we compare XML and relational databases in order to establish their fundamental differences and to contrast how each platform relates to and manages data. Next, the basics of data mapping are covered, as well as issues relating to performance and platform

disparity. We then get into the details of mapping the hierarchical XML data model to relational databases.

Finally, we take a look at the common ways in which current database products support XML through the use of proprietary extensions. The implications of using these extensions, along with some techniques on how to mitigate their impact, also are provided. This chapter concludes with an overview of native XML databases.

1.3.3 Part III: Integrating applications

We now carry the discussion forward into the realm of application integration. The chapters in Part **III** are almost entirely centered around the use of Web services, as numerous traditional and service-oriented integration scenarios are explored and contrasted. Figure 1.3 illustrates the scope of Part **III** chapters, as they relate to application tiers.

The mechanics of application integration (Chapter 8)
Fundamental integration concepts are introduced, and the differences between traditional and contemporary application integration architectures are discussed. This chapter is more of a primer for cross-application integration, as it also provides a guide to middleware products, and explores common paths for evolving an enterprise integration environment.

If integration architecture is new to you, I recommend you read through this chapter prior to proceeding with Chapters 9 and 10.

Service-oriented architectures for legacy integration (Chapter 9)
Here we dive into the multi-varied world of legacy integration architectures. We begin by describing the following set of XWIF services models:

- proxy services
- wrapper services
- coordination services (for atomic transactions)

We then explain the roles of common integration components, including:

- legacy adapters
- intermediary services
- service interceptors

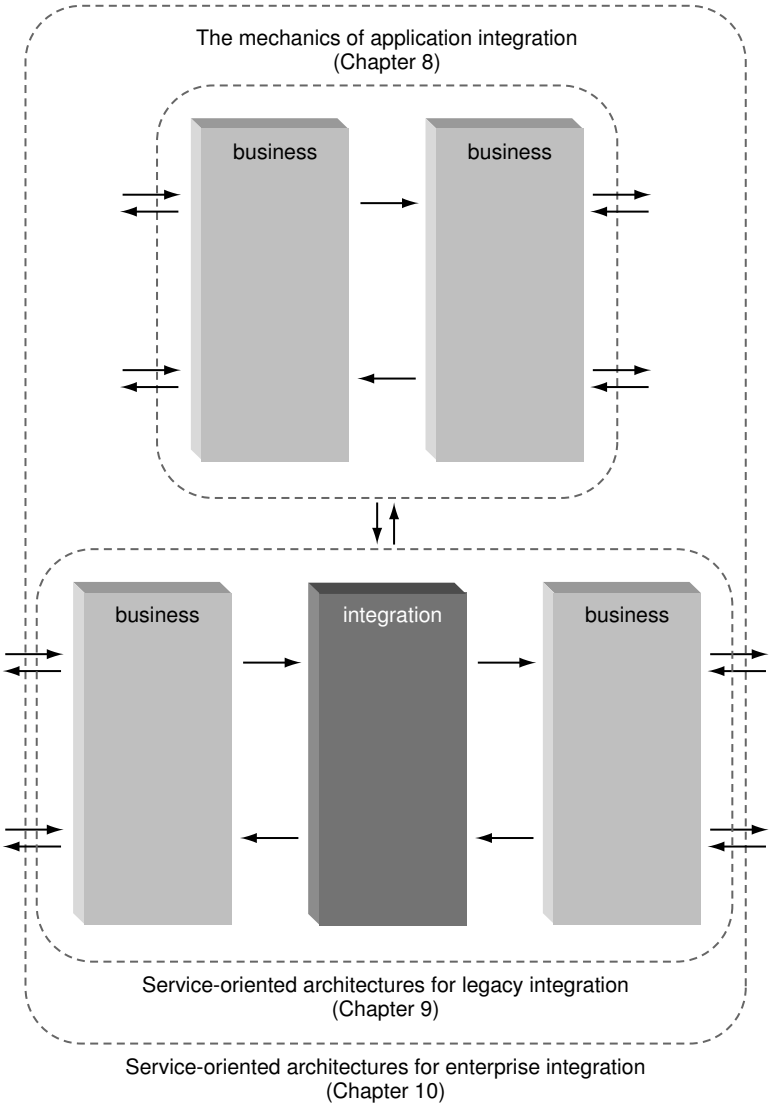

Figure 1.3
The architectural scopes of Part III chapters.

Many legacy integration architectures are then explored. Each of the following sections separately illustrates an integration architecture in a traditional and service-oriented state:

- one-way integration architecture: batch export and import
- one-way integration architecture: direct data access

- point-to-point architecture: tightly-coupled integration between homogenous legacy applications
- point-to-point architecture: tightly-coupled integration between heterogeneous applications
- point-to-point architecture: integration between homogenous component-based applications
- point-to-point architecture: integration between heterogeneous component-based applications
- centralized database architecture

These sections are supplemented further with architectural comparison matrices that contrast key architectural aspects within traditional and service-oriented contexts. The chapter concludes with an analysis process for assessing the feasibility of introducing service-oriented design principles within legacy architectures.

Service-oriented architectures for enterprise integration (Chapter 10)
The scope now broadens to encompass enterprise integration architectures. The roles of broker and orchestration components are demonstrated, along with the introduction of the following XWIF service models:

- process services
- coordination services (for business activities)

Both traditional and contemporary EAI architectures are then explored, including:

- hub and spoke
- messaging bus (publish and subscribe)
- enterprise service bus (ESB)

How Web services can be integrated within these environments is illustrated, and supplemented with numerous design considerations.

Service-oriented integration strategies (Chapter 11)
To supplement and expand on the topics covered in Chapters 9 and 10, the following collection of integration strategies are provided:

- strategies for streamlining integration endpoint interfaces
- strategies for optimizing integration endpoint services

- strategies for integrating legacy architectures
- strategies for enterprise solution integration
- strategies for integrating Web services security

1.3.4 Part IV: Integrating the enterprise

This part of the book provides a clear roadmap to a standardized service-oriented enterprise, and consists of a collection of best practices and processes for planning and implementing an enterprise-wide integration strategy.

Here's a brief overview of the chapters:

Thirty best practices for integrating XML (Chapter 12)
Chapter 12 describes a wide variety of best practices and recommendations for various aspects of XML integration, including:

- planning XML migration projects
- knowledge management within XML projects
- standardizing XML applications
- designing XML applications

Thirty best practices for integrating Web services (Chapter 13)
Chapter 13 details a set of best practices that provide guidance and insight for managing and integrating Web services. The following areas are covered:

- planning service-oriented projects
- standardizing Web services
- designing service-oriented environments
- managing service-oriented development projects
- implementing Web services

Building the service-oriented enterprise (SOE) (Chapter 14)
Our last chapter contains perhaps the most important information within this guide. First, it delves into the details of service-oriented modeling concepts and design principles. A detailed tutorial breaks down the components of a service-oriented architecture into activities, services, and processes.

It then applies these concepts to a service-oriented enterprise by establishing a series of business modeling and technology building blocks for the design of service-oriented

environments. Examples are provided, establishing problems that are then solved using these service-oriented design principles.

Finally, the XWIF Layered Scope Model (LSM) is introduced, establishing a comprehensive migration strategy for the controlled integration of XML and Web services technologies through a series of phases that gradually transition an organization toward the service-oriented enterprise. This last section draws upon information provided throughout the Field Guide, by listing the XWIF service models, processes, and strategies most appropriate for each LSM phase.

1.3.5 The extended enterprise

Service-oriented concepts and architecture allow an enterprise to be extended beyond its organizational boundaries. The enterprise standardization process in Chapter 14 identifies a migration path into the extended enterprise; however, the XWIF business-to-business interchange model is not an area of integration we cover in this guide.

1.4 www.serviceoriented.ws

Updates, samples, a glossary of terms, and various other supporting resources can be found at `www.serviceoriented.ws`. I am interested in your feedback. Any experiences you'd like to share, or suggestions you may have as to how I can continue to improve this book would be much appreciated.

1.5 Contact the author

To contact me directly, visit my bio site at `www.thomaserl.com`.

About the Author

Thomas Erl is an independent consultant with XMLTC Consulting in Vancouver, Canada. His previous book, *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, became the top-selling book of 2004 in both Web Services and SOA categories. This guide addresses numerous integration issues and provides strategies and best practices for transitioning toward SOA.

Thomas is a member of OASIS and is active in related research efforts, such as the XML & Web Services Integration Framework (XWIF). He is a speaker and instructor for private and public events and conferences, and has published numerous papers, including articles for the *Web Services Journal*, *WLDJ*, and *Application Development Trends*.

For more information, visit `http://www.thomaserl.com/technology/`.

About SOA Systems

SOA Systems Inc. is a consulting firm actively involved in the research and development of service-oriented architecture, service-orientation, XML, and Web services standards and technology. Through its research and enterprise solution projects SOA Systems has developed a recognized methodology for integrating and realizing service-oriented concepts, technology, and architecture.

For more information, visit `www.soasystems.com`.

One of the consulting services provided by SOA Systems is comprehensive SOA transition planning and the objective assessment of vendor technology products.

For more information, visit `www.soaplanning.com`.

The content in this book is the basis for a series of SOA seminars and workshops developed and offered by SOA Systems.

For more information, visit `www.soatraining.com`.