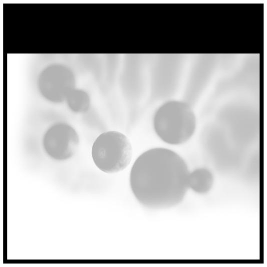
Business Culinary Architecture
Computer General Interest
Children Life Sciences Biography
Accounting Finance Mathematics
History Self-Improvement Health
Engineering Graphic Design
Applied Sciences Psychology
Interior Design Biology Chemistry

WILEY e BOOK

WILEY
JOSSEY-BASS
PFEIFFER
J.K.LASSER
CAPSTONE
WILEY-LISS
WILEY-VCH
WILEY-INTERSCIENCE



Quick CORBA™ 3

Jon Siegel

Wiley Computer Publishing

John Wiley & Sons, Inc.
NEW YORK • CHICHESTER • WEINHEIM • BRISBANE • SINGAPORE • TORONTO

In loving memory of my dad, Armand Siegel—
professor, teacher, thinker.

Publisher: Robert Ipsen
Editor: Robert M. Elliott
Assistant Editor: Emilie Herman
Managing Editor: John Atkins
Associate New Media Editor: Brian Snapp
Text Design & Composition: Publishers' Design and Production Services, Inc.

Published by John Wiley & Sons, Inc.

This title is also available in print as ISBN 0-471-38935-8

For more information about Wiley products, visit our web site at www.Wiley.com

OMG Press Advisory Board

OMG Press Books in Print

(For complete information about current and upcoming titles, go to www.wiley.com/compbooks/omg/)

- *Building Business Objects* by Peter Eeles and Oliver Sims, ISBN: 0471-191760.
- *Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise* by Peter Herzum and Oliver Sims, ISBN: 0471-327603.
- *Business Modeling with UML: Business Patterns at Work* by Hans-Erik Eriksson and Magnus Penker, ISBN: 0471-295515.
- *CORBA 3 Fundamentals and Programming, 2nd Edition* by Jon Siegel, ISBN: 0471-295183.
- *CORBA Design Patterns* by Thomas J. Mowbray and Raphael C. Malveau, ISBN: 0471-158828.
- *Enterprise Application Integration with CORBA: Component and Web-Based Solutions* by Ron Zahavi, ISBN: 0471-32704.
- *Enterprise Java with UML* by C.T. Arrington, ISBN: 0471-386804
- *The Essential CORBA: Systems Integration Using Distributed Objects* by Thomas J. Mowbray and Ron Zahavi, ISBN: 0471-106119.

- *Instant CORBA* by Robert Orfali, Dan Harkey, and Jeri Edwards, ISBN: 0471-183334.
- *Integrating CORBA and COM Applications* by Michael Rosen and David Curtis, ISBN: 0471-198277.
- *Java Programming with CORBA*, *Third Edition* by Gerald Brose, Andreas Vogel, and Keith Duddy, ISBN: 0471-247650.
- *The Object Technology Casebook: Lessons from Award-Winning Business Applications* by Paul Harmon and William Morrisey, ISBN: 0471-147176.
- *The Object Technology Revolution* by Michael Guttman and Jason Matthews, ISBN: 0471-606790.
- *Programming with Enterprise JavaBeans*, *JTS*, *and OTS: Building Distributed Transactions with Java and C*++ by Andreas Vogel and Madhavan Rangarao, ISBN: 0471-319724.
- *Programming with Java IDL* by Geoffrey Lewis, Steven Barber, and Ellen Siegel, ISBN: 0471-247979.
- *UML Toolkit* by Hans-Erik Eriksson and Magnus Penker, ISBN: 0471-191612.

About the OMG

The Object Management Group (OMG) was chartered to create and foster a component-based software marketplace through the standardization and promotion of object-oriented software. To achieve this goal, the OMG specifies open standards for every aspect of distributed object computing from analysis and design, through infrastructure, to application objects and components.

The well-established Common Object Request Broker Architecture (CORBA) standardizes a platform- and programming-language-independent distributed object computing environment. It is based on the OMG/ISO Interface Definition Language (OMG IDL) and the Internet Inter-ORB Protocol (IIOP). Now recognized as a mature technology, CORBA is represented on the marketplace by well over 70 Object Request Brokers (ORBs) plus hundreds of other products. Although most ORBs are tuned for general use, others are specialized for real-time or embedded applications, or are built into transaction processing systems where they provide scalability, high throughput, and reliability. Of the thousands of live, mission-critical CORBA applications in use today around the world, more than 300 are documented on the OMG's success-story web pages at www.corba.org.

CORBA 3, the OMG's latest release, adds a Component Model, quality-of-service control, a messaging invocation model, and tightened integration with the Internet, Enterprise Java Beans, and the Java programming laguage. Widely anticipated by the industry, CORBA 3 keeps this established architecture in the forefront of distributed computing, as will a new OMG specification integrating CORBA with XML. Well-known for its ability to integrate legacy systems into your network, along with the wide variety of heterogeneous hardware and software on the market today, CORBA enters the new millennium prepared to integrate the technologies on the horizon.

Augmenting this core infrastructure are the CORBAservices that standardize naming and directory services, event handling, transaction processing, security, and other functions. Building on this firm foundation, OMG Domain Facilities standardize common objects throughout the supply and service chains in industries such as Telecommunications, Healthcare, Manufacturing, Transportation, Finance/Insurance, Electronic Commerce, Life Science, and Utilities.

The OMG standards extend beyond programming. OMG specifications for analysis and design include the Unified Modeling Language (UML), the repository standard Meta-Object Facility (MOF), and XML-based Metadata Interchange (XMI). The UML is a result of fusing the concepts of the world's most prominent methodologists. Adopted as an OMG specification in 1997, it represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems and is a well-defined, widely accepted response to these business needs. The MOF is OMG's standard for metamodeling and metadata repositories. Fully integrated with UML, it uses the UML notation to describe repository metamodels. Extending this work, the XMI standard enables the exchange of objects defined using UML and the MOF. XMI can generate XML Data Type Definitions for any service specification that includes a normative, MOF-based metamodel.

In summary, the OMG provides the computing industry with an open, vendor-neutral, proven process for establishing and promoting standards. The OMG makes all of its specifications available without charge from its Web site, www.omg.org. With more than a decade of standard-making and consensus-building experience, the OMG now counts about 800 companies as members. Delegates from these companies convene at week-long meetings held five times each year at varying sites around the world to advance OMG technologies. The OMG welcomes guests to their meetings; for an invitation, send your email request to info@omg.org.

Membership in the OMG is open to end users, government organizations, academia, and technology vendors. For more information on the OMG, contact OMG headquarters by phone at +1 (781) 444-0404, by fax at +1 (781) 444-0320, by email at info@omg.org, or on the Web at www.omg.org.

Contents

Introduction

This book collects descriptions of the Object Management Group's most powerful new CORBA and modeling specifications into a single volume. These specifications combine to give CORBA the capabilities it needs to support advanced Internet and enterprise computing. It includes almost all of the ten specifications originally labeled CORBA 3 (most of which have already been issued by the OMG as part of various CORBA 2.X point releases), and adds important new ones including the XML/Value mapping that represents an XML document as a multiply linked list of CORBA `valuetype`s. It also includes descriptions of OMG's modeling standards: the Unified Modeling Language (UML), Meta-Object Facility (MOF), and XML Metadata Interchange (XMI). With these infrastructure and modeling specifications, OMG defines the state of the art in distributed enterprise and Internet computing.

There are two groups of people who should read this book:

First, if you are an experienced CORBA programmer or architect, curious about OMG's newest specifications, this is the book for you. As I just mentioned, it describes the new CORBA specifications that you need to know about, plus OMG's modeling specifications. Since the book does not include an introduction to basic CORBA, it is an "advanced" book even though the presentation of each specification is no more advanced than the presentation in my introductory book *CORBA 3 Fundamentals and Programming*. It provides experienced CORBA users with a book that discusses OMG's new specifications without yet another introduction to IDL, or how IIOP and the IOR let a CORBA client invoke an operation on another vendor's CORBA server over the network.

Second, if you are an enterprise architect and are considering using CORBA for your infrastructure, even if you haven't studied basic CORBA, you can benefit from this book as well but in a different way. The specifications described in this book give CORBA the capabilities it needs to play in the the enterprise and Internet worlds—in fact, they put it in the forefront. You will be able to understand the capabilities of the specifications from the descriptions and figure out how they fit into your enterprise architecture. But you will find the technical (as opposed to architectural and functional) parts of the descriptions opaque, so you should go back and study basic CORBA after you're done to fill in the blanks. As much as possible, I've kept the technical and programming aspects towards the end of each chapter so that architects can read farther into each chapter before coming to programming detail. (On the other hand, if you're a programmer and you find the beginning of a chapter a little thin technically, this is your clue to skip towards the end!)

If you're new to CORBA, you'll need to read another book first unless you're only interested in the capabilities and architecture of the specifications and not in programming details. If you're a programmer and you haven't studied basic CORBA, perhaps now's the time. You need to learn how OMG Interface Definition Language (OMG IDL) separates interface from implementation, how the CORBA Object Reference makes invocations location transparent, and other aspects of the CORBA infrastructure. My previous book, *CORBA 3 Fundamentals and Programming*, is a good place to learn this.

If you're an experienced CORBA programmer or distributed-systems architect, you may want to skip through this book and read the most interesting (to you!) parts first. Except for XML/Value and the CORBA Component Model (CCM), there are few dependencies from one chapter to another. (XML/Value in Part I, Chapter 2 depends on the `valuetype` presented in Part I,Chapter 1, and all of Part III builds to the presentation of the CCM in its final chapter.)

Here's how it's organized:

The book starts with a **Prologue**, which pulls together the many parts of CORBA 3 from an architectural point of view, covering both client and server sides. The prologue also lists all of the specifications originally included in the original CORBA 3 press release, plus the others presented in this book, and tells when each was adopted and formally issued by OMG as part of a numbered CORBA release. (More on this below.)

Part I covers **Integration with Java and the Internet**, including the `valuetype`, the reverse Java-to-IDL mapping, the XML/Value mapping, which lets you manipulate XML documents (including those described by DTDs) as a collection of native CORBA types, and the Interoperable Naming Service.

Part II covers **Quality-of-Service Control**, including asynchronous and messaging mode remote invocations, setting policies for timeout and priority control, and the three specialized CORBA modes: Real-Time, Fault Tolerant, and minimumCORBA for embedded systems.

Part III leads up to and presents the **CORBA Component Model (CCM).** The first two chapters present the Portable Object Adapter (POA) and Persistent State Service (PSS). The last presents the CCM itself, starting with a programming example and moving on to a description of all of its parts.

Part IV presents **OMG's modeling specifications**, starting with UML and the MOF, and finishing with OMG's new Model Driven Architecture (MDA). A recent extension to OMG's basic architecture (as I write this, OMG members have just voted to authorize drafting of technical documents that would enable this extension to be adopted as the organization's direction), the MDA defines services and facilities as a platform-indendependent UML model first, which it then maps to the various platforms for either implementation or interoperability. The end of this book, this chapter represents a new beginning for OMG, which soon expects to be adopting specifications for facility interfaces, based on platform-independent UML models, on a variety of middleware platforms such as EJB, XML/SOAP, COM/DCOM/MTS, .Net, and others.

Even though OMG may not have released a volume labeled CORBA 3 when you first pick this book up in your bookstore, the specifications described here are real. As already mentioned, the label CORBA 3 was first used in a 1998 press release from OMG declaring that 10 specifications—including most of the ones in this book, of course—would be issued, together, under that label sometime during 1999. All of the specifications were adopted on schedule or nearly so, but formal release (a subsequent step which requires a maintenance revision of the original adopted specification, and a vendor commitment to produce and market an implementation) occurred in stages, rather than all-at-once. The first few (including the important `valuetype`, foundation for the XML/Value mapping) were formally issued with CORBA 2.3 in June 1999; several more as part of CORBA 2.4 in late 2000. As this book goes to press, it appears likely that others will be issued as part of a CORBA 2.5 point release around mid-2001 (although this is developing too late for mention anywhere else in this book except here!), preceding the formal release of the few remaining (mainly the Persistent State Service and CORBA Component Model) as CORBA 3.0 late in the year. The CCM is being implemented in quite a number of places; check out www.ditec.um.es/~dsevilla/ccm/ for details. (After the page comes up, search for "CCM implementations.")

I've collected information on release dates (mostly in the past!) for all of these specifications and put them in the Prologue, where they appear both in the text and in a table.

Some of the material in this book was taken from my previous book, *CORBA 3 Fundamentals and Programming*, published in mid-2000 by John Wiley and Sons. These chapters are new: The XML/Value mapping which lets you manipulate an XML document as a collection of native CORBA types, Fault Tolerant CORBA, minimumCORBA for embedded systems, and OMG's new Model Driven Architecture. The discussion of the Interoperable Naming Service was updated to take into account a maintenance revision to the specification, as was the chapter on Real-Time CORBA.

Thank you for picking up this book. It collects my descriptions of the most powerful new CORBA specifications into a single slim (well, compared to my other books!) volume. I hope you enjoy reading it as much as I enjoyed putting it together.

Jon Siegel
Director, Technology Transfer
Object Management Group

Acknowledgments

It's not possible to acknowledge all of the people who contributed to this book. It's based on the OMG specifications, and hundreds of skilled people from hundreds of companies around the world have participated in their creation. Five times every year, more than five hundred CORBA experts gather at a hotel somewhere in the world at an OMG meeting to advance OMG's suite of standards. I can't name them all here, but I can at least start by acknowledging my debt to these people for their fine work.

Enthusiastic support from the Object Management Group made this book possible. In particular, I want to thank OMG CEO Dr. Richard Soley for his good advice and support. Others around the OMG offices who provided support and encouragement include COO Bill Hoffman, Technical Director Andrew Watson, Liaison Director Henry Lowe, Director of Standards Fred Waskeiwicz, Dody Keefe, and the rest of the staff. However, all of the opinions in the book are mine, and neither anyone else's nor OMG's.

Many thanks to Dan Frantz (BEA Systems, Inc., www.bea.com) who wrote the technical and programming sections of the POA chapter. (I wrote the beginning sections and Santa Claus story myself, and don't want Dan to get blamed for them!) Also thanks to Ed Cobb (BEA Systems, Inc.), Patrick Thompson (Rogue Wave Software, Inc., www.roguewave.com), and Patrick Ravenel (Persistence Software, www.persistence.com), who wrote the code for the mini-example in the CORBA Components section of Part III

Chapter 3 and reviewed my text for that section so many times, they might as well have written some of it themselves. Patrick Ravenel also wrote the PSDL code and reviewed the PSS writeup in Part III, Chapter 2, and so gets double thanks. Michael Cheng of IBM Corporation (www.ibm.com) started out helping with the C++ mapping for valuetypes but, by the time he was done, had also contributed the example for the Java mapping and the RMI/IIOP code example. I had a lot of help with the XML/Value mapping chapter, and want to mention in particular Alan Conway (one of my co-authors on *CORBA 3 Fundamentals and Programming*) and Darach Ennis of IONA Technologies (www.iona.com), and Patrick Thompson (already mentioned for his help with the CCM chapter). Alan and Darach wrote the code for the dynamic mapping example and checked my IDL for the static mapping; Patrick contributed some essential and very timely help with the static mapping.

On the Analysis and Design side, thanks to Cris Kobryn of TeleLogic (www.telelogic.com) for contributing Part IV, Chapter 1 on UML. Cris is one of the authors of the specification, and he chairs the current UML Revision Task Force at OMG. Also, thanks to Sridhar Iyengar of Unisys, Inc. (www.unisys.com) for Part IV, Chapter 2 on the MOF and XMI. Sridhar is the principal author of these specifications, and known around OMG as "Mr. MOF." I'm proud to have these two authorities contribute to this book.

Many thanks to the OMG members who reviewed my writeups of the various specifications. These reviewers get credit for the consistent correctness of the descriptions, but I take all of the blame myself for any errors that might have crept in while they weren't looking. Dock Allen of The Mitre Corporation (www.mitre.com) reviewed the Real-Time description. Vic Giddings and Bill Beckwith of Objective Interface Systems (www.ois.com) helped with minimumCORBA. Louise Moser and Michael Mellior-Smith of Eternal Systems, Inc. (www.eternal-systems.com), principal authors of the Fault Tolerant CORBA specification, reviewed that chapter. And finally, Jishnu Mukerji of Hewlett-Packard Corporation (www.hp.com) and Dave Frankel of IONA Technologies reviewed the chapter on the MDA.

Since books tend to be written during evening, weekends, and even vacations, authors' families contribute noticeably to the success of a project such as this one. I want to thank my wife Nancy and our younger son Adam for their support and patience during the writing and editing of this book. Their help and forbearance made this work possible. Even our older son Josh, who is off on his own (working as a programmer, of course!), managed to help out here and there and gets his own "thank you" too.

This is an exciting time for CORBA and OMG. Even after the additions presented in this book, CORBA continues to grow. I just checked OMG's Web site where 15 technology adoptions (out of the 100 currently in process) will extend CORBA's capabilities in one way or another. Another 20-plus processes will add new CORBA-based facilities in OMG's domains. But the big news is about the Model Driven Architecture described in the last chapter, which provides big benefits to enterprise computing everywhere by extending OMG specifications to middleware platforms beyond CORBA. That chapter, at the end of this book, is the beginning of a new chapter for OMG. I hope you get a lot out of this book, from beginning to end.

Enjoy!

About the Author

Dr. Jon Siegel is Director of Technology Transfer for the Object Management Group (OMG) where he has worked since late 1993. At OMG, he heads OMG's technology transfer program with the goal of teaching the technical aspects and benefits of the Object Management Architecture including CORBA, the CORBAservices, the Domain specifications in vertical markets ranging from healthcare, life sciences, and telecommunications to manufacturing and retail systems, and OMG's modeling specifications UML, MOF, XMI, and CWM. In this capacity, he presents tutorials, seminars, and company briefings around the world, and writes magazine articles and books. With OMG since 1993, Siegel was founding chair of the Domain Technology Committee responsible for OMG specifications in the vertical domains.

Before joining OMG, Dr. Siegel performed research and development in Computer Science for Shell Oil Company where he championed the use of standards to reduce software development time and cost, and was an early developer of distributed object software. While at Shell, he served as its representative to the Open Software Foundation for four years, and to the Object Management Group for two years. He holds a Ph.D. in Theoretical Chemistry from Boston University.

Dr. Siegel lives in rural eastern Massachusetts, about 40 miles outside of Boston, with his wife Nancy and their younger son, Adam. His older son, Josh, left the family nest a few years back and is now working as a computer programmer (of course!). Siegel's hobbies, which he abandons while working on a book, include bicycling and astronomy.

Trademarks

CORBA®, The Information Brokerage®, CORBA Academy®, IIOP® and the Object Management Group logo® are registered trademarks of the Object Management Group. OMG™, Object Management Group™, the CORBA Logo™, "The Middleware That's Everywhere™" , the CORBA Academy logo™, XMI™, MOF™, CWM™, OMG Interface Definition Language™, IDL™, CORBAservices™, CORBAfacilities™, CORBAmed™, CORBAnet™, UML™, the UML Cube Logo, "We're Known By The Companies We Connect™", Model Driven Architecture™, MDA™, OMG Model Driven Architecture™, OMG MDA™, "The Architecture of Choice for a Changing World™", and Unified Modeling Language™ are trademarks of the Object Management Group.

Enterprise JavaBeans, Java, Java Server Pages, and JDBC are trademarks or registered trademarks of of Sun Microsystems, Inc., in the United States and/or other countries.

Microsoft, MS, MS-DOS, Windows, Windows NT, COM, DCOM, MTS, and .Net are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

PROLOGUE

CORBA and Distributed Computing Grow Up Together

In the 11 years since the Object Management Group (OMG) first defined the Common Object Request Broker Architecture (CORBA), the standard has moved from the periphery of computing to the preferred architecture for enterprise and internet server applications. Hundreds of mission-critical applications, documented at www.corba.org, attest to the suitability of CORBA for applications that must be scalable, reliable, transactional, and secure.

CORBA and distributed, networked computing grew up together. In the early 1990s when CORBA was young, the internet was still a curiosity used by a small number of scientists to exchange email and send files using ftp. Desktop computer users were more likely to exchange files via floppy disks than networks, which were expensive, unusual, and hard to use.

And, network programming was difficult. There were a few network programs—telnet, FTP, and SMPT—that satisfied basic needs. To go beyond this, programmers needed to code directly to sockets or some other primitive abstraction of the network. Even Remote Procedure Call (RPC) implementations were off in the future.

Following the OSF's Distributed Computing Environment (DCE) to market by only a year or two, basic CORBA added advanced features characteristic of object orientation: *Location transparency* divorced objects not only from their location on the network, but even from the need to be on the network at all: objects could just as easily be local to the client that

WHAT ABOUT BASIC CORBA?

You need to understand basic CORBA first (and rather well, in fact) in order to understand the CORBA 3 extensions that are the subject of this book. But, we're not going to cover these basic aspects here; this is an intermediate-level book written for the thousands of architects and developers who have already mastered basic CORBA and deserve a book that doesn't force them to wade through (and carry around!) chapters of stuff they already know. If you aren't familiar with basic CORBA, we suggest that you switch to our basic-level book, *CORBA 3 Fundamentals and Programming*, ISBN 0-471-29518-3 (John Wiley & Sons, 2000).

called them. *Cascaded calls* allowed applications to be built up from componentized building blocks. *Instantiation* allowed the creation of instances when they were needed, and their destruction when they were not. Interfaces defined in OMG *Interface Definition Language* enabled programming language independence even if the first language binding was for the distinctly non-object–oriented language, C.

In this environment, it was a virtue that CORBA hid networking details from the programmer almost totally. Distributed application programming was difficult enough. Starting slowly, programmers generated the first useful business CORBA applications nearly as soon as vendors provided them with the basic tools.

As the 1990s progressed, intranet and Internet computing grew up together. Following a brief period of experimentation with a number of incompatible proprietary protocols (does anyone out there remember DECNET?), TCP/IP emerged as the transport protocol of choice and, with the arrival of inexpensive and easy-to-install twisted-pair cabling, offices everywhere began to replace “sneakernet” with ethernet. Wide-area-network providers that started by coupling thermal-printing terminals via 300-baud modem connections to text-based mainframes morphed into high-speed infrastructures that coupled powerful desktops and laptops with color displays and high-bandwidth DSL or cable connections to the millions of servers on the World Wide Web.

Eveyone noticed the emergence of HTTP as the application protocol (that is, the protocol that ran over TCP/IP) of choice over the Internet link from client to server, since that was what users came into contact with. Less obvious was the protocol that linked together the numerous machines that comprised the business infrastructure, which grew too big to run on a single machine even if it was a huge mainframe.

SETTING THE STAGE

With the additions collectively known as CORBA 3, the suite of Object Management Group (OMG) specifications keeps up with enterprise requirements in distributed computing. These first few pages describe how networked computing and CORBA grew up together. We'll keep the history short, and get to the architectural overview in the introduction to Part I. This high-level overview will establish the context for the rest of the book.

With the formal adoption of the CORBA Component Model (CCM) in June 1998, all of the CORBA 3 specifications became adopted OMG technology. However, this did not mean that the OMG issued a CORBA 3 specification book right away. To ensure that implementing companies have a robust specification to work from, OMG requires all newly-adopted specifications to undergo their first maintenance cycle before they are given a release number and issued as a formal document. Since the specifications that make up CORBA 3 completed this mandatory maintenance step individually rather than all at once, their releases happened the same way: A few specifications were issued early as part of CORBA 2.2; a few more in releases 2.3 and 2.4; and finally the remainder—including the key CORBA Component Model—in CORBA 3.0 scheduled for release in late 2001. We review the details of this event sequence in this Prologue.

This is where CORBA predominated. In this heterogeneous world where scalability and reliability were crucial, CORBA provided a tight, reliable, and secure coupling that enabled even transactional reliability in applications that spanned the enterprise and, where necessary, beyond it. CORBA 2 applications scaled to thousands of transactions per second and powered Web sites in electronic commerce and financial trading in public view, and some of the largest manufacturing installations in less visible situations.

But by the end of the 1990s, networks were no longer the curiosity they had been, and network programming experience had become a common skill. This was fortunate, because the network had morphed from a simple wire that linked client and server to a key infrastructure component that enabled the linking of business units in powerful new forms of commerce.

As the network changed, CORBA had to change. The pure location transparency that was such a benefit when networks were new and unknown, became a hindrance to programmers when they tried to take advantage of network characteristics in new and powerful server designs. And, when networks became overloaded, taking advantage was replaced by coping as Quality of Service control became essential.

As networks allowed more users to access popular servers, programmers developed techniques that allowed these servers to scale to meet the need. Proprietary at first, by the end of the decade these techniques had become common enough to be standardized and they, too, became a part of the new CORBA, particularly through the Portable Object Adapter (POA) and CORBA Component Model (CCM). OMG decided to collect these advances in a major release they planned to label CORBA 3.

P.1 The Power of CORBA 3

There's a lot to CORBA 3 (as you know just from scanning the table of contents of this book); so much so that if you look only at the details, you can miss the point—a "forest-and-trees" effect. To avoid this, we'll start here with a broad view of the power of CORBA 3; once we're done, we'll use the rest of the book to back up the promises and assertions that we'll make in this section.

P.1.1 The CORBA 2.0/2.1 Client

We've already introduced the CORBA 2.0/2.1 client: It's a simple construct with no view of the network. It invokes and passes objects by reference only, and invokes synchronously using either the Static Invocation Interface (SII), using stubs compiled from IDL, or Dynamic Invocation Interface (DII), using stubs interpreted at runtime from the IDL. Some programmers used the DII to make *deferred-synchronous* invocations—a polling-mode invocation that required polling to retrieve results.

P.1.2 The CORBA 3 Client

In contrast, the CORBA 3 client has a very sophisticated view of the network, and of other distributed resources hidden from (or, in some cases, not yet available to) its CORBA 2 ancestor. This client:

- Can invoke either synchronously or asynchronously on any CORBA object, using either SII or DII
- Can make time-independent invocations—that is, operate in CORBA's new loosely coupled networking environment
- Can take advantage of CORBA's newly defined reliable network infrastructure
- Can negotiate a quality-of-service level with the network and a server

- Can invoke and pass CORBA objects by reference, and CORBA Valuetypes by value
- Can find services using URL-format object references
- Can invoke operations on RMI/IDL Java objects
- Can deal with firewalls
- Can use both Factory and Finder patterns to interact with CORBA components (and objects)
- Can navigate among the multiple interfaces of a CORBA component

Originally defined to let clients take advantage of local compute cycles while server operations executed remotely, asynchronous invocation plays a much more important role in enterprise computing: Combined with a reliable network infrastructure—that is, CORBA invocations over a messaging-like protocol—this mode allows CORBA clients and servers to operate in the loosely coupled way that inter-enterprise computing requires.

Other capabilities—URL-like object references, coping with firewalls, and interoperability with RMI/IDL Java objects—further integrate our client with the world of the Internet and of enterprise computing. And the last two capabilities—using Factory and Finder patterns, and navigating among a component's multiple interfaces—equip our client to deal with important new capabilities of CORBA 3 servers.

P.1.3 The CORBA 2.0/2.1 Server

Programmers have been writing huge, scalable, reliable CORBA servers for years (OMG has a collection of writeups on its success story Web site, www.corba.org). So CORBA's ability to support these servers has never been in question. What has been questioned, however, is the amount of low-level programming skill needed to produce these servers, and the difficulty of finding this skill combined with the business knowledge needed to make them function in the enterprise. Programming this kind of capability into a CORBA 2 server did, we admit, require a certain amount of skill. In addition, infrastructure support for making an object's internal state persistent, and for associating the persistent state with the executing instance, was not fully developed.

P.1.4 The CORBA 3 Server

The salient advancement in CORBA 3 on the server side is the simplification of the programming model. Embodied mainly in the CORBA Component Model specification, with help from the Persistent State Service (PSS), CORBA 3 defines an environment that lets every capable business developer produce servers with the characteristics required by enterprise computing. In addition to simplifying the programming model, CORBA 3 standardizes the mechanisms that make servers scale, and the interfaces to these mechanisims. The CORBA 3 server:

- Simplifies server programming with new declarative languages for server-side functions and characteristics
- Automatically generates, for CORBA Components, a type-specific home object that implements class-like methods including instance creation and destruction and client-visible identity, and provides a place for queries across the extent of a type
- Allows CORBA Components to bear multiple interfaces, and supports client navigation among them
- Integrates object persistence, transactionality, security, and event handling into the CCM as *run-time* constructs, rather than *coding-time* constructs, relieving business programmers of the responsibility for dealing with them
- Integrates with Enterprise JavaBeans (EJB) and is, as we'll show in Section III.3.3.13, a *superset* of EJB
- Supports many flexible patterns of resource control for *servants* (a servant is the executing image of a CORBA object instance), with differing qualities of transience or persistence
- Associates an ObjectID with the Object Reference, allowing a servant to find and retrieve its persistent state on activation, and properly store it on deactivation
- Can negotiate quality of service with the network and the client
- Can deal with firewalls
- Can be specialized for small embedded systems, or for Real-Time applications, in addition to its capabilities in large scalable server applications

Aimed at enterprise servers—CORBA's most demanding environment—these new capabilities make CORBA one of the biggest of the big boys on the enterprise end of the wire. Supported by new capabilities in persistence handling, and working with the server-side analogues of the networking functions we examined on the client side, the features on this list define a very capable programming and execution environment indeed. What else could we possibly want? How about applicability to specialized environments, provided by minimal (for embedded systems) and Real-time CORBA, mentioned in the last bullet. Already used in mission-critical (read "life and death") applications including fly-by-wire and chemical process control, Real-Time CORBA brings standardization to the previously proprietary world of distributed real-time systems. We'll describe some of the most exciting applications in Part II, Chapter 3.

P.2 The Many Parts of CORBA 3

CORBA 3 was pre-announced by the OMG in a September 1998 press release. The release predicted that 10 new distinct but related specifications would be adopted nearly simultaneously and issued together. As we will show in the rest of this book, all of the predicted specifications and several more closely related ones were in fact adopted and issued, although not together. In fact, the 10-plus specifications spread over three point releases (CORBA 2.2, 2.3, and 2.4) and the major release 3.0, which was scheduled for late 2001 as we went to press. We've summarized the release schedule for all of these specifications in Table P.1.

The press release divides the specifications into three categories. This division fits so logically that we've used it to organize this book, which is

Table P.1 Release of Specifications Originally Grouped as CORBA 3

RELEASE	DATE	INCLUDED SPECIFICATION	ORIGINALLY ADOPTED
CORBA 2.2	Feb. 1998	POA	June 97
		IDL-to-Java Mapping	July 98
CORBA 2.3	June 1999	Valuetypes	May 98
		Java-to-IDL Mapping	Aug. 98
CORBA 2.4	Late 2000	Messaging	Sept. 98
		Interoperable Naming Svc	Mar. 99
CORBA 3.0	Early 2001	CORBA Component Model	Late 99
		CORBA Scripting	May 99
		Real-Time CORBA	May 99
		Minimal CORBA	Nov. 98

divided into a Part for each category plus a final one describing OMG's modeling specifications. The three categories, and our additional Part, are:

- Improved integration with Java and the Internet, presented in Part I
- Quality of Service control, presented in Part II
- The CORBA Component Model, presented in Part III
- OMG's modeling specifications, presented in Part IV

Here's a closer look at each category, with a list of the specifications included in each one and where (and whether) it appears in this book.

P.2.1 Improved Integration with Java and the Internet

The 1998 announcement included three specifications in this category. We're not going to follow the original categorization in this book. In fact, we won't present two of the original specifications (for the reasons given below), but we've moved one here from the third category and added two new ones. We present this category in Part I. Here is our revised list of specifications in this category:

- *Objects passable by value* (`valuetype`s): The press release mentions this specification in the third category along with the CCM, but we've moved it to here where you'll find our presentation in Part I, Chapter 1. It definitely improves integration with Java, and also serves as the basis for the XML/Value mapping. It was formally released as part of CORBA 2.3.
- *Java-to-IDL Mapping*: Based in part on the `valuetype`, this mapping allows Java RMI objects to interoperate over the network like CORBA objects. They have CORBA object references and emit the IIOP protocol. The Java-to-IDL Mapping was mentioned in the press release in this category; we present it in Section I.1.4 right after `valuetype`s. This specification was also released as part of CORBA 2.3.
- *XML/Value Mapping*: This mapping of the popular data language XML to CORBA `valuetype`s wasn't even in the plans when the original CORBA 3 press release was issued, but because XML is so important to Internet commerce, we're including it in this book and in this category. It standardizes the representation of an XML document as a collection of native CORBA types. The specification lets you take advantage of a DTD if you have one, but works perfectly well for DTD-less (that is, dynamic) XML documents if you don't. In this book, we present the mapping along with a programming example in Part I, Chapter 2. This

specification is brand-new as we go to press; although adopted, it has not yet been issued as part of a numbered CORBA release.

- *CORBA Firewall Specification*: This specification will allow firewalls to be configured for CORBA, passing IIOP traffic when they're supposed to and keeping it out when they're not. The firewall specification, although adopted, was undergoing a major revision as we went to press, so we decided not to include a chapter on it even though the press release mentions it in this category. Check the OMG Web site for information as the work progresses, and look for our writeup in a later book.
- *Interoperable Naming Service*: Another specification not mentioned in the press release, this service is key to allowing newly discovered CORBA objects to be invoked over the Internet. So we've added it to this category, where you'll find it in Part I, Chapter 3. All of the provisions of the Interoperable Naming Service were added to CORBA core in release 2.4.
- *DCE/CORBA Interworking Specification*: This one is mentioned in the press release and passed its OMG Platform Technology Committee (PTC) vote in September 1998, but wasn't embraced enough by the marketplace to become a fully adopted specification or rate a chapter of its own in this book.

P.2.2 Quality of Service Control

In the press release, this category included three specifications. OMG adopted one more in this category since then, so we've added it to this section. This category is presented in Part II. It includes these specifications:

- *CORBA Messaging*: This comprehensive specification encompasses both asynchronous and messaging-mode invocations (which you'll find in Part II, Chapter 1) and Quality of Service Control (Part II, Chapter 2). This specification was added to CORBA in Release 2.4, where it appears as Chapter 22.
- *Real-Time CORBA*: This specification extends the CORBA architecture to give Real-Time applications the resource control they need to guarantee end-to-end predictability for distributed applications built on Real-Time operating systems, running in a controlled environment. We describe it in Part II, Chapter 3. This specification was added to CORBA in Release 2.4, where it appears as Chapter 24.
- *Fault Tolerant CORBA*: New since the press release, the Fault Tolerant CORBA specification standardizes redundant software configurations and systems that, when run on redundant hardware, give CORBA

the robust and reliable performance that Enterprise applications need. Our description appears in Part II, Chapter 4. More recent than Real-Time and minimum CORBA, this adopted specification has not yet appeared in a numbered release.

- *minimumCORBA*: Aimed at embedded and card-based systems, this specification defines a small-footprint CORBA configuration by omitting dynamic features (DII, interface repository) not needed in a predictable environment. minimumCORBA is presented in Part II, Chapter 5. This specification was added to CORBA in Release 2.4, where it appears as Chapter 23.

P.2.3 The CORBA Component Model

The press release mentions four specifications in this category. We've moved one, the **valuetype**, into the first category. Another, multiple interfaces for CORBA objects, was absorbed into the CCM specification itself, and we've absorbed a third one, CORBA scripting languages, into the CCM chapter even though these are distinct specifications. Still, there are three chapters in Part III of this book, and they describe:

- *The Portable Object Adapter* (POA): Added to CORBA in release 2.2, the POA forms the basis for scalable CORBA servers. The CCM Container is a specialized POA. We've devoted Part III, Chapter 1 to the POA to ensure that we've presented the background necessary to understand the CCM, coming up two chapters hence.
- *The Persistent State Service* (PSS): Not mentioned in the original announcement, the PSS is another important building block in the foundation of the CCM. Replacing the now-deprecated Persistent Object Service, the excellently designed PSS is used by the CCM Container to store the state of persistent objects when executing code is deactivated between calls. (It's also available for programmers to invoke directly, of course, in both CCM and non-CCM CORBA applications.) The PSS was adopted in late 1999 and is scheduled for inclusion in a formal CORBAservices release before the end of 2001, intentionally synchronized with the formal release of the CCM as part of CORBA 3. We present the PSS in Part III, Chapter 2.
- *The CCM*: Considered by many to be the essence of CORBA 3 (even though so many other important specifications were included in the original announcement!), the CCM packages up transactionality, persistence, event handling, security, and POA-based server resource control into a development and runtime package that business pro-

grammers can handle. Intentionally matched feature-for-feature with Enterprise JavaBeans (EJBs) as far as they go, the CCM fulfills the last promise made by the Java and Internet integration theme that started our three categories. The CCM was adopted by OMG in late 1999 and is scheduled for formal release before the end of 2001 as the numbered release CORBA 3. The CCM, and a brief description of CORBA scripting languages, comprise Part III, Chapter 3.

These three categories, and the specifications we've listed, include everything originally labeled CORBA 3. But because modeling plays a key role in the success of the large software projects that benefit most from CORBA, we've added a fourth part devoted to OMG's modeling specifications, including the world standard for representation of Analysis and Design, the Unified Modeling Language (UML).

P.2.4 OMG's Modeling Specifications

There are three chapters in this part. The first two cover OMG's existing modeling specifications, and the last looks ahead at what OMG members expect, as we go to print, to be OMG's future direction: the *Model Driven Architecture* (MDA). Part IV covers:

- *The UML*: Part IV, Chapter 1 was written by Cris Kobryn, one of the authors of the UML specification. It introduces the UML and presents an example based on the programming example in our companion book, *CORBA 3 Fundamentals and Programming.* Although Release 1.3 of UML was current as we went to press, Release 1.4 had just been completed and was working its way through OMG's voting process, and RFPs for a major revision to be designated UML 2.0 were scheduled to complete in late 2001.
- *The Meta-Object Facility* (MOF): Part IV, Chapter 2 was written by Sridhar Iyengar, principal author of the MOF specification. It describes this important foundation that defines the meta-models for almost all of OMG's specifications, and a standard repository for models and meta-models. Release 1.3 of the MOF was current as this book went to press, with Release 1.4 due soon.
- *The MDA*: This final chapter brings our book to a close with a look at OMG's future: the MDA. Basing future specifications on the UML, OMG members will build a library of standards that hold their value as the computing infrastructure under them changes over time. Retaining its requirement that standards be implemented as they are adopted,

OMG will further require that every UML model be mapped to and implemented in at least one middleware platform, which might be CORBA, Java/EJB, XML/SOAP, or some other existing or future platform. Part IV, Chapter 3 describes (in a general way, since work on the MDA is just getting started) how UML will model specific middleware environments as OMG-standard UML profiles, allowing standard mappings to define the route from a platform-neutral UML representation to a platform-specific set of interface specifications and, ultimately, the application itself.

P.3 OMG's Adoption Process and Timing

OMG's technology adoption process is defined in the group's policies and procedures. It starts with the issuance of a requirements document called a Request for Proposals (RFP), and ends its first phase with the adoption of a submission through a series of votes by various subgroups of members with varying (i.e., technical or business) interests. This part of the process typically takes 14 to 18 months, and is followed by an ongoing series of maintenance revisions of which the first is the most important (at least for our discussion here).

All OMG specifications are maintained evergreen by the group's members, and in particular by the companies that contributed the technology. During the last month or two before a new specification is adopted, its submission document is in flux as submitting companies attempt to merge their individual technologies into a "best of breed" common standard. In the haste to make deadline, there is not always time to check that these changes get all of the details right. As a result, the first maintenance revision of an OMG specification usually requires more than the usual amount of change in order to fix bugs that have become apparent during initial implementations.

Until October 1998, OMG had been calling the initial adoption of each of its specifications the 1.0 release, and the first maintenance revision the 1.1 release. Members and others who followed OMG closely knew to expect more than modest changes between the 1.0 and 1.1 releases, and to wait for the 1.1 release before coding or releasing a product based on a new specification. Just before the CORBA 3 specifications were adopted, however, OMG changed the designations of these first key releases. Instead of calling the first adoption the 1.0 release, it was designated an *adopted specification* and not given a release number. In spite of the lack of a release number, adopted specifications are fully adopted OMG technology and available to all on the group's Web site. But, to emphasize the implementability of the first revised version of a new specification, its release was designated an *available specification* and bore a release number.

Helpful to companies that otherwise might have invested time and money implementing preliminary versions of OMG specs, this change in labeling had the unfortunate side effect of lengthening the time between initial adoption of a new specification and its formal issuance with a release number. Although every specification had had an interval between adoption and availability of products, the market had previously been able to point to the issued (albeit buggy) OMG specification with its release number for reassurance that the new spec was really there.

This explains why, between the initial adoption of the CCM in late 1999 and the scheduled issuance of the formal CORBA 3 book with the revised specification in late 2001, CORBA 3 exists as a specification but not as a numbered release nor a product. During the longer interval starting with the issuance of the CORBA 3 press release in September 1998 and ending with the formal release of CORBA 3 book, the 10 specifications that ultimately constituted the specification finished their initial maintenance revisions at widely spaced intervals and became ready for issuance. Since it didn't make sense to hold technologies back from release once they were ready, OMG included them in a number of point releases during 1999 and 2000 as we listed in Table P.1.

We've included parts of CORBA 2.2 in the table, with the POA and IDL to Java mapping, because these specifications are the foundation for much of CORBA 3 even though they're not part of CORBA 3 itself.

CORBA 2.3 is the first formal release that includes specifications originally slated for CORBA 3: **valuetype**s and the reverse mapping of Java to IDL.

About six months later, CORBA 2.4 added Messaging and the Interoperable Naming Service.

CORBA 3.0—with the CCM, Scripting, Firewall, Real-Time, and Minimal CORBA—is scheduled for release in late 2001.