SIGS REFERENCE LIBRARY

Donald G. Firesmith
Editor-in-Chief

1. Object Methodology Overview, CD-ROM, Doug Rosenberg

2. Directory of Object Technology, edited by Dale J. Gaumer

3. Dictionary of Object Technology: The Definitive Desk Reference, Donald G. Firesmith & Edward M. Eykholt

4. Next Generation Computing: Distributed Objects for Business, edited by Peter Fingar, Dennis Read, and Jim Stikeleather

5. C++ Gems, edited by Stanley B. Lippman


7. Best of Booch: Designing Strategies for Object Technology, Grady Booch (edited by Ed Eykholt)


Additional Volumes in Preparation
Best of Booch

Designing Strategies for Object Technology

Grady Booch

Edited by
Ed Eykholt
To Jan: My friend, my lover, my wife
— Grady

To my daughter Suzanne
— Ed
ABOUT THE AUTHORS

Grady Booch, inventor of the Booch method for object-oriented software engineering, is Chief Scientist at Rational Software Corporation, headquartered in Santa Clara, California. He has been with Rational since its founding in 1980. He has contributed significantly to improving the effectiveness of software development worldwide, through his seminal work in object modeling, iterative development, and software architecture.

The Booch method is being used to develop some of the most complex and demanding software systems, including air-traffic-control systems, commercial aircraft avionics, financial trading systems, telecommunication switching networks, and defense systems. Additionally, Grady developed a family of reusable components in both Ada and C++, which helped to popularize software reuse and make it economically feasible.

Grady was also involved in the development of Rational Environment, the company’s original software-engineering environment and its compiler technology. Grady was also the original architect for the Rational Rose object-oriented analysis and design tool. He has worked in a variety of object-oriented and object-based languages, including Ada, C++, Smalltalk, Visual Basic, and Java.

He served as a consultant on the highly successful FS 2000 shipboard command-and-control system developed by CelsiusTech, one of Sweden’s leading developers of command-and-control systems. On this project, Grady worked with the project’s chief architect and trained many of the company’s senior developers in applying object-oriented technology. This project became the basis of a profitable line of business for CelsiusTech and, as a result of applying Grady’s methods and Rational’s technology, the company has been able to reuse up to 70 percent of its code from project to project.

Grady has consulted on numerous projects worldwide, helping some organizations develop client/server systems and assisting others in customizing their own object-oriented methods. Some of his customers include the U.S. Government, Alcatel, Andersen Consulting, AT&T, MCI, Microsoft, Price Waterhouse, UBS (a large Swiss bank), the Orient Overseas Container Line (based in Hong Kong), and Xerox. Grady was also the project manager
for the Range Safety Display System, a 500,000-line-of-code (LOC) application at Vandenberg Air Force Base. He was also the project engineer for a 1.5 million LOC real-time telemetry processing application there.

Grady’s skill as an innovative computer scientist is matched by his skill as synthesizer, author, and popularizer on topics of object technology. Grady considers himself an architect. His books, articles, and talks have led to a great increase in the awareness and application of object technology.


Grady has also published more than 85 technical articles on object-oriented technology and software engineering, which appear regularly appearing in the *Report on Object Analysis and Design* (ROAD) and *Object Magazine*. Grady consults and lectures on object-oriented topics throughout the world.

Grady’s experience in software engineering and OO method development paved the way to allow Jim Rumbaugh, Ivar Jacobson, and him to join forces with the express goal of unifying their respective methods (Booch, OMT, andOOSE/Objectory) into one unified definition of notation and semantics: the Unified Modeling Language (UML). The UML is expected to be embraced within the industry. Version one of the UML is anticipated to be published by early 1997.

As much as possible, Grady has remained accessible to professionals in the object-technology industry. He regularly communicates with a wide audience in the industry via e-mail, the comp.object newsgroup, and the OTUG mailing list.

Grady is a distinguished graduate of the United States Air Force Academy, where he received a B.S. in computer science in 1977. He received his M.S.
in Electrical Engineering / Computer Engineering from the University of California at Santa Barbara in 1979. He is a member of the ACM, the IEEE, Computer Professionals for Software Responsibility, and the Association for Software Design. Grady is an ACM Fellow and a Rational Fellow. He can be contacted by e-mail: egb@rational.com

**Ed Eykholt** (pronounced “eye-colt”) is a sales engineer with Rational Software Corporation in McLean, Virginia, where he provides technical sales support for the Rose product line of object-oriented software engineering tools and delivers training for OO tools and methods. He was instrumental in collating feedback on the Unified Modeling Language and coordinating its standardization. With Don Firesmith he coauthored the *Dictionary of Object Technology: The Definitive Desk Reference*.

Prior to joining Rational in mid-1995, Ed was a Business Systems Consultant with NCR/AT&T in Dayton, Ohio. Ed facilitated dialog between business and information-systems professionals in order to achieve consistent understanding and expectations between them, assuring that IS supported business strategies and requirements.

He has used object-oriented analysis and design, business process reengineering, and other techniques. He has significant experience in accounting systems and product management.

Ed received his Bachelor of Science degree in Electrical Engineering in 1985 and Master of Science in Management (MBA) in 1987 from Purdue University and its Krannert School of Management, West Lafayette, Indiana. He can be contacted by e-mail: eykholt@rational.com
FOREWORD

I greatly admire Grady Booch’s contributions to the software development community. One might look no further than what he presented in *Object Oriented Modeling and Design with Applications* and begin to appreciate his contributions. I submit, however, that the essence of the contribution is in understanding all of the complexities associated with software development, managing the risks, and delivering tangible, real-world results. My measure of real-world results is: “Did we deliver what the customer needed, on time and on budget?” Anything else is suboptimal. The Booch Method has been proven to do this, time and time again, in real projects with real deliverables in the real world.

Over the years Grady has written and spoken almost nonstop about the challenges facing those of us who call ourselves members of the software development community. Many of these talks, books, and papers have centered on the benefits of Object Technology and the Booch Method. Many others have dealt with topics less concrete and in some ways much more difficult to address or even acknowledge: risk awareness and management, balancing complexities, team dynamics, roles and responsibilities for project team members, architecture, rigor and discipline, patterns, reuse, and so on. Not only are these topics often missed by project teams, they are often ignored, leading to predictable results. I think you, the reader, will find some additional awareness from this book, or perhaps some reinforcement or embellishment for your software development practices.

At Xerox, our teams do not believe that object technology is a way to deliver systems. They believe it is the only way. Since establishing a competency center for OT in mid-1994, we have been evangelizing OT, the Booch method, a Methodology Framework, and a holistic, balanced set of work practices, architecture, and many of the topics listed previously. We have been working with Grady and Rational Software Corporation continuously to help us reengineer the process of delivering information systems to our customers. In fact, the “Booch book” is required reading for our software engineers. This book will join Booch’s other works on our bookshelves.

ix
Our progress is, like our software development process, iterative and incremental. For us, this work is never “done.” The “surprise” of the Internet/WWW is as poignant an example as one might need to validate this approach. Our purpose is to create information systems to support the business of Xerox Corporation. We firmly believe that the technological and cultural foundations we are building in our software development community are required for a successful future. Through our work, we will partner with the business community and rapidly respond to the changing business environment. We could not even dream of this without object technology. We have committed our future to OT. This is a risk we have acknowledged and are managing. Because you are reading this book, perhaps you have come to the same conclusions, or are just frustrated with limited success with conventional approaches. In any case, read on and enjoy.

Alan F. Nugent
Vice President and CTO,
Xerox Corporation,
GP&IM Global Strategy and Advanced Technology
PREFACE

This volume contains a collection of essays on various object-oriented topics that we hope you will find interesting. These essays are based on articles published since Grady Booch’s 1993 book, *Object Oriented Modeling and Design with Applications*, second edition.

Grady has written and spoken on a host of topics key to the success of the object technology industry. Obviously, Grady has written on the Booch Method, and how a complete method must address a model’s notation and semantics as well as a process for creating that model. As he has observed, a method is necessary, but not sufficient, for the success of a project. The success of an object-oriented project depends on many other factors, including business, project and team dynamics, architecture, artifacts, and implementation. Over the years, Grady has addressed all of these topics and they are represented here in this collection. This book is organized according to these categories. Furthermore, the topic of managing OO projects was the focus of Grady’s earlier book, *Object Solutions: Managing the Object-Oriented Project*, so we only include more recent articles on this topic here.

Because some of these articles date back as far as 1993, you may notice that some of Grady’s comments about the industry, for example, may not jibe with today’s landscape. We decided to leave many of these comments in as is for their historical perspective, recognizing how quickly the industry is evolving. In other areas, we felt it was especially helpful to adjust the Booch Method notation and semantics to reflect the current thinking in the Unified Modeling Language (UML). Some areas of the UML are still evolving, but most of the notation presented herein is stable, except where noted. As an editor of this book, Ed Eykholt revisited each article, adjusting it, if necessary, to assure its applicability and current thinking. Personal pronouns (I, me, my) should be interpreted as referring to Grady, with the exception of text appearing in this *sans serif* font, such as in the introductions for each chapter, or footnotes in brackets, [ ], which were written by Ed.

INTENDED AUDIENCE
This collection will interest software professionals who are concerned about the success of object-oriented software projects, including anyone with a
role on the project team, and students. Because the book is a collection of a broad set of topics, it need not be read in order nor in its entirety to be of value.

Many of the topics are independent of an object-oriented approach altogether, but other articles do assume a working knowledge of the OO concepts. Booch’s *Object Oriented Modeling and Design with Applications* is the primary reference text for this background. Because we have updated these articles to reflect the current thinking of the Unified Modeling Language, those interested in understanding how the Booch Method has evolved into the current UML definition will find this book valuable. The latest information on the UML can be found on Rational Software’s web site, http://www.rational.com.

We encourage you to read on about good things object oriented and the challenges facing those who search for them.
## CONTENTS

*About the Authors* ......................................................... v

*Foreword by Alan F. Nugent* ........................................... ix

*Preface* .................................................................. xi

1 **Managing Complexity** ............................................. 1
   Coming of Age in an Object-Oriented World ................. 2
   A Question of Balance .............................................. 4

2 **The Business of Object Technology** ......................... 11
   The Software Landscape ........................................... 12
   Software as a Strategic Weapon ................................ 21
   Objectifying Information Technology ....................... 26
   The Business Case for Class Libraries ....................... 33
   The Microsoft Effect ................................................ 36

3 **Process and Products** ............................................ 41
   The Macro Process .................................................. 43
   Measures of Goodness .............................................. 54

4 **Projects and Teams** ............................................... 63
   The Object-Oriented Project .................................... 64
   Growing Mature Abstractionists ................................ 70

5 **Models** .............................................................. 77
   Why We Model ...................................................... 79
   The Evolution of the Booch Method ............................ 84
   Next Generation Methods ....................................... 94
   Unification ........................................................... 101
   Quality Software and the Unified Modeling Language ... 107
   Scenarios ............................................................. 111
   Properties and Stereotypes ...................................... 122
   Finite State Machines ............................................ 129

6 **Architecture** ....................................................... 133
   The End of Objects and the Last Programmer ............... 135
   Conducting a Software Architecture Assessment .......... 144
   Distributed Systems ............................................. 151

xiii
Patterns ......................................................... 162
Patterns & Protocols ........................................... 167

7 Implementation ................................................. 179
The OO Languages ........................................... 180
The Design of Reusable Class Libraries ................. 183
Object-Oriented Development with Visual Basic ........ 191
You Need a Software Recycling Program ................. 196
Development of the Web, by the Web, and for the Web .... 200
Java ................................................................. 205

8 Looking Ahead .................................................. 213
References .......................................................... 217
List of Figures .................................................... 221
Notable Quotes .................................................... 223
Index ............................................................... 227

xiv