

Object-Oriented Analysis and Design with Applications, Third Edition

Grady Booch, Robert A Maksimchuk, Michael W. Engle, Bobbi J. Young, Ph.D., Jim Connallen, Kelli A. Houston

Object Oriented Analysis and Design with Applications (3rd ed.) is written by Grady Booch; Robert A. Maksimchuk; Michael W. Engle; Bobbi J. Young, Ph.D; Jim Conallen; and Kelli A. Houston. Published by Addison-Wesley, © 2007, ISBN 0-201-89551-X, 691 pages, \$64.99 US.

This 3rd edition is the eagerly and long awaited update to the 2nd edition which was published in 1994. It provides thorough and practical coverage of concepts, techniques, notations and examples for modern object-oriented analysis and design. The material covered draws upon a solid foundation of theoretical work but is consistently pragmatic in approach. This book provides an essential body of knowledge for professionals responsible for the analysis and design of complex systems.

As with the second edition, the book is organized into three major sections – Concepts, Method and Applications. Concepts introduces the fundamental principles of object-oriented analysis and design (OOAD) such as creating abstractions, objects and classes, and how to address the complexities found in a variety of systems. The Method section focuses on how to analyze and design complex systems with an emphasis on using UML 2. This section also gives some coverage of OOAD as part of a development cycle and from the project management perspective. Finally, the Applications section provides five in-depth examples from different domains which provide illustrative approaches to sets of problems commonly faced by practitioners.

This edition follows a similar format and addresses many of the same topics as its predecessor but varies in several areas. Most noticeably, the famous “clouds” and other Booch notations used in the 2nd edition have all been replaced with UML. The UML diagrams also make frequent use of the newer UML 2 notations such as frames on sequence diagrams and ports on component diagrams. The applications examples have also been updated. The weather monitoring and cryptanalysis examples remain but the class library and inventory tracking examples have been replaced with satellite-based navigation, traffic management and a vacation tracking system. As a whole, the new set of applications nicely cover a variety of challenges found in modern systems design. There are also fewer code examples in this edition. However, as the frequent use of Courier font suggests, the text still sits conceptually just slightly above code level when that is necessary.

This book is very well organized, written and edited. For example, in the Methods section, the chapter on Notation doesn’t merely plod through the syntax of various shapes and line styles but explains each diagram set with regard to intended use and contribution to object-oriented models. There are also clear and informative distinctions between essential techniques and more advanced concepts. In addition to the substantive case studies in

the Applications sections, there are effective supporting examples used throughout the text. Though some concepts clearly build upon each other, the reader is not forced to read the material in a certain order – e.g., references to material in other chapters are clearly marked and summarized. The bibliography is nicely organized into major topics, there are extensive end-notes by chapter, and foot notes are used as needed.

Diagram styles vary somewhat from chapter to chapter but, as explained in the preface, this is deliberate in order to familiarize the reader with the output of commonly used tools. These variances in style also help one discern between superficial and substantive differences in notation. The process material, both in the Methods section and in the organization of the Applications examples, naturally draws from the Rational Unified Process (RUP). However, RUP is used effectively to convey ideas and not in a way that would prevent practitioners of another process from applying the core OOAD material being discussed. More extensive or sharply defined distinctions between what is common practice as compared to alternative approaches, including the risks or benefits of either, would have been nice to have but this omission doesn’t detract from what is a great book overall.

Though intended primarily for developers and architects of software systems, the material presented would also be highly valuable to analysts in non-engineering roles such as business systems analysts. It is also a worthwhile read for those working on systems without a software emphasis. Though obviously focused on analysis and design activities, it is an excellent supplemental text for requirements gatherers, developers and project managers.

Analysts, designers and architects of complex systems, will find this text provides broad and deep coverage in the current practice of OOAD. As a result, it should be regarded as mandatory reading for professionals in those fields.

Reviewed by Brian A. Lawler, Association of Computing Machinery and Johns Hopkins University, brian.lawler@acm.org and brian.lawler@jhu.edu.