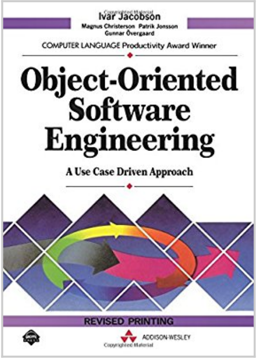

Object Oriented Software Engineering: A Use Case Driven Approach

By Ivar Jacobson

Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson

How can software developers, programmers and managers meet the challenges of the 90s and begin to resolve the software crisis? This book is based on Objectory which is the first commercially available comprehensive object-oriented process for developing large-scale industrial systems. Ivar Jacobson developed Objectory as a result of 20 years of experience building real software-based products. The approach takes a global view of system development and focuses on minimizing the system's life cycle cost. Objectory is an extensible industrial process that provides a method for building large industrial systems. This revised printing has been completely updated to make it as accessible and complete as possible. New material includes the revised Testing chapter, in which new product developments are discussed.

Object Oriented Software Engineering: A Use Case Driven Approach

By Ivar Jacobson

Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson

How can software developers, programmers and managers meet the challenges of the 90s and begin to resolve the software crisis? This book is based on Objectory which is the first commercially available comprehensive object-oriented process for developing large-scale industrial systems. Ivar Jacobson developed Objectory as a result of 20 years of experience building real software-based products. The approach takes a global view of system development and focuses on minimizing the system's life cycle cost. Objectory is an extensible industrial process that provides a method for building large industrial systems. This revised printing has been completely updated to make it as accessible and complete as possible. New material includes the revised Testing chapter, in which new product developments are discussed.

Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson Bibliography

- Sales Rank: #716811 in Books
- Published on: 1992-06-30
- Original language: English
- Number of items: 1
- Dimensions: 9.50" h x 1.68" w x 7.24" l, .0 pounds
- Binding: Paperback
- 552 pages

Download and Read Free Online Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson

Editorial Review

Amazon.com Review
A text on industrial system development using object- oriented techniques, rather than a book on object-oriented programming. Will be useful to systems developers and those seeking a deeper understanding of object orientation as it relates to the development process.

Review
Perhaps the most profound and deeply revealing volume on object technology to date...It is simply a must-own book. -- Steve Bilow -- *Journal of Object-Oriented Programming*

From the Inside Flap
This is a book on industrial system development using object-oriented techniques. It is not a book on object-oriented programming. We are convinced that the big benefits of object orientation can be gained only by the consistent use of object orientation throughout all steps in the development process. Therefore the emphasis is placed on the other parts of development such as analysis, design and testing.

You will benefit from this book if you are a system developer seeking ways to improve in your profession. If you are a student with no pervious experience in development methods, you will learn a robust framework which you can fill with details as you take part in future development projects. Since the focus on the text is on development, the book will be convenient to use in combination with other texts on object-oriented programming. Many examples illustrate the practical application of analysis and design techniques.

From this book you will get a thorough understanding of how to use object orientation as well as the basic technique throughout the development process. You will learn the benefits of seamless integration between the different development steps and how the basic object-oriented characteristics of class, inheritance and encapsulation are used in analysis, construction and testing. With this knowledge you are in a much better position to evaluate and select the way to develop your next data processing system.

Even though object orientation is the main theme of this book, it is not a panacea for successful system development. The change from craftsmanship to individualization does not come with the change to a new technique. The change must come on a more fundamental level which also includes the organization of the complete development process. Objectory is one example of how this can be done.

This book does not present Objectory. What we present is the fundamental ideas of Objectory and a simplified version of it. In this book we call this simplified method OOSE to distinguish it from Objectory. To use the process in production you will need the complete and detailed process description which, excluding large examples, amounts to more than 1200 pages. Introducing the process into an organization needs careful planning and dedication. It also requires that the process be adapted to the unique needs of the organization. Such process adaptations must of course be carefully specified, which can be done in a development case description, as will later be explained.

It is our hope that we have reached our goal with this book, namely to present a coherent picture of how to use object-orientation system development in a way which will make it accessible both to practitioners in the field and to students with no previous knowledge of system development. This has been done within a framework where system development is treated as an industrial activity and consequently must obey the same requirements as industry in general. The intention is to encourage more widespread use of object-oriented techniques and to inspire more work on improving the ideas expounded here. We are convinced that using these techniques will lead to better systems and a more industrial approach to system development.

Part I: Introduction. The book is divided into three parts. The first part covers the background, and contains the following chapters: System development as an industrial process

The system life cycle

What is object-orientation? Object-oriented system development Object-oriented programming

This part gives an introduction to system development and summarizes the requirements of an industrial process. It also discusses the system life cycle. The idea of object orientation is introduced, and how it can be used in system development and during programming is surveyed.

Part II: Concepts. The second part is the core of the book. It contains the following chapters: Architecture Analysis Construction Real-time specialization Database specialization Components Testing

The first chapter in this part introduces the fundamental concepts of OOSE and explains the reason why these concepts are chosen. The following chapter discuss the method of analysis and construction. The next two chapters discusses how the method may be adapted to real-time systems and database management systems. The components chapter discusses what components are and how they they can be used in the development process. Testing activities are discussed in a chapter of their own.

Part III: Applications. The third and last part covers applications of OOSE and how the introduction of the new development process may be organized and managed. This part ends with an overview of other object-oriented methods. This part comprises: Case study: warehouse management system Case study: Telecom Managing object-oriented software engineering Other object-oriented methods

Appendix. Finally we have an appendix which comments on our development of Objectory.

So, how should you read this book? Of course, to get a complete overview, the whole book should be read, including the appendix. But if you want to read only selected chapters the reading cases below could be used.

If you are an experienced object-oriented software engineer, you should be familiar with the basics. You could read the book as suggested in Figure P.1.

If you are a newcomer to object-orientation and software engineering you could read the book as in Figure P.2

If you are an experienced software engineer you could read the book as in Figure P.3

If you are a manager you could read the book as proposed in Figure P.4. Although the book is not object-oriented, it is written in a modularized way and can be configured in several different ways. Building systems in this way is the theme of the book, and the technique and notation used above is very similar to the technique used in this book.

A short history and acknowledgments

The work presented in this book was initiated in 1967 when I proposed a set of new modeling concepts (notation with associated semantics) for the development of large telecommunication switching systems. The main concepts were signals and blocks. A real-time system is an open system communicating with its environment by signals alone. A signal models the physical stimulus/response communication which a concrete system has when interacting with the outside world. Given a signal as input, a system performs internal actions such as executing algorithms, accessing internal information, storing results and sending output signals to the environment. This view presents the system in a very abstract way - as a black box. A less abstract view on a lower level models the system as a set of interconnected blocks. Blocks are modules which can be implemented in hardware or software or any combination of both. A block communicates with its environment only through signals. Signals between two blocks are internal, whereas signals modeling physical communication, that is, signals between a block and the environment of the system, are external. Internal signals are messengers conveying data from one block to another within the same system. All entries of a block were labelled and constituted the signal interface of that block, to be specified in a separate interface document. Hence the system can now be viewed as a set of interconnected blocks jointly offering the functions of the system. Each block has a program which it obeys on a receipt of an input signal, performing internal actions, that is, executing algorithms, storing and accessing block internal information, and sending internal and external signals to the environment.

The proposal can be summarized as an attempt to unify long experience of systems design with the possibilities offered by dramatically new computer technology. Since the two technologies were so different, this was not a self-evident method, neither within Ericsson nor within computer science. There was a rather strong belief that the two represented unrelated technological universes: the new one was so different that it would be meaningless and only a burden to make any attempt to learn from the old one. However, the two techniques were joined and a set of modeling concepts evolved.

The modeling constructs were soon followed by the skeleton of a new design method, the use of which was first demonstrated in the development of the AKE system put into service in Rotterdam in 1971, and more completely demonstrated in the AKE system put into service in Fredhall, Sweden, in 1974. Naturally this experience has guided subsequent work on the development of the successor to AKE, the AXE system, which is now in use in more than 80 countries worldwide. The modeling constructs were very important and, for the AXE system, a new programming language and a new computer system were developed in accordance to these early ideas.

Although it is a neighbouring country, the early development of object-oriented programming and Simula in the 1960s in Norway was done independently and in parallel with our work. It was not until 1979 that we "discovered" object-oriented programming and then it was in terms of Smalltalk. Although object-oriented ideas have influenced our recent work, basically two separate problems are being solved: "large-scale" and "small-scale".

The modeling constructs introduced during the 1960s were further formalized in research taking place between 1978 and 1985. The research resulted in a formally described language which offered support for object-orientation with two types of object and two types of communication mechanism, send/wait and send/no-wait semantics. The language supported concurrency with atomic transactions and a special semantic construction for the handling of events similar to the use case construct presented later. This work, reported in a PhD thesis in 1985, resulted in a number of new language constructs, initially developed from experience, being refined and formalized. This was a sound basis from which to continue and, taking a new approach, develop the method. The principles of Objectory were developed in 1985-7. I then further refined and simplified the ideas, generalized the technique used in the telecom applications, extended it with the inheritance concept and other important constructs like extensions, and coupled to it an analysis technique and object-oriented programming.

Today these concepts have been further redefined. The Objectory process, of which this book describes some fundamental ideas, is the result of work by many individuals, most of whom today work at Objective Systems SF AB, Sweden. Gunnar Overgaard and Patrik Jonsson did much of the writing of the first process description of Objectory analysis and design, respectively. Magnus Christerson did much to condense and rewrite the material into the form of this book. They have all contributed to Objectory; especially the formalization of the concepts. Magnus has also related the ideas of Objectory to other areas as presented in this book. Fredrik Lindstrom has also been involved in the condensation of the material of this book. Agneta Jacobson, Bud Lawson amd Lars Wiktorin have prepared material for some of the chapters.

Marten Gustafsson has substantially contributed to the analysis part of Objectory. Valuable contributions to Objectory have also been made by the following people: Sten-Erik Bergner, Per Bjork, Ann Carlbrand, Hakan Dyrhage, Christian Ehrenborg, Agneta Jacobson, Sten Jacobson, Mikael Larsson, Fredrik Lindstrom, Lars Lindroos, Benny Odenteg, Karin Palmkvist, Janne Pettersson, Birgitta Spiridon, Per Sundquist, Lars Wetterborg and Lars Wiktorin. The following users of Objectory have also contributed by feeding back experiences and ideas to enable improvements: Staffan Ehnebom, Per Hedfors, Jorgen Hellberg, Per Kilgren, Haken Lidstrom, Christian Meck, Christer Nilsson, Rune Nilsson, Goran Schefte, Fredrik Stromberg, Karin Villers, Stefan Wallin and Charoltte Wranne. The following persons have done a lot to support the tehnology described in this book: Kjell S. Andersson, Hans Brandtberg, Ingemar Carlsson, Hakan Dahl, Gunnar M. Eriksson, Bjorn Gullbrand, Lars Hallmarken, Bo Hedfors, Barbara Hedlund, Hakan Jansson, Christer Johansson, Ingemar Johnsson, Kurt Katzeff, Rolf Leidhammar, Jorma Mobrin, Jan-Erik Nordin, Anders Rockstrom, Kjell Sorme, Goran Sundelof, Per-Olof Thysselius, Ctirad Vrana and Erik Ornulf. The following people have given me strong personal inspiration and support: Dines Bjorner, Tore Bingefors, Dave Bulman, Larry Constantine, Goran Hemdal, Tom Love, Nils Lennmarker, Lars-Olof Noren, Dave Thomas and Lars-Erik Thorelli. In Sweden we do not normally thank family and friends in these circumstances, but no one beleives that results like these can be achieved without exceptional support from them. We are also grateful to the support we have been given from STU (Swedish National Board iof Industrial Development, now recognized to NUTEK) through the IT-4 program which has been part of the financial support and sponsorship for the writing of this book.

Changes to this revised printing, apart from minor general corrections and improvements, are: The testing chapter has been restructured and in parts rewritten, also an emphasis on early testing has been added. The discussion of robust object structures have been increased and also an example has been added. We hope this will better clarify why such an object structure gives more robust systems. The notion of a development case have been introduced as a way to adapt a general process to the specific needs of an organization or a project. Some people we would like to thank were unfortunately left out in the first printing and have now been added to the acknowledgment section, particularly Dave Bulman and Nils Lennmarker who have inspired the technology presented in this book.

0201544350P04062001

Users Review

From reader reviews:

Lori Morgan:

The book Object Oriented Software Engineering: A Use Case Driven Approach can give more knowledge and also the precise product information about everything you want. Why then must we leave the good thing like a book Object Oriented Software Engineering: A Use Case Driven Approach? A number of you have a different opinion about reserve. But one aim that book can give many details for us. It is absolutely correct. Right now, try to closer together with your book. Knowledge or data that you take for that, you can give for each other; you may share all of these. Book Object Oriented Software Engineering: A Use Case Driven Approach has simple shape however you know: it has great and massive function for you. You can search the enormous world by open up and read a publication. So it is very wonderful.

Troy Jones:

Information is provisions for individuals to get better life, information presently can get by anyone on everywhere. The information can be a information or any news even a problem. What people must be consider whenever those information which is within the former life are challenging be find than now is taking seriously which one is suitable to believe or which one the actual resource are convinced. If you obtain the unstable resource then you buy it as your main information you will have huge disadvantage for you. All of those possibilities will not happen in you if you take Object Oriented Software Engineering: A Use Case Driven Approach as the daily resource information.

Jean Gaitan:

As we know that book is significant thing to add our knowledge for everything. By a guide we can know everything you want. A book is a list of written, printed, illustrated or maybe blank sheet. Every year ended up being exactly added. This reserve Object Oriented Software Engineering: A Use Case Driven Approach was filled about science. Spend your time to add your knowledge about your technology competence. Some people has diverse feel when they reading a new book. If you know how big good thing about a book, you can feel enjoy to read a guide. In the modern era like at this point, many ways to get book which you wanted.

Samantha Smith:

Reading a guide make you to get more knowledge as a result. You can take knowledge and information from the book. Book is composed or printed or outlined from each source which filled update of news. Within this modern era like now, many ways to get information are available for an individual. From media social including newspaper, magazines, science book, encyclopedia, reference book, book and comic. You can add your understanding by that book. Are you ready to spend your spare time to spread out your book? Or just seeking the Object Oriented Software Engineering: A Use Case Driven Approach when you essential it?

Download and Read Online Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson #JLFVN0I6A3H

Read Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson for online ebook

Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson books to read online.

Online Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson ebook PDF download

Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson Doc

Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson Mobipocket

Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson EPub

JLFVN0I6A3H: Object Oriented Software Engineering: A Use Case Driven Approach By Ivar Jacobson