

---

## Preface

Object-oriented design is common in computing systems development. There is a plethora of techniques and an abundance of texts on the subject. Why, then, a book on *validated* designs for object-oriented systems? Our experience with specification, modelling and analysis techniques over the last decade tells us that object-oriented design is much more than mere “boxes and arrows.” Systems developers should be able to combine techniques and tools to achieve designs that are not merely the subject of heated argument but can be improved by careful, rigorous and machine-supported analysis. This book describes an approach to the design of object-oriented systems which combines the benefits of abstract modelling with the analytic power of formal methods to give designs that can be rigorously validated and assured with automated support.

System modelling has a valuable contribution to make in the development of computing systems. It encourages abstract, systematic and comprehensive consideration of system aspects, including the description of functionality, interactions with the environment and temporal behaviour. The rising level of interest in modelling is evidenced by the central role being taken by model-driven development approaches within the software engineering communities, such as the model-driven architecture work by the Object Management Group.

Formal methods can bring further tangible benefits to systems and software development, including objectivity and rigour of analysis, without compromising abstraction. However, these benefits can only be realised in practice if formal techniques are carefully targeted. In particular, formal techniques cannot be applied in isolation but must work alongside other analysis and design tools. The successful application of these techniques is regularly reported. This book stands in the tradition of “Lightweight” formal methods. Its aim is to equip readers with the ability to take advantage of formal modelling techniques without necessarily having to deploy the whole panoply of formal methods. We show how a formal notation (VDM++) can be used to complement and enhance object-oriented class models, with the engineer free to move between the class structure view and the functional view in VDM++. The approach shows how formal techniques can be used to take practitioners on from where they are, rather than requiring a radical shake-up of existing practice.

Although still relatively novel, this synthesis of object-oriented design and formal methods is a credible technology which has been applied in a wide variety of industrial projects. The style and content of the book are intended to bear this out. The elements of formal model-oriented specification, basic types, collections, abstract specification of functionality and system structuring are illustrated by examples derived from industrial experience. The material in the text is supported by the use of industry-strength tools. Many exercises are woven into the text, and more will be made available on the accompanying Web site (<http://www.vdmbook.com>).

### Structure of the Text

Part I provides motivation for modelling systems and enhancing models with descriptions of functionality. The introduction discusses the role of modelling in the development process and tool support. The reader's first glimpse of VDM++ comes in Chapter 2 when a realistic model of a chemical plant alarm management system is presented. Chapter 3 sets up the tool support that the reader will use throughout the remainder of the book.

Part II shows how the modelling approach is realised in a specific language, VDM++. Chapters 4 and 5 introduce the basics of modelling data and functionality in an object-oriented design. Chapters 6 to 8 introduce the key abstractions for modelling collections and relationships. The technical content is introduced through examples based on real applications of the technology (robots maneuvering around obstacles and congestion monitoring in road transport).

Part III is in many respects the core of the book. It situates the modelling approach in the industrial context by means of three case studies concentrating on different aspects of the application of modelling technology. Chapter 9 shows the use of modelling abstractions in understanding the architecture and operation of the famous Enigma cipher machine. Importantly, it also shows how established testing approaches can be applied in the context of a formal object-oriented design. Chapter 10 uses a continuous train speed monitoring function as an example to illustrate the links between the levels of detail in a formal model in VDM++ and the associated design-level class diagrams in UML. Chapter 11 gives an in-depth discussion of a real trading management ("back-office") system, including a review of the pros and cons of the modelling approach.

Part IV moves forward from the production of a model to deal with more advanced topics in the development process. Chapter 12 introduces the facilities for handling concurrency in VDM++. Chapter 13 looks at the facilities for gaining confidence in the accuracy of a model, both by simulation and by systematic consistency checking, a partly automated process. Chapter 14 discusses the implementation of models in Java.

### Using the Book

The book is aimed at software engineers with some prior experience in object-oriented design/programming, including intermediate or advanced students and researchers

studying object-oriented design. Readers are assumed to have some experience of programming in an imperative and object-oriented language such as Java or C++. Some familiarity with the basic concepts of class, attribute, method and inheritance is assumed. However, no specialised knowledge of formal methods is expected.

Readers new to object-orientation, for example those wishing to study formal modelling in conjunction with design in UML and programming in Java, can still benefit from the book. Such readers may omit the later parts of Chapter 9, all of Chapters 10 and all of Part IV on a first reading, revisiting these when greater confidence has been gained in programming.

### **Background to the Book**

This book is a product of a very long line of research dating to the development of the Vienna Definition Language (VDL) in IBM's Vienna Laboratory in the 1970s. Originally targeted at language definitions and compiler development VDL and its successor the Vienna Development Method (VDM) gradually found a wider range of applications. Seminal texts, notably those by Dines Bjørner and Cliff Jones [Jones80, Bjørner&82, Jones90], presented, in a formal context, many notions which are mainstream now, such as pre- and postcondition specification, data and operation refinement. VDM's specification language, VDM-SL, achieved ISO standardisation of both its syntax and formal semantics in 1996.

From the 1990s, two of us (John Fitzgerald and Peter Gorm Larsen) were engaged with the development of formal methods strong enough for application outside specialised areas. We both had experience at applying such methods in industry, and we both wanted to bring this experience to bear on a subject that was widely regarded as arcane and irrelevant to professional practice. The collaboration led to a text [Fitzgerald&98] which introduced modelling principles using a VDM-SL subset and which was driven by examples of industrial application and supported by the commercial tools developed by IFAD A/S. Although the VDM-SL work was successful in a range of applications, it had already become apparent that the link between plain formal modelling and object-oriented design could be bridged. Nico Plat, Paul Mukherjee and, later, Marcel Verhoef, had been applying the extended object-oriented version of VDM, called VDM++, initially developed through the European Commission's Afrodite project and then in a wide range of commercial and research projects and in the development of tool support, again at IFAD.

This text represents a synthesis of those strands of work. Its aim is to present an achievable improvement to existing practice through the use of formal design techniques in the context of mainstream object-oriented design, and to demonstrate this through realistic case studies inspired by industrial practice.

### **Accompanying Web Site**

Supporting material for the book, including VDMTools, manuals and further exercises is available from a variety of sources, many over the Web. The book's supporting website <http://www.vdmbook.com> acts as a portal, providing links to all the relevant sites.

## Acknowledgements

We are grateful to many wonderful colleagues whose work has made it possible to produce this book, and all those whose patience we have tried in the process. We apologise in advance to any we have inadvertently omitted from our list.

IFAD A/S of Odense in Denmark developed the first industrial-strength tools for VDM and VDM++. We are profoundly grateful to all our colleagues at IFAD over many years for the opportunity to explore the potential of this technology.

Our work is constantly inspired by the experiences of practitioners in system design and formal methods. We are grateful to the technology users who have provided inspiration for the case studies reported in the book. We would particularly like to thank the Japan Future Information Technology and Systems Co. Ltd., and in particular Mr. Shin Sahara, for kindly providing information on the TradeOne development experience.

We have had positive and professional support from Beverly Ford, Catherine Drury, Michael Koy, Frank McGuckin, their colleagues and reviewers at Springer-Verlag during the production cycle for the book. We gladly acknowledge the help of many colleagues who commented on drafts of the text: Bernhard Aichernig, Ian Bayley, Barrett Bryant, Ian Cottam, Dave Hastings, Neil Henderson, Jozef Hooman, Kevin Lano, Mikhail Lebedev, Richard Moore, Luis Neves, Erik Toubro Nielsen, Oliver Opitz, Stephen Paynter, Alexander Petrenko, Daan Rijsenbrij, Shin Sahara, Kim Sørensen, John Turner and Georg Weissenbacher.

Our respective institutes and employers have been very supportive during the production of the book. John Fitzgerald is grateful to Transitive Ltd. and the School of Computing Science at the University of Newcastle upon Tyne; Peter Gorm Larsen and Paul Mukherjee thank Systematic Software Engineering A/S; Nico Plat thanks West Consulting B.V.; Marcel Verhoef thanks Chess Information Technology B.V.

At a personal level, we are deeply grateful to our families and friends for their perseverance since the genesis of this book several years ago, to John Hudson, Michel Overbeeke, Marcelle van Valkenburg, and in particular to Yvonne Mukherjee and Margit Sandvang Larsen for their great hospitality during those many writing “conferences” in Odense!

Newcastle upon Tyne, UK  
Århus, Denmark  
Fort Worth, Texas, USA  
Rotterdam, The Netherlands  
Dordrecht, The Netherlands

*John Fitzgerald*  
*Peter Gorm Larsen*  
*Paul Mukherjee*  
*Nico Plat*  
*Marcel Verhoef*

September 2004

Validated Designs for Object-oriented Systems

Fitzgerald, J.; Larsen, P.G.; Mukherjee, P.; Plat, N.;  
Verhoef, M.

2005, XII, 404 p. 65 illus., Hardcover

ISBN: 978-1-85233-881-7