



# Model-Based Requirements Engineering

Jon Holt, Simon A. Perry  
and Mike Brownsword

# Model-Based Requirements Engineering

Jon Holt, Simon A. Perry  
and Mike Brownsword

The Institution of Engineering and Technology

Published by The Institution of Engineering and Technology, London, United Kingdom  
The Institution of Engineering and Technology is registered as a Charity in England & Wales  
(no. 211014) and Scotland (no. SC038698).

© 2012 The Institution of Engineering and Technology

First published 2012

This publication is copyright under the Berne Convention and the Universal Copyright Convention. All rights reserved. Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may be reproduced, stored or transmitted, in any form or by any means, only with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publisher at the undermentioned address:

The Institution of Engineering and Technology  
Michael Faraday House  
Six Hills Way, Stevenage  
Herts, SG1 2AY, United Kingdom

[www.theiet.org](http://www.theiet.org)

While the author and publisher believe that the information and guidance given in this work are correct, all parties must rely upon their own skill and judgement when making use of them. Neither the author nor publisher assumes any liability to anyone for any loss or damage caused by any error or omission in the work, whether such an error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

The moral rights of the author to be identified as author of this work have been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

#### **British Library Cataloguing in Publication Data**

A catalogue record for this product is available from the British Library

**ISBN 978-1-84919-487-7 (paperback)**

**ISBN 978-1-84919-488-4 (PDF)**

Typeset in India by MPS Ltd, a Macmillan Company  
Printed and bound by CPI Group (UK) Ltd, Croydon, CR0 4YY

**IET PROFESSIONAL APPLICATIONS OF COMPUTING SERIES 9**

# **Model-Based Requirements Engineering**

## Other volumes in this series:

- |          |   |
|----------|---|
| Volume 1 | <b>Knowledge discovery and data mining</b> M.A. Bramer (Editor)                                       |
| Volume 3 | <b>Troubled IT projects: prevention and turnaround</b> J.M. Smith                                     |
| Volume 4 | <b>UML for systems engineering: watching the wheels, 2nd edition</b> J. Holt                          |
| Volume 5 | <b>Intelligent distributed video surveillance systems</b> S.A. Velastin and<br>P. Remagnino (Editors) |
| Volume 6 | <b>Trusted computing</b> C. Mitchell (Editor)   |
| Volume 7 | <b>SysML for systems engineering</b> J. Holt and S. Perry   |
| Volume 8 | <b>Modelling enterprise architectures</b> J. Holt and S. Perry  |

This book is dedicated to Jude, Eliza and Roo.

JDH

To Hannah, Austin, Jacob, Isabella, Beatrice, Grace and Lillian.  
With love from Uncle Jolly.

SAP

To my wonderful wife, Jean.

MJB

---

## Acknowledgement

---

Wise men talk because they have something to say. Fools talk because they have to say something.

*Plato*

As I write more books I find that the list of people who should be acknowledged grows longer, whilst the list of people that I actually mention becomes shorter.

The information in this book is the result of more than 12 years of work, from when I first realized what a use case actually *was*, rather than what everyone else thought it was. There have been so many people who contributed to this work over the years that thanking them seems impossible, but there are a few stalwarts who deserve a mention: Duncan and his colleagues, Dipesh and Mick, Rick and his people at the Defence Academy, all at INCOSE, Pemby and his analogy, all at the people from the IET PN for systems engineering (Ian, Andrew, Doug, Steve, Michael, and the rest of the crew), my BCS colleagues and, of course, all (rather some) of my colleagues at Atego. Thanks to Kay, who drew the original Brontosaurus picture, based on my rubbish sketch, that we still use to this day.

Special thanks go to Lisa and her team at IET publishing for putting up with us for 10 years now!

As ever, I cannot thank Mike and Sue enough for all that they have done for me over the years, both professionally and personally. Rather worryingly, I think that I am the same age now as Mike was when I first met him!

I could not have achieved what I have today without two very special individuals – my co-authors and cohorts for many years now Simon A. Perry and Mike Brownsword. People say that you can judge a person by the company that they keep and I feel truly honoured to share the front cover with both of them today and, hopefully, for many years to come.

Finally, the reason why I do all of this, I must thank my beautiful wife Rebecca and my three children Jude, Eliza and Roo for all of their love and support over the years. Olive and Betty also deserve a mention for their love and devotion to whoever controls the cat food.

**Jon Holt**, July 2011

There have been many people who have contributed to the ideas captured in this book and Jon has done an excellent job of naming the most influential. Rather than adding to this list, I would like to say a big thank you to all who have attended training courses that I have been lucky enough to deliver and who have challenged,

questioned and contributed to the often lively discussions that have taken place over the last 6 years. Your thoughts and opinions have been invaluable.

Without turning this into a love-fest, I must take this opportunity to thank Jon and Mike for being great friends, as well as colleagues and coauthors, for more than 10 and 5 years, respectively. I hope that we have many more years working together.

To conclude, Motley, the cat, is still, just about, with us and continues to serve as an excellent feline alarm clock, if one is with no sense of time.

Finally, as ever, I must thank my wife Sally for all the love and encouragement she has given me over the last 20 years. I am, without doubt, a better person for having met her.

**Simon A. Perry**, July 2011

In writing my first book I find myself in awe about the number of people who have contributed, through comment, discussion, model and criticism, whether on site, during a course or in the office. I would like to thank them all. However, I will simply echo Jon and Simon's thanks and extend it a little to the iFEST partners who have been exposed to some of these ideas more recently.

To complete the triangle, I will take this opportunity to thank Jon for letting me hang him upside down from a crane, and Simon for reviewing all my work. They have made writing this book, as with most of my work, an enjoyable task. I hope they will let me work with them on more books in the future.

Finally, I thank my wonderful wife Jean and my two energetic boys Samuel and Daniel for all the love and laughter they give. Also to continue the theme of cats, I would like to thank Shelly, our little cat, who never fails to remind me that it's late at night and I really should stop modelling or writing to feed her.

**Mike Brownsword**, July 2011



---

# Contents

---

## Acknowledgement

xv

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction to requirements	1
1.1.1	The need for requirements engineering	1
1.2	Concepts and terms	4
1.2.1	Defining ‘requirement’	4
1.2.1.1	Dictionary definition	4
1.2.1.2	INCOSE definition	4
1.2.1.3	Modelling definition	5
1.2.1.4	Standards definition	5
1.2.1.5	Definition for this book	6
1.2.2	Types of requirements	6
1.2.2.1	Basic types of requirement	6
1.2.2.2	Business requirements	7
1.2.2.3	Functional requirements	7
1.2.2.4	Non-functional requirements	8
1.2.2.5	Differentiating between types of requirement	8
1.2.3	Stakeholders	9
1.2.4	Context modelling	11
1.2.4.1	Realising model-based systems engineering	12
1.3	Overview of the book	13
	References	14
<b>PART I</b>	<b>Introduction</b>	<b>15</b>
<b>2</b>	<b>Modelling</b>	<b>17</b>
2.1	Introduction	17
2.2	Why we model	17
2.2.1	The kennel (doghouse)	17
2.2.2	The house	19
2.2.3	The office block	20
2.2.4	The point	22
2.3	The three evils	23
2.3.1	Complexity	23
2.3.2	Lack of understanding	25

2.3.3	Communication	26
2.3.4	The vicious triangle	27
2.4	Modelling	28
2.4.1	Defining modelling	28
2.4.2	The choice of model	29
2.4.3	The level of abstraction	29
2.4.4	Connection to reality	30
2.4.5	Independent views of the same system	30
2.5	The SysML	31
2.5.1	The two aspects of a SysML model	31
2.6	Summary	32
	References	32
<b>3</b>	<b>The SysML notation</b>	<b>33</b>
3.1	Introduction	33
3.1.1	Structural modelling	36
3.1.1.1	Adding more detail to relationships	40
3.1.2	Behavioural modelling	43
3.1.3	Stereotypes	50
3.2	Block definition diagrams	51
3.2.1	Overview	52
3.2.2	Notation	52
3.2.3	Usage	56
3.2.4	Summary	60
3.3	Requirement diagrams	61
3.3.1	Overview	61
3.3.2	Notation	61
3.3.3	Usage	64
3.3.4	Summary	68
3.4	Parametric diagrams	68
3.4.1	Overview	68
3.4.2	Notation	68
3.4.3	Usage	70
3.4.4	Summary	74
3.5	Use case diagrams	74
3.5.1	Overview	74
3.5.2	Notation	76
3.5.3	Usage	78
3.5.3.1	Use case at too high a level	81
3.5.3.2	Actor at too high a level	82
3.5.3.3	Repeated actors	83
3.5.3.4	Something missing – Use cases without actors and actors without use cases	84
3.5.4	Summary	85

3.6	Sequence diagrams	85
3.6.1	Overview	85
3.6.2	Notation	85
3.6.3	Usage	87
3.6.4	Summary	90
3.7	Conclusions	90
	References	91
<b>PART II</b>	<b>Concepts</b>	<b>93</b>
<b>4</b>	<b>Requirements engineering concepts</b>	<b>95</b>
4.1	Introduction	95
4.2	The ontology	95
4.2.1	The 'Requirement' concept	96
4.2.1.1	The 'Business Requirement'	98
4.2.1.2	The 'Functional Requirement'	98
4.2.1.3	The 'Non-functional Requirement'	99
4.2.2	The 'Requirement Description' concept	101
4.2.2.1	Typical properties of a requirement	102
4.2.3	The 'Source Element' concept	103
4.2.4	The 'Rule' concept	105
4.2.5	The 'Context' concept	107
4.2.5.1	Stakeholder context	107
4.2.5.2	System contexts	110
4.2.5.3	Other contexts	113
4.2.6	The 'Use Case' concept	113
4.2.7	The 'Scenario' concept	115
4.3	The ontology revisited	116
4.4	Summary	117
	References	117
<b>5</b>	<b>A framework for model-based requirements engineering</b>	<b>119</b>
5.1	Introduction	119
5.2	Introducing the framework	119
5.3	Introduction to the framework views	120
5.4	The views	122
5.4.1	The Source Element View	122
5.4.1.1	Rational	122
5.4.1.2	The ontology	123
5.4.1.3	Relationships with other views	125
5.4.1.4	Visualising the view	125
5.4.1.5	Discussion	126
5.4.2	Requirement Description View	127
5.4.2.1	Rational	127
5.4.2.2	The ontology	127

5.4.2.3	Relationships with other views	129
5.4.2.4	Visualising the views	131
5.4.2.5	Discussion	132
5.4.3	Definition Rule Set View	133
5.4.3.1	Rational	133
5.4.3.2	The ontology	133
5.4.3.3	Relationships with other views	135
5.4.3.4	Visualising the view	136
5.4.3.5	Discussion	137
5.4.4	Requirement Context View	139
5.4.4.1	Rational	139
5.4.4.2	The ontology	139
5.4.4.3	Relationships with other views	140
5.4.4.4	Visualising the view	142
5.4.4.5	Discussion	142
5.4.5	Context Definition View	144
5.4.5.1	Rational	144
5.4.5.2	The ontology	144
5.4.5.3	Relationships with other views	145
5.4.5.4	Visualising the view	146
5.4.5.5	Discussion	148
5.4.6	Validation View	148
5.4.6.1	Rational	148
5.4.6.2	The ontology	149
5.4.6.3	Relationships with other views	151
5.4.6.4	Visualising the 'Stakeholder Scenario View'	151
5.4.6.5	Visualising the 'System Scenario View'	152
5.4.6.6	Visualising the 'Constraint Validation View'	152
5.4.6.7	Visualising the 'Constraint Usage View'	154
5.4.6.8	Discussion	154
5.5	The Traceability Views	157
5.5.1	Relationships to other views in the framework	158
5.5.2	Visualising the view	158
5.6	Conclusions	159
	References	159

### **PART III Realisation** 161

<b>6</b>	<b>Case study</b>	<b>163</b>
6.1	Introduction	163
6.2	Background	166
6.3	The Source Element View	166
6.4	The Requirement Description View	171
6.4.1	The Traceability View (Requirement Description View to Source Element View)	175

6.5	The definition rule set view (testing the Requirement Descriptions)	177
6.6	The Context Definition View (identification of stakeholder contexts)	183
6.7	The Requirement Context View	186
6.7.1	The Traceability View (Stakeholder Requirement Context View to Requirement Description View)	193
6.8	The Validation View (definition of stakeholder scenarios)	196
6.8.1	The Traceability View (Stakeholder Scenario View to Requirement Context View)	200
6.9	The Context Definition View (identification of system context)	203
6.10	The Requirement Context View (system requirements)	203
6.10.1	The Traceability View (system Requirement Context View to Stakeholder Requirement Context View)	212
6.11	The Validation View (System Scenarios View)	215
6.11.1	The Traceability View (System Scenario View validates Requirement Context View)	219
6.12	The Validation View (Constraint Validation View)	221
6.12.1	The Traceability View (System Constraint Validation view to System Requirements Context View)	227
6.13	Conclusions	229
	References	231
<b>7</b>	<b>Realising the process</b>	<b>233</b>
7.1	Introduction	233
7.2	The 'Person'	234
7.2.1	Teaching guide	234
7.2.1.1	Different types of teaching	235
7.2.2	Professional training	236
7.2.2.1	Teaching requirements	236
7.2.3	Teaching as part of an undergraduate/postgraduate course	238
7.2.3.1	Teaching requirements and stakeholders	238
7.2.3.2	A generic course structure	239
7.2.3.3	Conclusions	243
7.2.4	Competence	243
7.2.4.1	Generic competencies	244
7.2.4.2	Bespoke competencies	250
7.2.4.3	Generic versus specific competencies	256
7.2.5	Summary of 'people'	256
7.3	The 'process'	257
7.3.1	Defining the process	257
7.3.1.1	The ACRE process: the Requirement View	258
7.3.1.2	The ACRE process: the Process Structure View	260
7.3.1.3	The ACRE process: the Process Content View	263
7.3.1.4	The ACRE process: the stakeholder view	265
7.3.1.5	The ACRE process: the Information View	270

7.3.1.6	The ACRE process: the Process Behaviour View	270
7.3.1.7	The ACRE process: the Process Instance View	273
7.3.2	Mapping the process to source standards	273
7.3.3	Using the process	275
7.3.3.1	Example use: quick and dirty process	277
7.3.3.2	Example use: semi-formal process	278
7.3.3.3	Example use: formal process	280
7.3.4	Deploying the process	280
7.3.5	Summary of 'Process'	283
7.4	The 'Tool'	283
7.4.1	Considerations for tool availability	283
7.4.2	Use of tools with the process	286
7.4.2.1	Example tool implementation: quick and dirty process	287
7.4.2.2	Example tool implementation: semi-formal process	289
7.4.2.3	Example tool implementation: formal process	293
7.4.3	Tool selection	297
7.4.3.1	Providing modelling capability	297
7.4.3.2	Providing system functionality	298
7.4.3.3	Providing interoperability capability	298
7.4.3.4	Ensuring vendor's quality of service	298
7.4.3.5	Being compatible with process model	298
7.5	Conclusions	299
	References	299
<b>Appendix A Summary of notation</b>		<b>301</b>
A.1	Introduction	301
A.2	Block definition diagrams	301
A.3	Requirement diagrams	304
A.4	Parametric constraint diagrams	306
A.5	Use case diagrams	308
A.6	Sequence diagrams	309
	Reference	310
<b>Appendix B Advanced sequence diagram notation</b>		<b>311</b>
B.1	Introduction	311
B.2	Showing parallel processing	311
B.3	Referencing other diagrams	312
B.4	Showing alternatives	313
B.5	Showing loops	313
	Reference	315
<b>Appendix C A seven views summary of the UCAM processes</b>		<b>317</b>
C.1	Introduction	317
C.2	Overview of the seven views approach	317
C.2.1	The Requirements View	319
C.2.2	The Stakeholder View	319

C.2.3	The Process Structure View	319
C.2.4	The Process Content View	319
C.2.5	The Process Behaviour View	320
C.2.6	The Information View	320
C.2.7	The Process Instance View	320
C.3	The Seven Views of UCAM	320
C.3.1	The Requirements View	321
C.3.2	The Stakeholder View	322
C.3.3	The Process Structure View	322
C.3.4	The Process Content Views	323
C.3.5	The Process Behaviour Views	325
C.3.6	The Information Views	329
C.3.7	The Process Instance Views	331
References		332
<b>Index</b>		<b>333</b>

---

## Chapter 1

# Introduction

---

What you want and what you get are two different things.

*Christine Holt*

### 1.1 Introduction to requirements

The whole area of requirements is one that is fraught with conflicts, misunderstandings and many contradictory views and opinions.

It is widely acknowledged that getting the requirements ‘right’ is crucial to the success of any project or to realising any form of successful system, but what exactly is meant by this?

It is clear that a good understanding of requirements is important, so why do so few people actually apply any sort of rigour to their requirements engineering activities?

This section discusses these two issues at a high level by looking at exactly what we mean by ‘requirements’ and ‘requirements engineering’ and then looking at some key concepts that are required to ensure good requirements and rigorous requirements engineering activities.

#### 1.1.1 *The need for requirements engineering*

To understand the requirements for a system, it is not enough to simply state the needs of a group of users, but it is necessary to *engineer* the requirements of all relevant stakeholders.

‘Requirements engineering’, therefore, is the discipline that enables this understanding and is important for a number of reasons:

- *Systems approach.* All systems approaches identify a need to understand the requirements properly. This relates to all types of systems whether they are technical, social, financial etc.
- *Quality.* There are many definitions of quality, but the two that are considered here are from the International Standards Organisation (ISO) [1]. This first definition, which is arguably the most common definition of quality used in the world, is ‘fitness for purpose’. Fitness for purpose means that the system does what it is supposed to do or, to put it another way, the system satisfies its



original requirements. The second definition that is cited here is ‘conformance to requirements’ – no explanation needed.

- *Requirements drive the project.* Every aspect of the project should be traceable back to the source requirements. The requirements should drive everything that is done in the project, and if they are not, then some serious questions need to be asked.
- *Benchmark for acceptance.* Acceptance testing – the tests that provide the customer with the confidence that the system is fit for purpose are based solely on requirements. Acceptance tests are not based on the design or the implementation techniques that are applied (unless they are constraints) but are based entirely on the requirements set for the system. Therefore, if the requirements are not fully understood, how on earth can the final system be accepted?
- *For increased confidence.* One of the least tangible, yet most powerful benefits of applying an effective systems approach is one of confidence. When the requirements of the system are understood, the system can be demonstrated and accepted. When the requirements are understood, they can be agreed by the customer somewhere in the initial, or conceptual, stage of a project. This confidence refers to the confidence of both the customer and the supplier sides of the relationships. In other words, the confidence of all the stakeholders will be increased. Although difficult to measure, confidence is an immensely valuable attribute for any relationship.

The points raised here make the case for understanding requirements. To understand the requirements, it is necessary to engineer them, and this gives rise to the discipline known as ‘requirements engineering’. An effective approach to requirements engineering will result in a concise, consistent and lucid definition of the requirements of any system and will yield the following benefits:

- *Increased probability of optimum solution.* If the requirements for any system are unknown or badly defined, then there is no way that the system under development can meet them. There is also no way that the system can be validated, as validation is based solely on the requirements.
- *Full traceability.* It is important that any aspect of the system can be traced back to its source requirement(s). Not only is traceability essential for effective verification and validation of the system, but it is also a crucial part of quality assurance.
- *Requirements that are independent of the solution.* A good set of requirements should be independent of any specific solution. In the real world, this is rarely completely achievable, as there will almost always be some aspect of the requirements where the solution, part of the solution or the technology used by the solution has already been decided. When and where this does occur, such requirements are considered to be ‘constraints’ because they will limit, in some way, how that requirement may be realised. In some cases this may refer to a solution or partial solution. In others, there may be a quality constraint, such as meeting a particular standard or following a specific process. Otherwise, the