

University of Groningen

## Software architecture analysis of usability

Folmer, Eelke

**IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.**

*Document Version*

Publisher's PDF, also known as Version of record

*Publication date:*

2005

[Link to publication in University of Groningen/UMCG research database](#)

*Citation for published version (APA):*

Folmer, E. (2005). *Software architecture analysis of usability*. s.n.

### Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

### Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.*

# **Software Architecture Analysis of Usability**



**Eelke Folmer**



RUG

The work presented in this thesis has been funded by the European Union funded STATUS (Software Architecture that supports Usability) project. STATUS was an ESPRIT project (IST-2001-32298) financed by the European Commission in its Information Society Technologies Program, Action Line IV.3. The project started on 1-12-2001 and its duration was 36 months.

<b>Title</b>	Software Architecture Analysis of Usability
<b>Author</b>	Eelke Folmer
<b>Email</b>	Email(at)eelke.com
<b>ISBN</b>	90-367-2361-2
<b>Electronic version</b>	<a href="http://www.eelke.com/thesis.pdf">http://www.eelke.com/thesis.pdf</a>
<b>Keywords</b>	Software Architecture, Software Architecture Analysis, Software Quality, Usability, HCI, Usability Patterns.
<b>Cover Design</b>	Eelke Folmer/ Wouter Kiel/ Based upon the painting the tower of Babel by Flemish painter Pieter Brueghel the Elder (1525-1569)
<b>Printing</b>	PrintPartners Ipskamp, Enschede

Copyright © by Eelke Folmer

*All rights reserved. Subject to exceptions provided for by law, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the copyright owner. No part of this publication may be adapted in whole or in part without the prior written permission of the author.*

Rijksuniversiteit Groningen

**Software Architecture Analysis of Usability**

Proefschrift

ter verkrijging van het doctoraat in de  
Wiskunde en Natuurwetenschappen  
aan de Rijksuniversiteit Groningen  
op gezag van de  
Rector Magnificus, dr. F. Zwarts,  
in het openbaar te verdedigen op  
maandag 26 september 2005  
om 14.45 uur

door

Eelke Folmer  
geboren op 28 februari 1977  
te Balk

Promotor:

Prof Dr. Ir. J. Bosch

Beoordelingscommissie:

Prof Dr. D.K. Hammer

Prof Dr. N.A.M. Maiden

Prof Dr. J.C. van Vliet

ISBN: 90-367-2361-2

# Acknowledgements

Wow! Four years of hard work and I can't believe it's finally finished! In these years I have got to know some outstanding individuals which contributed to my work and which therefore I would like to thank. First and foremost I would like to thank my advisor Jan Bosch. You have been a great mentor and a great friend. In the first year I was very skeptical about the results we achieved but you taught me an important lesson: *"there is no point in doing research if you don't believe in yourself"*. I am grateful for the freedom you gave me for exploring my own ideas. Without your knowledge, inspiring one-liners and cracking-of-the-whip I would never have finished.

Furthermore I would like to thank Jilles van Gorp who helped me put my research on track in the first two years. Next I would like to thank Renzo Akkerman for all the "bike dates" we had which made sure we both started our work on time (and yes you were right about doing my thesis in Latex). Also thank you both for proofreading my introduction. Thanks go to Lisette Bakalis for pulling me out of my "ivory tower" and making sure engineers could actually use my method. Thanks to Martijn van Welie for co-authoring and providing feedback on my research. I want to thank the reading committee for providing feedback and approving my thesis: Dieter Hammer (RuG), Neil Maiden (City University) and Hans van Vliet (VU Amsterdam). Thanks go to Frits Greuter for interesting me for Software Engineering during my master's project.

The research in this thesis has been funded by the STATUS project. I especially would like to thank George Marks and Dimitris Tsirikos of LogicDIS, Alberto de Andres, Rafael Pous and Fernando Vaquerizo of IHG, Ana Moreno, Natalia Jurista, Xavier Ferre of the Universidad Politecnica de Madrid, Jeff Magee, Jeff Kramer and Robert Chatley of Imperial College London. Much respect goes to my colleagues (and hopefully still friends), Michel, Sybren, Marco, Jos, Anton, Rein, Ivor, Johanneke, Jan Salvador and Mugurel for putting up with me these years. I would also like to thank the support staff, especially Esmee Elshof and Desiree Hansen for letting me be part of "het kippenhok" when I was kicked out of my office to make room for old computer parts.

Where would one be without friends? I would like to thank Juansi Dominguez for the sportive breaks in the gym which kept my mind of working and my body to working. Next my buddy Philip Jonkers you inspired me to do my PhD. Thanks for your wisdom. Thank you both for accepting to be paranimf for my PhD. defense. Special thanks to my close mates David, Tsjipke, Renzo, Paul, Wouter and Beep and my "Bolsward" buddies Erwin and Marijke. And the rest of my friends: Albert, Anita, Arjen, Bart, Neef Eelke, Eelco, Evert Jan, Gijs, Guido, Marrit, Marieke Ploegh, Marieke Boumans, Michiel, Onno, Patrick, Paul, Rik, Roeland, Ronald and MSN friends "A", "Sopje" and "Svoelle".

Beyond friends there are family. My "mem" Clasien, thanks for your love and support for no matter what I decided to do. My "heit" Andries, thanks for your advice, which kept my spirits up, but my feet firmly on the ground. Love goes to my brother Paulus, his girlfriend Ellen, my sister Atty and our dog Famke. The last words in this section are for those closest to me. Much love goes to Evelien for the very special person she is. Sjoukje and Femke, I am still grateful to have been part of your lives. Finally, this thesis is dedicated to Geertrui Peek, to express my gratitude for your love, understanding and support that you gave me during these past four years. Sorry for all the stress that my work put on our relationship, but you showed me true love by always standing by my side.

Is that everyone?? - Love and Peace 🙌 Eelke



# Table of Contents

<b>Chapter 1 Introduction.....</b>	<b>15</b>
1.1 Software Systems .....	15
1.1.1 Feature race .....	15
1.1.2 Competition .....	16
1.1.3 Engineering challenges .....	16
1.2 Software Quality .....	17
1.2.1 What is quality? .....	17
1.2.2 Product acceptance.....	18
1.2.3 What is software quality? .....	18
1.2.4 Software quality models .....	19
1.2.5 Usability.....	22
1.2.6 What is usability? .....	22
1.2.7 Design for usability.....	23
1.2.8 Product oriented design .....	23
1.2.9 Process oriented design.....	25
1.2.10 Challenges.....	25
1.3 Software Architecture .....	26
1.3.1 Software architecture description.....	27
1.3.2 Software architecture design.....	32
1.3.3 Features / evolution .....	33
1.3.4 Software quality.....	33
1.3.5 Cost & time to market.....	34
1.3.6 Complexity.....	34
1.3.7 Reuse .....	35
1.3.8 Software product families .....	35
1.3.9 Design constraints.....	35
1.3.10 Capture design knowledge .....	36
1.3.11 Turn architecture design into a process.....	40
1.3.12 Challenges.....	43
1.4 Research Context .....	44
1.4.1 STATUS project .....	45
1.4.2 STATUS partners.....	45
1.4.3 STATUS objectives .....	46
1.5 Research Questions .....	47
1.5.1 Investigating the relationship between SA and usability .....	48
1.5.2 Development of an SA assessment technique for usability .....	48
1.6 Research Methods.....	48
1.6.1 Case studies .....	49
1.6.2 Action research.....	50
1.7 Thesis Structure.....	50
1.7.1 Investigating the relationship between usability and SA. ....	50
1.7.2 Development of an SA assessment technique for usability.....	51
1.7.3 Related publications.....	52
<b>Chapter 2 Architecting for Usability .....</b>	<b>53</b>
2.1 Introduction.....	53
2.1.1 Current software still has low usability.....	54



2.1.2	Traditional design approaches fail.....	55
2.1.3	Software architecture restricts usability .....	57
2.1.4	Architecting for quality .....	59
2.2	What is Usability?.....	61
2.2.1	Shackel.....	61
2.2.2	Nielsen .....	62
2.2.3	ISO 9241-11.....	63
2.2.4	ISO 9126 .....	64
2.2.5	Overview .....	65
2.3	Evaluating Usability.....	67
2.3.1	Usability testing.....	67
2.3.2	Usability inspection.....	68
2.3.3	Usability inquiry .....	68
2.3.4	Overview .....	69
2.4	Design for Usability .....	70
2.4.1	Process oriented .....	70
2.4.2	Product oriented.....	71
2.4.3	Interface guidelines .....	71
2.4.4	Design heuristics and principles .....	72
2.4.5	Usability patterns .....	73
2.4.6	Overview .....	74
2.5	Usability Framework .....	74
2.5.1	Problem domain .....	76
2.5.2	Solution domain .....	77
2.5.3	Relating the problem to solution domain .....	77
2.6	Research Issues.....	78
2.7	Conclusions.....	79

## **Chapter 3 Usability Patterns in SA..... 81**

3.1	Introduction .....	81
3.2	Usability Framework .....	82
3.3	Usability Attributes.....	84
3.4	Usability Properties .....	84
3.5	Usability Patterns .....	85
3.6	Summary and Conclusions .....	85

## **Chapter 4 The SAU Framework..... 87**

4.1	Introduction .....	87
4.2	The Relationship between SA and Usability .....	89
4.3	Usability Attributes.....	91
4.4	Usability Properties .....	92
4.5	Architecturally Sensitive Usability Patterns.....	97
4.6	Putting it Together: Relation between SA and Usability .....	106
4.7	Related Work .....	107
4.8	Conclusions.....	109
4.9	Acknowledgements.....	111

## **Chapter 5 Bridging Patterns.....113**

5.1	Introduction.....	113
5.2	Bridging Patterns.....	114

5.3	Multi Level Undo .....	117
5.4	Multi Channel Access.....	122
5.5	Wizard.....	126
5.6	Single Sign On.....	129
5.7	Discussion .....	133
5.8	Future Work.....	134
<b>Chapter 6 Reusable SA Tradeoff Solutions .....</b>		<b>137</b>
6.1	Introduction.....	137
6.2	The Software Architecture - Quality Framework.....	139
6.3	The Security-Safety-Usability Framework .....	143
6.4	Discussion .....	152
6.5	Related Work .....	154
6.6	Conclusions.....	156
<b>Chapter 7 SALUTA: Webplatform.....</b>		<b>157</b>
7.1	Introduction.....	157
7.2	Relationship between Usability and SA .....	159
7.2.1	Usability attributes.....	159
7.2.2	Usability properties.....	159
7.2.3	Architecture sensitive usability patterns .....	160
7.3	Software Architecture Assessment .....	161
7.3.1	Usability specification .....	162
7.3.2	Usage profiles .....	163
7.4	SALUTA .....	163
7.4.1	Create usage profile.....	164
7.4.2	Describe provided usability.....	165
7.4.3	Evaluate scenarios.....	166
7.4.4	Interpret the results .....	167
7.5	Validation.....	167
7.5.1	Usage profile creation.....	168
7.5.2	Architecture description.....	169
7.5.3	Evaluate scenarios.....	171
7.5.4	Interpret the results .....	172
7.5.5	Validation .....	173
7.6	Conclusions.....	173
7.7	Acknowledgments.....	174
<b>Chapter 8 SALUTA: Compressor &amp; Esuite .....</b>		<b>175</b>
8.1	Introduction.....	175
8.2	SAU Framework.....	176
8.2.1	Architecture sensitive usability patterns .....	176
8.2.2	Usability properties.....	177
8.2.3	Usability attributes.....	177
8.3	SALUTA .....	179
8.3.1	Create usage profile.....	179
8.3.2	Describe provided usability.....	180
8.3.3	Evaluate scenarios.....	181
8.3.4	Interpret the results .....	181
8.4	Case Studies .....	182

8.4.1	Usage profile creation.....	183
8.4.2	Architecture description.....	184
8.4.3	Evaluate scenarios.....	184
8.4.4	Interpret the results .....	185
8.4.5	Validation .....	185
8.5	Conclusions.....	186
<b>Chapter 9 Experiences with SALUTA.....</b>		<b>187</b>
9.1	Introduction.....	187
9.2	The SAU Framework.....	189
9.2.1	Usability attributes .....	190
9.2.2	Usability properties .....	190
9.2.3	Architecture sensitive usability patterns: .....	191
9.2.4	Relationships in the SAU framework.....	193
9.3	Overview of SALUTA .....	194
9.3.1	Usage profile creation.....	195
9.3.2	Analyze the software architecture.....	196
9.3.3	Scenario evaluation .....	197
9.3.4	Interpretation of the results .....	198
9.4	Case Descriptions.....	199
9.4.1	Webplatform.....	200
9.4.2	Compressor.....	200
9.4.3	ESuite.....	201
9.4.4	Assessment results .....	201
9.5	Experiences .....	201
9.5.1	Usage profile creation.....	202
9.5.2	Architecture analysis .....	205
9.5.3	Scenario evaluation .....	206
9.5.4	Interpretation .....	207
9.5.5	General experiences .....	208
9.6	Related Work .....	210
9.7	Conclusions.....	211
9.8	Acknowledgements .....	212
<b>Chapter 10 Conclusions .....</b>		<b>213</b>
10.1	Research Questions .....	213
10.1.1	Investigating the relationship between SA and usability .....	213
10.1.2	Development of an SA assessment technique for usability .....	217
10.1.3	Overall research question.....	220
10.2	Contributions.....	220
10.2.1	Investigating the relationship between SA and usability .....	220
10.2.2	Development of an SA assessment technique for usability .....	221
10.3	Future Work and Open Issues .....	221
<b>Chapter 11 References.....</b>		<b>225</b>
<b>Chapter 12 Samenvatting .....</b>		<b>237</b>

# Table of Figures

Figure 1: McCall Quality Model.....	20
Figure 2: Boehm Quality Model .....	20
Figure 3: ISO 9126 Quality Model.....	21
Figure 4: Apple Interface.....	24
Figure 5: Windows XP Program Compatibility Wizard.....	24
Figure 6: Java Source Code .....	27
Figure 7: Box and Line Diagram .....	28
Figure 8: the 4+1 View Model and Stakeholders .....	29
Figure 9: the SNH View Model.....	30
Figure 10: Data Abstraction Layer .....	34
Figure 11: Software Architecture Design Constraints .....	36
Figure 12: 3-Tier Architecture .....	37
Figure 13: Database Management System .....	38
Figure 14: Model-View-Controller .....	38
Figure 15: Struts Framework.....	39
Figure 16: Glue Code .....	39
Figure 17: QASAR Architecture Design Method.....	42
Figure 18: Action Research Refinement Cycle .....	50
Figure 19: Approaches to Usability Engineering .....	55
Figure 20: Current Design for Usability .....	56
Figure 21: Waterfall Model and its Representations .....	56
Figure 22: Software Architecture Design Method .....	59
Figure 23: Desired Design Approach .....	60
Figure 24: Shackel's Definition of Usability.....	62
Figure 25: Nielsen's Definition of Usability.....	62
Figure 26: ISO 9241-11 Definition of Usability .....	63
Figure 27: ISO 9126-1 Quality Model.....	64
Figure 28: Overview of Evaluation Methods .....	70
Figure 29: Usability Framework .....	75
Figure 30: Usability Framework .....	83
Figure 31: Wizard Pattern .....	90
Figure 32: Multi Channeling Pattern .....	91
Figure 33: Usability Framework .....	106
Figure 34: Undo in Microsoft Word.....	117
Figure 35: Undo in Word.....	119
Figure 36: Undo in Photoshop .....	119
Figure 37: Command Pattern .....	120
Figure 38: Undo Framework.....	121
Figure 39: Amazon Accessible through a WAP Browser .....	122
Figure 40: Ebay on PC, WAP and I-Mode Phone .....	123
Figure 41: 2T/3T and MVC Architectures.....	124
Figure 42: Model 1 Architecture.....	125
Figure 43: Model 2 Architecture .....	125
Figure 44: a Typical Wizard .....	126
Figure 45: Wire Frame Wizard.....	127
Figure 46: Pack-and-go Wizard in PowerPoint .....	127
Figure 47: UML Model of a Wizard.....	128
Figure 48: .NET Passport Providing SSO Capability.....	129
Figure 49: .NET Password.....	130

---

Figure 50: Windows 2000 SSO.....	130
Figure 51: Encapsulation on Each System. ....	131
Figure 52: Overview of Passport Mechanism .....	132
Figure 53: UML Model Passport Manager.....	132
Figure 54: Relation between Quality and Software Architecture.....	140
Figure 55: Relationships between the Elements in the Framework.....	142
Figure 56: Relationships for Wizard/ Usability.....	143
Figure 57: Intra Quality Relations between Usability, Security and Safety .....	147
Figure 58: Inter Quality Relations between Usability, Security and Safety.....	148
Figure 59: Framework Showing SSO as a "Boundary" Pattern .....	151
Figure 60: Usability Framework .....	161
Figure 61: Example Usage Profile .....	165
Figure 62: Snapshot Evaluation Example.....	166
Figure 63: Webplatform Software Architecture .....	170
Figure 64: Provide Multiple Views/ & Visual/Functional Consistency .....	171
Figure 65: SAU Framework.....	178
Figure 66: Multi Level Undo UML Model .....	192
Figure 67: Relationships between Attributes, Properties and Patterns. ....	193
Figure 68: Example Usage Scenario .....	195
Figure 69: Compressor Architecture.....	197
Figure 70: Snapshot Assessment Example .....	198
Figure 71: Transforming Requirements to a Usage Profile .....	203

# Table of Tables

Table 1: Definitions of Usability in Chronological Order .....	23
Table 2: Nielsen's' Heuristics .....	24
Table 3: Apple Interface Guidelines .....	24
Table 4: Wizard from Welie Pattern Collection .....	24
Table 5: STATUS workpackages overview .....	46
Table 6: Overview of Usability Definitions .....	66
Table 7: Overview of Authors .....	92
Table 8: Providing Feedback .....	94
Table 9: Error Management .....	94
Table 10: Consistency .....	94
Table 11: Guidance .....	95
Table 12: Minimize Cognitive Load .....	95
Table 13: Explicit User Control .....	95
Table 14: Natural Mapping .....	96
Table 15: Accessibility .....	96
Table 16: Adaptability .....	96
Table 17: System Feedback .....	99
Table 18: Actions for Multiple Objects .....	99
Table 19: Cancel .....	99
Table 20: Data Validation .....	100
Table 21: History Logging .....	100
Table 22: Scripting .....	101
Table 23: Multiple Views .....	101
Table 24: Multi Channeling .....	102
Table 25: Undo .....	102
Table 26: User Modes .....	102
Table 27: User Profiles .....	103
Table 28: Wizard .....	103
Table 29: Workflow Model .....	104
Table 30: Emulation .....	104
Table 31: Context Sensitive Help .....	105
Table 32: Multi Level Undo .....	117
Table 33: Multi Channel Access .....	122
Table 34: Wizard .....	126
Table 35: Single Sign On .....	129
Table 36: Single Access Point .....	145
Table 37: Warning .....	146
Table 38: Multi-Level Undo .....	146
Table 39: Single Sign-on .....	150
Table 40: Selected Tasks Webplatform .....	168
Table 41: Usage Profile Webplatform .....	169
Table 42: Pattern and Property Implementation Details .....	171
Table 43: Architecture Support Matrix .....	172
Table 44: Undo .....	176
Table 45: Error Management .....	177
Table 46: ASUP's and Properties Identified .....	178
Table 47: Comparison of Both Cases .....	182
Table 48: eSuite Usage Profile .....	183
Table 49: ASUP / Properties Present in SA .....	184

---

Table 50: Example Usage Profile Evaluation.....	184
Table 51: Evaluation Results Assessments.....	185
Table 52: Consistency.....	190
Table 53: Multi-Level Undo .....	191
Table 54: Comparison of System Characteristics .....	199
Table 55: Assessment Results .....	201
Table 56: ESuite Usage Scenario.....	207