

DOCTORAATSPROEFSCHRIFT

2011 | School voor Informatietechnologie
Kennistechnologie, Informatica, Wiskunde, ICT

User-Centered Process Framework and Techniques to Support the Realization of Interactive Systems by Multi-Disciplinary Teams

Proefschrift voorgelegd tot het behalen van de graad van
Doctor in de Wetenschappen, Informatica, te verdedigen door:

Mieke HAESEN

Promotor: prof. dr. Karin Coninx
Copromotor: prof. dr. Kris Luyten



Acknowledgements

If someone would have told me ten years ago that I would obtain a PhD, I probably would not have believed it. Seven years ago, I started working as a researcher at the Expertise centre for Digital Media (EDM), a research institute at Hasselt University. During these first years as a researcher, several people and my love for research convinced me to pursue my PhD. Four years ago I actually took on this new challenge. Before describing my research in this dissertation, I would like to thank several people for their advise, interesting insights, cooperation and support.

First of all, I would like to address special thanks to my advisor, Prof. dr. Karin Coninx. After giving me the opportunity to be part of a multidisciplinary user-centered design team in the VIP-lab project, she convinced me to obtain a PhD and supported me in my work. She gave me the necessary amount of freedom in my research, while sharing interesting insights and valuable feedback regarding my publications and this dissertation. Second, I want to thank my co-advisor, Prof. dr. Kris Luyten. His broad view on research stimulated me to concentrate on user-centered software engineering and his open-mindedness in research inspired me to combine creativity and more traditional research methods in my PhD.

I also would like to thank Prof. dr. Eddy Flerackers, our managing director, and Prof. dr. Frank Van Reeth, our deputy managing director, for giving me the opportunity to work as a researcher and PhD student in the pleasant work environment that is offered at EDM.

I want to address special thanks to Prof. dr. Panos Markopoulos and Prof. dr. Gilbert Cockton for their very detailed and valuable comments with respect to this dissertation. Thanks also to the other members of my jury: dr. David Geerts and Prof. dr. Marc Gyssens.

The major part of this PhD was funded by the IWT in the context of the AMASS++ project (IWT SBO 060051). In this project I collaborated with

dr. Jan Meskens who immediately after the start of the project shared his professional skills to conduct research and to co-author deliverables and papers. He also taught me the basics of creating clear figures and nice slide shows for scientific publications and presentations. It was a pleasant experience to collaborate with him in this project. Furthermore, I want to thank Prof. dr. Marie-Francine Moens and the other AMASS++ team members of KULeuven for their collaboration.

In my time as a researcher and PhD student, I had the opportunity to collaborate with several interesting people. I would like to thank dr. Jan Van den Bergh for sharing his expertise and interesting remarks with respect to my research. Thanks also to dr. Chris Raymaekers, dr. Joan De Boeck, Prof. dr. Serge Demeyer and Sylvain Degrandt for their collaboration. Special thanks goes to Karel Robert who helped me out with graphic design and shared novel ideas regarding my research (projects). Furthermore, I want to thank the representatives of Androme, Cegeka, Human Interface Group, Namahn, Marlon, and Tinkertouch, who all helped me to gain insights and feedback from practitioners in the field.

Doing this PhD would not have been half as pleasant without my colleagues at EDM. Their direct or indirect contributions to my research, the fun and inspiring lunch meetings and the informal talks were very valuable to me. I would like to say thanks for this to (in alphabetical order): Stijn Agten, Petr Aksenov, Anastasiia Beznosyk, Maarten Cardinaels, Tom De Weyer, Kris Gabriëls, Nasim Mahmud, Sofie Notelaers, Johanna Renny Octavia, Deepak Sahni, Jolien Schroyen, Jan Schneider, Sean Tan, Daniël Teunkens, Davy Vanacken, dr. Lode Vanacken, dr. Geert Vanderhulst, and Jo Vermeulen. I also want to thank Luc Adriaens for assisting me whenever videos needed to be captured and Ingrid Konings and Roger Claes for taking care of the administrative tasks and logistics.

Also some people who were not directly involved in my research or work environment, should be mentioned here. I would like to thank Niels, Liesbeth and Barbara for their understanding, support, friendship and the pleasant moments spent together. I also want to express my gratitude to my parents. They always supported me in the choices I made in life and gave me the opportunity to study at university, which was a first step towards this PhD. Last, but certainly not least, I want to thank my partner Kevin and our son Casper for their absolute support and ensuring the balance between research and life.

There is no doubt that without this “*multi-disciplinary team*”, consisting of all the people mentioned above, my PhD would not have been the same!

Abstract

The last decades, the use of interactive systems has increased. Current technologies allow a wide range of end users to use several systems in various contexts. Interactive systems are not restricted to professional use anymore and end users expect to obtain a positive user experience from using these systems. Consequently, it is recommended to carefully consider the end user needs during the design and development of systems and their user interfaces.

User-centered approaches for the design and development of interactive systems have proved effective when design teams can properly take into account user needs and the context of use of a system. These approaches typically involve multi-disciplinary teams in order to include complementary points of view in a system's design and development. However, difficulties occur for some development teams when multiple backgrounds and disciplines are involved in a software design and development project.

In this dissertation we investigated the combination of user-centered design (UCD) and software engineering (SE) and the involvement of multi-disciplinary teams, which are indispensable in order to obtain systems that carefully take into account end user needs and contextual information. We presented MuiCSer¹, a process framework for multi-disciplinary user-centered software engineering that was used to specify practical user-centered software engineering (UCSE) processes. Furthermore, we used MuiCSer to study the use of tools and artifacts in UCSE as well as UCSE in practice. Resulting weaknesses and difficulties in UCSE included a lack of tools and notations that support the collaboration in multi-disciplinary UCSE teams and support the transition from informal artifacts to formal models. Additionally, our studies based on MuiCSer revealed a lack of notations that incorporate a broad range of user needs and requirements.

¹MuiCSer is pronounced as “mixer”.

We introduced and investigated the COMuICSer² storyboarding approach and tool in order to overcome known weaknesses and difficulties in UCSE. First, two user studies were conducted in order to investigate the creation and use of storyboards in multi-disciplinary teams. The results of these studies inspired us to conduct further research concerning the incorporation of comics techniques in the COMuICSer approach and tool. Besides a communication tool, COMuICSer storyboards are also considered as a notation that facilitates the transition of informal artifacts into formal models. We investigated techniques that use COMuICSer storyboards for the creation of formal artifacts in UCSE. Furthermore, connecting COMuICSer storyboards with the user stories used in agile software engineering was introduced to include non-functional requirements and contextual information in requirements for the development of a system.

The highly visual notation used by COMuICSer to depict requirements of interactive systems was shown to be suitable for the communication within multi-disciplinary teams. Furthermore, COMuICSer storyboards can be considered as central documents in UCSE processes. The storyboarding techniques studied in this PhD were evaluated by conducting several user studies. By reflecting on different aspects of COMuICSer storyboarding, we described the strengths and the weaknesses of COMuICSer and suggested several opportunities for future research and development.

²COMuICSer is pronounced as “comics-er”.

Contents

Acknowledgements	i
Abstract	iii
Contents	x
List of Figures	xv
List of Tables	xviii
1 Introduction	1
1.1 User-Centered Design	1
1.1.1 The User Experience and Context of Use	2
1.1.2 Roles Involved in Multi-disciplinary Teams	2
1.1.3 User-Centered Design Processes	3
1.2 Combining User-Centered Design and Software Engineering for Multi-disciplinary Teams	5
1.3 Overview	6
 I A Process Framework for Multi-disciplinary User-Centered Software Engineering	 7
2 The MuiCSer Process Framework	9
2.1 Introduction	9
2.2 MuiCSer	10
2.3 Models and Tools	13
2.3.1 Existing User-centered Processes	13
2.3.2 Artifact transformation tools	16

2.3.3	Requirements and User Needs	20
2.3.4	Structured Interaction Models	20
2.3.5	Low-fidelity Prototypes	20
2.3.6	High-fidelity Prototypes	21
2.3.7	Final Interactive System	21
2.4	Conclusion	21
3	User-Centered Software Engineering in Practice	23
3.1	Introduction	23
3.2	Case Studies	24
3.2.1	NewsWizard	24
3.2.2	Mobile Game for Children	27
3.2.3	Lessons Learned	30
3.3	UCSE in Industry	31
3.3.1	Roles	32
3.3.2	Models and Artifacts	33
3.3.3	Tools	34
3.3.4	Fit with the MuiCSer Process Framework	35
3.3.5	Discussion	35
3.4	Conclusion	36
4	Weaknesses and Difficulties in UCSE	37
4.1	Weaknesses and Difficulties	37
4.2	Can Storyboards be Considered as a Possible Answer?	38
II	Storyboards as a Common Language for User-Centered Software Engineering	41
5	COMuICSer Storyboarding	43
5.1	Introduction	43
5.2	Visualizing Requirements and User Needs	44
5.3	COMuICSer	45
5.3.1	Definition	45
5.3.2	Bridging the Early Stages of UCSE Processes	47
5.4	Tool Support for COMuICSer Storyboards	48
5.5	Research Challenges	50
5.5.1	RC A: Storyboarding for Multi-disciplinary Teams	51
5.5.2	RC B: A Visual Storyboarding Language	51
5.5.3	RC C: Storyboarding to Support Artifact Transformations	52

5.5.4	RC D: Storyboarding to Connect UCD and SE	53
5.6	Conclusion	53
6	Storyboarding in Multi-disciplinary Teams	55
6.1	Introduction	55
6.2	First User Study	56
6.2.1	Participants and Procedure	57
6.2.2	Results	60
6.2.3	Discussion	61
6.3	Observational Study	62
6.3.1	Participants, Setup and Procedure	63
6.3.2	Results	65
6.3.3	Discussion	70
6.4	Conclusion	72
7	The Visual Storyboarding Language	75
7.1	Introduction	75
7.2	The Language of Comics in Storyboarding	76
7.2.1	Visualizing and Communicating Scenarios	77
7.2.2	Applying Techniques of Comics	77
7.3	User Experiment	82
7.3.1	Participants and Procedure	83
7.3.2	Results and Discussion	84
7.4	Incorporating Techniques of Comics in a Storyboarding Tool	87
7.4.1	Composing the storyboard	87
7.4.2	Storyboards for Later Use	89
7.5	Conclusion	91
8	From Informal to Formal Artifacts with Storyboards	93
8.1	Introduction	93
8.2	Informal and Formal Artifacts in User-Centered Software Engineering	94
8.2.1	Informal Artifacts	94
8.2.2	Formal Artifacts	96
8.2.3	Transforming Informal Design Knowledge into Formal Artefacts	98
8.2.4	Introducing a Common Language that Supports the Transformation	101

8.3	Storyboarding to Bridge the Gap Between Informal Design Knowledge and Formal Models	102
8.3.1	A Storyboard Meta-model	103
8.3.2	Mapping Storyboards to Models	104
8.3.3	From Storyboard to High-fidelity Prototype	107
8.4	Conclusion	109
9	Connecting Storyboards and Agile Practices	111
9.1	Introduction	111
9.2	Agile UCD Processes and Requirements Notations	113
9.2.1	Agile UCD Processes	114
9.2.2	Requirements Notations	115
9.3	Interview with Agile SE Practitioners	116
9.3.1	User Stories and the High-level Analysis Document	117
9.3.2	COMuICSer in Agile SE Teams	118
9.3.3	Discussion	118
9.4	Connecting User Stories to COMuICSer Storyboards	119
9.5	Tool Support for Connected Storyboards	121
9.5.1	Agile Planning Tools	121
9.5.2	User Stories in the COMuICSer Tool	122
9.6	First User Study	125
9.6.1	Creating the Storyboard	125
9.6.2	Specifying User Stories	128
9.6.3	Using the Storyboard	129
9.6.4	Discussion	129
9.7	Conclusion	131
III	Reflections and Conclusions	133
10	Interpreting Storyboards	135
10.1	Introduction	135
10.2	Collecting Storyboards	136
10.2.1	Storyboard A: Location-aware Application for Museums	139
10.2.2	Storyboard B: Customized Museum Tour	139
10.2.3	Storyboard C: Serious Game for MS Patients	141
10.2.4	Storyboard D: Alarm and Help System for People Suffering from Dementia	141
10.2.5	Storyboard E: Querying through Social Networks	143

10.3	UI Designs Informed and Inspired by Storyboards	146
10.3.1	Participants	146
10.3.2	Setup and Procedure	147
10.4	Observations and Results	148
10.4.1	Creating UI Designs	148
10.4.2	Discussions with the Authors	152
10.4.3	Storyboard-specific Results	152
10.5	Discussion and Lessons Learned	154
10.5.1	Interpreting the Storyboards	154
10.5.2	Translating the Storyboards into UI Designs	155
10.5.3	Misinterpretations	155
10.5.4	Preferences for Team Meetings	156
10.5.5	Opportunities for Further Research	156
10.6	Conclusion	157
11	Reflections	159
11.1	Introduction	159
11.2	COMuICSer Storyboards	159
11.2.1	Structuring Scenarios and Annotating Personas	160
11.2.2	COMuICSer Storyboards for Multi-disciplinary Teams	160
11.2.3	The Transformation to Formal Artifacts	162
11.2.4	Including Non-functional Requirements	163
11.3	COMuICSer Storyboarding and the MuiCSer Process Framework	164
11.3.1	Mapping COMuICSer Storyboards to the MuiCSer Process Framework	164
11.3.2	Using COMuICSer Storyboards in Usability Evaluations	167
11.3.3	Assessing Artifacts by Means of COMuICSer Storyboards	169
11.4	COMuICSer Tool Support	170
11.4.1	Using the COMuICSer Tool in UCSE Projects	170
11.4.2	Cognitive Dimensions of the COMuICSer Tool	171
11.4.3	Multi-touch Storyboarding	174
11.5	Conclusion	177
12	Conclusions and Future Work	179
12.1	Summary	179
12.2	Answers to the Research Questions	182
12.2.1	RQ 1a: Are Storyboards and Accompanying Tool Support Useful for UCSE Practitioners?	182

12.2.2	RQ 1b: How are Storyboards Created in a Multi-disciplinary Team?	182
12.2.3	RQ 2: What Aspects of a Visual Language can Contribute to Storyboards in UCSE?	183
12.2.4	RQ 3: How can Storyboards be Used for the Transformation from Informal to Formal Artifacts?	184
12.2.5	RQ 4: How can Storyboards be Connected with Software Engineering Artifacts?	184
12.3	Future Work	185
12.3.1	Extended Tool Support for COMuICSer Storyboards	185
12.3.2	Longitudinal User Studies	186
12.4	Scientific Contributions and Publications	187
A	Documents of User Studies and Interviews	193
A.1	Semi-structured Interviews with UCSE Practitioners	193
A.2	First User Study	194
A.2.1	Personas	194
A.2.2	Scenario	198
A.2.3	Questionnaire	199
A.3	Observational Study	204
A.3.1	Personas	204
A.3.2	Scenario	206
A.3.3	Questionnaire	208
A.4	Interview and User Study with Agile SE Practitioners	212
A.4.1	Semi-structured Interview	212
A.4.2	First User Study	212
A.5	Interpreting Storyboards	212
B	Samenvatting (Dutch summary)	217
C	Glossary	219
	Bibliography	239

List of Figures

1.1	The process for human-centered design, recommended by ISO 13407 [Int99].	4
2.1	Our MuiCSer process framework. The dark arrow indicates the overall design and development direction. The light arrows indicate feedback from evaluation, verification and validation efforts.	12
2.2	Mapping of models and artifacts of <i>GUIDE</i> (Graphical User Interface Design and Evaluation) [RPM95] on the MuiCSer process framework.	14
2.3	Mapping of models and artifacts of <i>The Usability Engineering Lifecycle</i> [May99] on the MuiCSer process framework.	14
2.4	Mapping of models and artifacts of <i>Effective Prototyping</i> [AAB06] on the MuiCSer process framework.	15
2.5	Mapping of models and artifacts of <i>Rapid Contextual Design</i> [HWW04] on the MuiCSer process framework.	15
2.6	A timeline presenting the stages of MuiCSer and how artifact transformation tools can be mapped on it. The white area in a bar that represents a tool indicates a lacking or limited support of artifacts at that stage of MuiCSer.	19
3.1	MuiCSer process instances for two case studies.	25
3.2	Low- and high-fidelity prototype of the NewsWizard interface. The main part of the user interface concerns the wizard. The user can navigate between steps using arrow-buttons or tab pages.	27
3.3	Three levels of prototypes for one specific screen. From left to right: a Canonical Abstract Prototype (CAP), a low-fidelity prototype and a high-fidelity prototype.	29

5.1	A storyboard and its interrelationship with other artifacts in the UCSE process.	46
5.2	A screenshot of the COMuICSer storyboarding tool. This tool supports storyboarding by connecting a storyboard with a scenario, personas and other annotations.	49
6.1	First part of the storyboard that was used for the walkthrough of our study. The scenes and their titles are presented in a chronological order.	58
6.2	Second part of the storyboard that was used for the walkthrough of our study. The scenes and their titles are presented in a chronological order.	59
6.3	Setup of the observational study (A) and contents of the toolbox that was provided to each participant (B).	64
6.4	Frames of the videos that were recorded during each storyboarding session.	67
7.1	<i>Facial expressions</i> in a scene, show the emotions of a user. This user is frustrated by the application on his PDA.	78
7.2	<i>Body language</i> can be incorporated in storyboard scenes to amplify the emotions of users. The body language used in this scene, stresses that the user is frustrated.	79
7.3	<i>A photograph in combination with an iconic character</i> preserves the possibility to identify with the user, but shows a lot of contextual information.	80
7.4	<i>Differentiating characters</i> is possible by giving them contrasting looks. This allows readers to easily recognize different characters.	81
7.5	The <i>transition</i> between two scenes is automatically filled by the human brain. Although it is not explicitly depicted by these scenes, the reader deduces from the transition that the user in the scene moves to the desk and sits down during the transition.	83
7.6	The three storyboards that were analyzed in our user experiment. (To increase readability, we translated some words in the scenes.)	86
7.7	GUI elements of the tool that support detailed backgrounds and iconic characters in storyboard sketches.	88
7.8	Screenshot of the COMuICSer tool. The part of the screenshot that is enlarged shows the features to label (1) time information and (2) transitions.	90

8.1	Examples of informal and formal artifacts.	95
8.2	The MuiCSer process that was used for the design and development of <i>News Video Explorer</i> . Extracts of the most important artifacts are presented for each stage of MuiCSer. In this diagram, informal artifacts have a solid border, while formal artifacts are distinguished by a dashed border.	99
8.3	A timeline presenting the stages of MuiCSer and how artifact transformation tools can be mapped on it (repetition). The white area in a bar that represents a tool indicates a lacking or limited support of artifacts at that stage of MuiCSer.	100
8.4	Storyboard created for the development of the News Video Explorer, an application to visually explore video archives.	102
8.5	Our COMuICSer storyboard meta-model. It contains the graphical depiction with the objects of interest (context), personas, devices and activities. Scenes are related using the Allen interval algebra operators.	104
8.6	The model transformations based on a COMuICSer storyboard.	105
8.7	Activity to task transformation.	106
8.8	The Jelly multi-device design environment, showing a video browser design for a desktop computer (a). Jelly allows designers to copy widgets from one device (b) and to paste these widgets as a similar component on a different device (c).	108
8.9	The relationships between a COMuICSer storyboard and the UI design(s) created in the Jelly tool.	109
8.10	The Jelly multi-device design environment, showing an imported COMuICSer storyboard that can be used during the UI design.	110
9.1	Connecting user stories to a storyboard. Each storyboard scene can be connected to several user stories. This highly visual relationship between storyboard scenes and user stories, add contextual information to the user stories (e.g. in what environment is the system used?, is the system used by individuals?).	120
9.2	Example of the connection between a storyboard scene and a user story. The scene represents the context of use, including personas and device information, while the functional requirements are included in the user story.	121
9.3	User story creation in the COMuICSer tool, and the coupling to an agile planning tool such as Scrumdo [Scr].	124

9.4	Extract of the sketches that were drawn before the storyboard for the agile team was created.	126
9.5	Four scenes, extracted from the whole storyboard that was created during this study. Only limited descriptions and details are shown because of anonymity reasons.	127
10.1	Storyboard A - scenes and their titles: Location-aware Application for Museums.	138
10.2	Storyboard B - scenes and their titles: Customized Museum Tour.	140
10.3	Storyboard C - scenes and their titles: Serious Game for MS Patients. The face of the patient was anonymized because of privacy issues.	142
10.4	Storyboard D - scenes and their titles: Alarm and Help System for People Suffering from Dementia.	144
10.5	Storyboard E - scenes and their titles: Querying through Social Networks.	145
10.6	Two UI designs, created by the participants of this user study.	150
11.1	An overview of the extent to which our COMuICSer storyboarding research covers the MuiCSer process framework.	165
11.2	The user-centred process that was adopted for the development of novel video information retrieval visualizations for the TV broadcasting domain.	167
11.3	First part of a storyboard presenting how a multi-disciplinary team creates / discusses a COMuICSer storyboard in a meeting. The tool presented in this storyboard supports collocated storyboarding workshops.	175
11.4	Second part of a storyboard presenting how a multi-disciplinary team creates / discusses a COMuICSer storyboard in a meeting. The colors shown on the tabletop display, indicate the different contributions of the different members of a multi-disciplinary team.	176
A.1	Photograph of Mary, persona for an owner of a bistro, used for the first user study presented in Chapter 6 Section 6.2.	194
A.2	Photograph of Ann, persona for a usability engineer, used for the first user study presented in Chapter 6 Section 6.2.	195
A.3	Photograph of Thomas, persona for a developer, used for the first user study presented in Chapter 6 Section 6.2.	196

A.4	Photograph of Michael, persona for a graphic designer, used for the first user study presented in Chapter 6 Section 6.2.	197
A.5	Questionnaire used for the first user study presented in Chapter 6 Section 6.2	200
A.6	Photograph of Bob, persona for a banker, father, used for the observational study presented in Chapter 6, Section 6.3.	204
A.7	Photograph of Mary, persona for a teacher, mother, used for the observational study presented in Chapter 6, Section 6.3.	205
A.8	Photograph of Kate, student, used for the observational study presented in Chapter 6, Section 6.3.	206
A.9	Photograph of Benjamin, student, used for the observational study presented in Chapter 6, Section 6.3.	207
A.10	Questionnaire used for the observational study presented in Chapter 6, Section 6.3.	209
A.11	Questionnaire used for the observational study presented in Chapter 6, Section 6.3.	213

List of Tables

2.1	An association of tools that can be used to support MuiCSer and their accessibility for different roles in a multi-disciplinary team.	17
2.2	Overview of artifacts supported by artifact transformation tools.	18
3.1	The roles that are involved in the teams of the companies we interviewed. The different fills refer to different team members in the teams. The brackets depict team members that are only part of the team if the customer has one of these roles.	32
3.2	The models and artifacts that are created and used by the companies. The models/artifacts are sorted, based on the order in which they may be created in a MuiCSer process.	34
3.3	The tools that are used by the companies.	35
6.1	Overview of the most important results (observations) of our observational study.	68
9.1	A comparison of User-Centered Design and Agile Software Engineering	113
10.1	An overview of the storyboards that were collected for our study.	137
10.2	An overview of the observations for each storyboard in average (μ) and standard deviation (σ).	149
10.3	An overview of the responses to the questionnaire for each storyboard in average (μ) and standard deviation (σ). Participants were asked to answer to each question using a 5-point Likert scale.	151

10.4	An overview of the participants' satisfaction regarding the UI designs that were created in average (μ) and standard deviation (σ). Participants were asked to answer to each question using a 5-point Likert scale before and after the discussion with the author.	153
11.1	The the number of participants with particular roles that were involved in and the presence of collaborative aspects in the studies conducted within the context of the storyboarding research presented in this dissertation.	161
12.1	Overview of problems we identified in UCSE and the research challenges and questions for storyboarding that can be related to these weaknesses and difficulties.	180

Chapter 1

Introduction

This dissertation will present our research regarding the combination of user-centered design (UCD) and software engineering (SE). Before further describing our work, we will first introduce UCD and the need for combining UCD and SE in the next sections. Finally, we present an overview of the chapters presented in this dissertation.

1.1 User-Centered Design

Designers may not think of themselves as typical users but instead they should carefully consider the end users and their needs. In the late eighties, Norman stated that the designer of an everyday thing is not its user [Nor88]. This idea is not limited to everyday things, also software applications and their user interfaces (UIs) need to be designed with the end user in mind. Norman presented this as three models: the design model, the user's model and the system image. Furthermore, he proposed seven principles to support user-centered design.

In the years that followed, the UCD principles and ideas were gradually but slowly introduced in practice. It took a while until the general idea of UCD was translated into a UCD process [Kar97]. Nevertheless, individual techniques to support UCD such as Contextual Design, Participatory Design, Task Analysis, and Usability Testing were steadily introduced [But96].

In 1999, an ISO standard was published to guide practitioners in Human-centered Design [Int99]. This ISO standard defines human-centered design

as a design process that actively involves users and a multi-disciplinary team. Furthermore, this ISO standard recommends an analysis of users and the technology they will use and iterative design. In the remainder of this dissertation we will use the words *user-centered design* when we refer to practices that suit with the specifications of this ISO standard. Simultaneously, the Usability Engineering Lifecycle [May99] was introduced in order to recommend a process for UCD.

UCD gained popularity and importance in the last decade. Vredenburg et al. concluded in 2002 that practitioners prefer to apply UCD in multi-disciplinary teams and a set of common UCD methods was identified [VMSC02]. In 2003, Gulliksen et al. presented the key principles for user-centered design of systems [GGB⁺03]. Many of these principles fit with the ISO standard [Int99]. Preece et al. devoted a chapter to user-centered approaches [PRS02] in their handbook “Interaction Design - beyond human-computer interaction”.

We will briefly explain the typical characteristics of UCD in the following sections in order to obtain a clear understanding of UCD for the remainder of this dissertation.

1.1.1 The User Experience and Context of Use

The major reason to apply UCD processes in the past was to increase the usability of software systems. However, nowadays, the usability is only one of many quality attributes of software systems and user interfaces. The whole of quality attributes for user interfaces is also referred to as the user experience and includes sociability, playability, accessibility, utility, learnability and likeability [CHB05].

While the first software systems were only intended for professional use, the past decades systems have become available for a plethora of computing platforms and a wide range of users. The usage of software systems is not limited to professional use anymore and a diversity of computing devices and platforms allows the usage of computers almost everywhere. Consequently, more quality attributes than ease of use have become important for end users of software systems. Because the usage of computing devices is extended from work environments to a wider environment including home, public places and the car, the context in which a system can be used is also gaining importance.

1.1.2 Roles Involved in Multi-disciplinary Teams

UCD processes should involve multi-disciplinary teams because several aspects are considered when creating technical systems for human beings. When a

multi-disciplinary team is involved, many complementary ideas are suggested, creativity is encouraged and new methods arise [PRS02]. Disciplines involved in UCD include computer science, human-computer interaction (HCI), graphic design, product design, cognitive ergonomics and engineering. Often, these disciplines involved in UCD fall under the heading *interaction design*, and consequently, UCD practitioners are often called interaction designers.

The ISO standard recommends the involvement of several roles in UCD: end user, purchaser / manager of user, application domain specialist / business analyst, systems analyst / systems engineer / programmer, marketer / salesperson, UI designer / visual designer, human factors and ergonomics expert / HCI specialist, technical author / trainer / support personnel [Int99]. However, it is not necessary that all these roles are represented by different people in the team. Some roles may overlap, or in particular projects, some roles not necessarily need to be involved (e.g. technical author).

In this dissertation, we will refer to these roles of a multi-disciplinary team, as presented in the ISO standard. One difficulty in this list of roles, is to identify the exact responsibilities of a designer. Since most of these roles concern interaction design, almost all team members can be considered as interaction designers. The different types of designers that are part of the list in the ISO standard can create the visual design of a UI, can be responsible for the design of the system architecture and/or can design the UI and its behavior. Since this dissertation concentrates on the different backgrounds involved in UCD, we clearly have to distinguish who we consider if we mention the involvement of a designer. When the role of a designer is considered in this dissertation, we consider a designer that creates the UI and its behavior, unless we explicitly specify their expertise (e.g. visual designer).

1.1.3 User-Centered Design Processes

The ISO standard defines User-Centered Design as an iterative process [Int99]. Furthermore, this standard suggests several design activities that should be taken for the development of a system. These activities are visualized in a circular and iterative process, as shown in Figure 1.1. Figure 1.1 shows that the process emphasizes the involvement of the user from the beginning.

Other examples of processes or approaches that support the principles of UCD are GUIDE [RPM95], Rapid Contextual Design [HWW04] and Effective Prototyping for Software Makers [AAB06]. A comparison of these processes is discussed in Chapter 2.

The involvement of UCD is not limited to conducting usability tests after

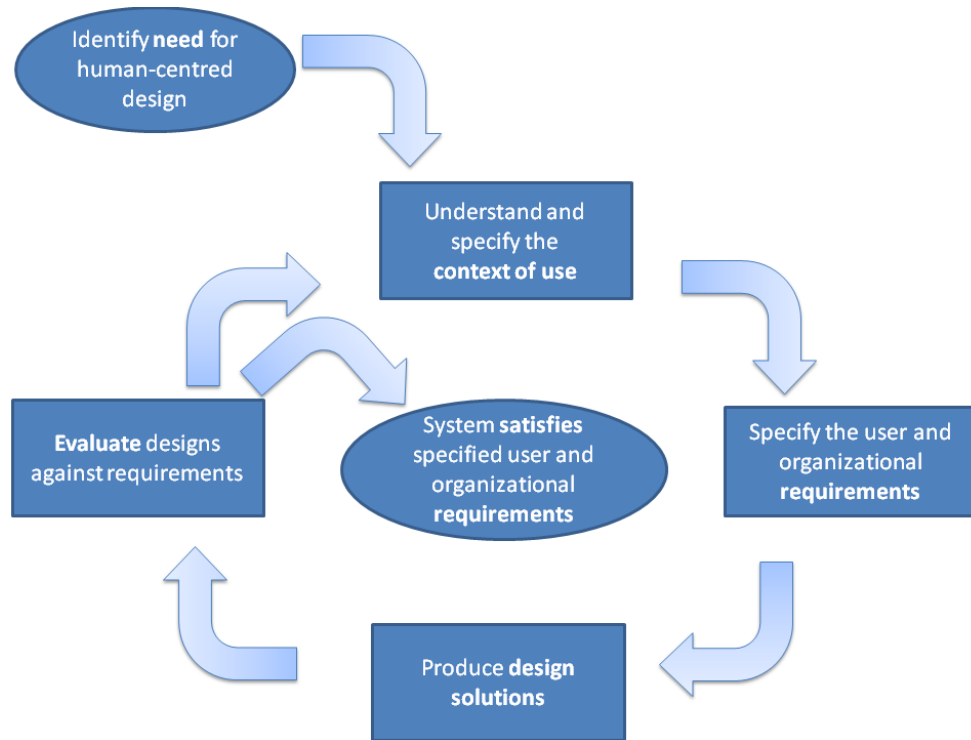


Figure 1.1: The process for human-centered design, recommended by ISO 13407 [Int99].

the design and development of a system. All UCD processes emphasize the involvement of end users from the beginning of the design and development process. Before the elicitation of requirements and user needs, an analysis of a representative group of end users is suggested. Contextual Design [BH98] is an example of a technique that is used for conducting user needs analysis. Usually, a user needs analysis is conducted by UI designers or HCI specialists. Based on the results of the user needs analysis, user needs and requirements are elicited. Typical artifacts that are used to represent the results of a user needs are reports in natural language, scenarios of use representing how the future system can be used [Car00] and personas that represent target users in a fictional and hypothetical description [PA06].

Once the user needs and requirements are specified, the UI design can take place. UI designs can have several styles and usually evolve from sketchy

low-fidelity prototypes representing the first UI screens and behavior to high-fidelity prototypes representing a realistic interactive UI. For each design artifact that is created during the design of a UI, it is recommended to evaluate these design artifacts in usability tests that involve representative end users.

1.2 Combining User-Centered Design and Software Engineering for Multi-disciplinary Teams

The collaboration of multi-disciplinary team members is beneficial for UCD because of complementary viewpoints. However, the involvement of a wide range of disciplinary backgrounds may also impede the communication in the team and the transfer of information [HCB⁺06].

Many techniques that originated from UCD concentrate more on the end user needs and requirements than the functional requirements and aspects of a system. Since software engineering is indispensable in the development of software systems, the combination of UCD techniques and practices from software engineering has to be facilitated.

Existing work presents several approaches and techniques that are suited to combine user-centered design and software engineering. Constantine launched the term “Usage-centered design” [Con96] which aims to combine HCI and software engineering. Usage-centered design concentrates particularly on the use of a system rather than understanding the users, because for the design and development of a system, it is important to understand what users (try to) do instead of who the users are. Constantine introduced essential use cases [Con95] and canonical abstract prototypes [Con03], as notations to support usage-centered design.

Sutcliffe addressed possible techniques to integrate software engineering and HCI [Sut05]. He states that expert designers often rely on their skills rather than documenting knowledge during a development process and he stresses that the use of models and scenarios may facilitate the representation of reusable knowledge. The use of scenarios to support design in multi-disciplinary teams and in several stages of a design and development process was described and exemplified by Carroll [Car00].

Although these and several other approaches and techniques were introduced in the decade before the start of this PhD, multi-disciplinary teams that combine UCD and software engineering are still facing difficulties. In this dissertation, we first identify difficulties and weaknesses in user-centered software engineering. Following, we propose possible approaches for user-centered soft-

ware engineering in multi-disciplinary teams. The next section presents an overview of the research that is described in this dissertation.

1.3 Overview

In Part I of this dissertation, Chapter 2 proposes MuiCSer, a process framework for multi-disciplinary user-centered software engineering. This process framework is used for applied user-centered software engineering (UCSE) projects and for investigating the weaknesses and difficulties of UCSE. Chapter 3 describes case studies and the responses of interviews with practitioners in UCSE. The results of the studies based on the MuiCSer process framework led to an overview of weaknesses and difficulties in UCSE, which is presented in Chapter 4.

Several approaches to overcome the weaknesses and difficulties in UCSE by using storyboards are proposed in Part II of this dissertation. Chapter 5 introduces COMuICSer storyboarding, and an accompanying tool, which can be used for UCSE. Furthermore, we list research challenges and related research questions with respect to COMuICSer in this chapter.

In order to investigate storyboarding in multi-disciplinary teams, we conducted two user studies, presented in Chapter 6. The results of these user studies inspired us for further research regarding storyboarding. Chapter 7 describes how techniques of comics can be incorporated into the visual storyboarding language. By incorporating these techniques into the COMuICSer tool, conveying a message by using a COMuICSer storyboard should be simplified.

Besides using storyboards as a communication tool, COMuICSer storyboards are also intended to structure requirements and facilitate the transitions between artifacts in UCSE. In Chapter 8 we present techniques that use storyboards to transform informal artifacts into formal models. Chapter 9 concentrates on connecting storyboards and user stories used in agile software engineering to capture the requirements of a system that will be developed.

The last and concluding part of this dissertation, Part III, assesses some aspects of storyboarding and reflects on several facets of our COMuICSer storyboarding approach. Chapter 10 presents a user study that evaluated some aspects regarding the interpretation of storyboards. Further reflections on COMuICSer are discussed in Chapter 11. Finally, Chapter 12 describes possible directions for future work as well as our conclusions.

Part I

A Process Framework for Multi-disciplinary User-Centered Software Engineering

Chapter 2

The MuiCSer Process Framework

2.1 Introduction

User-Centered Design (UCD) approaches have proven their value for interactive systems development for new as well as for legacy systems. Because diagrams and models related to the application back end (e.g. UML diagrams) often result from a software engineering (SE) process, there is a search to combine principles and practices from the SE domain and UCD approaches in order to define an overall process that fulfills the needs of a multi-disciplinary design team. We coin the processes that unite both HCI and SE perspectives as *User-Centered Software Engineering (UCSE) Processes*.

Based on former research results, we explore extensions of model-based user interface development approaches to bridge the gap with SE approaches such as model-based design [Pat99]. A model-based approach typically employs different types of models, thereby conveying enough information to generate the skeletons for concrete user interfaces. Models still tend to emphasize the more technical phases in application development over the creative design phase and overall development cycle. Overcoming these shortcomings in a unified HCI and SE approach, and paying attention to multi-disciplinary teams are a necessity to allow for a pragmatic approach and applicability of model-based techniques in real-world projects.

To accommodate for both flexibility in selecting the techniques for one particular UCSE process and consistency in models and consecutive developments, we prefer starting from a conceptual process framework rather than a

single, exhaustively defined UCSE process. The conceptual process framework can be considered as a generic process that can be customized or instantiated for the specific design task at hand. Though UCD research in the HCI community is focused on processes, process frameworks are gaining importance in the software engineering community (e.g. The Eclipse Process Framework [Ecl]). Furthermore, Gulliksen et al. stated that user-centered system design processes should be customized for each organization [GGB⁺03]. Therefore, we believe this approach is helpful to strive at the same time for practical processes for applied research and for a comparison and evaluation framework, driving research activities regarding models, development artifacts and tools. In this chapter, we propose a UCSE process framework and detail the tools, models and artifacts that support the approach.

2.2 MuiCSer

We propose a process framework that focuses on the end user needs during the entire design and development cycle. This process framework for Multi-disciplinary user-Centered Software engineering processes is called MuiCSer, which is pronounced as “mixer”. One of the major goals of MuiCSer is to support design of high quality user experiences, provided by the software that is delivered. The user experience is typically determined by measuring aspects such as usability, accessibility and availability of required functionality of the delivered application.

In former applied research and software development projects with industry, we gradually introduced model-based approaches while we were part of multi-disciplinary teams. Our conceptual process framework, which is partly based on these experiences, organizes the creation of interactive software systems by a multi-disciplinary team. We support different models throughout processes that are derived from the framework, where each model describes a specific aspect of an interactive system and represents the viewpoint of one or more specific roles in the multi-disciplinary team. The need for communication with end users or customers results in additional models or artifacts (e.g. low-fidelity and high-fidelity prototypes) on top of the commonly used models in a model-based approach (e.g. task model or domain model [Pat00]). This has also a positive effect on the visibility and traceability of the processes that are based on our process framework, in particular when artifacts are stored in a central repository: the models and artifacts describe the status of the system being designed at various stages, support the design decisions made during these phases and are ready for use in the next iteration.

Both functional and non-functional requirements are tackled by the process framework and unlike traditional software engineering processes, it supports processes with a continuous and smooth integration between user interface design and software development. Figure 2.1 gives an overview of the proposed process framework. The MuiCSer process framework supports several stages, which on their turn allow for the creation and use of particular models and artifacts. The next paragraphs describe the stages of the MuiCSer process framework in more detail. Models and artifacts are discussed in Section 2.3

MuiCSer processes typically start with an analysis phase in an initial iteration where the users' tasks, goals and the related objects or resources that are important to perform these tasks are specified (Figure 2.1 A). If the quality of user experiences of a legacy system needs to be improved, the functionality of such a system can be often found in existing manuals and software documentation, which also contribute to the analysis. Several notations are used to express *the requirements and user needs* that result from this analysis phase: HCI experts take a user-centered approach and commonly express usability requirements in narrative documents, a conceptual model, personas [PA06], a scenario [Car00] and a task specification [PRS02, Gre02]. The software engineers specify the required behavior of the system with use cases and behavior diagrams (e.g. UML). Although the relationship between both is clear, linking them in an engineering process remains a difficult issue. However, as the process framework assists in defining what artifacts are important in which stages and how progress from abstract to concrete models can be realized, the framework facilitates to identify, create and relate the required models in each stage.

The results of the analysis are used to proceed towards *structured interaction models*, including task models [Pat00], system interaction models and presentation models (Figure 2.1 B). These models can be expressed using the UML notation [Mar02], thus maintaining alignment with the traditional SE models.

Since user needs as well as functional information are specified in the first two stages (Figure 2.1 A and B), they can both serve as input for the *low-fidelity prototypes* (Figure 2.1 C). User interface designers create mockups of the user interface, based on the information contained in the task and interaction models, while using design guidelines and their experience. In subsequent phases, low-fidelity prototypes are transformed into *high-fidelity prototypes* (Figure 2.1 D), which in their turn evolve into the *final interactive system* (Figure 2.1 E).

In a UCSE process, that is supported by MuiCSer, each stage is related to

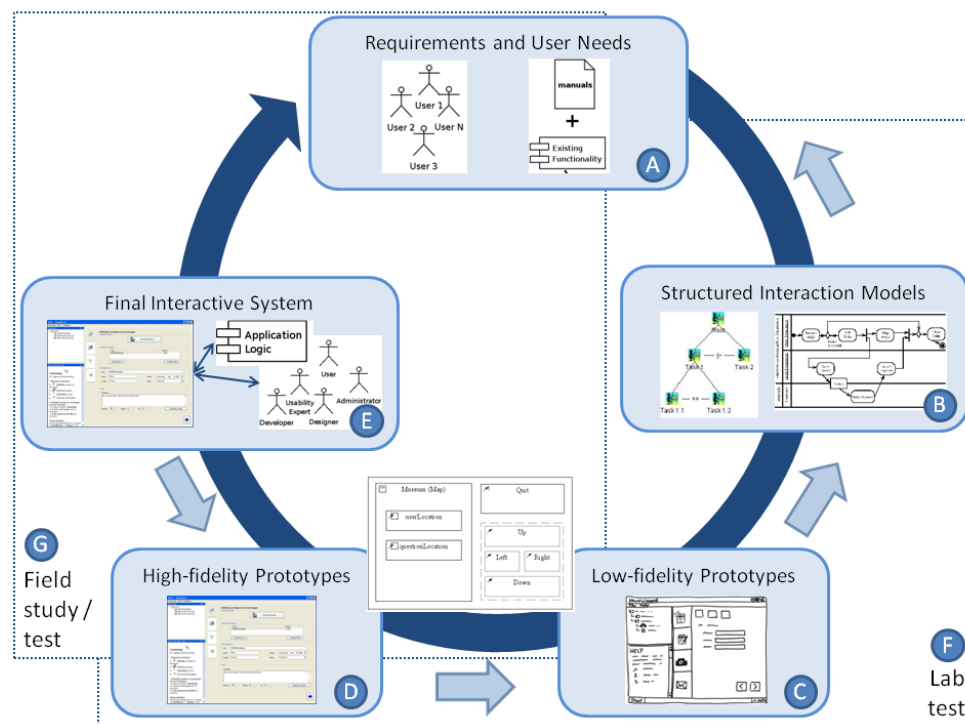


Figure 2.1: Our MuiCSer process framework. The dark arrow indicates the overall design and development direction. The light arrows indicate feedback from evaluation, verification and validation efforts.

the artifacts created in a previous stage. By evaluating the result of each stage, the support for user needs and goals and the presence of required features is verified. These evaluations in the MuiCSer process framework are indicated by the light blue arrows in Figure 2.1. If possible, an evaluation with target users can be very useful to get feedback from the end user directly. Because most of the artifacts do not present a fully functional system, part of the evaluations concern *lab testing* (Figure 2.1 F). In a lab test, the prototype is evaluated in a controlled study which usually takes place in a usability lab that simulates the natural environment of end users. To evaluate some advanced prototypes, *field testing* (Figure 2.1 G) can be used as a technique to examine the user interface in more realistic situations. If the results of a phase are not suited (e.g. too complex) to involve an end user during evaluation, it is still necessary to evaluate, verify or validate the models or prototypes, e.g. in meetings with domain experts or by performing an expert evaluation.

2.3 Models and Tools

Now we introduced the MuiCSer process framework, we exemplify to what extent it is covered by existing user-centered processes and tools for the creation and transformation of artifacts and models used in UCSE. The study of the current availability of tools reveals in what stages tool support should be improved and how the collaboration within multi-disciplinary teams can be supported.

2.3.1 Existing User-centered Processes

Existing UCD approaches such as GUIDE [RPM95], The Usability Engineering Lifecycle [May99], Effective Prototyping [AAB06] and Rapid Contextual Design [HWW04] can be represented using this framework. We investigated these approaches, based on the artifacts and models they suggest.

Figures 2.2 to 2.5 show the mapping of the models and artifacts of existing user-centered approaches on our MuiCSer process framework. These figures show how the different approaches can be situated on the MuiCSer process framework. Rapid Contextual Design, for instance concentrates on the first three stages of MuiCSer (Figure 2.5), while GUIDE also includes high-fidelity prototyping (Figure 2.2). Effective prototyping focuses on the low- and high-fidelity prototyping stages (Figure 2.4) and the Usability Engineering Lifecycle covers all stages of MuiCSer (Figure 2.5). Although not all MuiCSer stages are covered by these approaches, they do support the principles of MuiCSer. Like-

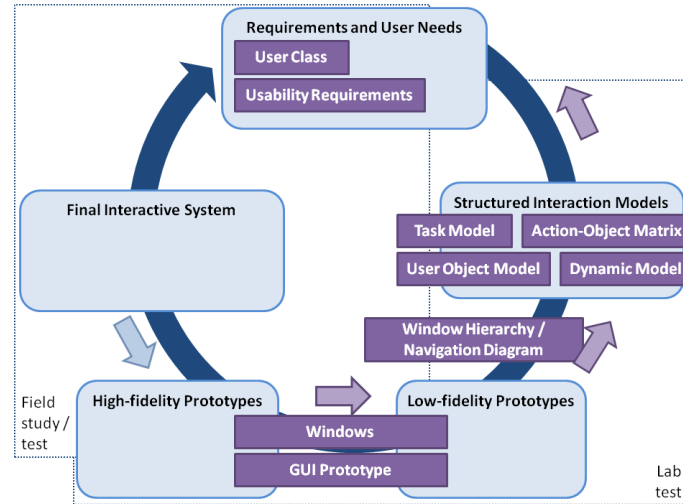


Figure 2.2: Mapping of models and artifacts of *GUIDE* (Graphical User Interface Design and Evaluation) [RPM95] on the MuiCSer process framework.

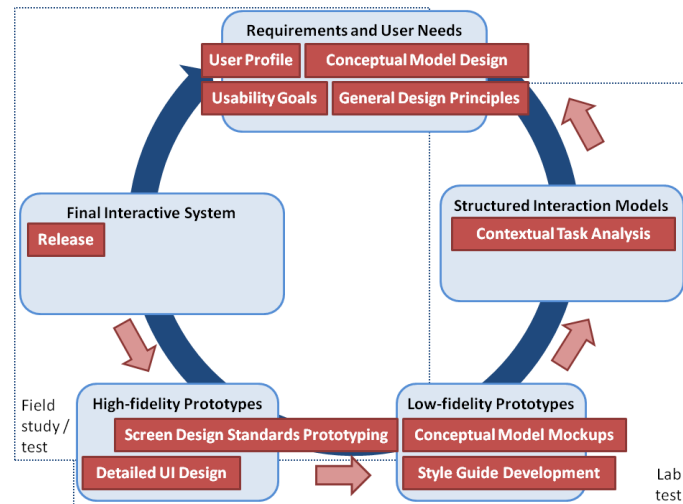


Figure 2.3: Mapping of models and artifacts of *The Usability Engineering Lifecycle* [May99] on the MuiCSer process framework.

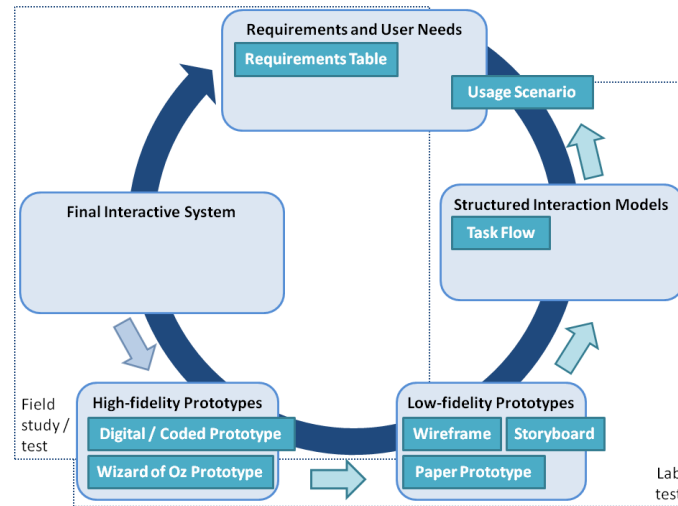


Figure 2.4: Mapping of models and artifacts of *Effective Prototyping* [AAB06] on the MuiCSer process framework.

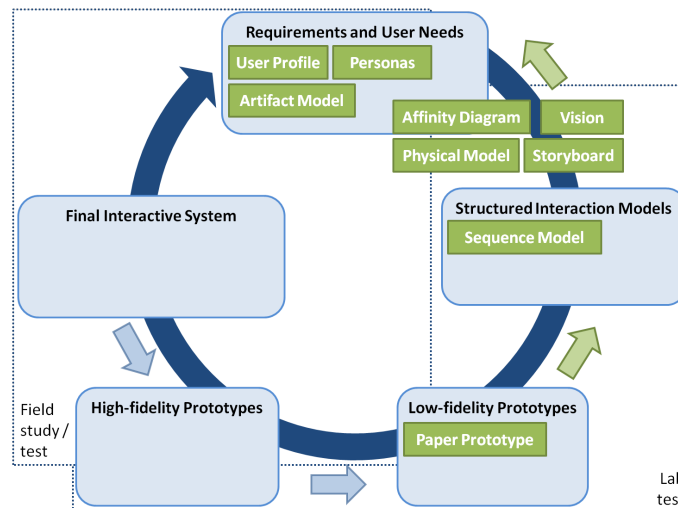


Figure 2.5: Mapping of models and artifacts of *Rapid Contextual Design* [HWW04] on the MuiCSer process framework.

wise, when UCSE processes are derived from MuiCSer, the multi-disciplinary team can decide to what extent particular stages of MuiCSer are covered. Nevertheless, it is highly recommended to carefully consider all stages of MuiCSer in order to obtain a system that corresponds to the requirements and user needs.

2.3.2 Artifact transformation tools

A literature survey, conducted in order to investigate models and tools used in UCD and UCSE that fit parts of the MuiCSer process framework gave rise to Table 2.1. We started from the roles that typically can be part of a multi-disciplinary UCD team, as described by ISO 13407 [Int99] and other sources that report about tools used by particular roles in multi-disciplinary teams [AAB06, HWW04]. Following, we associated the tools with the roles, based on the tools' descriptions in literature.

Table 2.1 shows the first six tools are widespread. Consequently these tools are very accessible for different roles of the multi-disciplinary team, which is confirmed by Campos and Nunes [CN07a]. In contrast to the widespread tools, the other fourteen tools shown in the table are mainly supporting practices of designers and developers. The last ten tools of Table 2.1 are specifically developed in order to support user-centered approaches and connect at least two artifacts or models created and used in UCSE. We use the term *artifact transformation tool* to describe this group of tools which allow UCD practitioners to progress the design and development of an interactive system involving different roles, often by providing different views on the same artifact or model. The ways in which a tool can manipulate, create relations or transform between artifacts and models are summarized in [CLC04].

By first mapping these artifact transformation tools on the artifacts they support as shown in Table 2.2, we can see what stages of MuiCSer are supported by the particular tools. The latter is shown in the timeline of Figure 2.6. Most tools that are suitable for interactive, incremental and multi-disciplinary user-centered processes are artifact transformation tools, which comes as no surprise. Figure 2.6 also shows that it is possible to combine two or three tools to cover most stages of MuiCSer. While Teresa [MPS04], for instance, can be used to model tasks of a multi-platform application and generate a system task model, an abstract user interface and a concrete user interface, Gummy [MVL08] can be used by designers to add creative aspects to the medium- to high-fidelity prototypes for multi-platform user interfaces.

The overview of tools in Figure 2.6 also reveals that limited tool support is

Tools	Roles multi-disciplinary team							
	End user	Purchaser, manager of user	Application domain specialist	Systems analyst, programmer	Marketer, salesperson	UI designer, visual designer	HCI & ergonomics specialist	Technical author, trainer, etc.
Word processor [AAB06]		✓	✓	✓	✓	✓	✓	✓
Presentation [AAB06]		✓	✓	✓	✓	✓	✓	✓
Spreadsheet [AAB06]		✓	✓	✓	✓	✓	✓	✓
Drawing [AAB06]		✓	✓	✓	✓	✓	✓	✓
Paper [AAB06]	✓	✓	✓	✓	✓	✓	✓	✓
PDF viewer [AAB06]		✓	✓	✓	✓	✓	✓	✓
Paint Program [AAB06]						✓		✓
Simple Programming [AAB06]				✓				
HTML (site) editor [AAB06]				✓		✓		
Animation tool [AAB06]						✓		
Advance programming [AAB06]				✓				
CTTE [MPS02]				✓		✓		
ActivityDesigner [LL08a]				✓		✓		
TaskSketch [CN07a]				✓		✓		
Vista Environment [BGW98]				✓		✓		
CanonSketch [CN07a]				✓		✓		
Teresa [MPS04]				✓		✓		
SketchiXML [CKV07]	✓			✓		✓		
Damask [LL08b]				✓		✓		
GrafiXML [MV08]				✓		✓		
Gummy [MVLC08]				✓		✓		
IntuiKit [CSV ⁺ 04]				✓		✓		

Table 2.1: An association of tools that can be used to support MuiCSer and their accessibility for different roles in a multi-disciplinary team.

Artifact transformation tools	Artifacts								
	Scenario [MPS02]	Use Case [MPS02]	Task Model [MPS02, BGW98, CN07a]	Domain Model [CN07a]	Activity Diagrams [CN07a]	User Interface Architecture [BGW98]	System Architecture [CN07a]	Abstract UI [MPS04, CN07a]	Concrete UI [MPS04, MVLC08, CSV ⁺ 04]
CTTE [MPS02]	✓	✓	✓						
ActivityDesigner [LL08a]	✓				✓				✓
TaskSketch [CN07a]		✓	✓		✓		✓		
Vista [BGW98]			✓	✓		✓			
CanonSketch [CN07a]				✓				✓	✓
Teresa [MPS04]			✓					✓	✓
SketchiXML [CKV07]			✓	✓				✓	✓
Damask [LL08b]									✓
Gummy [MVLC08]									✓
GrafiXML [MV08]									✓
IntuiKit [CSV ⁺ 04]									✓

Table 2.2: Overview of artifacts supported by artifact transformation tools.

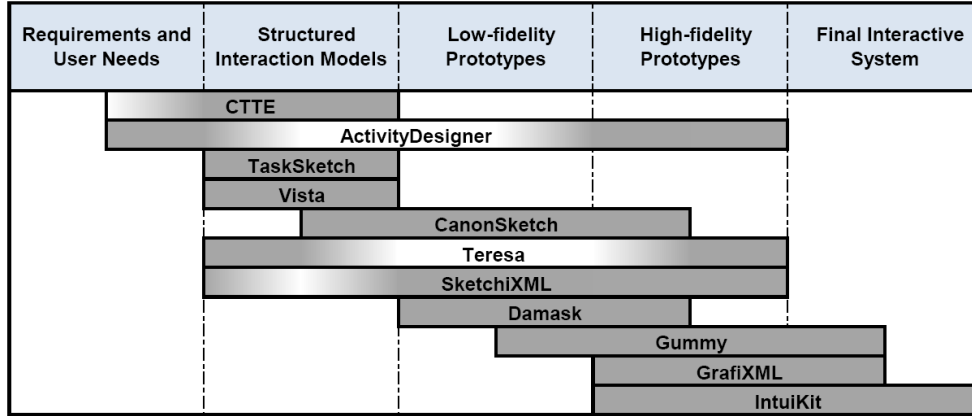


Figure 2.6: A timeline presenting the stages of MuiCSer and how artifact transformation tools can be mapped on it. The white area in a bar that represents a tool indicates a lacking or limited support of artifacts at that stage of MuiCSer.

available for the transformation of requirements and user needs into structured interaction models. Furthermore, there is no single tool that completely covers MuiCSer, which could be beneficial when a new iteration takes place after an interactive system is deployed. The main drawbacks of most of these tools are their inaccessibility for non-experts because of their specialized notations and their relative immaturity for real-world software development processes. Several of the aforementioned tools are increasingly being used in industrial projects, so we expect that the influence of practitioners will improve this situation. SketchiXML for instance is already suitable to be used by a wider range of roles including designers and end users [CKV07]. Gummy supports the roles of software developers and designers but this tool is gradually being extended to be used by application domain specialists [LMVC08].

The following sections describe different models being created, changed and transformed during the execution of MuiCSer processes in order to support a smooth transition towards the final interactive system. The use of these models and tools is not required, but they provide a clear idea of how MuiCSer processes can be instantiated with concrete models, notations and tools.

2.3.3 Requirements and User Needs

The first stage of the MuiCSer process framework starts with an analysis of the user needs and possibly a study of the legacy system. The user needs analysis usually results in report written in a narrative style. By narrative style we mean documents that contain structured natural language, this style of presenting information usually is very accessible to the members of multi-disciplinary teams, but often hard to abstract for design and development purposes. Some other narrative artifacts that are created right after a user needs analysis are *personas* [PA06] that represent hypothetical archetypes of users and *scenarios* [Car00] that may include personas and describe how a future system is used. Both personas and scenarios are presented in a narrative style. No specific tools are available to specify personas or scenarios, the available widespread tools such as a word processor suffice for the HCI or application domain that create these artifacts.

2.3.4 Structured Interaction Models

Task models are frequently used to specify requirements for an application from a user's point of view. Most task models have a hierarchical structure, allowing a gradual refinement of the high-level tasks and goals into fine-grained actions and activities. A task specification for a system can be found by transforming the requirements text and the scenarios into a hierarchical task model with temporal operators, such as the ConcurTaskTrees notation [Pat00]. Although this step is not automated, the UI designer performing this step uses a set of (informal) rules and is supported by a tool such as CTTE [MPS02].

This task model can be related to other user interface and software engineering models expressed using e.g. UML diagrams, which are widely known by software analysts and programmers. These *user interface models* can provide an alternative view on the information captured in the task model [NNC05, VC05] or additional information [NeC00, VC05].

2.3.5 Low-fidelity Prototypes

Since the creativity of designers and other members of a multi-disciplinary team may influence the user experience in a positive way, MuiCSer does not imply the use of specific tools or technologies to create low-fidelity prototypes. The first *mockups* can be created using pencil and paper or using a tool. Tools such as SketchiXML [CKV07] or CanonSketch [CN07a] have the advantage that they provide support for the transition to high-fidelity pro-

totypes. The ability to make the transition from low-fidelity to high-fidelity using these tools and notations is illustrated by the drawing of a Canonical Abstract Prototype [Con03, VDBBM⁺07] between the low-fidelity and the high-fidelity stage in Figure 2.1.

2.3.6 High-fidelity Prototypes

For the high-fidelity prototyping stage, design and development tools that support serialization of the user interface design to (high-level) XML-based languages are preferred. This allows more rapid prototyping of user interfaces that support a common set of tasks. Tools such as Gummy [MVLC08] or GrafiXML [MV08] additionally have specific support for adapting the designs to different platforms, screen sizes or in general different contexts of use. A loose coupling with the application logic is preferred to enable reuse.

2.3.7 Final Interactive System

To speed up development of the final user interface and to make it as flexible as possible, it is preferable to reuse the developed artifacts (e.g. XML-based high-fidelity prototypes and even selected models) as much as possible. A flexible user interface management system allows the use of these models at runtime and supports all stages of the MuiCSer process framework. Coupling for example the task model to the user interface descriptions allows to check for task coverage of the user interface and even selection of a subset of features for certain users while ensuring that all remaining tasks are still valid. Using these artifacts in the final interactive system also ensures that they are still available and up-to-date for the development of future increments.

2.4 Conclusion

In this chapter we introduced our MuiCSer process framework for multidisciplinary user-centered software engineering. The goal of the process framework is twofold. On the one hand, UCSE processes can be derived from MuiCSer for UCSE practices. On the other hand, the process framework is defined in order to structure and associate models and tools used in UCSE. The definition of the framework stimulates the use of customized processes that pay explicit attention to consistency of design and development artifacts throughout the different cycles of the process. Furthermore the MuiCSer

process framework focuses on multi-disciplinarity, which is indispensable in UCSE.

Processes derived from MuiCSer support techniques used by developers as well as the creativity of designers and developers. Furthermore, similar to UCD processes [Int99], a positive user experience is likely to be obtained from MuiCSer processes. Because of the flexibility offered by the MuiCSer process framework, UCSE processes that are derived from MuiCSer can include models and artifacts depending on the project characteristics, the future system that is to be developed and the project team that is involved. In contrast to existing user-centered approaches, MuiCSer does not constrain practitioners in the use of models, artifacts and tools. Each iteration of a MuiCSer process produces one or more interactive prototypes to allow continuous user involvement and evaluation and to enhance the visibility of this process.

The comparison and association of existing user-centered approaches and UCSE models and tools based on the MuiCSer process framework, revealed that not one single tool covers all stages of MuiCSer. Furthermore, tools to transform informal artifacts (e.g. user needs) into formal structured interaction models are lacking. We used our MuiCSer process framework for further investigations regarding UCSE. In Chapter 3 we describe how MuiCSer was used to investigate the shortcomings of UCSE approaches. Following, the challenges in UCSE, that are identified in this first part of this dissertation are described in Chapter 4. Part II proposes storyboarding approaches to overcome these challenges in MuiCSer processes.

Chapter 3

User-Centered Software Engineering in Practice

3.1 Introduction

The process framework for multi-disciplinary user-centered software engineering, MuiCSer, which was introduced in Chapter 2 aims to facilitate a flexible selection of techniques used in user-centered software engineering (UCSE) projects while stimulating multi-disciplinary teams to maintain consistency of artifacts throughout different process cycles. As already described, the framework provided by MuiCSer allowed us to investigate the use of models and tools in UCSE, based on literature. This revealed some interesting issues for further research. Nevertheless, we also want to examine what issues arise in UCSE practices.

In this chapter we describe how MuiCSer processes are used in practice. First we present some case studies in which processes of MuiCSer were derived for applied research. These processes for UCSE exemplify the use of the MuiCSer framework. Lessons learned, which are based on our experiences when applying these processes, are described. Next, we will describe interviews with practitioners which provided us with an overview of the roles involved in their teams and their use of artifacts, models and tools. Both the case studies and the interviews resulted in a better understanding of the needs of multi-disciplinary teams that practice UCSE.

3.2 Case Studies

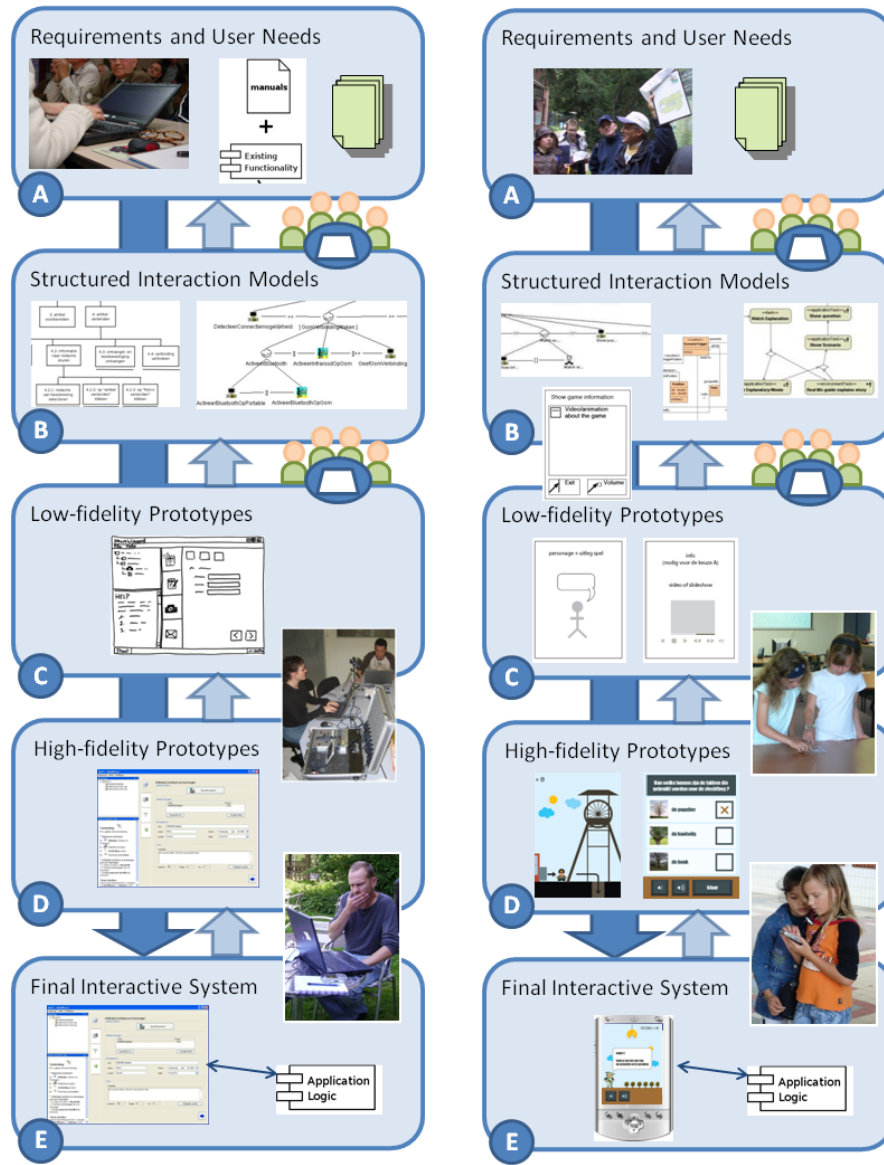
We explain how MuiCSer can be used by describing two MuiCSer processes that are instantiated for two case studies, carried out in 2005 to 2007 within the VIP-lab project [CHB05]. The aim of the project was to disseminate the user-centered approach of applied projects to companies and organizations in Flanders and the Netherlands. The result of each case study was an evaluated high-fidelity prototype. The company or organization, involved in the case study was responsible for the development and deployment of a final interactive system. Our initial versions of MuiCSer were realized simultaneously with these case studies.

The first case study concerns the redesign of a legacy system while the second case study presents the approach that has been used for the design of a new system. The project team was not limited to computer scientists but also psychologists and social scientists and in some case studies a graphic designer were involved. Figure 3.1 shows an overview of the MuiCSer processes that are employed for these case studies. For sake of clarity of the presentation and to allow comparison, both processes are shown as a linear path without emphasis on the intra- and inter-stage iterations.

3.2.1 NewsWizard

When a mobile reporter is on location, she has more concerns than writing an article. A major challenge is often to configure a network connection to send the article to the editorial staff. When the study was carried out, in 2005-2007 this was certainly a realistic situation. The NewsWizard prototype, which was developed in this case study in cooperation with a news publishing agency, should ease the job of a journalist on location by guiding her while making the appropriate network connection and sending the article(s) and photos.

As recommended by MuiCSer, first the legacy systems have been explored in order to obtain *requirements and user needs*. Manuals of the existing editing software for journalists that support writing and sending articles have been studied and the systems were demonstrated to the project team. Next, journalists and photographers were observed and interviewed by social scientists while they were collecting information and sending it to the editorial office. Besides the comparison of the job of a contemporary journalist and a photographer, this contextual inquiry [BH98] resulted in personas [PA06] and scenarios [Car00] (Figure 3.1(a)-A)). For these activities a word processor and



(a) The NewsWizard case study.

(b) The mobile game case study.

Figure 3.1: MuiCSer process instances for two case studies.

PDF viewer were used.

At the second stage of this process which concerned the creation of *structured interaction models*, task models were created by developers using the Hierarchical Task Analysis (HTA) [PRS02] and ConcurTaskTrees (CTT) [MPS02] notations (Figure 3.1(a)-B). The tools used for this stage concerned a common drawing tool for the HTA diagram and CTTE [MPS04] for the CTT. The evaluation of these task models was carried out during meetings with the project team. The social scientists checked consistency with the observations, the personas and the scenarios while the computer scientists examined the technical possibilities. Representatives of the news publishing agency evaluated the design according to the needs of the journalists and their own expectations. The task models were refined within two iterations. The threshold for progression in this project was the agreement of the team members on structure and content of the task model, scenarios and personas.

By putting together requirements, user needs and structured interaction models, it became clear that journalists mainly experience problems when they need to send an article on location. Consequently a user interface in wizard-style was designed to collect articles and load pictures (e.g. when the journalist is not accompanied by a photographer), and to send the data successfully. The relations between the task model and the *low-fidelity prototype* on paper (Figure 3.1(a)-C and Figure 3.2, left) were determined manually and the prototype was checked for completeness with relation to the task model during meetings, similar to the meetings held during the structured interaction analysis stage.

In order to have a prototype that could be evaluated by journalists in a usability lab, the low-fidelity prototype of NewsWizard evolved soon into a *high-fidelity prototype* (Figure 3.1(a)-D and Figure 3.2). Although this was done manually by developers in an advance programming tool, there is a clear one-to-one mapping from each component in the low-fidelity prototype to each component in the high-fidelity prototype. By consequence, the high-fidelity prototype is also complete with respect to the task model. In three iterations and increments the NewsWizard prototype was developed and functionality was added. After each iteration and increment the UI was evaluated by journalists during a lab test (Figure 3.1(a)). In order to evaluate the prototype in the natural environment of a reporter, some field tests were carried out (Figure 3.1(a)-E). During these field tests, the participating journalists were observed and interviewed while accomplishing a realistic assignment on location using NewsWizard. The general observations showed that the use of NewsWizard was much more intuitive than using the existing system. Most

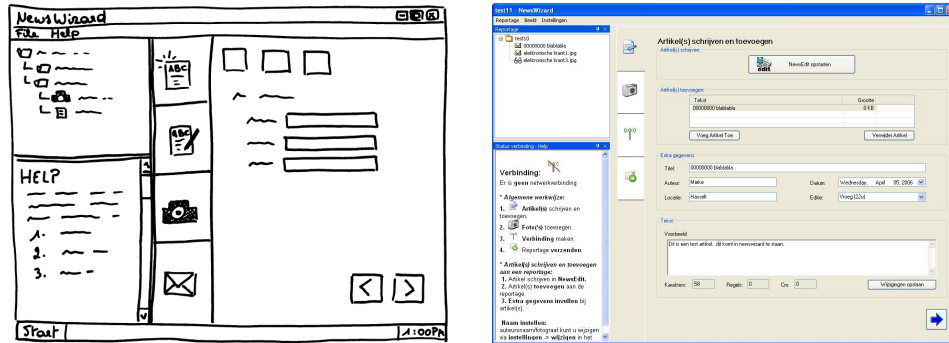


Figure 3.2: Low- and high-fidelity prototype of the NewsWizard interface. The main part of the user interface concerns the wizard. The user can navigate between steps using arrow-buttons or tab pages.

journalists confirmed that in the future they would rather send articles from location instead of going back to their desk if they could use the NewsWizard.

3.2.2 Mobile Game for Children

A second case study concerns the development of a prototype for a mobile game, and was carried out in collaboration with two local cultural / tourist organizations. Because both sites required different technical facilities in order to support (indoor / outdoor) localization for the game, we decided to concentrate on the client UI in the game. This allowed us to build the UI on a general framework, independent of the technical localization aspects. Furthermore, we decided to develop the application logic in general, while providing different UI representations for the two organizations. In the remainder of this section, we will refer to *the game* as the general application logic and *the game concepts* as the two individual games which are represented in the separate UIs.

The goal of both game concepts was to make educational excursions more interesting and informative for children. Since a new system had to be developed in this case study, it was impossible to examine manuals and existing functionality. Mainly the results from a user and task analysis could contribute to the structured interaction models. During the user and task analysis, school groups were observed and interviewed while they were visiting museums and zoos. These observations showed us that the addressed target users prefer being guided throughout the visit in a narrative style, based on a story they can identify themselves with. After several brainstorming sessions,

the multi-disciplinary team including a graphic designer and representatives of the cultural and tourist organizations, came up with two game concepts for a PDA application (Figure 3.1(b)-A). The goal of one game concept was to save the trees in a nature resort, while the other game concept challenged children visiting a mining museum to help a miner to have a safe working day. Scenarios ensured that all team members had the same understanding of the game concepts to be designed. In this stage, the *requirements and user needs* were specified using a word processor and were consulted by the team using a PDF viewer.

The scenarios of the game concepts proved to be very useful to structure the user tasks and to create a task model created in CTTE using the CTT notation (Figure 3.1(b)-B). Even though both game concepts were substantially different, the same user interface components should be necessary. This matched the decision to create a general framework containing the application logic for both game concepts.

Besides the task model, other HCI engineering models were created to present the relationship between the user interface and the application logic (Figure 3.1(b)-B). The application model ensured the application logic would be suitable for both game concepts. The system interaction model, based on the user task and application model gave an overview of the flow of actions carried out by the system and the user. The abstract presentation model, was based on the preceding models and represented the user interface components. These UI components could be used in a Canonical Abstract Prototype (CAP) [L. 03]. This CAP (Figure 3.3, left) was a first graphical representation of the functional parts of the user interface, independent of the game concept. The CAP was created using the CanonSketch tool, while a drawing tool was used for the creation of the other models. During the evaluation of the models, the scenarios were used to ensure the models did meet the requirements of each game concept. After the creation of these *structured interaction models*, the task was handed over to the graphic designer. He translated the CAPs into *low-fidelity prototypes* (Figure 3.1(b)-C), which evolved into a design of the prototypes for both game concepts (Figure 3.3, center and left) created in a paint program after adding layout and style information.

In order to get some early feedback of the end users, the prototypes for both game concepts were tested in a lab environment with materials similar to what is being used in participatory approaches such as PICTIVE [Mul91] (Figure 3.1(b)-D). The tests showed children were amused by the game, but revealed problems concerning the size and behavior of buttons and the content. Locations of the sites were simulated in the lab by means of photo projec-

tions [MHC08]. In a second iteration of this project (after our field tests), we used a large screen projection of a virtual environment that represented the site of the mining museum [HDBCR09]. Virtually walking around in the mining museum, increased the subjects' awareness of playing a location-based mobile game.



Figure 3.3: Three levels of prototypes for one specific screen. From left to right: a Canonical Abstract Prototype (CAP), a low-fidelity prototype and a high-fidelity prototype.

Based on the test results, the designer adjusted the designs of the user interfaces and complemented them with animations using an animation tool, while the models of the structured interaction analysis were used by computer scientists for the development of the application logic of the game in an advance programming tool. Following, the prototypes of the game concepts and the application logic were coupled in order to obtain an interactive high-fidelity prototype. The resulting *high-fidelity prototypes* were evaluated by children in a nature resort and a mining museum. Localization was realized using a Wizard of Oz [DJA93] application. During these field tests few user interface problems were detected. However, playing the game concept for the mining museum, revealed that extra content to cover the walk between two locations was missing. We believe that the virtual environment, which was used in the second iteration of lab testing, would have revealed this problem before the field tests took place [HDBCR09]. Besides this UI problem, we may conclude that the model-based approach, and the evaluation in early stages influenced the high-fidelity prototype in a beneficial way.

3.2.3 Lessons Learned

The case studies presented above, were carried out using MuiCSer processes. In the NewsWizard case study, a MuiCSer process was used for the redesign of an existing system, while the second case study concerned the design of a new system. In both case studies we experienced that it was hard to structure the information to get a complete overview of the requirements and user needs, which confirms our findings regarding tool support for UCSE (Chapter 2). Since the usage of personas and scenarios implies narrative information that is only partially structured, it was helpful to transform the information into structured interaction models. Based on former experiences in applied research projects, task models seemed a suitable notation to structure the narrative scenarios. These task models made it possible to abstract the most important goals of the future prototype. However, by doing so, some information contained by the personas and scenarios could be overlooked. Therefore, the task models were evaluated while keeping different viewpoints in mind during meetings which involved the computer scientists as well as team members with other roles.

By carrying out different case studies we had the opportunity to fine-tune the approach in our multi-disciplinary team. In the NewsWizard case study it became clear that task models were understandable for all team members, including the customer and thus could be evaluated during meetings. Nevertheless, computer scientists experienced that the information of task models was insufficient for the development of the high-fidelity prototypes. During the structured interaction analysis and prototyping of the mobile game, models presenting the links between the user interface and the application logic were helpful to get more insights into the functional requirements. Furthermore, these models evolved gradually into a first graphical representation, the CAP, which was also presented to and used by the graphic designer.

The low-fidelity prototypes of both case studies were created by putting together the artifacts of earlier MuiCSer stages. The design of the first prototypes was discussed and evaluated during meetings attended by the multi-disciplinary team.

Journalists and children were asked to participate in the evaluation of the high-fidelity prototypes of both case studies (the NewsWizard and the mobile game for respectively). Our experience from other case studies taught us that field tests give more information on the entire user experience compared to lab testing. By evaluating a prototype in the natural environment of the end user, a broader user experience is taken into account and context dependent

actions can be observed.

When comparing the processes shown in Figure 3.1 we discover that both are in line with the MuiCSer framework from the start where the user studies take place, until the high-fidelity prototyping phase. Several artifacts were created as a result of the process stages. This illustrates the fact that the MuiCSer framework suggests some models and artifacts, but that the design team decides about the particular results for the process at hand. All artifacts proved useful as input to artifacts created in the next phase. The transition of these artifacts required some human intervention that is difficult or impossible to automate.

The creation of the artifacts, was carried out using several tools. Evaluation, verification and validation of artifacts can be facilitated by tools, but often the transformations supported by the tools are incomplete and the involvement of the team is required. In the case studies, the computer scientists and designers used CTTE, CanonSketch, drawing tools, animation tools and advance programming tools for the development of HCI models and coded prototypes. Widespread tools such as pencil and paper, a word processor and a PDF viewer were useful for the other artifacts as the entire project team, including representatives of the participating companies, was familiar with these common tools. This is in line with the conclusions of our literature study in Chapter 2 and emphasizes the difficulties that arise when the transition between informal and formal artifacts and vice versa has to be realized.

3.3 UCSE in Industry

The MuiCSer process framework allowed us to investigate UCSE and its artifacts and tools. This led to some findings regarding current challenges in UCSE. However, these findings are based on literature (Chapter 2) and our own experiences as part of a multi-disciplinary team (Section 3.2). In order to check whether our findings correspond with common practice, we interviewed companies that are active in UCD / UCSE.

Three leading Belgian companies that are specialized in user-centered projects, were involved in the interviews. In order to guarantee anonymity, which was requested by one of these companies, we call them C1, C2 and C3 in the remainder of this section. C1 is a company that designs and develops websites in a user-centered way, while C2 and C3 are companies that conduct usability and user interface consultancy in larger projects.

At the time of our semi-structured interviews (2008), these companies had between five and sixteen years of experience, employed ten to forty profession-

als with various backgrounds, and had completed projects in a wide range of application domains. In total, we interviewed eight employees of the companies: four project managers, three interaction designers and one web designer. Some of the questions that were used for these semi-structured interviews are listed in Appendix A, Section A.1.

3.3.1 Roles

Table 3.1 shows an association of the roles of ISO 13407 and the roles that are typically involved in projects conducted by the three companies. Each different pawn in the table represents a different team member (including purchaser and/or end user). The table shows that some team members have two or more roles. Usually, one team member combines the roles of a UI designer, HCI specialist and technical author.





















Roles	Companies		
	C1	C2	C3
End user	()		
Purchaser, manager of user			
Application domain specialist	()		
Systems analyst / engineer, programmer			
Marketer, salesperson	()		
UI designer, visual designer	 		
HCI / Human factors & ergonomics specialist			
Technical author, trainer & support personnel			

Table 3.1: The roles that are involved in the teams of the companies we interviewed. The different fills refer to different team members in the teams. The brackets depict team members that are only part of the team if the customer has one of these roles.

During the interview, C1 explained that the purchaser is always part of the team. The roles of end user, marketer or application domain specialist are only involved if the purchaser has one of these roles, which is indicated by the pawns in parentheses in Table 3.1. C2 and C3 always make the distinction between the purchaser and the end user. C2 explained that an end user is considered as an application domain specialist if she is part of the team, while C3 considers the purchaser as an application domain specialist.

When comparing C1, C2 and C3 regarding the involvement of all roles, we see that C1 may involve all roles in its team, while C2 and C3 leave out

a systems analyst and a marketer. These roles are usually not involved in the services provided by C2 and C3, which does not mean that these roles are not involved in the projects as a whole. Usually, at the moment that the development starts, C2 and C3, deliver artifacts containing the specifications, requirements, user needs and UI designs of the system to the purchaser, who assigns (external) programmers to develop the system.

Since the interviewed companies often collaborate with external partners for the actual development of an application, they indicated there is no sufficient support to translate their artifacts into a notation appropriate for systems analysts or programmers. Shortcomings of existing notations require close collaboration between all team members to avoid misconception.

3.3.2 Models and Artifacts

All interviewees explained that the use of particular models and artifacts has grown from their own experiences in UCD projects. Table 3.2 shows the models and artifacts that are created and used by the companies we interviewed. The table shows that all three companies consider non-functional requirements. Since C1 is taking care of both the design and the development, and little team members are involved, they pragmatically implement the non-functional requirements by discussing the first mockups and wireframes in meetings that involve all team members as well as the purchaser. In contrast to the approach of C1, the other two companies do not develop the system themselves and thus have to pass all results of their analysis and design activities to the purchaser who delegates this to systems analysts and programmers. This implies that C2 and C3 usually perform an extensive analysis in order to capture the user needs and to obtain a suitable UI design. They create artifacts such as scenarios, personas, storyboards, user stories and dialog models. Consequently, all their conclusions are documented in many artifacts which are needed to communicate their findings with the purchaser who on her turn passes the artifacts to the systems analysts / engineers and programmers.

As shown by Table 3.2, C1 tends to concentrate most on the models and artifacts created later in UCSE processes, while C2 and C3 focus more on the stages of UCSE that exclude the development of the system. Although C2 and C3 create many artifacts, the interviewees of C2 and C3 emphasized during the interviews that many meetings and a close collaboration with the purchaser's company are indispensable in order to make sure that their UI designs and other artifacts meet the user needs and requirements. Both C2 and C3 select notations for their projects that fit the notations the purchaser and

Models and Artifacts	Companies		
	C1	C2	C3
Scenarios [Car00] / Personas [PA06]		✓	✓
Storyboards [PRS02]		✓	✓
Mental model [PRS02]		✓	✓
Non-functional requirements [PRS02]	✓	✓	✓
Use cases [PRS02] / user stories [Coh04]		✓	✓
HTA [PRS02]		✓	✓
Information architecture [AAB06]	✓	✓	✓
Process flow [PRS02]	✓	✓	✓
Navigation / dialog model [AAB06]		✓	✓
Mockups [PRS02]	✓	✓	✓
Wireframes [AAB06]	✓	✓	✓
UI designs [PRS02]	✓	✓	✓
Animated UI designs	✓	✓	
Final interactive system	✓		
Manuals	✓		✓

Table 3.2: The models and artifacts that are created and used by the companies. The models/artifacts are sorted, based on the order in which they may be created in a MuiCSer process.

programmers are accustomed to. However, these notations do not always allow the interviewees to include all information that is resulting from their usability and user interface consultancy. This explains why face to face communication with the purchaser, and if possible the programmers, is necessary.

3.3.3 Tools

Most of the tools used by the three companies can be considered as widespread tools. One type of tool used by the interviewees that is particularly developed for UCD purposes is a prototyping tool. However, none of the interviewees uses artifact transformation tools because they want to have full control over the artifacts and in particular the designs they create.

Similar to the selection of models and artifacts, the tools used by the companies are chosen in a pragmatical way. The three companies explain that the tool has to fit the practices of the purchaser and the development team. Widespread tools and tools that allow artifacts to be saved in an accessible format are favored by the interviewees.

Tools	Companies		
	C1	C2	C3
Word processor [AAB06]	✓	✓	✓
Presentation [AAB06]	✓	✓	✓
Spreadsheet [AAB06]		✓	✓
Drawing [AAB06]	✓	✓	✓
Paper [AAB06]	✓	✓	✓
PDF viewer [AAB06]	✓	✓	✓
Paint program [AAB06]	✓	✓	✓
Prototyping tool (e.g. Axure [Axu])	✓	✓	✓
Simple programming [AAB06]	✓	✓	✓
HTML (site) editor [AAB06]	✓		
Animation tool [AAB06]	✓	✓	
Advance programming [AAB06]	✓		

Table 3.3: The tools that are used by the companies.

3.3.4 Fit with the MuiCSer Process Framework

Based on the description of roles involved, and models, artifacts and tools used by the three companies, we can conclude that their approaches fit the MuiCSer process framework. The user-centered processes followed by the three companies involved in our semi-structured interview include all stages of the MuiCSer process framework, at least until the high-fidelity prototyping stage. The three companies carefully evaluate or verify each artifact at several moments during the process. If possible, the companies conduct usability tests, but additionally frequent communication within the multi-disciplinary team is considered as very important by all interviewees.

The interviews confirmed the need for flexibility when selecting models, artifacts and tools for user-centered projects, as we proposed in the MuiCSer process framework. According to the companies, the characteristics of the project and the composition of the multi-disciplinary team influence the choice of models, artifacts and tools in their user-centered approaches.

3.3.5 Discussion

The interviews described above, confirm our findings regarding UCSE described in Chapter 2 and Section 3.2. The companies expressed the difficulties they experience when they have to collaborate with technical people such as

programmers. They confirm that this is partly caused by the lack of suitable models and notations. Two of the three companies have to pass the results of their analysis to the purchaser, who on their turn passes these results to external systems analysts and software developers. Consequently, the companies try to use notations that systems analysts are familiar with. However, they are facing the limitations of these notations that cannot include particular design decisions. Similar difficulties with respect to the involvement of domain experts and developers in a team are also addressed by Evans [Eva03].

Although the use of tools is closely related to the type of artifacts that is created, the companies prefer the use of widespread and accessible tools. The artifacts team members are accustomed to, and the know-how of the team influence the use of models, artifacts and tools. This learns us that the availability of a particular tool alone does not suffice to overcome the difficulties in UCSE practices. Tools that support UCSE and in particular MuiCSer, should not only support the notations used by UCSE practitioners but also need to consider the involvement of the entire team.

3.4 Conclusion

Two case studies as instances of the MuiCSer process framework were described in this chapter. These case studies exemplified the use of MuiCSer. Furthermore, we conducted interviews with practitioners in UCD. Both the case studies and the interviews confirmed that the stages proposed in the MuiCSer process framework, as well as its flexibility concerning the use of artifacts, models and tools are relevant. Additionally, difficulties were revealed that arise in UCSE processes: the lack of notations that support the transition from informal design knowledge to formal artifacts and the need to verify the consistency of artifacts together with the multi-disciplinary team in order to avoid a loss of information.

Our MuiCSer process framework in combination with the results of our studies regarding UCSE in literature and practice provide us with some challenges for UCSE. Before we propose a solution in Part II, we recapitulate these challenges in Chapter 4.

Chapter 4

Weaknesses and Difficulties in UCSE

The MuiCSer process framework introduced in Chapter 2 was in the first place developed to derive processes that are suitable for applied user-centered software engineering (UCSE) projects. In Section 3.2 we described two case studies that were conducted by applying UCSE processes as instances of MuiCSer. Furthermore, the process framework was used in different ways to investigate UCSE processes, artifacts, tools (Section 2.3) and practices (Section 3.3) and to identify weaknesses and difficulties in UCSE. In this chapter we will list these limitations of current UCSE practices. Following we will explain why we consider the work described in the following part of this dissertation as a possible approach that can be used to overcome the difficulties in UCSE.

4.1 Weaknesses and Difficulties

Notations and tools that support the collaboration in multi-disciplinary UCSE teams are lacking. Our findings of the comparison of UCSE tools regarding the inaccessibility of artifact transformation tools, was confirmed by the user-centered design (UCD) practitioners that participated in our interviews. Most tools used in a UCSE project should be accessible to all members of a multi-disciplinary team. Furthermore, the lack of notations and tools that support multi-disciplinary teams requires a close collaboration within the team for the creation of artifacts. By involving all team members in the creation of artifacts and prototypes, misconceptions and a loss of information can be avoided. Although the involvement of multi-disciplinary

teams is inevitable in UCSE, a notation and tool that can be used by all team members may enhance and increase efficiency of the communication within the multi-disciplinary team.

Notations and tools that support the transition from informal artifacts into formal models are lacking. The comparison of existing UCSE processes, models and tools, revealed that a tool is missing to translate informal design knowledge into formal artifacts. The interviewees confirmed this and emphasized the lack of notations and tools that translate artifacts of the first stages in MuiCSer into notations that are suitable for team members with a technical background. Our case studies showed that the informal design knowledge, which is often expressed in a natural language, is hard to structure without any loss of information.

Notations that incorporate all user needs and requirements are lacking. Our case studies and the interviews with practitioners uncovered the lack of a notation that includes all information from the first analysis stages. This type of notation would benefit the creation of artifacts while diminishing the amount of information that might get lost or overlooked. The types of information that should be considered for this notation include non-functional requirements, functional requirements and contextual information regarding a system's use.

4.2 Can Storyboards be Considered as a Possible Answer?

A possible technique to overcome the weaknesses and difficulties in UCSE is introducing a common language for multi-disciplinary teams, which can be connected to languages that are specific for each discipline, which was also suggested by Evans [Eva03]. For this language, we suggest storyboards. Storyboarding is an existing UCD technique [LM96, PRS02], which is also used in practice (Section 3.3). Although it is not a new technique, little is known about their ideal form, creation and usage in user-centered approaches.

The language used in storyboarding is highly understandable for all team members of a multi-disciplinary team, including end users. Earlier work reported that storyboarding is even applicable as a technique for participatory design with children [MLD⁺07]. Thanks to their natural language, storyboards can express several aspects of requirements and user needs. Furthermore, the

visual representation and the set of scenes may be the first step in a UCSE process towards a formal artifact that carefully considers several aspects of requirements and user needs.

If storyboards can be considered as a notation that is accessible for a wide range of disciplines, they can be introduced in UCSE teams as a communication tool. We expect that the use of storyboards will not eliminate the close collaboration in a multi-disciplinary team, but may increase the communication within the team.

Our assumptions regarding storyboarding were further investigated in this PhD. We do not claim that storyboards are the exclusive notation that answers all weaknesses and difficulties in UCSE, but the following part of this dissertation will show the opportunities provided by storyboarding to overcome these problems. In Chapter 5 we introduce our definition of storyboarding, followed by a formulation of research challenges and accompanying research questions that are relevant for storyboarding in UCSE. Our investigations regarding these different research challenges of storyboarding are described in Chapters 6 to 9.

Part II

Storyboards as a Common Language for User-Centered Software Engineering

Chapter 5

COMuICSer Storyboarding

5.1 Introduction

When combining user-centered design (UCD) techniques and software engineering (SE) principles in user-centered software engineering (UCSE), the biggest challenge is the communication within a multi-disciplinary team including the end users. MuiCSer, a framework for Multidisciplinary user-centered Software Engineering processes, which was introduced in Part I, focuses on the benefits of both disciplines, and was presented to investigate the features and shortcomings of current UCSE models and tools. One missing link in most user-centered processes is a tool to progress from informal design artifacts (e.g. scenario) toward more structured and formal design artifacts (e.g. task model), without any loss of information. Most tools and techniques require specific knowledge about specialized notations or models, thus exclude most team members not familiar with these notations or models. Furthermore, functional information may be missing in informal design artifacts while structured design artifacts may not always contain all non-functional information. We propose storyboards as a comprehensible notation to overcome these shortcomings.

In this chapter, we introduce storyboarding as a solution for the aforementioned problems in UCSE. We first provide a definition of storyboards. Next, we present accompanying tool support for these storyboards. Finally, we research challenges and research questions regarding storyboarding and UCSE which will be investigated in this dissertation.

5.2 Visualizing Requirements and User Needs

The early design stages of user-centered design (UCD) include a user needs analysis and generally result in several artifacts such as usability requirements [RPM95], scenarios [Car00] and personas [PA06] containing the user needs. These artifacts are written in natural language, have a narrative style and are usually created by interaction designers. Similar artifacts are used in software engineering and agile development [HWW04] (e.g. essential use cases, scenarios and user stories). Although several disciplines provide notations to describe user needs, the notations are not always suitable to pass information of the user needs to other members of the multi-disciplinary team without any misconceptions.

Lindgaard et al. [LDT⁺06] address the difficulties in presenting user needs for requirements engineering based on a case study. The wide interpretation of tasks and user needs analysis confuses multi-disciplinary team members, which impedes communication within the teams. Brown et al. [BLB08] conducted an ethnographic study to investigate the collaboration between interaction designers and developers. The study describes the benefits of using stories and sketches in the early stages of user-centered approaches and emphasizes the power of combining both. Assembling stories and sketches is a powerful technique to reveal errors, and to consider temporal and contextual information. A study of Myers et al. [MPN⁺08] reports that designers are experiencing difficulties when designing the behavior of user interfaces. While prototyping the appearance of user interfaces is straightforward, designing and communicating the interactive behavior (e.g. the navigation within the UI) is an ongoing process. Furthermore, the survey revealed designers frequently use sketches and storyboards.

The professional use of storyboards originates from the film-industry and is getting introduced in several disciplines such as advertisement and product design [vdL06]. For similar visualization purposes, storyboards are used in UCD, where they can have different forms. On the one hand, storyboards visually express scenarios of use. On the other hand, storyboards represent the flow for UI designs to clarify interactivity in the early stages of UCD [LM96]. We will concentrate on the former type of storyboards, which considers storyboards as a technique to complement scenarios that results in a visual depiction of how a user carries out a task using a system that is to be developed [PRS02, KJ07].

Storyboards can be used in UCSE to clarify user needs. In particular, storyboards can be useful to depict systems that are used in several contexts of use or on multiple devices and systems that need to support business processes.

In earlier work, storyboards were used for the design of mobile systems [PS05], to provoke empathy in a design team [MGM03] and to validate conceptual ideas of new interactive systems [DLDZ07]. The visual representation used for storyboards allows members of a multi-disciplinary team to be creative, but in the meanwhile it provides a concrete specification of a scenario of use. In the next section we will describe how this type of storyboards can be specified and used in order to support multi-disciplinary teams in UCSE.

5.3 COMuICSer

We propose storyboarding notation that is named COLlaborative MultI-disciplinary user-Centered Software engineering, COMuICSer. COMuICSer is pronounced as “co-mixer” and refers to comics, which have a similar notation as storyboards, as will be explicitly described in Chapter 7.

5.3.1 Definition

By specifying the components of COMuICSer storyboards, we aim to obtain a clear definition of storyboards, which allows us to investigate how storyboards can contribute to UCSE from several viewpoints. We define a COMuICSer storyboard as *sketches of real life situations, depicting users carrying out several activities by using devices in a certain context*.

Real life situations that depict in which circumstances the future system will be used, are the main component of a storyboard, because these situations explain to the reader by means of a scenario in what realistic circumstances a certain system is or will be used. Storyboards representing a future scenario of use are the most interesting type of storyboards in UCSE. However a storyboard representing a current scenario of use can be helpful as well to identify or clarify challenges and problems in an existing system. The visualizations of real life situations will show the end user needs to a multi-disciplinary team and may provoke empathy with the users in the team.

In user-centered approaches, the focus is on the users from the beginning of a project, so the *users* have a prominent role in a COMuICSer storyboard. If available, personas can be linked to the storyboard. However, as stated in ISO 13407 which specifies human-centered design, not only the user, but also the *activities* carried out by the user, the technology or *devices* that are provided, and the *context* in which a system is used, should be considered.

An example of a simple storyboard is presented in the center of Figure 5.1. This storyboard depicts a few hours of a journalist’s working day. In the first

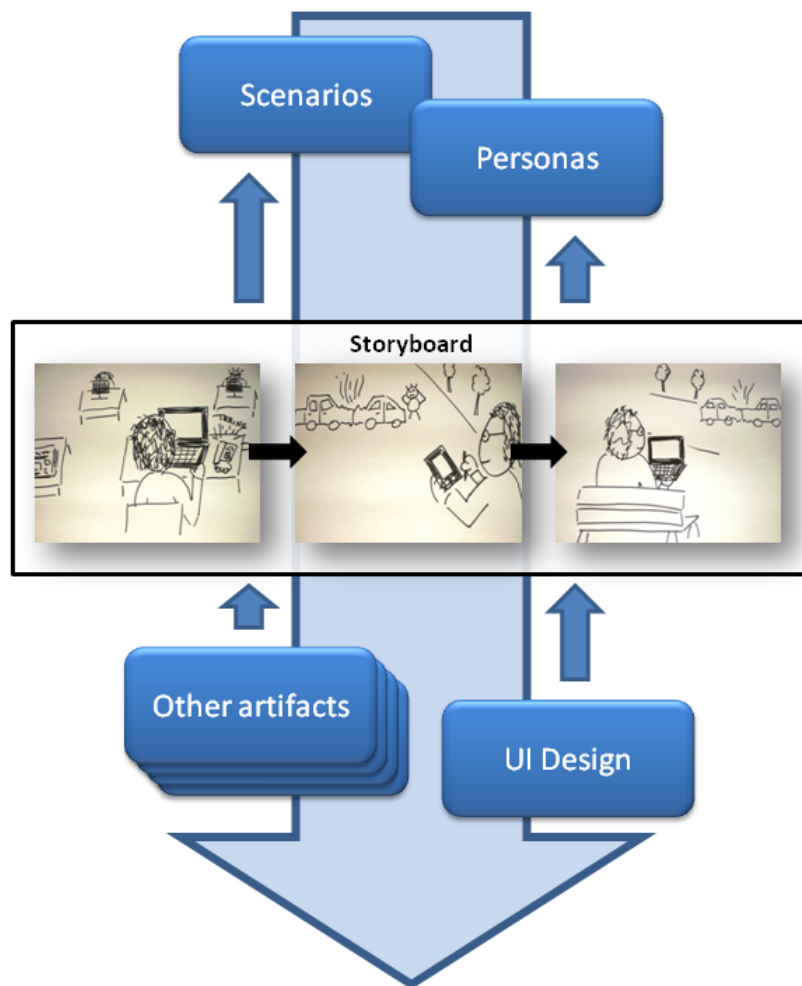


Figure 5.1: A storyboard and its interrelationship with other artifacts in the UCSE process.

scene, the journalist is working behind his desk. This is how he usually starts his day at work. However, very often he receives a phone call that notifies him about a certain incident in the neighborhood (e.g. a car accident). Following, the journalist hurries to the place of the incident, where he takes notes on his personal device. The second scene shows the journalist taking notes about a car accident. Because the police requests the journalist to go further away from the cars that collided, he searches a park bench and finalizes his article for the newspaper using his laptop, as shown in the third scene. Note that the third scene exemplifies the context in which the NewsWizard prototype, presented in Chapter 3 could be used.

5.3.2 Bridging the Early Stages of UCSE Processes

The creation of storyboards happens at the early stages of a UCSE process, after the user needs observation or analysis and the creation of scenarios and personas. The available information for the creation of the storyboard consists of requirements and user needs. This information does not necessarily have to be concluded, but can be discussed and adjusted in several iterations. An example storyboard and the interrelationship between a storyboard and other artifacts are presented in Figure 5.1. A storyboard can be built by splitting up the *scenario* into scenes and presenting the scenes as sketches depicting users interacting with the future system. Connecting scenes of a storyboard, structures the narrative information of the scenario, while the accessible notation allows all team members, including end users, to be involved in a UCSE process.

In all scenes, added to the storyboard, *personas* and *devices* can be highlighted. The storyboard in the center of Figure 5.1 shows a persona (e.g. Bart, journalist, 43 years old) in the three scenes. Furthermore, the device used in the first and the third scene is a laptop, while a personal device is used in the second scene. Highlighting personas and devices enriches the information contained by the storyboard and is useful to make the transition to other artifacts. At a later stage of the development process, the storyboard can guide the UI design and development. By carefully considering the situation of each scene, designers and developers can build an application corresponding to contextual information, requirements and constraints contained by the storyboard. Interaction designers can use a storyboard to verify that the UI designs take into account all requirements. A storyboard also contributes to the preparation of the usability tests.

Using storyboards in UCSE processes increases the visibility of the project.

New team members for instance, can explore the requirements of the project at a glance by looking at the storyboard. The aforementioned aspects of storyboarding will be described in chapter 6.

5.4 Tool Support for COMuICSer Storyboards

In Part I we addressed the need for tools that support UCSE processes in the early stages of design. Since storyboards are created during the requirements elicitation, storyboarding tools should at least partly be able to cover transitions between the early stages of UCSE. When suitable tool support is available for all team members, storyboards may become more powerful and the visibility and traceability of a project can increase.

Currently little tool support is available for storyboarding in multi-disciplinary teams. The ActivityDesigner [LL08a] tool allows storyboarding at the early stages of design. In this tool, designers can extract activities from concrete scenarios making it possible to include rich contextual information about everyday life as scenes. Based on the scenes, higher level structures and prototypes can be created. However, in the ActivityDesigner not all information is visually represented by the scenes and which components need to be available in a scene, is not specified. We are interested in tool support for the creation of storyboards that visually represent the context of use. Digitizing storyboards has the advantage that they can be used as input for later stages of UCSE.

More generally, several tools such as Comic Life [Pla], Celtx [Cel] and Kar2ouche [Imm] support storyboarding and include interesting features to create storyboard scenes. These tools mainly focus on the storyboard and do not provide any features that consider the use of storyboards in UCSE processes. Furthermore, Ozenc et al. address the need for tools that support refining rough designs and a scenario-driven process [OKZ⁺10].

We propose the COMuICSer tool to support the creation and use of COMuICSer storyboards and passing contextual information to other artifacts in UCSE processes (e.g. UI designs and their interaction sequences). A proof of concept of the COMuICSer tool has been implemented as part of the activities in the context of this PhD research.

In our COMuICSer tool, a team member, e.g. an interaction designer starts the creation of the storyboard by writing or loading a narrative scenario in the tool using the scenario panel (Figure 5.2-1). Following, a sequence in the textbox of the scenario can be selected and consequently, a new scene can be created. The new scene appears in the storyboard panel (Figure 5.2-2) and the

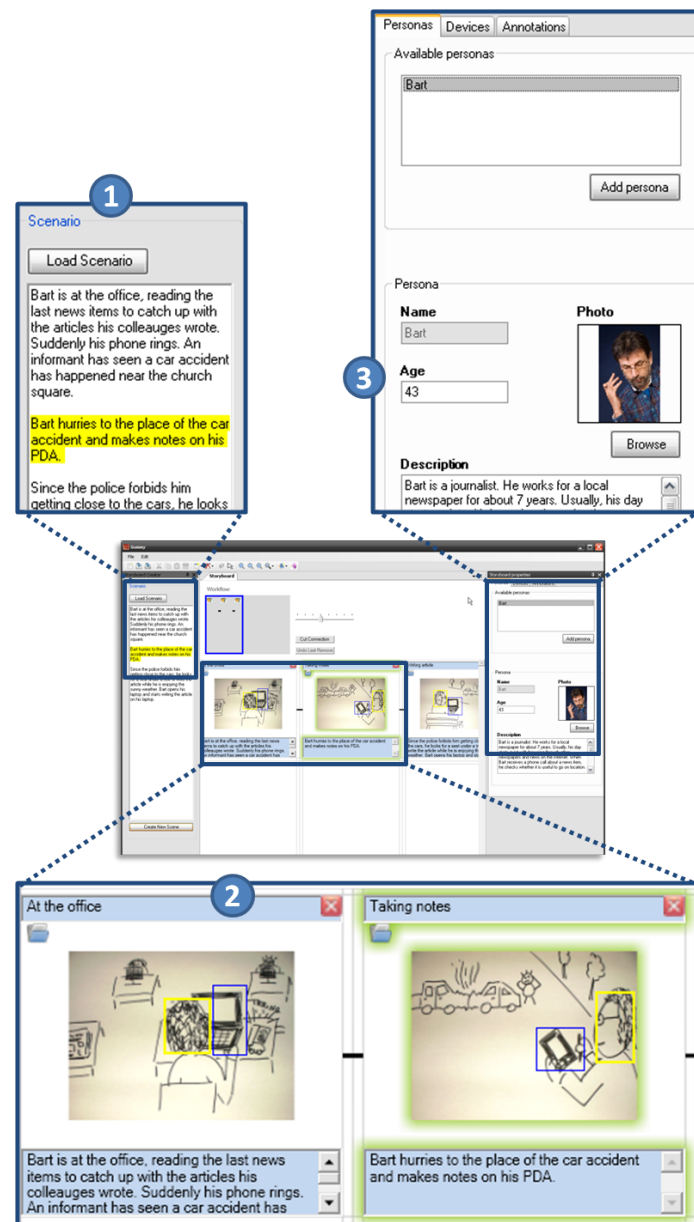


Figure 5.2: A screenshot of the COMuICSer storyboarding tool. This tool supports storyboarding by connecting a storyboard with a scenario, personas and other annotations.

sequence of the scenario is automatically added to the scene as a description. Now, the interaction designer can load an image and add a title. The image of the scene can be a photo of the user observations or a scanned sketch, which encourages designers to sketch in a creative and informal way [NL00].

Scenes that are loaded in the COMuICSer tool can be annotated by all team members with device information, personas and free annotations. These annotations are made in a similar way as the *photo tagging* feature on Facebook or Flickr. In each scene, personas or devices are annotated by drawing a rectangle around it. Next, each rectangle can be linked to a persona, device specifications or some free annotation text, which can be specified in the properties panel (Figure 5.2-3).

Since our COMuICSer storyboards depict users as well as devices, one scene clearly shows what activity users need to be able to perform on a particular device in a certain context. This way, the storyboard scenes provide interesting information for the UI design. We decided to connect the COMuICSer tool to the Gummy GUI builder tool [MVLC08]. Gummy is a tool, developed in our research lab, which supports the graphical UI design of multi-device and context-aware user interfaces. By connecting our COMuICSer tool to Gummy, a device annotation in the storyboard (e.g. a target platform and a screen resolution), can be considered when the design workspace of Gummy is loaded and thus allows designers to create user interfaces for a wide range of devices without having to change their work practices. Furthermore, the contextual information of the scenes (e.g. sketches presenting the environment or courses of communication) can be used as guidelines for the UI design without obstructing the creativity of UI designers.

The COMuICSer tool makes it possible to include the results of the first UCSE stages (e.g. user needs analysis) and helps to process and structure these narrative artifacts for later stages in the design and development process. Our introduction of COMuICSer storyboarding and the accompanying tool, provide the basics, which can be used for further research regarding storyboarding in UCSE. The next section lists and explains research challenges regarding storyboarding and UCSE.

5.5 Research Challenges

We introduced COMuICSer storyboarding in order to support UCSE and to overcome the problems when combining UCD and SE, which were considered in Chapter 4. Our first assumptions are that storyboards may enhance communication in multi-disciplinary teams and provide connections between several

artifacts created and used in UCSE. However, further research is needed to investigate how storyboards can benefit UCSE. Before we present our work regarding storyboarding, we list important research questions related to storyboarding and UCSE in this section. Each research challenge considers one or more research questions.

5.5.1 RC A: Storyboarding for Multi-disciplinary Teams

We assume that storyboards may be helpful for multi-disciplinary teams. However, only little information is available about the use of storyboards in teams that involve people having different backgrounds. Earlier studies of Johansson et al. [JA07] and Brown et al. [BLB08] showed the benefits of combining stories and sketches when a team involves people having technical and people having non-technical backgrounds. However, no studies investigate whether the notation used for storyboards is considered useful for all disciplines involved in UCSE. Furthermore, there are no studies that report the way a storyboard is created within a multi-disciplinary team. In order to extend our COMuICSer tool for multi-disciplinary teams, we are interested in the needs of a multi-disciplinary team that creates storyboards and want to know whether particular team members have more influence on the resulting storyboard than other team members. For this research challenge, we consider the following questions:

- *RQ 1a: Are storyboards and accompanying tool support useful for UCSE practitioners?*
- *RQ 1b: How are storyboards created in a multi-disciplinary team?*

We conducted two user studies in order to investigate the questions mentioned above. One study was conducted in order to observe how UCSE practitioners judge the used of storyboards and our COMuICSer tool in UCSE. Another study was an observational study that aimed to investigate how multi-disciplinary teams collaboratively create storyboards. Chapter 6 describes these user studies and their results, which inspired us to investigate other research challenges.

5.5.2 RC B: A Visual Storyboarding Language

One of the advantages of storyboards is its visual language which is assumed to be understandable to all stakeholders involved in a UCSE project. The use of storyboards in other domains shows that storyboarding does not require

much knowledge regarding its notation. However, since this visual notation allows much freedom and creativity, it is difficult to specify what components of the notation are interesting for storyboarding in UCSE. Consequently, the following research question can be asked:

- *RQ 2: What aspects of a visual language can contribute to storyboards in UCSE?*

Since storyboards resemble comics [Bux07, DSLL06], we investigated which techniques used in comics can be incorporated in COMuICSer storyboards in order to reinforce storyboarding in UCSE. We describe in Chapter 7 what principles of comics, presented by Scott McCloud [McC93, Mcc06], are applicable to COMuICSer storyboarding and how these principles could be supported by our COMuICSer tool.

5.5.3 RC C: Storyboarding to Support Artifact Transformations

The combination of user-centered design and model-based approaches, involves the creation and use of informal as well as formal artifacts. Although these informal artifacts and formal HCI models are complementary and their combination is beneficial for the user experience of the resulting software product, transforming one artifact into another is not straightforward. The gap between UCD and model-based approaches is mainly caused by differences in “design” language of practitioners. We assume that storyboards can bridge part of the gap between informal and formal artifacts. Consequently, the following research question is considered:

- *RQ 3: How can storyboards be used for the transformation from informal to formal artifacts?*

The visual notation used in storyboards can detail a narrative scenario by extensively removing ambiguities. Furthermore, by splitting up a narrative scenarios in scenes, some preliminary structure is available, which can be used for the transformation to formal models. Nevertheless, more formal approaches are needed to support the transition from informal to formal artifacts. In Chapter 8 we present some techniques that allow the transformation from storyboard to formal artifacts.

5.5.4 RC D: Storyboarding to Connect UCD and SE

Not only formal models, which are used in HCI, need to be considered in UCSE. The main goal of UCSE is to support techniques from both UCD and SE. Since UCD focuses on quality attributes that can be considered as non-functional requirements and SE typically concentrates on functional requirements, techniques are needed to connect functional and non-functional requirements. Furthermore, the context in which a system is used, should also be considered in UCD as well as in SE. These ideas led to the following research question:

- *RQ 4: How can storyboards be connected with software engineering artifacts?*

We investigated how storyboards can be connected to requirements specified in agile software engineering in order to combine functional as well as non-functional requirements. In Chapter 9 we describe an approach to connect storyboards and user stories, which decreases the loss of information when requirements are passed within a team.

5.6 Conclusion

Storyboards are implicitly used in different ways by multi-disciplinary teams that take part in UCD processes. This partly explains the many interpretations of storyboarding and reveals the challenges in developing a storyboarding tool for multi-disciplinary teams. In order to provide a clear starting point in the use of storyboards in UCSE, we introduced our definition for COMuICSer storyboards. COMuICSer storyboards preserve creativity in the way a visualized scenario of use is depicted. By highlighting parts of freely created storyboards, the communication in a multi-disciplinary team and the transition to formal artifacts can be facilitated. We proposed tool support for digitizing and annotating COMuICSer storyboards.

Our definition and accompanying tool support for COMuICSer storyboards will be used for further research corresponding to the four research challenges described in this chapter. By investigating these research challenges and accompanying questions, we will describe the creation and the use of storyboards in multi-disciplinary teams (Chapter 6). Following, we will describe how the visual language of comics can be incorporated in storyboarding techniques and tool support for UCSE in Chapter 7. One of the major problems in UCSE is the transition from informal artifacts to formal artifacts, which is discussed in

Chapter 8. Finally, Chapter 9 describes how storyboards can be used as a connection between non-functional requirements as used in UCD and functional requirements as used in agile SE.

Chapter 6

Storyboarding in Multi-disciplinary Teams

6.1 Introduction

The visual language and the rich content that are supported by storyboards let us assume that storyboarding is a suitable language for communication in multi-disciplinary teams. The visual representation allows team members to understand a storyboard at a glance and the notation is accessible for team members having different backgrounds. Although the interviews described in Chapter 3, revealed that practitioners use storyboards, they do not use any specific tools for storyboarding. Since a storyboard usually grows from creative ideas of one or more team members, there is no general approach for storyboarding.

Earlier studies that were conducted, investigated the creation of storyboards. However, these studies were limited with respect to the aspects of storyboarding we want to investigate. In a field study, conducted by Dow et al., participants with different backgrounds were interviewed about suitable representations for design in ubiquitous computing [DSLL06]. However, storyboarding is only one of the notations that are discussed in their work. Moraveji et al. conducted participatory design sessions with children which included storyboarding to consider design problems [MLD⁺07]. This study resulted in particular insights regarding storyboarding for children in participatory design sessions. Wahid et al. conducted several studies regarding the reuse of images and design knowledge for storyboarding [WBC⁺09, WBMH10]. Finally, the work of Truong et al. provided some guidelines for storyboarding but only involved designers and HCI specialists [THA06]. Although in these

studies, storyboarding is the main technique that is investigated, the background of participants is limited to HCI specialists or end users and it is not clear to what extent developers and other stakeholders can be part of their storyboarding approaches.

Since our assumptions regarding COMuICSer storyboarding need to be investigated before further research on storyboarding can be started, we conducted two user studies. The central questions to be examined in these user studies were: Are storyboards and accompanying tool support useful for practitioners in user-centered design (UCD)? How are storyboards created in multi-disciplinary teams? A first user study aimed to establish how easily multi-disciplinary team members would understand the ideas represented by a storyboard. Furthermore, this study examined the participants' need for storyboarding in multi-disciplinary teams and evaluated the tool described in Chapter 5. The second user experiment was an observational study, conducted to observe how storyboards can be created in meetings that involve several members of a multi-disciplinary team.

6.2 First User Study

To investigate whether if the storyboarding approaches proposed in Chapter 5 fit in common practice, we created a storyboard describing the usage of storyboards in a multi-disciplinary team. This storyboard was based on the results of the aforementioned interviews with practitioners (Chapter 3) and describes how storyboards are created and used in all stages of a user-centered software engineering (UCSE) project. Accompanying personas personify the basic disciplines involved in UCSE projects. This storyboard was used for this user study because of two reasons. First of all, we wanted to confront the participants with a COMuICSer storyboard to establish its understandability for people having different backgrounds. Second, the storyboard allowed us to inform and consult the participants regarding the use of COMuICSer storyboards and the use of an accompanying tool in a multi-disciplinary team.

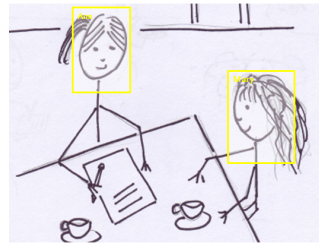
The storyboard visualized a scenario describing the user-centered process that was followed in order to design and develop a system for waiters in restaurants. Figures 6.1 and 6.2 present this storyboard. In the process represented by the storyboard, the focus was on our COMuICSer storyboarding tool as a central system for UCSE. The primary personas involved in the storyboard had different roles including the restaurant owner who was the customer, an HCI specialist who was also the project manager, a developer and a designer. The storyboard consisted of several scenes representing the different steps that

needed to be taken in a UCSE project. Each scene clearly presented the users that were involved and the systems that were used in a particular context. The scenes ranged from a first meeting with the customer and the HCI specialist discussing some ideas of the future system, over the creation of the COMuICSer storyboard by the HCI specialist, meetings and collaborations with team members and users, to the usage of the final system by waiters. We assured that several types of usage of storyboards were depicted by our storyboard, including, creating, editing, managing, sharing and presenting storyboards. The personas and scenario used for this user study are presented in Appendix A, Section A.2.

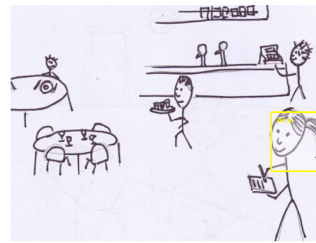
6.2.1 Participants and Procedure

In order to assess the concept of storyboarding in an entire UCSE process, we evaluated the storyboard in an informal study consisting of two phases: an informal evaluation of our COMuICSer storyboarding tool, which was presented in Chapter 5 and a walkthrough to validate the storyboard. The latter enabled us to establish whether storyboards can benefit the common understanding of UCD practitioners and whether our tool provides suitable visualizations to discuss the use of a future system. Four female and three male UCD practitioners participated. Participants ranged in age from 23 to 39, with an average age of 29. Their backgrounds were very diverse, including interaction design, graphic design, computer science and media studies. The participants' experience in multi-disciplinary project teams ranged from a few months to more than five years. All but one participant had at least some experience in using storyboards before this user study. The one participant that was not familiar with storyboarding, had a technical background.

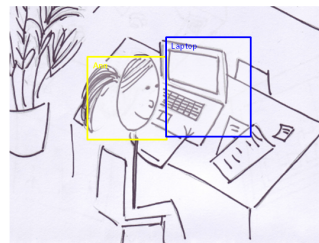
Each session took approximately one hour and started with a pre-study questionnaire asking the participant about their expertise and their experiences of being part of multi-disciplinary teams. Next, the participant was briefed about the study, was provided with a manual of the storyboarding tool and was asked to explore the features of the tool by creating a storyboard. Since our evaluation of the COMuICSer tool focused on the features to digitize and annotate a storyboard, and not its creation, the participants could use an existing scenario, sketches and personas. Once the participant got a first impression of storyboarding and the transition to UI designs, she participated in the storyboard walkthrough. During this walkthrough, each participant was asked to keep in mind a given persona description while discussing the storyboard with the facilitator scene by scene, using the tool. The personas



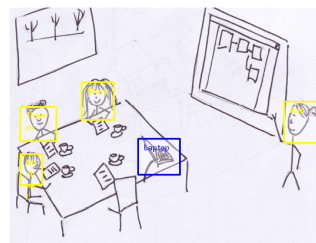
(a) Exploratory meeting.



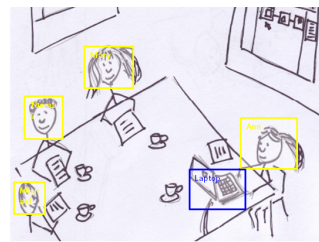
(b) Contextual inquiry.



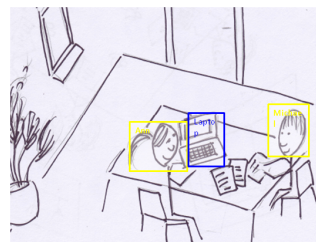
(c) User needs analysis - first iteration.



(d) Stakeholder meeting - presenting storyboard.



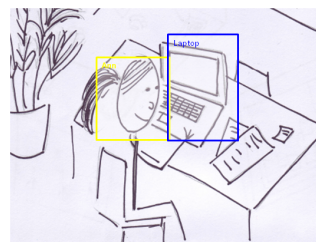
(e) Stakeholder meeting - editing storyboard.



(f) User needs analysis - second iteration.

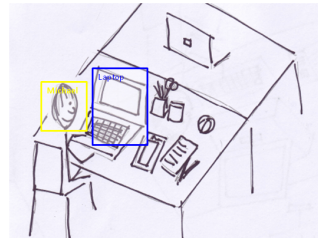


(g) Feedback meeting with customer.

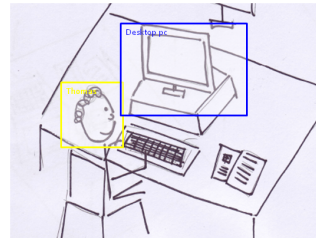


(h) Low fidelity prototyping - first iteration.

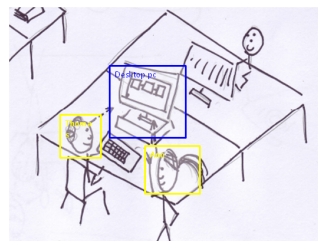
Figure 6.1: First part of the storyboard that was used for the walkthrough of our study. The scenes and their titles are presented in a chronological order.



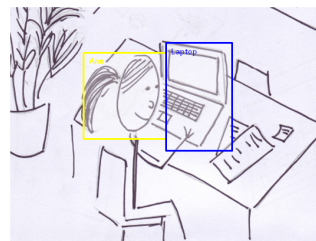
(a) Visual design.



(b) Structured interaction analysis.



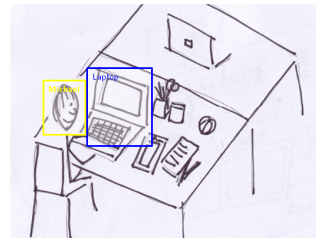
(c) Discussion of structured interaction models.



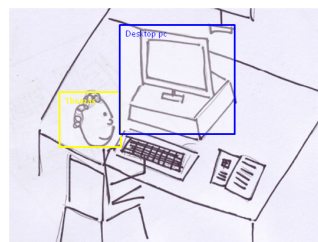
(d) Low fidelity prototyping - second iteration.



(e) First usability test with the low-fidelity prototypes.



(f) Detailed UI design.



(g) High fidelity prototyping.



(h) Field test.

Figure 6.2: Second part of the storyboard that was used for the walkthrough of our study. The scenes and their titles are presented in a chronological order.

assigned to the participants corresponded at least partly to the participants' current jobs. Three participants were asked to keep in mind the persona of the HCI specialist, three participants had to consider the storyboard from the perspective of the developer and one participant discussed the storyboard from the point of view of the designer. The discussions of the storyboard were recorded using a voice recorder and after the walkthrough, the participants filled in a post-study questionnaire concerning the storyboard, accompanying personas and the storyboarding tool. This post-test questionnaire asked the participants about their preferences regarding the features of our COMuICSer tool, the type of tool and approach they would use for storyboarding and their understanding of the storyboard during the walkthrough. The questionnaires used for this user study are presented in Appendix A, Section A.2. The following sections present our observations as well as the answers to the questionnaires.

6.2.2 Results

The answers to the pre-study questions asking about the participants' experiences while being part of multi-disciplinary teams confirmed that communication is the main difficulty. According to most participants, communication problems result from the gap in domain knowledge between team members. Four participants also mentioned that this problem stems from using different vocabularies while three participants refer to the differences in goals as a cause for these difficulties. Nevertheless, all participants agreed that despite these communication problems, involving the different perspectives present in multi-disciplinary teams is beneficial for the user experience of the resulting system.

The exploration of the COMuICSer tool resulted in positive remarks of the participants regarding the possibility to digitize storyboards. The participants did not claim that the tool is better than pencil and paper, but they noted: *"the tool is as quick as pencil and paper if you have everything in the tool to make the storyboard"*, *"pencil and paper is faster, but the tool is more flexible to handle changes"* and *"when sketches are available, the tool is very useful and flexible"*. Although for this study, the sketches for the scenes were provided to the participants, some participants appreciated the fact that the sketches of the storyboard could be created freely using pencil and paper. However, other participants were concerned that they were not very talented in making sketches and when considering the use of a storyboarding tool in future projects, they requested features that would facilitate the creation of story-

board sketches in the tool. This exploration uncovered some minor usability issues and general remarks regarding the tool. The participants considered the features to add and connect annotations (Figure 5.2) as easy to use and intuitive, one participant even states: *“This type of storyboard is some kind of map, suggesting and showing considerations that are important for the future system.”*

The visual representation of the storyboard presented during the walk-through stimulated discussion and the tool facilitated the understanding: clicking the annotations available in a scene enabled the exploration of personas and devices. While discussing the storyboard scene by scene, we observed that a focus view on a particular scene was missing in the tool. After the walkthrough, all participants declared it was easy to understand the approach depicted by the storyboard and they all confirmed that the persona and device annotations that are visualized in the tool (Figure 5.2) contributed to their understanding. All participants agreed that the general approach presented by the storyboard was realistic and feasible. They all accepted the use of storyboards in a multi-disciplinary team. Most participants preferred various disciplines to be involved during the creation of the storyboard. They think that not only the HCI specialist but also other roles in a team can contribute to the creation of the storyboard and this might also avoid misunderstandings at later stages. Some participants even clearly stated that storyboarding should be a group activity. The participants with a technical background, that were asked to consider the storyboard from the perspective of the developer persona, confirmed that storyboards are useful for them as well. Their work could benefit from storyboards because this notation provides more explicit information concerning activities carried out by users, which can be used for structured interaction modeling and to obtain more concrete requirements.

6.2.3 Discussion

The storyboard was a valuable artifact for discussion and proved to be very understandable for UCD practitioners with different backgrounds. The annotations, highlighted in the tool, contributed to the understanding of the storyboard. The participants confirmed that digitizing the storyboard and adding annotations would be useful in user-centered approaches and agreed that the storyboard could be used by several stakeholders for several design, development and evaluation activities.

The discussions of the walkthroughs, revealed that the types of storyboarding activities presented in our storyboard, can be classified as follows:

(1) creating, (2) editing and managing and (3) sharing and presenting. The COMuICSer storyboarding tool that was evaluated in this study, only partly supports the creation of storyboards because the scenes need to be sketched or photographed beforehand. The major features of the tool concern editing and managing storyboards' structure and annotations, which is interesting when the storyboard is used by several team members for several user-centered activities. Presenting the storyboard was initially included as one of the goals of the tool, but the walkthrough revealed that the current visualizations for storyboards are not completely suitable for presenting storyboards. Some participants suggested to support the transformation of the storyboards to a slide show or movie to present storyboards to stakeholders. Four participants noticed that the tool should adapt its view and annotations according to the background and goals of the team member working with the tool or the purpose of the storyboard at a particular stage.

Although it became clear that storyboarding is a creative activity and there is no general approach to obtain a storyboard, many participants emphasized that the storyboard should be created by people having different backgrounds. Some participants prefer to create storyboards in group meetings, in order to obtain a common understanding and avoid misunderstandings in the team. This emphasized to us that this first user study only provides preliminary results regarding storyboarding for multi-disciplinary teams. This first user study gave us an impression of how UCD practitioners would accept and use COMuICSer storyboards and the tool. Furthermore, we observed that the notation was suitable to explain the context of use of a future system to people having different backgrounds. However, since many UCD practitioners mention that the creation of a storyboard should be a group activity that involves several different disciplines, additional research is needed. The next section describes an observational study that examines the creation of storyboards in multi-disciplinary teams.

6.3 Observational Study

The results of our first user study described above, show the interest of practitioners to use COMuICSer storyboards throughout a user-centered project and the need for storyboarding tools. However, some participants emphasized that the creation of storyboards is a responsibility that should be shared among different team members and preferably should be done in collaborative meetings that involve several disciplines.

Based on our survey, described in Chapter 5, we have no clear understand-

ing of how storyboards are created collaboratively when a multi-disciplinary team is involved. We assume that the understanding of team members would benefit from adding annotations and connecting the storyboard to other artifacts, but we do not have clear evidence for this.

The creative process of storyboarding for individual designers was investigated in earlier work [THA06]. Storyboarding in team was observed to explore how highly integrated work can be supported by digital tabletops [PGS06]. A similar study was conducted to investigate the relationship between images and design rationale [WBMH10]. However, none of the aforementioned studies consider the respective contributions of team members having different backgrounds, roles and goals in a user-centered approach, therefore, we conducted an observational study of storyboarding sessions involving multi-disciplinary teams.

Observing real-world situations can provide us with valuable information on how multi-disciplinary teams organize their tasks, what kind of artifacts they produce, what tools they commonly use, etc. However, for this study, it was not possible to conduct a contextual inquiry regarding the use of storyboards from the beginning to the end of user-centered processes. Consequently, in this observational study we concentrate on the creation of the storyboard, and take the scenario and personas as starting point for this task. Observing the participants of our study during the collaborative creation of storyboards, will enable us to gain insights into how we can design storyboarding tools to effectively support this particular style of work.

6.3.1 Participants, Setup and Procedure

Twelve participants, four female and eight male, were divided into three groups of four people. Participants ranged in age from 22 to 46, with an average age of 31. Nine participants had a background in computer science, while the others had a background in history, visual arts and product design.

Multi-disciplinary teams were obtained by asking each participant to take on a particular role in the role-play of the study, so that each team consisted of an HCI specialist, a user interface designer, a systems analyst, and a stakeholder (end user or application domain specialist). These roles were selected with the different backgrounds of UCD [Int99] and the practices of industry (Chapter 3) in mind. We assigned the roles based on the participant's current role or job. The results from the post-study questionnaire indicate that most participants felt 'comfortable' to 'very comfortable' in their role. Only three participants felt 'neutral' toward the attributed role. All but one participant

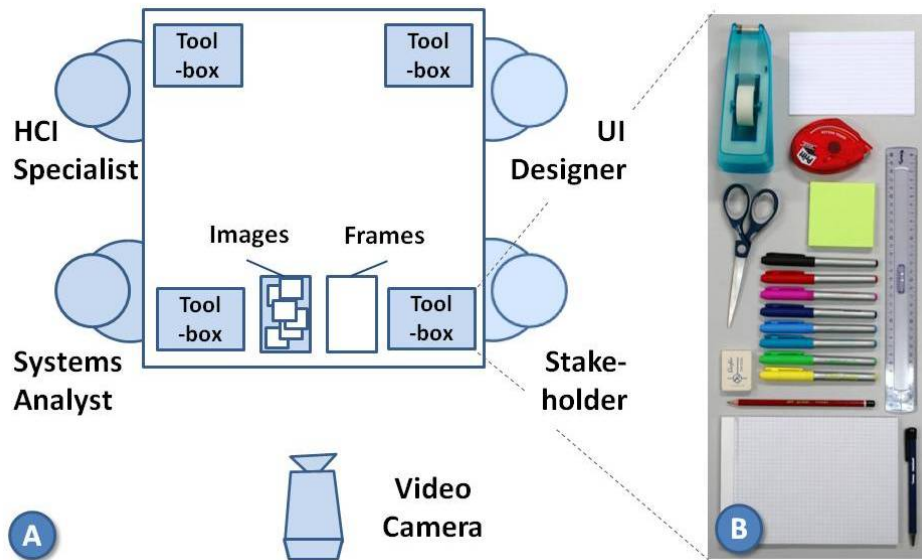


Figure 6.3: Setup of the observational study (A) and contents of the toolbox that was provided to each participant (B).

had at least one year of experience in being part of a multi-disciplinary team. All participants in the role of HCI specialist had experience in creating storyboards, while approximately two-thirds of the other participants did not.

In order to fine-tune the setup and procedure of our study, we conducted a pilot study that involved four participants. The setup and procedure of this pilot study did not differ much from our final approach which is described below, but inspired us to explicitly include an individual preparation phase in each session in order to force each team member to get prepared for the storyboarding session with her background in mind.

The sessions were carried out in a room equipped with a table (160 cm x 160 cm) and four chairs positioned around it so that each group member would initially be seated at a different corner of the table (Figure 6.3, A). A video camera was mounted in such a way that the whole table could be recorded for later analysis. The table was covered with a paper tablecloth. On top of it, participants could find a stack of A4 sheets to create scenes in the storyboard and a box with images representing persons and items from a scenario. Each participant was also provided with a box of ‘tools’: a regular pencil and

ballpoint pen, colored pens and highlighters, an eraser, a ruler, scissors, glue, adhesive tape, a notebook, post-it notes, and index cards (Figure 6.3, B).

A briefing and instructions for the task, a short description of the participant's role, personas, and a scenario were provided to each participant. The personas and scenario, which are added in Appendix A, Section A.3, revolved around a family that uses a home automation system to control the heating and lighting, and that can assist in saving money on energy consumption. The system can be controlled by different people (differing in age and technological aptitude), using different devices (e.g. touchscreen, laptop, smartphone). Although it was not explicitly mentioned, the description of the scenario suggested that the home automation system should take into account settings related to personal profiles and activities, settings of a profile should be merged in particular situations, and the system should be able to detect the presence of people in particular rooms.

Each group was asked to create a storyboard that represents the given scenario. Once they read and understood all instructions, including personas and scenario, each participant had 15 minutes to prepare individually. They were asked to write down or sketch anything they considered to be important, bearing in mind their specific role and goals. After the preparation, the participants were permitted to start the storyboarding task. Each group was told they had 60 minutes, and we stopped groups that went over the time limit. Upon completion of the storyboarding task, the participants were asked to fill in a questionnaire asking about their former experiences and collecting their findings regarding the workshop, the storyboarding task, the collaboration within the multi-disciplinary team and opportunities for future storyboarding tool support. A copy of this questionnaire is available in Appendix A, Section A.3. Throughout each session, two observers took notes about the actions and things said by the participants. During the storyboarding task, they only answered questions that related to the instructions. Other questions were left up to the participants to resolve using their own judgment.

6.3.2 Results

In this section we present the findings of our observational study, based on the results of the questionnaire and the observations made throughout the three storyboarding sessions. Table 6.1 provides an overview of our observations, of which the most important results are discussed in this section and Figure 6.4 shows frames of the videos that were recorded during each session. Since each frame belongs to the end of the video, they give an impression of the materials

used during the session and the resulting storyboards.

During the individual preparation prior to the cooperative storyboarding task, several participants highlighted phrases in the text that was provided in the beginning of a session. All participants structured the information in a certain way: a few participants used bulleted lists, while the others represented it in a graphical way, ranging from diagrams to freehand sketches. Materials used during this preparation included the available images, colored pens, notebooks, post-it notes and index cards.

In terms of content, the roles of the participants were clearly expressed in the artifacts they prepared. The HCI specialists mainly focused on the relationship between personas, devices and tasks, the designers on UI aspects or designs and accompanying requirements, the systems analysts on the devices and their connections, and the stakeholders on general requirements and the needs of the personas. In all sessions, participants began to explain their prepared artifacts to each other once the cooperation started, but in two out of three sessions, not all participants presented their preparation. Artifacts were rarely explicitly added to the storyboard, but participants did use them to explain their ideas during discussions.

The approach to the storyboarding task differed in the three teams. Team A started by shortly discussing their strategy for storyboarding and decided to first depict the equipment and users in the different rooms of the house. Almost immediately after this decision, they started creating the first scene collaboratively. Next, the team implicitly split in two to prepare other scenes. Awareness was maintained, since participants frequently switched between cooperating with their neighbor and cooperating with the entire team, and most of the work was done in the middle of the table. The HCI specialist maintained the relationship between the storyboard and the available scenario.

Team B first discussed the system based on the requirements mentioned by the application domain specialist. After a discussion of approximately 15 minutes, in which some decisions regarding the system were already made, the HCI specialist reminded the team that the storyboard should depict the scenario and took the lead in creating scenes. The other team members were actively involved in the discussion and handed required images to the HCI specialist. Once the HCI specialist started creating a new scene, the application domain specialist finalized the former scene.

Similarly, team C first discussed the system based on the requirements presented by the application domain specialist. This discussion lasted almost 30 minutes before a first scene was created. However, while discussing the devices for the system, the available images were put in the middle of the table

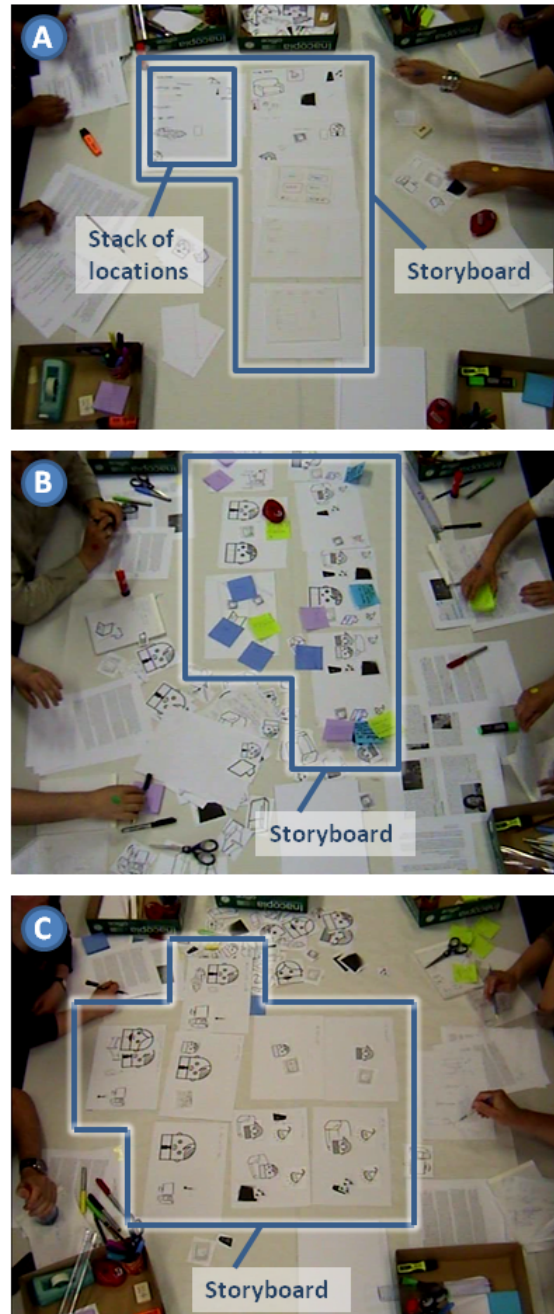


Figure 6.4: Frames of the videos that were recorded during each storyboarding session.

	Team A	Team B	Team C
<i>Prepared Artifacts</i>			
HCI specialist	overview of personas	overview of devices, UI and usability aspects	notes of goals, tasks, sketches devices
Systems analyst	devices and personas that use them	connected devices, hierarchy profiles	sketches displays, preliminary storyboard
Designer	sketches locations, personas, devices	notes of requirements, UI aspects	UI designs, preliminary storyboard
Stakeholder	overview of personas and devices	highlight in scenario, notes of requirements	highlight in scenario, notes of devices, requirements
<i>Storyboarding Observations</i>			
Lead	HCI specialist	HCI specialist	HCI specialist
Collaboration Activities	discussion, subteams prepare scenes	discussion, subteams prepare scenes	discussion, individuals select images
Round-up Activities	restructure, add post-its	check needs, add map with overview	add UI designs, check order, annotate scenes
<i>Resulting Storyboards</i>			
Number of scenes	7	8	10
Structure	spatial, stack of location scenes, expandable to show details	chronological order, numbered scenes	chronological order, numbered scenes
Use of Words	locations, some devices	post-its with reminders	titles of scenes
Use of UI designs	for particular location	post-its on top of the scene	shown on the screen of the device

Table 6.1: Overview of the most important results (observations) of our observational study.

to debate the different options and to decide which devices should be used. Again, it was the HCI specialist who reminded the team of the storyboard and who started creating the different scenes, based on the discussion.

When considering the storyboarding tasks as a whole, all teams discussed the system's features that were implicitly described in the scenario, such as using different profiles and activities, merging profiles and sensing someone's presence in a room. Sometimes it was just one team member that noticed a particular requirement, but in many cases, these discussions were inspired by visually representing situations.

The resulting storyboards consisted of seven to ten scenes. All storyboards contained scenes representing personas and devices, and showed the status of particular devices (e.g. a light that was switched on or off). Two storyboards contained text to indicate the location or the general situation of scenes. One team added post-it notes to the storyboard that would remind team members of particular features, difficulties and decisions.

Structuring the scenes of the storyboard was done in different ways. Team A created a visual representation of all rooms, and for each room the situation was depicted. Teams B and C tried to put the scenes in a chronological order, based on the flow of events in the scenario. Extra scenes were inserted into the storyboard sequence when considered necessary. Scenes were labeled with numbers, and in team C, titles were added to each scene as well.

When asking about the storyboarding task and the collaboration in the questionnaire, HCI specialists and stakeholders rated their direct contribution to the storyboard highly and were satisfied with the extent of this contribution. Systems analysts and designers, however, rated their contribution and satisfaction significantly lower, as they prepared artifacts that were used to a lesser extent during the session. Stakeholders and systems analysts rated their general influence on the storyboard notably higher than their direct contribution. Frequent discussions between stakeholders and analysts about the feasibility and costs of particular approaches account for this difference. The generally high involvement of HCI specialists is likely attributed to their former experience and leading role.

The participants confirmed in the questionnaire that being part of a multi-disciplinary team had a positive impact on the storyboarding session: *"Being part of the multi-disciplinary team made us look at the storyboard from each perspective"* (HCI specialist), *"By having storyboarding sessions that involve multi-disciplinary teams, the technical and economical constraints are known in the early stages of a project."* (systems analyst), *"Sharing ideas in a multi-disciplinary team results in other new ideas."* (designer), *"Different perspec-*

tives gave more ideas.” (stakeholder). Acting as a multi-disciplinary team clearly resulted in different perspectives, ideas and considerations.

When we asked the participants about the aspects of a storyboard that would be useful for their following tasks during the project given the roles they had during the workshop, the HCI specialists answered: *“I would use the description of interaction, and the interaction steps in the UI”, “I would use the storyboard to evaluate the system against the storyboard.”* One designer answered that he would not use components of the storyboard that would give exact information for his designs. Two other designers answered that could extract elements of the UI by consulting the visualization of the software system provided by the storyboard. The systems analysts answered: *“I would use the storyboard to see the devices that are used and how the users interact with them.”*, *“Mainly the technical limitations with respect to the hardware are useful (e.g details regarding the hardware that is used and the installation of the hardware).”* Finally, the stakeholders answered: *“Although this storyboard is still in a preliminary state, I would use the whole storyboard.”*, *“I would use it to keep in mind all different parameters.”*

All participants responded positively when asked about the usefulness of a digital storyboarding system for multi-disciplinary teams. Features that were often reported as being essential point toward the traditional benefits of a digital system, include the ability to locate tools or images more easily and editing operations such as moving or scaling items in a scene, cloning or reorganizing scenes and other artifacts in the storyboard, and saving scenes and artifacts for later reuse.

This observational study showed us possible approaches to collaboratively create storyboards in multi-disciplinary teams. Many suggestions for tool support are suggested in this study. However, in the remainder of this dissertation, we will mainly focus on storyboarding approaches that support the involvement of multi-disciplinary teams in UCSE.

6.3.3 Discussion

The artifacts prepared by the participants show that people with different backgrounds and goals create different artifacts to prepare their job. We observed that the focus differs for each discipline. HCI specialists focus on the relationship between personas, devices and tasks and are familiar with notations such as a storyboard. Stakeholders also carefully bear in mind the needs of personas, but they also think in terms of requirements that need to be considered in the resulting system. Designers mainly concentrate on UI designs,

and systems analysts are interested in devices and their connections. Some of the artifacts that were prepared, were not used for the storyboard, however, they were used during the discussion. These observations show the need for connecting the storyboard to other artifacts in collaborative or individual sessions.

Our observations of the storyboarding task confirm the assumption that a highly creative activity such as storyboarding is not easily generalizable. Although the teams created storyboards in different ways, the approaches and results are alike to some extent. In all teams, the HCI specialist took the lead to guide the team during the creation of the storyboard. The storyboarding task stimulated discussion of requirements and helped the team to make decisions regarding the system, the devices, and their use in a particular context. Visually presenting situations by using available images provoked the team to clarify parts of the scenario and to make decisions about the system that needed to be designed and developed.

The storyboards that resulted from the three storyboarding sessions had a similar number of scenes. However, one team preferred a spatial structure, while two teams structured the scenes in a chronological order. After the storyboarding session, the participants' ratings of their own contribution to, influence to and satisfaction of the storyboard varied. In general, the systems analysts were less satisfied about the resulting storyboard and they rated their contribution lower than their influence, which is confirmed by our observations. We observed that the systems analysts actively participated in the discussion. Their limited contribution to the storyboard can be explained by the fact that the systems analysts' are usually involved later in the user-centered project. Nevertheless, all participants confirmed that the involvement of multiple disciplines benefited the storyboarding sessions.

The participants did not only favor the involvement of a multi-disciplinary team because of the involvement of different viewpoints, they also were very positive about the idea to have tool support for the creation and use of storyboards. Possible directions for tool support are extensions of the individual storyboarding tool and a collaborative tabletop system. The features suggested by the participants do not tend to one specific platform but are applicable for individual as well as collaborative storyboarding tools. Furthermore, based on the fact that not all artifacts prepared by the participants were used in the storyboard may imply that team members of a multi-disciplinary team would prefer to connect some artifacts to a storyboard in a storyboarding tool that is used individually.

6.4 Conclusion

Two complementary user studies were presented in this chapter. These user studies were conducted in order to evaluate our initial ideas regarding storyboarding and tool support. On the one hand, we investigated how storyboards could be used to support multi-disciplinary teams. On the other hand, we obtained insights regarding the involvement of multiple disciplines in the use and creation of storyboards.

While the first user study confirmed most of our assumptions regarding the use of storyboards in a user-centered design and development process, not all participants of the study agreed with the way a storyboard was created. We assumed that a HCI specialist prepared a storyboard before presenting it to the other team members, but many participants of the first user study stressed that the creation of a storyboard should involve people having different backgrounds. Preferably, the storyboard should be created collaboratively. Since our literature survey (Chapter 2) and interviews regarding UCSE in practice (Chapter 3) did not answer the question how a storyboard can be created collaboratively, we decided to conduct an observational study to find out how storyboards can be created collaboratively by multi-disciplinary teams. The observational study showed that the involvement of different backgrounds has a positive impact on the creation of the storyboard. Every team member actively participated in the discussion, but it became clear that usually, the HCI specialist took the lead in the creation of the storyboard and maintaining the relationship between the storyboard and the scenario. Furthermore, the observational study confirmed that team members with a particular role use similar artifacts, which can be related to a storyboard.

Both studies showed that the visual representation of storyboards stimulates discussion, which confirms earlier findings presented in literature [Cil11, KJ07]. In our studies, we observed that not only during the creation of the storyboard, but also when a storyboard was created beforehand, discussion is provoked when it is shown to stakeholders. Furthermore, we can conclude from the two studies that the relationship between storyboards and multi-disciplinary teams is twofold. The communication within a multi-disciplinary team can benefit from the use of storyboards, which is confirmed by the participants of the first user study. The storyboard can be used throughout a user-centered project in order to obtain and maintain a common understanding about the context in which the future system will be used.

In this chapter we investigated two research questions that were introduced in Chapter 5. RQ 1a: *Are storyboards and accompanying tool support useful*

for UCSE practitioners?, was answered by the first user study. The participants of our study confirmed that the storyboard could be used by members of a multi-disciplinary team for several design, development and evaluation activities and they agreed that digitizing the storyboard in a tool and adding annotations would be useful in user-centered approaches. Our observational study partly answered RQ 1b: *How are storyboards created in a multi-disciplinary team?*. We observed that the involvement of different roles and disciplines involved lead to different contributions to a storyboard. Furthermore, the approach to structure the storyboard differed for the participating teams.

The results of the studies discussed in this chapter will be used as a starting point for the research that is presented in the next chapters. Chapter 7 will describe the storyboarding language and accompanying tool support. The connection and transition between storyboards and formal artifacts will be presented in Chapter 8. Finally, a possible approach to combine user-centered design and software engineering is proposed in Chapter 9.

Chapter 7

The Visual Storyboarding Language

7.1 Introduction

A visual language allows people, regardless of their backgrounds, to see, imagine, present and share ideas with others [Roa08]. Storyboards are examples of scenarios expressed in a graphical language using rich and iconic pictures, which can increase the ability to recognize and understand their meaning [vdL06, THA06]. This characteristic of storyboards corresponds to the principle of *semantic transparency*, introduced by Moody [Moo09], and consequently infers that the cognitive load to understand the notation is reduced. We also observed in the user studies described in Chapter 6 that the notation of storyboards was easy to understand.

Although the visual language of storyboards is very accessible, creating a storyboard is not as straightforward as reading it. As we observed in the user studies described in Chapter 6, many HCI practitioners are not trained as a designer and are hesitant when sketching storyboards because of limited drawing skills. Roam [Roa08] states that artistic trainings or talent are not needed to create insightful pictures: being able to draw some basics, should suffice to express things in a visual language. However, most of the participants of our studies gratefully used sketches or images that were provided to them in order to create a storyboard.

A study of Truong et al. [THA06] showed the differences in storyboarding by novice designers and experienced designers. While novices fear the lack of drawing skills and avoid creating hand drawn storyboards by using tools, expert designers are not hesitant to draw storyboards having different levels of

detail by hand or using advanced tools. Consequently Truong et al. address the need for a tool that offers all the functionality needed to create storyboards. Existing storyboarding tools such as Comic Life [Pla], Celtx [Cel] and Kar2ouche [Imm] support the creation of storyboard scenes, but do not support the link to later stages in user-centered software engineering (UCSE) projects. Also, it is not possible to extract information (e.g. personas) from these storyboard that is useful for later stages in UCSE.

We want to provide tool support for the creation of storyboard scenes for UCSE purposes, without limiting the creativity of designers, other team members or stakeholders. There are multiple ways to visually present a usage scenario. Van der Lelie [vdL06] explains that a sketchy storyboard describing the use of a product provokes discussion, while a detailed storyboard is rather suited for an evaluation phase in design. Buxton [Bux07] also mentions different styles in visual storytelling. He proposes sketches as well as hybrid photographic compositions. Combining photos and sketches, might help people with limited drawing skills to explain their ideas [Bux07] and might allow people to quickly create storyboards [KJ07].

We investigated the language used in comics in order to learn what aspects of a visual language can contribute to storyboards, and how these aspects can be incorporated into storyboards. In this chapter, we provide an elaborate overview of techniques that are common in comics and can facilitate storyboarding. Comic drawing techniques will provide us with a frame to discuss scene creation in storyboarding. This frame was tested during a user experiment. Both the frame and the results of the user experiment resulted in the inclusion of features to facilitate the creation of storyboard scenes in our storyboarding tool.

7.2 The Language of Comics in Storyboarding

The principles of the invisible art of comics, described by McCloud [McC93], were used by Truong et al. as an established framework for visual storytelling [THA06]. These principles, in combination with the techniques to draw comics [McC06], provide basics for storyboarding. Some techniques of comics were already applied in Comicboarding to facilitate participatory design with children [MLD⁺07]. Furthermore, the approach of comics has been used to visually describe the features of the Google Chrome browser when it was launched [GCt], which according to McCloud's site [McC] was called "*one of the friendliest technical descriptions the software industry has yet produced*" (*Forbes / Newsweek*). In this section we describe a set of principles

and techniques from comics that were proposed by McCloud [McC93, Mcc06] that facilitate the graphical language of storyboarding.

7.2.1 Visualizing and Communicating Scenarios

McCloud describes principles and techniques of comics and defines comics as: “*Juxtaposed pictorial and other images in deliberate sequence, intended to convey information and/or to produce an aesthetic response in the viewer.*” He describes the *vocabulary* of comics as a combination of words, pictures and other icons. When comparing pictures with writing, *pictures are received, while writing is perceived*. People can almost immediately understand a message depicted by a picture, while a written message needs to be decoded before it is understood. In comics, usually abstracted pictures are combined with more direct words to obtain the specific language of comics.

This indicates that in user-centered design (UCD) a similar vocabulary for storyboards can be a suitable complement to a narrative scenario in order to convey information. Team members of a multi-disciplinary team can understand at a glance what is depicted by a storyboard and efficiency in brainstorm meetings or discussions can be increased. The COMuICSer storyboarding approach, which was introduced in Chapter 5 considers scenes that contain a comparable vocabulary, which is easy to understand for all team members. Each scene can be complemented with a title and a description, giving more information of what is depicted in the scene. Usually, this description can be considered as a word-specific, picture-specific or duo-specific *combination of words and pictures*, that is known in comics. Truong et al. [THA06] suggest limiting the description to one sentence.

One of the difficulties in all communication media, and thus also in comics, is the *inability to communicate directly from one mind to another*. It is almost impossible to pass a message on paper without affecting it. A similar problem also occurs in HCI and is known as the gap between the designer’s model and the user’s model [Nor88]. In UCD, storyboards are used to obtain a uniform idea among all stakeholders, so discussing storyboards in UCD could be a remedy to the problem mentioned above. When storyboards are treated as a central document in UCD both the requirements elicitation and the general understanding within a multi-disciplinary team can be improved.

7.2.2 Applying Techniques of Comics

The following will describe several techniques of comics. For each technique, we discuss how it can contribute to COMuICSer storyboards. Some techniques

can only be used to draw the scenes of a storyboard, while for other techniques, the COMuICSer tool can be employed to ease its use.

Facial Expressions

In comics, *facial expressions* are important to show the emotions of characters in a story, and consequently provoke emotion in the reader. There is a plethora of possible facial expressions, but six basic types of expressions can be mixed as in a color palette to obtain other emotions [McC06]. The basic expressions depict whether a character is happy, in pain, shocked, angry, stern or distressed. Speech balloons can be used to add gradations to these expressions.

In storyboarding, facial expressions have a lower priority than in comics. In COMuICSer, emotions of users are expressed in a textual description that accompanies each scene of a storyboard. Nevertheless, the aim of facial expressions - provoking emotion in the reader - can also be beneficial for multidisciplinary teams. If a team member can empathize with the characters, she is more likely to understand the situation of end users and will take this into account for the resulting UI design. Figure 7.1 shows how a facial expression can show the emotions of the end user in a storyboard, the scene depicts a frustrated user.



Figure 7.1: *Facial expressions* in a scene, show the emotions of a user. This user is frustrated by the application on his PDA.

Body Language

Similar to facial expressions, *body language* can be used to express the temperament and mood of a character [McC06]. Characters existing of simple brush strokes can even express differences in body language. Besides the body language of a single character, distance and relationships between several characters can be shown through body language.

Body language has, just like facial expressions, not a high priority in CO-MuICSer storyboarding. But when including body language in scenes of a storyboard, the personality of a persona is amplified. Furthermore, body language will also help team members to empathize with characters. Figure 7.2 depicts the same frustrated user of Figure 7.1. By expressing his body language, emotions are amplified and a team member will more quickly understand that this user is frustrated. Because body language and facial expressions usually are expressed in the drawings of a storyboard's scene, providing tool support for was not within the scope of this research. A possible tool to support facial expressions in storyboarding tool is the Grimace Project [SF].

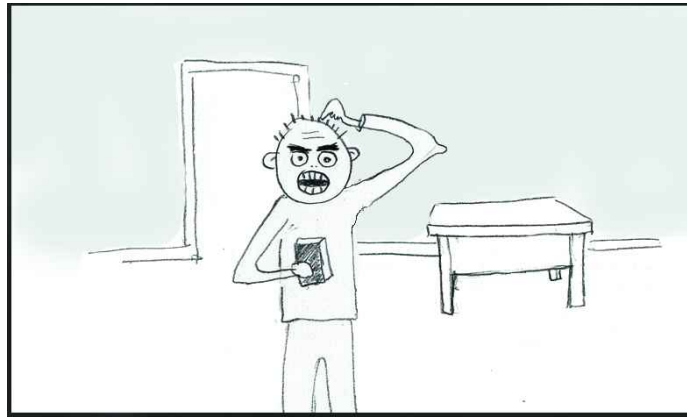


Figure 7.2: *Body language* can be incorporated in storyboard scenes to amplify the emotions of users. The body language used in this scene, stresses that the user is frustrated.

Iconic Characters on a Realistic Background

Comics artists often draw *iconic characters* because authors want their readers to identify with the actors of a story [McC93]. By leaving out very many details in a character's depiction, a reader is more likely to empathize with

the character. Several comics styles (e.g. Hergé's Adventures of Tintin) combine iconic characters with very detailed and *realistic backgrounds*. Realistic backgrounds provide the reader with as much information as possible about the context in which something happens. Consequently, the combination of iconic characters and realistic backgrounds allows readers to identify and thus empathize with the character and to imagine that they enter the world that is presented on the background.

These ideas fit storyboarding: UCD practitioners also want their team members to identify and empathize with the end users and to imagine the situation depicted by the storyboard as accurately as possible. Using photographs can also be helpful for team members with little drawing skills, who can for instance use photographs of user observations, complemented with iconic pictures of actors [Bux07, DSL06]. This approach allows the representation of a realistic environment for the end users that are considered during the entire UCD process. Figure 7.3 shows an example of this.



Figure 7.3: A *photograph in combination with an iconic character* preserves the possibility to identify with the user, but shows a lot of contextual information.

Differentiating Characters

Inner life, visual distinction and expressive traits are defined as the three components of successful character design for comics. These components allow characters to be *different* from other *characters* in a story [McC06]. By consistently applying these components, the reader will easily recognize actors in a comic. Before creating a character, an inner life or personality for this

character needs to be determined. This also helps readers to identify with a character. Providing visual distinctive characters allows readers to recognize actors of a story. This effect can be amplified by creating characters with contrasting looks. Finally, a character can be distinguished, based on expressive traits, which can be shown through facial expressions and body language.

COMuICSer storyboards allow personas [PA06] to be part of the scenes in the storyboard. First of all, personas describe the personality of a hypothetical archetype, so the inner life of personas can also be part of storyboards. Furthermore, these personas can have a distinctive appearance in scenes of a storyboard. Figure 7.4 shows an example scene, containing visual distinctive characters. However, in sketchy storyboards this might be hard to express. Therefore, annotations in COMuICSer storyboards can highlight personas (Chapter 5). By linking these annotations to the persona descriptions in the COMuICSer tool, it is easy for team members to identify particular personas.

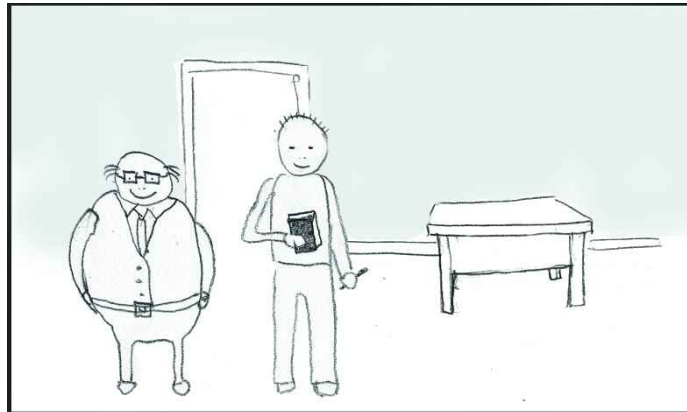


Figure 7.4: *Differentiating characters* is possible by giving them contrasting looks. This allows readers to easily recognize different characters.

Transitions

Observing parts of a picture, but perceiving the whole, is described as *closure* [McC93]. This means that particular parts of a whole do not need to be visible. The parts that are shown, allow the reader to imagine that the whole is present. One form of closure concerns the space between the panels of a comic, which is called *“the gutter”*. Although this space is empty, we

unconsciously add a transition between each two panels. The empty spaces define several possibilities for panel-to-panel *transitions* and can be considered as the heart of comics because they stimulate the reader's imagination. Buxton [Bux07] drew an analogy to this and declared that transitions are also the most fundamental component in an interface when creating the UI designs.

Since closure creates an illusion of time, time can be perceived spatially in comics. By changing the panel shape, it is possible to give the impression that much or little time passes during a panel. Although the spaces between panels also provide some time information, there is no rule that defines how much time passes between two scenes.

The principle of closure can be used in storyboarding to limit the number of scenes that need to be created. When transitions between scenes explicitly provide conditions or timing information, they can be used for the creation of models and UI designs. Figure 7.5 shows the user in only two situations, but the human brain automatically converts this to a continuous sequence that also includes the situation in which the character walks to the desk. However, since this type of transitions may be too vague for UCD activities, labeling the transitions or labeling scenes with timing information in COMuICSer storyboards, can enhance the meaning of subsequent scenes and the information they contain. Similar to comics, storyboards show sequential scenes, but the COMuICSer tool also supports parallel scenes, which is interesting to depict different remote users that simultaneously use, need or influence a system. This information about transitions is useful for the translation from a storyboard to structured interaction models or UI designs.

For instance, the case study that applied a MuiCSer process for the design and development of a prototype for a mobile game, presented in Chapter 3 [HDBCR09], would have benefited from the use of a storyboard that included labeled transitions and timing information. When a storyboard would have been available, and the time to walk from one location to another was labeled, the missing content in the game would have been noticed a long time before the field test took place and an important shortcoming of the user interface would be avoided.

7.3 User Experiment

In order to establish the applicability of comics in the context of storyboards, we conducted a user experiment. Similar to the approach taken by Truong et al. [THA06], we collected storyboards and identified key features in these storyboards. This experiment aimed to investigate whether techniques of comics

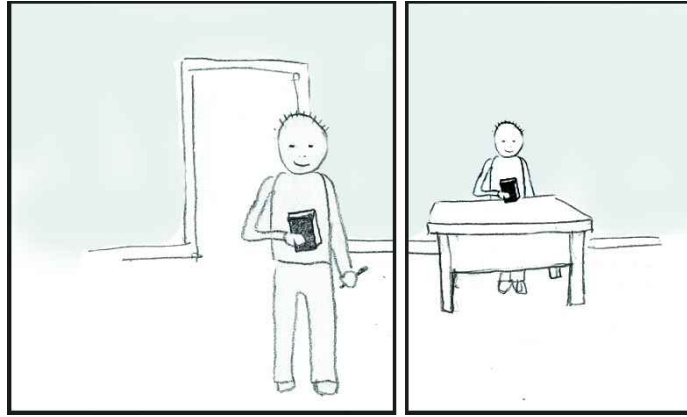


Figure 7.5: The *transition* between two scenes is automatically filled by the human brain. Although it is not explicitly depicted by these scenes, the reader deduces from the transition that the user in the scene moves to the desk and sits down during the transition.

were intuitively applied by three teams of novices without a visual design education.

7.3.1 Participants and Procedure

Seven students of our Master Computer Science-HCI, whose ages ranged from 22 to 25, were involved in our user experiment. The representativeness of students for this task may be questioned because they have little experience in UCD. However, our goal of this experiment was to see what techniques of comics intuitively would be used for the creation of the storyboards. In the context of a course, the participants had to use a range of UCD techniques in order to complete an assignment. The final result of the assignment was an evaluated, interactive high-fidelity prototype for a multi-device setting. The target devices of the assignment included a large multi-touch display and a smartphone. During the process, several artifacts needed to be delivered. For this experiment, we will focus on the early stages of the UCD process, that were part of the assignment.

We taught the students what personas, scenarios and storyboards are and what they are used for. Furthermore, we briefly referred to comics as a foundation for storyboarding, but the students were not obliged to use this for their storyboards.

The participants were split up in three teams for the assignment. Each team got a different assignment. In the first stage, the teams had to fine-tune the characteristics of the assignment in cooperation with one or two stakeholders. Three out of the four stakeholders involved, had a non-technical background. We observed the teams during these meetings with the stakeholders and analyzed their artifacts.

For this assignment, the participants were free to use any medium or tool for the creation of the personas, scenario and storyboard. Because we did not want to influence the choice of a medium for storyboarding, we did not introduce our COMuICSer tool or any other tools. First, each team organized a meeting to fine-tune the concept together with the stakeholders. In the second meeting, the teams presented their concept of the future system to the stakeholders, using personas, scenario and storyboard.

7.3.2 Results and Discussion

During the stakeholder meetings, we observed that the feedback of the stakeholders about *personas* was rather limited. Only one or two characteristics concerning the personas' job or personal situation, were discussed. We think this is due to the fact that the persona describes an archetype of the users of the future system, but does not include any characteristics of the system. Once the *scenario* was presented, an example of the future application was presented, and the stakeholders gave feedback about how realistic the situation was, compared to the current situation. It is notable that once the visual representation of the scenario, *the storyboard* was shown, the stakeholders started to discuss technical feasibility, user interaction, the number of users involved and the need for devices in the future application. While discussing the storyboards, stakeholders asked questions such as: "*How can a user enter his name on a multi-touch table? And what about privacy?*", "*Is it technically feasible to transfer data from a multi-touch table to your smartphone?*". This confirms our previous findings (Chapter 6): a storyboard stimulates discussion and is very useful to obtain a mutual understanding between all stakeholders. However, we do not want to claim that personas and a scenario are redundant in UCD projects.

The storyboards created by the participants were based on their narrative scenarios and were drawn using pencil and paper. Consequently, all artifacts were presented to the stakeholders on paper. The teams explained that they preferred pencil and paper for the storyboard because it provides freedom to draw quickly what they want to express.

Although all storyboards contained characteristics of comics, we could differentiate three styles of storyboards. Figure 7.6 shows the structure of storyboards A, B and C and one scene of each storyboard. The number of scenes in the three storyboards ranged from 6 to 12. Each storyboard included some kind of preliminary UI design, related to a scene. Storyboard A contained stick figures and sketchy objects, while storyboard C included elaborate characters and objects. The style of storyboard B could be situated between the aforementioned styles.

For a further comparison of the storyboards, we analyzed the use of the techniques mentioned in Section 7.2. *Facial expressions* and *body language* are not shown in storyboard A, but both storyboards B and C contain basic facial expressions in combination with speech balloons and limited expressions of body language. Storyboard A shows only the setting that is important for the future system, while storyboards B and C also consider the experience of users in that setting.

Since the three storyboards are sketched on paper, *iconic characters* are part of storyboards A, B and C. *Realistic backgrounds* are not fully applied to the storyboards, but storyboards B and C depict the context that is important to empathize with the end users. All storyboards include a *visual distinction of characters*. Storyboard A differentiates between characters by adding labels to the scene with the persona name, while storyboards B and C also differentiate in the appearance of characters. The small differences between storyboard A and the other two storyboards result in a different empathy while reading the storyboard. In the meeting where storyboard A was discussed, the future system was discussed from the viewpoint of “a user”, while the other storyboards were discussed from the personas’ points of view.

The *transitions* in the three storyboards are presented by arrows between scenes or panels for each scene. It is notable that all storyboards contain at least two transitions to present a concurrent and more detailed sketch of the user interface. Storyboard C distinguishes the transitions as follows: adjacent panels present concurrent scenes or a zoom on the UI, while a large “gutter” between panels presents a certain time interval between scenes. Furthermore we observed in storyboards B and C that panels are labeled with timing information concerning the start or duration of a scene.

This user study was an interesting experiment to analyze storyboards created by people having a technical background and different drawing skills and showed us that the techniques from comics presented in section 7.2 can facilitate the creation of storyboards. Since the participants freely applied several techniques, it is obvious that it is not difficult to apply them. Nevertheless,

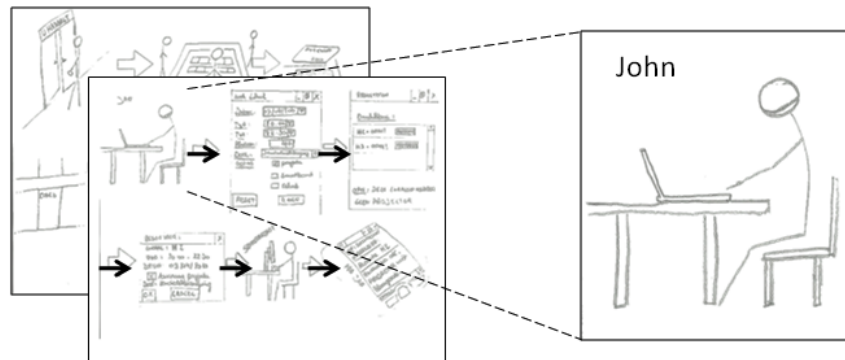
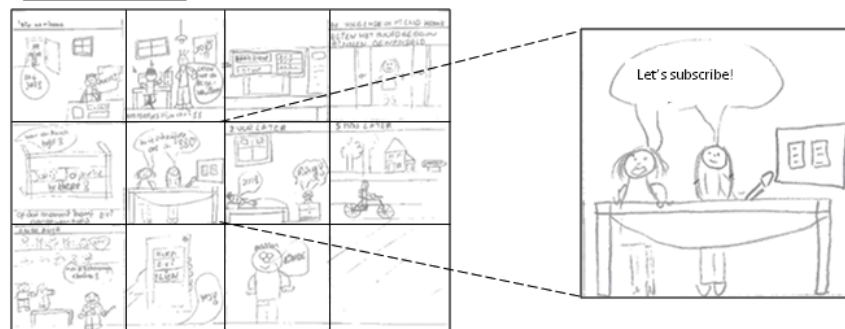
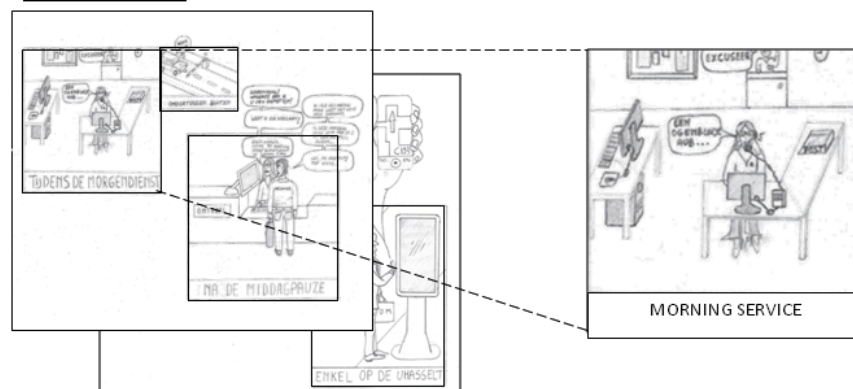
Storyboard AStoryboard BStoryboard C

Figure 7.6: The three storyboards that were analyzed in our user experiment. (To increase readability, we translated some words in the scenes.)

7.4 Incorporating Techniques of Comics in a Storyboarding Tool 87

the sketchy scenes of a storyboard do not always contain as much detail as comics. Quickly sketched storyboards are not rare in UCD. Consequently, tool support for storyboarding can be useful to enrich storyboards that are used in UCD. Certainly the *differentiation of characters*, *combining iconic characters with realistic backgrounds* and *labeling transitions and timing information*, are techniques that could be supported by a tool in order to encourage practitioners to apply techniques of comics to storyboards and to attach as much information as possible to the storyboard. The other two techniques *facial expressions* and *body language* are also interesting to be incorporated into a storyboarding tool, but focus more on drawing scenes than the other techniques. Some existing tools [Sne, SF] exemplify how facial expressions and body language can be included in a storyboard. However, besides limited support of facial expressions and body language, extended tool support for these techniques is out of the scope of our research. The next section describes how the techniques of comics are incorporated in our COMuICSer tool.

7.4 Incorporating Techniques of Comics in a Storyboarding Tool

Our COMuICSer tool, introduced in Chapter 5, mainly focuses on editing and managing storyboards by supporting the connection between the storyboard and artifacts created at later stages of UCD. Our earlier user studies (Chapter 6) and available literature clearly address the need for storyboarding tools that support the creation of storyboards. The correspondence between comics and storyboards inspired us to extend the storyboarding tool with features that facilitate creating, editing and managing storyboards. Furthermore, these features may implicitly remind team members to apply techniques of comics in their storyboards. The specific GUI elements and interactions that support this, are discussed in this section.

7.4.1 Composing the storyboard

Initially, our storyboarding tool allowed its users to load scenes that were sketched using pencil and paper. Since a plethora of useful drawing tools is available and the development of drawing features was out of the scope of our work, we did not include features to draw storyboard scenes. Often, teams will stick with pencil and paper, but at the same time find it very valuable to transform this in a digital artifact that fits well in a UCD process. However, we did include some features that can facilitate the creation of storyboard scenes

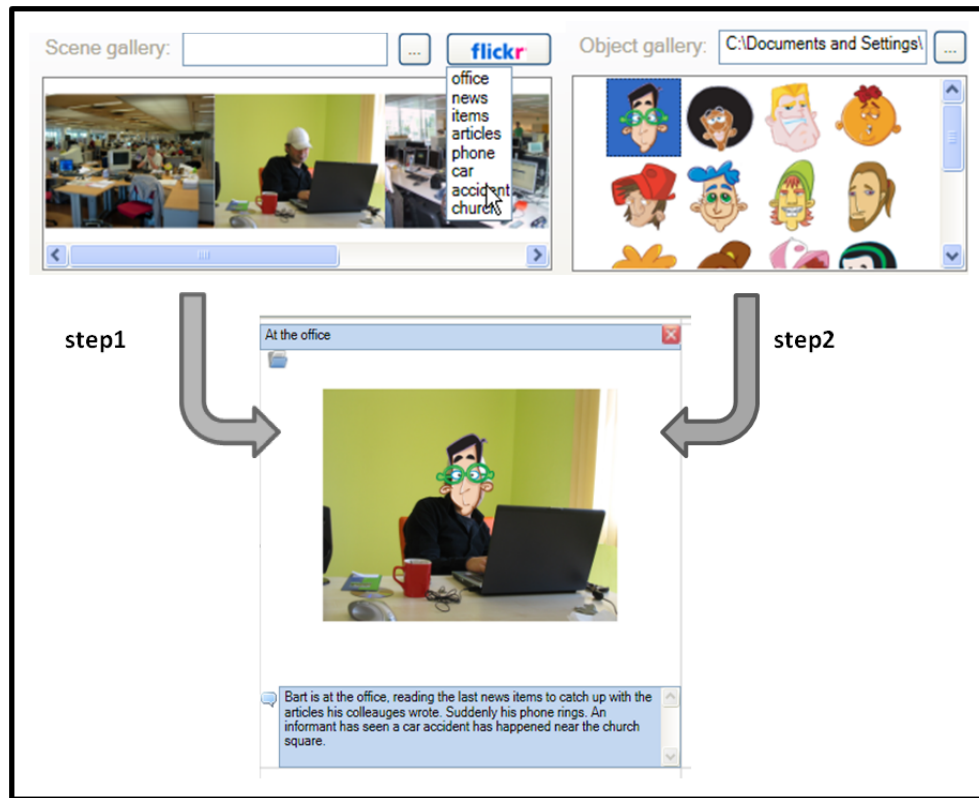


Figure 7.7: GUI elements of the tool that support detailed backgrounds and iconic characters in storyboard sketches.

for people that are hesitant to sketch the scenes. Besides loading sketched scenes, photographs of e.g. the user needs analysis could also be loaded in the tool as scenes.

In order to incorporate techniques from comics, we extended the tool with a Flickr search feature. A Flickr search button automatically suggests the nouns that are part of the narrative scenario, these nouns are extracted from the scenario by using the WordNet.Net library [Wor]. The first thirty search results of Flickr are extracted to the scene gallery by the Flickr.Net API library [Fli]. Following, these photographs can be dragged and dropped as a storyboard scene (Figure 7.7, step 1). Since the participants of our observational study, presented in Chapter 6, were eager to use the images that were provided for storyboarding, we included a feature in the tool to drag and drop

7.4 Incorporating Techniques of Comics in a Storyboarding Tool 89

images from a personal object gallery on top of the photograph or sketch that is loaded as a scene (Figure 7.7, step 2). These images of the object gallery allow practitioners to use pictures of faces or people that have a different style than the photograph. In this way, detailed backgrounds and iconic characters are combined in one scene in order to increase the empathy of team members with the end users. Consequently, people on the original photograph can be anonymized, which can be useful when photographs of a user needs analysis are reused in a storyboard. By loading images of characters with different facial expressions or body language, this object gallery allows its users to apply these techniques of comics as well. This feature also allows to differentiate characters. Images of objects or devices can also be dragged on top of a scene in order to depict the new system in the environment of the users. With our tool we do not want to limit the creativity of team members. The scenes of the storyboard can be created using pencil and paper, or can be composed from graphical representations in the tool itself.

7.4.2 Storyboards for Later Use

Once the storyboard contains its scenes and images, information for later UCD stages such as the creation of models and UI designs can be added. As described in Chapter 5, annotations can be added by means of tagging, in order to provide useful information for the later stages of the UCD project. By annotating personas for instance, team members do not need to take care about including differentiating characters in their storyboard scenes, the main differences are specified in the personas, which are connected to the storyboard.

Since storyboards are in the first place a communication tool, and the first user study (Section 6.2) showed our tool was lacking some suitable views to present a storyboard, the COMuICSer tool allows team members to export a storyboard to a Microsoft PowerPoint slideshow. The PowerPoint slideshow, that is automatically generated, contains an overview of the personas involved in the storyboard, followed by the storyboard scenes. Hyperlinks between the scenes allow team members to navigate quickly between slides containing the scenes and slides containing the personas. This can provoke and enhance the discussion of a storyboard during meetings with the project team. By linking the scenes and the personas, the idea of differentiating characters is maintained in the slideshow.

Transitions in a storyboard may provide important information for the development of a system or UI, in which the developer explicitly needs to know when a transition takes place, or how it is caused. Since interpreting

the implicit transitions does not suffice, the extension of our tool allows team members to label transitions between scenes and add timing information that is not implicitly contained by the storyboard scenes (Figure 7.8). This information can for instance be used to link the storyboard to a preliminary task model. Tasks can be extracted by the depiction of a scene and its description, while transitions or timing information, that is explicitly included using the tool, can be used to structure the task model.

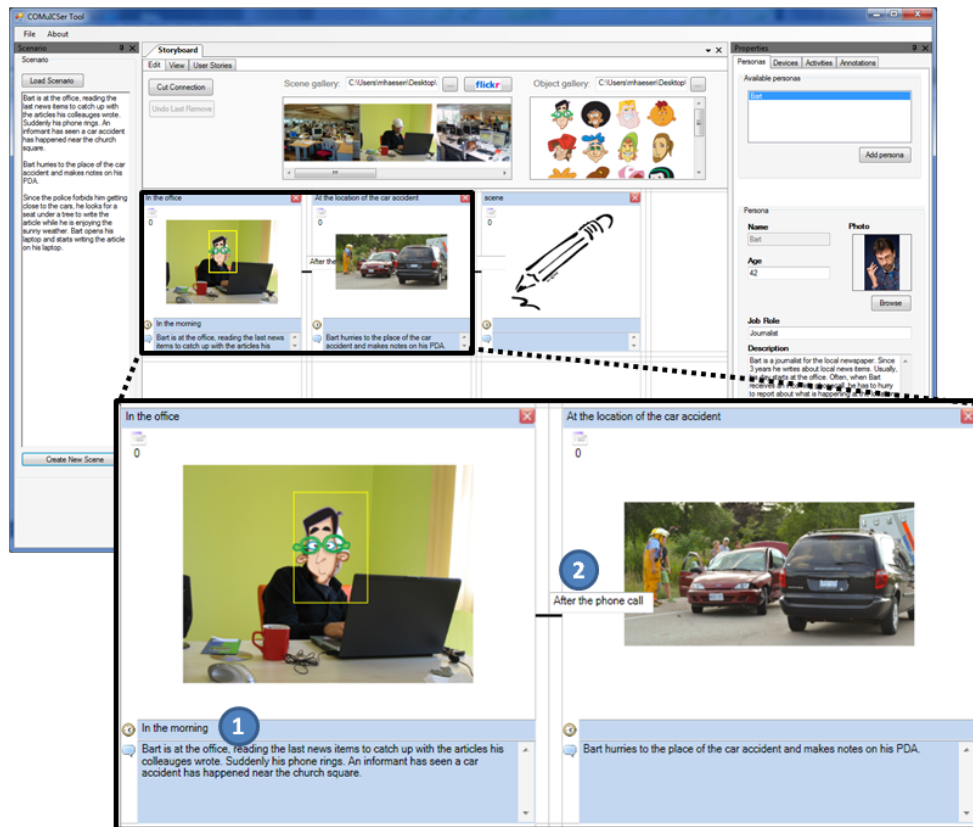


Figure 7.8: Screenshot of the COMuICSer tool. The part of the screenshot that is enlarged shows the features to label (1) time information and (2) transitions.

7.5 Conclusion

We presented techniques used to convey a message in comics that are applicable to COMuICSer storyboards. These techniques used in comics support the goals of COMuICSer. In a user experiment, we observed the use of storyboards and the way techniques of comics are incorporated in storyboards. In combination with the COMuICSer tool, the techniques of comics can contribute to a better integration of storyboards with other common artifacts in a typical UCD process (e.g. structured modeling and UI design).

Since storyboarding has the potential to play an important role in UCD processes, we sought for a better understanding of the universal graphical language that is used to construct scenes in a storyboard. In comics we found an established framework of principles and techniques that are generally accepted by most people and thus are likely to be applicable in storyboard construction. Part of the techniques can only be used to express a message in drawings, but some techniques can be supported by our tool to speed up the creation and enrichment of storyboards. In contrast to comics, storyboards are created quickly to explain a particular scenario. Our tool can be used to create and manage the storyboards and supports features that were inspired by comics techniques. Furthermore, these features of the tool may remind team members to incorporate some techniques of comics in order to enhance the understandability and enrich the storyboard for later stages of UCD. Annotating storyboards and labeling transitions between scenes, support the transition from narrative specifications to more technical models and UI designs.

By investigating comics, we answered RQ 2: *What aspects of a visual language can contribute to storyboards in UCSE?*, the third research question that was introduced in Chapter 5. Incorporating the aforementioned techniques of comics in storyboards empowers team members to empathize with end users. The following chapters describe how storyboards can contribute to notations used in HCI and software engineering. Chapter 8 presents how storyboards can be used as a notation to bridge informal and more formal notations used in HCI, while Chapter 9 concentrates on connecting storyboards and artifacts used in agile software engineering methodologies.

Chapter 8

From Informal to Formal Artifacts with Storyboards

8.1 Introduction

In order to unite User-Centered Design (UCD) and Software Engineering (SE), practices and approaches of both domains need to be considered. The former of this dissertation mainly focused on UCD practices and opportunities that may allow the transition to SE, without providing many details regarding the SE perspective. One domain in SE which is also considered in HCI and should be able to support UCD, is model-based user interface (UI) design.

Because of the growing diversity of computing platforms, model-based design of user interfaces is receiving an increasing amount of attention in the computer science and more specifically the software engineering community. Model-based UI design can offer benefits in terms of reusability, separation of concern, traceability, efficiency and consistency. However, one difficulty concerning the incorporation of model-based UI design in UCD, is the involvement of the multi-disciplinary team. The notations used for model-based UI design are not suited for all members of a multi-disciplinary team, which may cause misconceptions or a loss of information. Since the involvement of a multi-disciplinary team is the most common way user interfaces are created, model-based approaches have to fit into an overall multi-disciplinary design approach to become usable for UCD practitioners.

Some challenges concerning the incorporation of artifacts of other communities into the model-based design approach have already been tackled. User interface sketches can already be integrated into a model-based approach by

using sketch recognition of widgets within a UI mockup [CKV07]. Furthermore, tools like Microsoft Expression Blend support the transition from early prototypes (SketchFlow) to final designs and integration with software development (through Visual Studio). But designers and other people involved in a user interface development process also use other, often informal, artifacts such as personas and scenarios. Consequently, it is necessary to bridge the gap between this informal design knowledge and model-based user interface design.

In this chapter, we present a model-based UI design approach supporting a transition from informal to formal artifacts using storyboards. This approach has the advantage that experts in a multi-disciplinary team can continue to use the artifacts they are accustomed to, yet using storyboards allows for a smooth transition between them and consistency checking. More specifically, we propose how storyboards can be connected to other models through an underlying meta-model and cross-toolkit design support based on a abstract user interface model expressed in UIML [HA08] and UsiXML [LVM⁺04].

8.2 Informal and Formal Artifacts in User-Centered Software Engineering

As already stated in Part I, connecting typical SE processes with UCD processes has been an ongoing challenge tackled by many researchers [BGW98, CSV⁺04, RPM95]. The gap between interface design and engineering interactive systems is mainly a matter of “design” language differences: an interface designer uses different artifacts than a software engineer. In fact, each domain of expertise uses its own vocabulary, which complicates collaboration between people having different backgrounds. Artifacts created and used in user-centered software engineering (UCSE) can be categorized in two groups: *Informal* and *Formal Artifacts*, which are presented in Figure 8.1 and will be described below. Furthermore, we will describe existing work regarding the transformation of artifacts and how this inspired us to use storyboards to bridge the gap between informal and formal artifacts.

8.2.1 Informal Artifacts

Informal artifacts contribute to informal design knowledge, which usually is obtained by knowing the user needs from a user needs analysis combined with studying several artifacts. Informal artifacts include personas, scenario

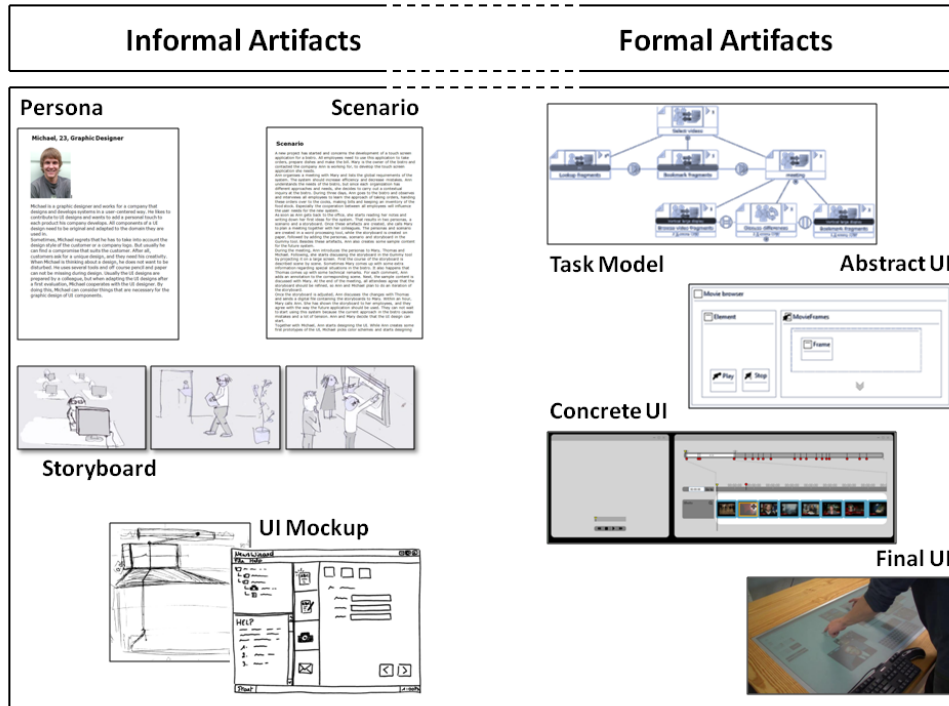


Figure 8.1: Examples of informal and formal artifacts.

descriptions, storyboards, preliminary user interface designs and diagrams. The left part of Figure 8.1 shows an overview of these artifacts.

Although most of the artifacts contributing to the informal design knowledge are created in the first stages of a UCD project, they are supposed to be used during the entire process. Their primary goal is to communicate part of the user needs and design decisions to all team members right after the user needs analysis. Furthermore, these artifacts should be used for guidance in several other stages, such as UI design and development. The notations used for these artifacts concern two types of languages often used to aid a universal comprehensibility: narrative and graphical notations.

The narrative notation may cause difficulties in the communication of user needs and requirements. When e.g. *personas* or *scenarios* are poorly communicated or accepted by the leadership team, or when other team members do not know how to use them, a lot of information contained by these artifacts can get lost during a UCD process. Furthermore, it may be problematic for

people with a technical background to translate narrative stories into technical specifications [DSL06]. Johansson et al. conducted a study which showed that people with a technical background more easily relate to narratives when these narratives are supplemented by sketches and graphs [JA07].

In UCD, a graphical notation is used for several artifacts such as storyboards, UI designs or diagrams. The advantage of *storyboards* is that they provide a depiction of how a future system can be used that is less ambiguous than a scenario [THA06]. *UI designs* can be used to share ideas about what the UI should look like and to discuss several UI considerations. Furthermore, these UI designs can be used to verify whether the first UI decisions meet the user needs. In practice, this can be done during stakeholder meetings, but also during informal evaluations together with end users. When several screens in UI designs are connected, they do not only present the look and feel of a UI, but also part of its behavior in a *diagram*.

Although many of the artifacts discussed in this section can be classified as informal artifacts, part of their exact meaning is open for interpretation, as they probably also contain unambiguous information. A notable example of artifacts containing this unambiguous information, is the formalization of sketched interface designs using the formal specification language Z presented by Bowen and Reeves [BR06]. Since our major concern is the communication in multi-disciplinary teams involved in UCSE, our approach does not strive for this level of formality but rather aims at structuring and extracting the information that can be linked to the models typically used in a model-based engineering approach. The next section presents these models relevant for the design of user interfaces.

8.2.2 Formal Artifacts

Several models are commonly used in the model-based design and development of user interfaces. Thevenin and Coutaz introduced the word “*plasticity*” for user interfaces that adapt to their context of use, while maintaining usability [TC99]. The Cameleon reference framework for plastic user interfaces [CCT⁺02] lists the different kinds of models and their role in the development process. UsiXML [LVM⁺04] is a single language that integrates most of these models such as task models, abstract user interfaces, concrete user interfaces and final user interfaces, which are shown in the right part of Figure 8.1. The remainder of this chapter will describe these models that are related to UsiXML.

Task models allow to express a hierarchical decomposition of a goal, into

activities or tasks, that can be translated into actions at the lowest level. In the activity theory, an hierarchical relationship exists between activities, tasks and actions [Nor05], while we consider activities and tasks at a similar level. The COMM task model notation [JLN10] illustrates this by allowing specification of modalities or interaction devices by *modal tasks*, which are leafs in the task tree. They are mostly used as a first step in designing an application to identify the tasks and later actions that have to be performed to reach a certain goal. Furthermore, task models support the UCD that is not limited to the consideration of user needs, but also includes activities, as suggested by Norman [Nor05].

Abstract user interfaces are high-level descriptions of user interfaces that are independent of a device or modality. They consist of *Abstract Interaction Objects* (AIOs) arranged in presentation units. An AIO supports the execution of a leaf task: it allows the user to give input to the system (e.g. enter a search term), to start the execution of some function (e.g. start the search), or allows the system to present output to the user (e.g. “searching” or the search result). Abstract user interfaces usually also define the transitions between presentation units, although the latter fact is usually not emphasized in graphical representations of the abstract user interface. Most abstract user interface languages have a formal basis in the form of a meta-model. For example, the Canonical Abstract Prototypes (CAP) notation [Con03] was first defined to allow more abstract paper prototyping to encourage creativity and later was integrated into a modeling environment with a proper meta-model [NNC06]. Van den Bergh et al. proposed the CAP3 [VdBLC11] notation that extends the CAP notation with structural and behavioral information. Furthermore, CAP3 supports relationships with other formal models.

Concrete user interfaces realize abstract user interfaces for a specific context of use, such as a desktop PC used by a journalist. It already represents the final look-and-feel (e.g. following the Windows User Experience Interaction Guidelines) but is independent of the user interface toolkit (e.g. Qt, MFC or GTK). A concrete user interface is especially useful to port user interfaces among different toolkits. Porting user interfaces between different platforms or modalities may require using a higher level of abstraction (such as the abstract user interface) to enable transition between radically different interaction objects.

Final user interfaces are instances of the concrete user interface for a specific toolkit (e.g. Qt on a HP PC running Windows 7). They can be interpreted or compiled to run on the target device. Final user interfaces are not considered in UsiXML or any other user interface description language supporting

multiple abstractions.

Besides these abstractions there are some other models that are important to user interface design, such as the domain model, describing concepts from the domain that are relevant to the user interface, and context models that capture the context of use in terms of user, platform and environment. Although UsiXML covers a wide range of models that describe various aspects of an interactive system, it has little support for informal artifacts other than low-fidelity prototypes [CKV07]. Artifacts such as storyboards and personas are not covered by UsiXML. We think a model-based engineering approach for interactive systems needs to include other informal artifacts in the overall development process in order to support a complete design and development process and to avoid a loss of informal design knowledge. Up until now, there was no suitable solution for this. Before presenting our work that supports the transition of storyboards to formal artifacts, we describe existing tools that support transformations between informal and formal artifacts in the next section.

8.2.3 Transforming Informal Design Knowledge into Formal Artefacts

The artifacts mentioned in sections 8.2.2 and 8.2.1, can be created and used in the context of the MuiCSer process framework (Part I). The process framework can be used to specify at what stage artifacts, originating from UCD and software engineering, are created by a multi-disciplinary team. Furthermore, this framework shows what transitions are needed to move from unstructured and informal artifacts toward structured and formal artifacts that define the interactive system. Figure 8.2 shows the process framework and includes extracts of the aforementioned artifacts and models.

In Chapter 2 we presented our literature survey concerning artifact transformation tools. As a reminder, we repeat the concluding timeline representing artifact transformation tools in Figure 8.3. In this section we will discuss how the transformation of informal design knowledge into formal artifacts is supported by some of the the artifact transformation tools shown in Figure 8.3.

Some work has been proposed to support transitioning from informal artifacts to an initial task model. The CTTE tool [MPS02] has limited support to identify tasks in a scenario description, which eases the creation of a task model based on a scenario. Although this is a useful feature, it identifies only part of the information contained in a scenario. Since much information con-

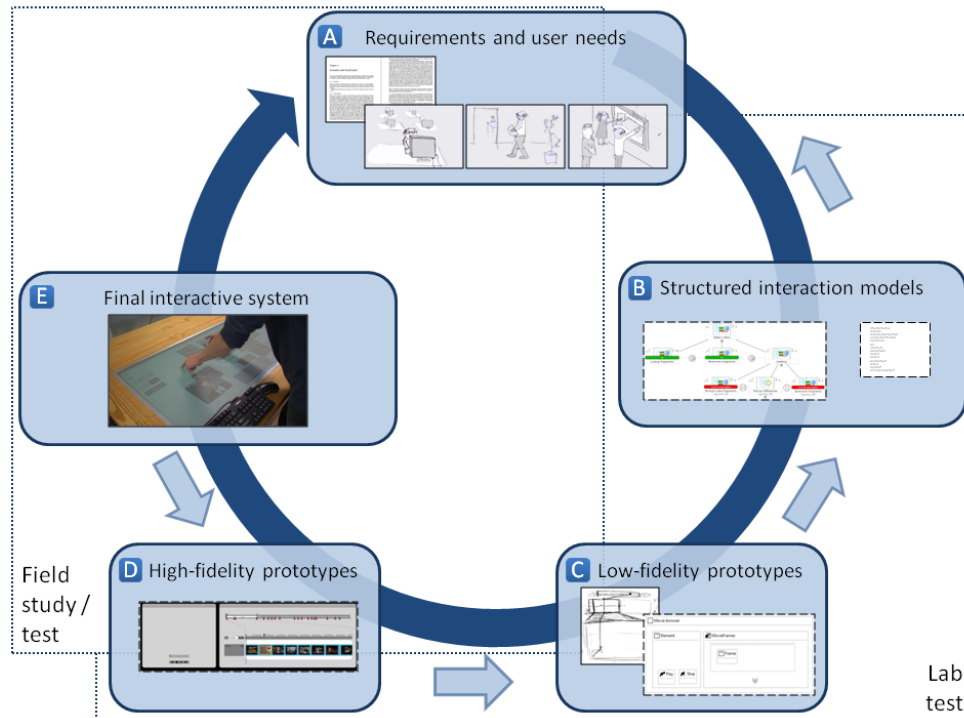


Figure 8.2: The MuiCSer process that was used for the design and development of *News Video Explorer*. Extracts of the most important artifacts are presented for each stage of MuiCSer. In this diagram, informal artifacts have a solid border, while formal artifacts are distinguished by a dashed border.

tained in the scenario is not important to represent directly in a task model, it is likely that some information of the scenario such as contextual information easily gets lost when the task model on its turn is transitioned into another model. While this work also addresses the transition from step *A* (Requirements and user needs) to *B* (Structured interaction models) of the MuiCSer process framework (Figure 8.2), it starts from a purely narrative scenario to extract tasks for the task model.

Dow et al [DSLL06] present a preliminary next generation storyboarding tool for ubiquitous computing. This concept of a tool provides a communication mechanism for different roles in a multi-disciplinary team and supports the connection with other tools. These ideas are implemented in Activity-Designer [LL08a], a tool that supports an activity-based prototyping process.

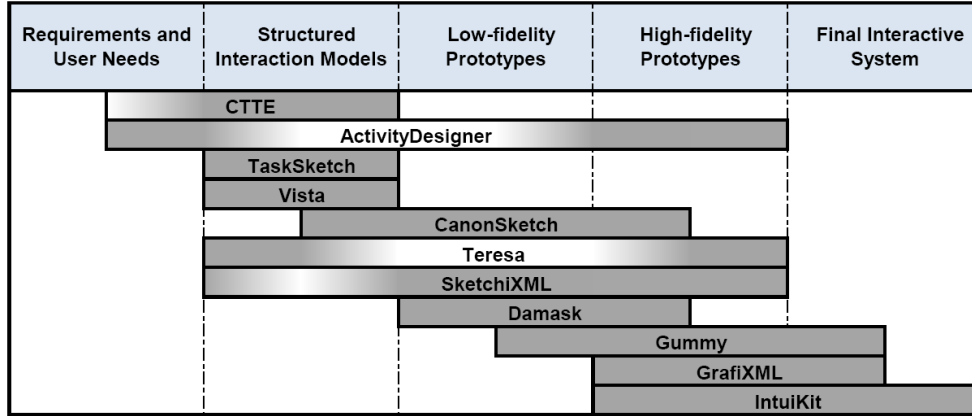


Figure 8.3: A timeline presenting the stages of MuiCSer and how artifact transformation tools can be mapped on it (repetition). The white area in a bar that represents a tool indicates a lacking or limited support of artifacts at that stage of MuiCSer.

One of the first steps supported by the tool, is the creation of scenes, based on everyday observations. These scenes contain textual labels describing actions and situations as well as an accompanying visual representation of the scene. These first activity models can contribute to the first prototypes, that can include interaction sequences. Furthermore, the tool supports the evaluation of the prototypes created. This work overlaps stages *A* (Requirements and user needs), *B* (Structured interaction models) and *D* (High-fidelity prototyping) in the MuiCSer process framework (Figure 8.2).

SketchiXML [CKV07] allows creating concrete UI models through sketches by using sketch recognition. CanonSketch [CN07b] offers different synchronized views on a user interface presentation model (UML class diagrams with Wisdom stereotypes [Nun03], Canonical Abstract Prototypes [Con03] and final user interface). This approach allows immediate switching between the different views on the user interface presentation model. Both SketchiXML and CanonSketch mainly address step *C* (Low-fidelity prototypes) and the transition to *D* (High-fidelity prototypes) of the MuiCSer process framework in Figure 8.2, but in different ways. While SketchiXML mainly focuses on informal specification of models, CanonSketch focuses on fluent transitioning between working styles; from detailed design to more high-level and abstract (re-)structuring of the user interface.

For the design of UIs that are intended to be used on multiple devices,

multi-device design approaches are suggested. In these approaches, UIs are automatically generated from a higher level model and thus support transformations from B (Structured interaction models) to D (High-fidelity prototypes). It was shown that such approaches can be effective in some specific application domains (e.g. Supple [GWW10] and PUC [NHA07]). The Damask tool [LL08b], a multi-device design extension for Denim [NLHL03], addresses C (low-fidelity prototypes) instead of high-fidelity prototypes.

8.2.4 Introducing a Common Language that Supports the Transformation

Supporting the creation, use and transformation of both informal and formal artifacts in one process model for UCSE is one step towards bridging the gap between informal design knowledge and formal models. Nevertheless, one of the biggest challenges in integrating informal design knowledge and formal models is introducing a common language that enables multi-disciplinary teams to collaborate in creating advanced user interfaces. In the following sections, we will describe how storyboards can be considered as a common language in order to support transformation to other artifacts.

In contrast to the tools mentioned in section 8.2.3, we will maintain contextual information and support a rather gradual approach to extract tasks and temporal relations by first structuring a scenario in an annotated storyboard and then guiding the transition to a task and a context model using model transformations. Furthermore, the use of COMuICSer supports creativity, which is addressed as one of the needs for tools that support multi-disciplinary UCSE teams [CN07a, JH98]. Our visual notation used for storyboard scenes supports the creativity of team members and the annotations directly connect visual depictions in a scene with labels. Since a major drawback of the prototyping tools mentioned in the previous section is the lack of support for UI designers, who are not accustomed to the models and algorithms that are used to generate final UIs [MHP00], we would like to use a storyboard for the design of prototypes that hide models from designers and allows them to work on the concrete representation of a UI while taking into account contextual information. Our COMuICSer storyboards (Chapters 5 and 7) can be connected to formal artifacts in order to maintain a considerable amount of informal design knowledge during an entire UCSE process. In the next section we describe how these connections can be realized.

8.3 Storyboarding to Bridge the Gap Between Informal Design Knowledge and Formal Models

We found inspiration in the intersection of storyboarding, comics and model-based engineering to connect informal design knowledge and formal models. In this section, we describe how our storyboarding approach and tools proposed in Chapters 5, 6 and 7 are taken into account for the creation of more formal artifacts that can be used in MuiCSer processes. The proposed tools that can be used for this are the COMuICSer tool for creating storyboards, mapping and transformation support for UsiXML and the Jelly high-fidelity prototyping tool.



(a) Doing an individual search.



(b) Walking to the meeting room.



(c) Discussing and extending search results.



(d) Assembling selected videos.

Figure 8.4: Storyboard created for the development of the News Video Explorer, an application to visually explore video archives.

The examples, presented in this section, are related to the design and development of News Video Explorer, an interactive prototype that was developed in the context of the AMASS++ project [MBTM08] and allows professional video searchers to visually explore news videos. The storyboard used for this

project is shown in Figure 8.4. The storyboard exemplifies how one integrated future application can be used for searching archives, browsing an archive video and adding video fragments to a favorites folder. In this storyboard, a video searcher has to search and browse video fragments of a sports person in order to assemble a news item concerning the career of this person. First, the video searcher retrieves interesting videos in his office, behind his own desk (Figure 8.4(a)). He saves the files on his mobile device. Following, he takes the mobile device and hurries to a meeting room where he will discuss the video fragments with some colleagues (Figure 8.4(b)). In this meeting, a large multi-touch display is used to browse through the available video fragments and to talk about the most suitable fragments (Figure 8.4(c)). Afterwards, the video searcher goes back to his office and starts assembling the resulting video (Figure 8.4(d)).

8.3.1 A Storyboard Meta-model

Once a storyboard is created and digitized using our COMuICSer tool, the storyboard, its structure and annotations are available for the design and development of a user interface. The informal design knowledge contained by the storyboard can be passed to more formal models. We structured the informal design knowledge by means of a storyboard meta-model, shown in Figure 8.5. This meta-model is MOF-compliant¹.

There is one central element in the meta-model: the *Scene*. A scene is a graphical representation of a part of the scenario. A set of scenes are related to each other using *TemporalRelationship* elements in a *Storyboard*. The *TemporalRelationship* element can be based on Allen's interval algebra [All83]. The *before* relationship indicates one scene happened before another, and there is undefined time progress in between scenes. The *meets* relationship indicates one scene is immediately followed by another scene, and the time progress between two scenes is virtually none. Although the most common relationships used in storyboarding are *before* and *meets*, we think parallel activities should be supported since they are common in collaborative and multi-user activities. Defining these temporal relationships between scenes allows us to exploit them later on, e.g. by mapping them on the temporal relationships that are used in the task model.

The drawings or photographs used during the construction of a storyboard often contain a lot of contextual information. Dow et al. [DSL06] show storyboarding, especially contextual storytelling, is useful for context-aware ap-

¹MOF is an industry standard established by the Object Management Group

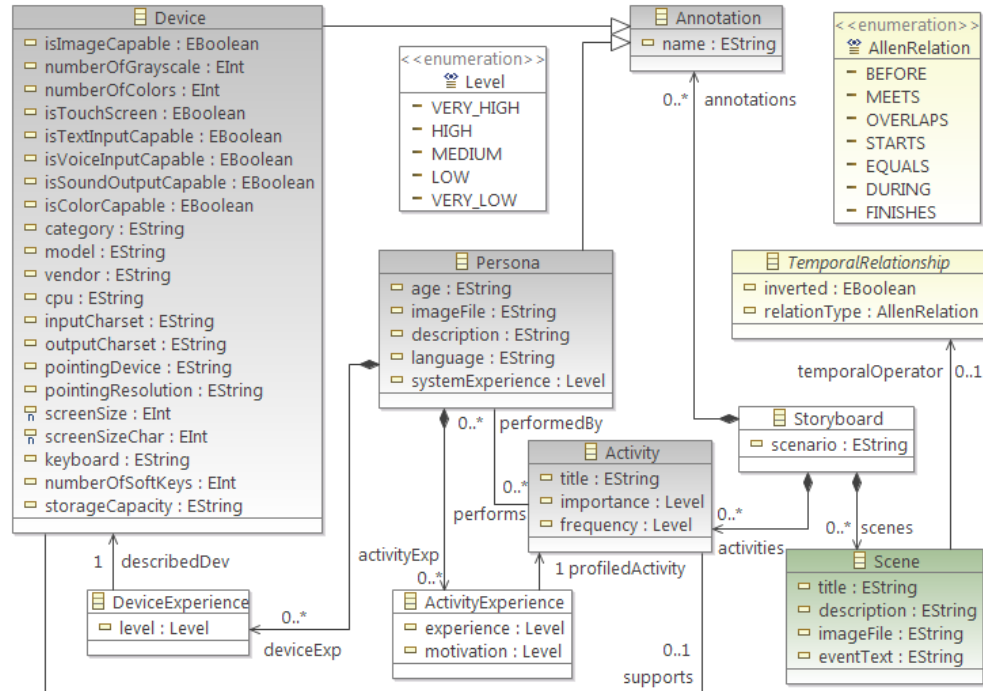


Figure 8.5: Our COMuICSer storyboard meta-model. It contains the graphical depiction with the objects of interest (context), personas, devices and activities. Scenes are related using the Allen interval algebra operators.

plication design (in their case ubiquitous computing applications) but lacks a good way of formalizing the context data. By providing the feature of tagging and annotating parts of scenes, we support a rudimentary way of translating the context inferred from the graphical depiction of a scene into a readable format. In COMuICSer, a scene is annotated with different types of information: *Persona* specifies archetypical users, *Device* presents computing devices and systems and *Activity* represents what happens in a scene. Vanderhulst et al. showed that a graphical depiction can have high value to obtain a usable model of the context of use [VLC09].

8.3.2 Mapping Storyboards to Models

As one instance of a transition from informal to formal artifacts, we introduce a mapping from COMuICSer storyboards to UsiXml [LVM⁺04]. UsiXML was

selected as the target model because it contains a consistent set of the models used in UCSE including task model, abstract user interface model and context model. Furthermore, UsiXML is supported by many tools [Usi].

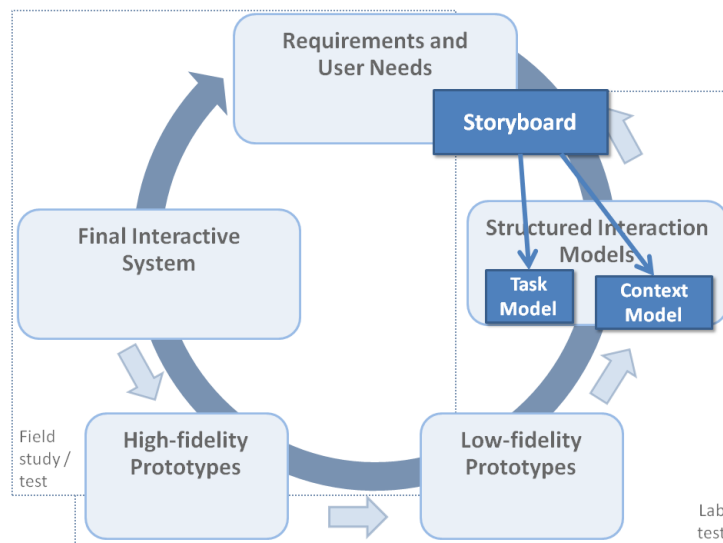


Figure 8.6: The model transformations based on a COMuICSer storyboard.

The two UsiXml models that can be partially generated from a COMuICSer storyboard are: the *task model*, describing task sequences required to reach the user's goal, and the *context model*, specifying the application's context of use. Sylvain Degrandart and Serge Demeyer developed two formal transformations of information contained by a storyboard to formal models. Figure 8.6 shows the relationship between the storyboard and these models as well as their mapping to the MuiCSer process framework.

Tasks of the UsiXml task model are in direct relation with activities that are depicted in a scene of a storyboard. For each activity, the *activity2task* transformation (formalized in Figure 8.7) creates a task in the task model with a *name* that corresponds to the activity *title*. Some domain-independent properties of an activity are also transformed into task properties through a one-to-one mapping. For example, the activity *importance* and *frequency* are directly mapped to their task equivalent, after a format adaptation. However, not all mappings are that straightforward: for instance the type of task can be either a system task, a user task or an interactive task. The task *type* can be inferred by the number of devices and personas related to the activity. If

```

rule activity2task{
  from a : Mstoryboard3!Activity
  to task : MUsiXmlTask!Task (name <- a.title,
    importance <- a.getUsiXmlImportance(),
    multidevice <- (a.performedUsing->size())>1),
    frequency <- a.getUsiXmlFrequency(),
    cooperative <- (a.performedBy->size())>1),
    type <- if (a.performedBy->size()==0) then 'system'
      else if (a.performedUsing->size()==0) then 'user'
      else 'interactive' endif endif)
}

```

Figure 8.7: Activity to task transformation.

an activity is performed by a persona using no device, the corresponding task is a user task, if no persona is performing the activity, the task is a system task, otherwise it is an interactive task. A task is set as *cooperative* if multiple personas are involved in the performance of corresponding activities. The scene shown in Figure 8.4(c), includes a cooperative task which involves two personas carrying out the same activity: browsing the videos in News Video Explorer. The same reasoning on the number of devices is used to infer if a task is a *multi-device* task or not.

Two other generative transformations, *persona2userStereotype* and *device2-platform*, focus on the partial generation of a context model from information specified in a COMuICSer storyboard. Firstly, the *persona2userStereotype* transformation creates a new *userStereoType* element for each *Persona* element, with direct mapping of properties such as *DeviceExperience* and *ActivityExperience*. Secondly, each *Device* is transformed to *newhardwarePlatform* and *softwarePlatform* elements. All the device properties related to hardware are mapped to their hardwarePlatform equivalent: *category*, *cpu*, *inputCharset*, *isColorCapable*, *screenSize*, *storageCapacity*, ... The rest of the device properties (*osName*, *osVersion*, *isJavaEnabled*, *jvmVersion*, ...) are mapped to softwarePlatform properties.

The context model does not retrieve all the information from the COMuICSer storyboard it has been mapped on. The storyboard model provides by definition a lot more context information: mainly all the entities in the environment. We generate a new context model for every scene in a storyboard, even though it is likely that part of this context can be shared across

scenes. Nevertheless, UsiXml does not allow to do this because variations are not possible within one context model.

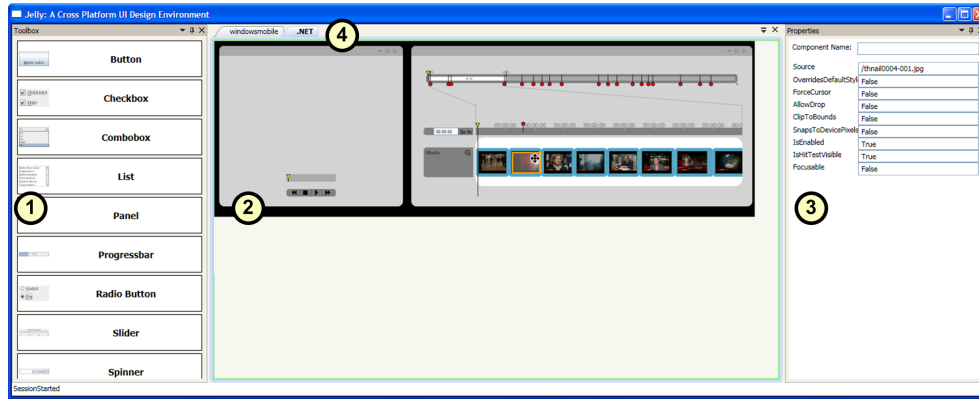
8.3.3 From Storyboard to High-fidelity Prototype

As prototyping a user interface is another crucial stage in UCSE, we consider connecting COMuICSer storyboards with high-fidelity prototypes (Figure 8.2-D). The multi-device Jelly design tool, developed by Meskens et al. [MLC10] (Figure 8.8 (a)) allows designers to design a concrete user interface (concrete UI), while it automatically maintains an abstract UI presentation model for this concrete design. This allows designers to use a notation they are accustomed to, in this case a graphical user interface design, and to use this artifact in a model-based engineering process without forcing them to change their working practices considerably. The abstract UI presentation model is a structured well-defined model based on UIML [HA08].

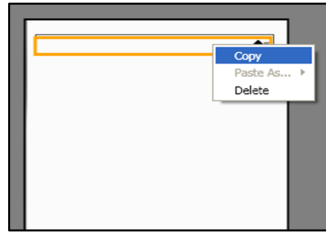
Figure 8.9 shows the relationship between the COMuICSer storyboard and the UI design. This UI design, which is created in the Jelly tool, is also connected to *abstract user interface* and *a presentation model*. To increase legibility, Figure 8.9 limits its representation to the artifacts that are supported by the Jelly tool. However, this does not infer that other stages and artifacts in the MuiCSer process framework are excluded. We highly recommend the support for the evolution from a UI design to a high-fidelity prototype by structured interaction models and low-fidelity prototypes that are evaluated in one or more iterations.

We extended the Jelly tool with the incorporation of our COMuICSer storyboards. This allows designers to keep track of the contextual information that is provided by the storyboard. Figure 8.10 shows a screenshot of the Jelly tool in which a COMuICSer storyboard was imported that was created in our COMuICSer storyboarding tool. For every device that is *tagged* in the storyboard, Jelly can provide a separate design workspace. This way, Jelly automatically takes into account the different contexts of devices. Designers are reminded of other contextual information by the visual representation of the storyboard in the tool.

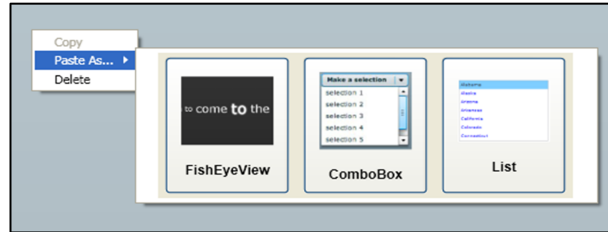
In each of these workspaces, designers create User Interfaces (UIs) by placing widgets from the *toolbox* (Figure 8.8 (a)-1) on the *canvas* (Figure 8.8 (a)-2) and dragging them around until the resulting layout is visually appealing. These widgets' properties can be changed through the *properties panel* (Figure 8.8 (a)-3). This usage model is very similar to traditional GUI builders, allowing designers to reuse their knowledge of single-platform UI design tools



(a) The Jelly multi-device design environment



(b) copy widgets in one device-specific UI



(c) Paste widgets to other device-specific UIs

Figure 8.8: The Jelly multi-device design environment, showing a video browser design for a desktop computer (a). Jelly allows designers to copy widgets from one device (b) and to paste these widgets as a similar component on a different device (c).

in a multi-device design environment. Under the hood, Jelly builds an underlying presentation model which can be connected with other artifacts included in the MuiCser process.

The mappings between Jelly’s underlying presentation model and formal artifacts such as Abstract Interaction Objects (AIOs, as introduced by the reference framework in section 8.2.2) also help designers during the creation of high-fidelity prototypes. It introduces the flexibility to *copy* a part of a UI on one device (Figure 8.8 (b)) and to *paste* it *as* a similar part on another device (Figure 8.8 (c)). A component is considered as similar if it has the same *AIO type* and *content data type* as the given component [VVC⁺07]. Jelly currently supports four types of AIOs, differentiated according to the functionality they offer to the user: (1) *input* components allow users to enter or manipulate content; (2) *output* components present content to the user; (3)

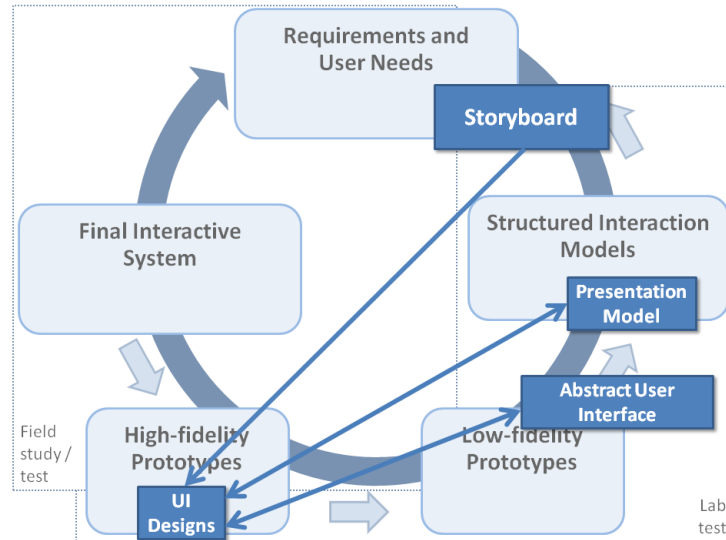


Figure 8.9: The relationships between a COMuICSer storyboard and the UI design(s) created in the Jelly tool.

action components allow users to trigger an action and; (4) *group* components group other components into a hierarchical structure.

As content datatypes, Jelly supports the five primitive types of XML Schema (e.g. decimal, string and void), a number of data types that are often used in user interfaces (e.g. Image and Color) and container data types that group content items of a certain type together (e.g. a list of strings and a tree of images).

8.4 Conclusion

To effectively integrate informal design knowledge in formal models, an increased involvement of a multi-disciplinary team during the entire process is recommended in order to evaluate the completeness of artifacts regarding the requirements and user needs created in the early staged of UCD. This means that techniques to translate informal design knowledge into more formal models are necessary to transition artifacts in a project. Nevertheless, a common language that is understandable by all team members in order to keep track of decisions made during the process is also needed to support the communication in the multi-disciplinary team. To accomplish this, we proposed a

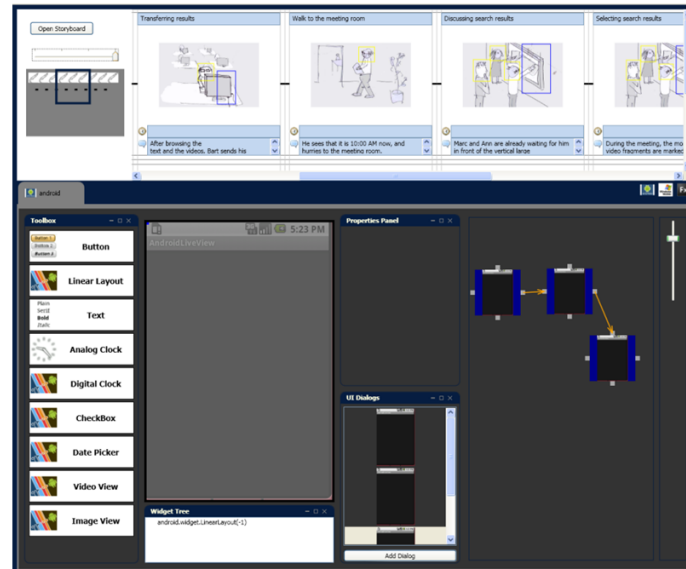


Figure 8.10: The Jelly multi-device design environment, showing an imported COMuICSer storyboard that can be used during the UI design.

meta-model for COMuICSer storyboards that was used to formalize the storyboarding notation and to provide mappings and transformations to formal models. The meta-model was used as a tool to support transformations of information contained by COMuICSer storyboards to a task and context model.

The incorporation of storyboards in the Jelly tool leverages the knowledge captured by the AIOs and their relation to concrete and final interface objects that are contained by the tool, which eases the creation of multi-platform user interfaces. Furthermore, the possibility to keep an eye on a storyboard while using the Jelly tool, allows designers to keep in mind the user requirements and context of use during the creation and evaluation of UI designs.

In this chapter we discussed how storyboards can bridge the gap between informal design knowledge and formal artifacts on the level of artifacts as well as tools. This answers one of the research questions introduced in Chapter 5, RQ 3: *How can storyboards be used for the transformation from informal to formal artifacts?* The artifacts and tools considered in this chapter are mainly used within the field of HCI. Since the aim of our work is to bridge UCD and SE, Chapter 9 will propose an approach to incorporate storyboards in agile software engineering practices.

Chapter 9

Connecting Storyboards and Agile Practices

9.1 Introduction

Nowadays, software applications are not limited to professional use anymore. Software is widely used by people having various backgrounds and is available in several environments and situations of everyday life. Due to the shift from desktop computers to applications that are spread across several devices (e.g. mobile devices) for a diverse group of end users, requirements elicitation has become even more complex than it was before. The different possible contexts of use of an application should be included in the requirements in order to take into account this contextual information during design and development activities.

The contexts of use, and non-functional requirements, which mainly express important aspects regarding the quality of a software product should be added to the functional requirements considered in software engineering (SE). Since SE results in the development of functional features, the main focus is on functional requirements and often non-functional requirements may be overlooked. This is partly due to the lack of notations in SE that include functional as well as non-functional requirements in a comprehensible and expressive way. Classical requirements notations such as natural language documents and use cases do not thoroughly include non-functional requirements or do not fully take into account the context of use. Moreover, non-functional requirements are difficult to express in a measurable way [NE00]. Cheng and Atlee [CA07] address the need for new requirements specification notations that consider the environment as a whole, including software, hardware and people. As stated

by Chung et al. [CPL09], non-functional requirements can help software engineers to make and justify design decisions and therefore, these non-functional requirements should be closely related to functional requirements. Furthermore, research and tools are needed to extend requirements elicitation with creativity techniques [MJK⁺10].

User-Centered Design (UCD) techniques analyze end user needs, including non-functional requirements and the context of use, and deal with them from the requirements specification to the final deployment of the software. UCD projects typically include several iterations of user interface design [Int99]. However, due to the involvement of a multi-disciplinary team, often the results of the different iterations are approached from a non-technical point of view. Although the UCD approach is very helpful to design applications that consider the context of use and non-functional requirements, combining UCD and SE practices is often an obstacle in the development of software applications that meet both user needs and functional requirements. Amongst others, the complementary viewpoints within the multi-disciplinary team can influence the resulting application in a positive way, but the different backgrounds can impede the communication within the team. Since UCD rather focuses on requirements from a user perspective, and SE mainly concentrates on the technical aspects of requirements, these requirements are hard to pass between both teams without any loss of information.

We introduce connected storyboards to combine visual storytelling (Chapters 5 and 7) and user stories [Coh04] used in agile methods. The starting point to introduce COMuICSer storyboards in SE is the domain agile SE because some similarities between UCD and agile approaches may ease this connection. Nevertheless, as we concluded from interviews with agile SE practitioners, agile approaches are still lacking notations that consider non-functional requirements as well as the different contexts of use of a system. In a survey of agile practices, participants acknowledged that non-functional requirements are often ignored [CR08]. By connecting existing techniques from agile SE and UCD, we aim to improve communication within a multi-disciplinary team without introducing extra effort in agile software development processes.

In this chapter, we will first describe agile SE methods and their relationship with UCD based on a focused survey of current literature complemented with interviews with practitioners. Following, we will propose the connection of storyboards with user stories and we will describe how this can be supported by our COMuICSer tool. Since this tool is intended to be used by UCD teams as well as agile SE teams, we conducted an informal first user study within an agile SE team. Findings of this first user study are discussed in order to

conclude how our connected storyboards can contribute to agile SE and UCD.

9.2 Agile UCD Processes and Requirements Notations

In contrast to traditional SE, agile SE methods pay attention to requirements from a user perspective, and customers are actively involved in a project. Agile methods apply strict and short iterations that result in parts of code that are delivered after every iteration [BBvB⁺01, CR08]. The aim of agile methods is to respond to changing requirements during development. Consequently, documentation is limited in agile SE methods, while frequent and efficient face-to-face communication within the team is encouraged [SR04].

Table 9.1: A comparison of User-Centered Design and Agile Software Engineering

	User-Centred Design	Agile Software Engineering
Team	multi-disciplinary, including user interface designer and user	developers, testers, customer, user
Communication	pass artifacts, team meetings	frequent and efficient face-to-face communication, evolutionary prototypes
Type of requirements	non-functional and functional user needs	brief functional requirements, from customer perspective driven by business values
Artifacts	many artifacts (natural language documents, prototypes)	software code, limited documentation (backlog including user stories)
Evaluation	usability testing, verification, validation, in several stages	acceptance testing of functional requirements, after each iteration
Iterations	iterative design activities	short development iterations

Table 9.1 shows a comparison of UCD and agile SE, based on existing research [MM05, MGR07]. This table provides information, which is interesting

to consider when investigating the combination of both methods. Particular agile approaches allow an integration of UCD, as we will present in the following section. However, the constrained documentation in agile SE methods and the use of many artifacts and field studies in UCD may impede the communication within a project that combines agile and UCD methods. The following sections will describe how UCD and agile SE approaches can be combined and what difficulties practitioners are facing when working in this way. Part of the literature study of this chapter is based on Cohn's book about applied user stories [Coh04], which is also used as a handbook by agile SE practitioners.

9.2.1 Agile UCD Processes

Similar to UCD processes, agile SE methods are iterative. Nevertheless, UCD typically results in many artifacts and iterations in UCD are not planned as strictly as in agile methods, which implies that some rules or adjustments are needed in order to intertwine UCD and agile SE. Constantine and Lockwood [CL02] introduced *Usage-centred Design* as an approach to combine UCD and agile methods. Usage-centred Design focuses on the creation of essential use cases and formal models, which requires team members to have a rather technical background.

Holtzblatt et al. [HWW04] introduced *Rapic Contextual Design*, as a variant of Contextual Design [BH98] that integrates techniques of contextual design with agile SE. As shown in Figure 2.5 of Chapter 2 that mapped Rapid Contextual Design on the MuiCSer process framework, Rapid Contextual Design mainly concentrates on the early stages of a user-centered software engineering process.

In contrast to the aforementioned approaches, other studies suggest to combine UCD and agile SE while preserving the use of notations from both domains, which allows the team members to use the notations they are familiar with. The *CRUISER lifecycle* was introduced by Memmel et al. [MGR07], which is intended to combine agile SE and human-centered design. CRUISER concerns an iterative lifecycle that integrates principles of both domains.

Paelke et al. propose a methodology for integrating UCD in agile methods for mixed reality design [PN08]. This methodology suggests to conduct user studies before the first agile iteration. Furthermore, each iteration includes design activities that typically take place in the later stages of UCD. The UCD documentation in this approach should be as lightweight as possible and usually includes *sketches*, *paper-prototypes* and *mockups*. Sy describes a comparable agile UCD approach, that adds more rules and structure than the

aforementioned approach [Sy07]. In this approach, designs are communicated between both teams by using *design cards* and *issue cards*. On the one hand, the approaches proposed by Memmel et al., Paelke et al. and Sy [MGR07, PN08, Sy07] suggest to combine existing techniques from UCD and agile SE in order to support multi-disciplinary teams. On the other hand, they do not suggest the use of a notation that facilitates a straightforward connection of functional as well as non-functional requirements. We believe that a central notation used by both the UCD and agile SE team can be helpful to record design decisions and to keep an eye on contextual information, functional requirements and non-functional requirements and their relationships.

9.2.2 Requirements Notations

When considering notations that both contain functional and non-functional requirements, we can take a closer look at *scenarios* and *personas*, which are used in UCD. Scenarios [Car00] are often used to describe the use of a future system in a natural language and may contain functional as well as non-functional requirements. By including personas [PA06] in scenarios, the user characteristics are specified in a way that evokes empathy in a project team and further non-functional requirements can be taken into account for the development of a system. Although scenarios and personas are very accessible notations for multi-disciplinary teams, their structure and narrative descriptions do not allow a direct connection or transition to more structured models, used for UI or software development purposes.

Scenarios, personas and storyboards can be considered as complementary artifacts in UCD [SM10]. Also in agile SE, storyboards can be used as a requirements notation. Design studio approaches, which are not exceptional for agile SE, are very suitable to create lightweight documentation such as storyboards in order to explore potential solutions from different perspectives before the functional requirements are elicited, or development is started [UW08, MJ10]. However, storyboards are not a widespread notation in agile SE.

User stories are frequently used in agile SE (e.g. Extreme Programming [BA04]) and support requirements from a customer perspective [Coh04]. User stories contain a written description of a story, and are used for planning and reminder purposes. Furthermore they also represent the discussion about the story and reminders for acceptance testing. Not all this information is available on a user story card, but because of the close interaction within an agile SE team, a reminder about a user story should be sufficient to remind

its details for further planning and development.

Cohn explains that some practitioners explicitly structure their user stories in such a way that they clearly express the function that is required, the role of the user that benefits from this function and the business value that is related to this function [Coh04]. Nevertheless, user stories provide limited support for non-functional requirements and contextual information [MJ10], which impedes communication with the UCD team. Moreover, sometimes it is hard to understand the relationship between user stories, and user stories do not scale well on large project teams [Coh04].

Agile SE limits documentation, processes and tools, while the focus is on interaction within the team and delivering parts of code after short iterations. Furthermore, the customer plays a central role [BBvB⁺01]. However, when considering an integration of UCD and agile SE methods, one should imagine the team becomes more diverse, interaction and face-to-face communication with all team members is not always feasible, and the customer is mostly not the equivalent of an end user. Since in several agile UCD approaches, the UCD team and agile developers work in parallel, some documentation is needed to keep all team members informed about requirements, the progress of the project, and artifacts.

We believe that a combination of both UCD and SE, as described by the aforementioned agile UCD approaches of Paelke et al. and Sy, should preserve the use of notations of both domains. However, we think a common notation is needed in order to connect functional and non-functional requirements and contextual information, to have a document that contains the design rationale and to support communication in the team that involves people having different backgrounds. Several characteristics of user stories inspire us to realize a connection between our COMuICSer storyboards and user stories. The next section describes the results of interviews with agile SE practitioners regarding agile approaches and the connecting user stories and storyboards.

9.3 Interview with Agile SE Practitioners

In order to understand the needs of agile SE practitioners regarding non-functional requirements, we interviewed four project and proxy managers of agile development teams of a large Belgian company that is active in the software industry. This company has many years of experience in applying traditional as well as agile software engineering approaches. A proxy manager is a team member that schedules the iterations of an agile project but also participates in pair programming, and is the first contact of the customers.

During this semi-structured interview, we asked the interviewees about the agile approach, their practices in requirements elicitation and we presented our COMuICSer tool in order to know their ideas regarding storyboarding in agile SE. In Appendix A, Section A.4, we list some of the questions used for this semi-structured interview.

9.3.1 User Stories and the High-level Analysis Document

When an agile project is started, a proxy manager first discusses the project with the customer. After this first meeting, the proxy manager uses the first two weeks of the project to collect the details regarding the system in a document. The agile SE teams avoid the creation of documentation as deliverable for the customer, but their documentation created during the preparation of a project is not limited to user stories. In the past, user stories appeared to be insufficient to provide an overview of all aspects of a project for developers that joined the agile team when the project already started. Furthermore, user stories proved to be too low-level to explain the status of a project to the customer. Consequently, the teams started creating high-level analysis documents that explain how user stories are linked to several parts of the system and the business process models. Typical artifacts that are contained by the high-level analysis include an interaction diagram, a use case diagram, a conceptual model and an initial backlog with a list of user stories. Sometimes screen flows or UI designs are added to the high-level analysis as well. Only limited versions of these models are included, without paying a lot of attention to the notational accuracy of the models.

Once the high-level analysis is created, epic user stories [Coh04] are formulated. These high-level user stories are entered into a MS Excel sheet and are split up into more detailed stories. All ideas, remarks and acceptance tests that need to be considered for the detailed user stories, are entered in an extra field of the epic user story.

A workshop with the team and the customer is organized to fine-tune the high-level analysis and to obtain information for the detailed stories, which are prepared by the team. A detailed user story contains a function, its business value, and a job role of people that the function is intended for. Besides this short description of a requirement, user stories include their location in the high-level analysis, diagrams and UI mockups that are related to this user story. Each user story is documented in a MS Word file.

User stories are not only complemented with other artifacts in order to clarify and provide more details. These additional artifacts are also helpful

during meetings with the customers. Often, the customers have problems to understand and situate user stories in the overall project, because the user stories are expressed on a level that is too detailed with respect to the general structure of the project. Consequently, the user stories are usually accompanied by a business process model when they explain the status of the project to the customers. Nevertheless, they observe that customers that are not familiar with business process models still have difficulties to understand the notation used for these models, thus a more general notation to communicate efficiently with the customer is needed.

9.3.2 COMuICSer in Agile SE Teams

After asking the interviewees about the agile SE approach that is common in their company, we presented COMuICSer storyboards and the accompanying tool, presented in Chapter 5 and 7. The interviewees were pleasantly surprised by the opportunities of our COMuICSer storyboards and tool. The agile SE teams are using personas already, and a storyboard would be an interesting additional artifact to visualize the high-level analysis.

Earlier experiences of making sketches of diagrams already showed the interviewees that a visual representation may help the customer and the agile team to get a better understanding of a future system. They confirm that by connecting a storyboard to user stories, functional and non-functional requirements can be linked. When this connection would be supported in a tool, it may be interesting to check consistency with a storyboard when user stories are updated. Furthermore, a storyboard may be helpful to provide information regarding the context of use of an evolutionary prototype, which is produced at the end of an iteration to communicate the results with customers and/or end users or which is evaluated during exploratory tests. One remaining question is whether the proxy manager or other team members should create the storyboard if the project does not involve a UCD team. Some interviewees are hesitant to draw sketches and are skeptical regarding the return on investment when they have to create storyboards, but they are interested in trying storyboarding for their projects.

9.3.3 Discussion

We can conclude from this interview that agile SE lacks notations that provide an overview of the contexts of use of a system, allow a smooth communication within multi-disciplinary teams, and take into account functional as well as non-functional requirements, which is also supported by our focused literature

study. The interviewees are convinced that the use of a visual representation of the requirements improves understandability for all stakeholders. In the remainder of this chapter, we will describe how storyboards can be connected to user stories and how tool support can allow both UCD and agile SE teams to use artifacts they are familiar with, during the project.

9.4 Connecting User Stories to COMuiCSer Storyboards

The COMuICSer storyboarding approach and accompanying tool for multidisciplinary teams in software development projects may result in a greater involvement of all team members and end users in engineering processes. This visual language specifies the way that storyboards are created, while preserving creative aspects of storyboarding. The similarities to comics, provide multidisciplinary teams with an accessible notation for requirements of applications for diverse platforms and a varying context of use [HMLC10] (Chapter 7). This simplifies the relation of storyboards with other activities in a UCD process and SE activities [HdB⁺11] (Chapter 8). Because efficiency is one of the keywords in agile approaches, we propose a technique to combine storyboards and user stories, in order to facilitate the communication in agile UCD teams.

Figure 9.1 shows a storyboard for a search application for professional TV searchers which was already introduced in Chapter 8 and Figure 8.4. Each scene of the storyboard is connected to one or more user stories. The depiction of contextual information is also suitable to visualize business process models in a way that is more accessible to all stakeholders, without demanding any knowledge about the notation.

A typical formulation of a user story starts with “*A user can*”, followed by a functional requirement. If applicable, user roles can be included in a user story, which implies that developers keep in mind the user and her role. One of the early adopters of Extreme Programming, introduced the following template to assure that a user role was part of the story [Coh04]:

```
As a (role)
I want (function)
so that (business value)
```

When the user story is created while a relevant COMuICSer storyboard scene is shown, it is very likely that this user story also connects to one or more personas. Personas can be considered as a role in a user story and consequently

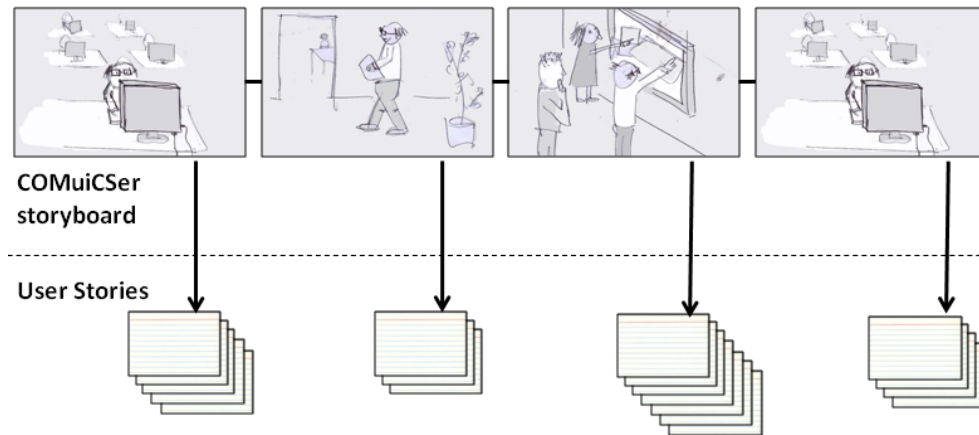


Figure 9.1: Connecting user stories to a storyboard. Each storyboard scene can be connected to several user stories. This highly visual relationship between storyboard scenes and user stories, add contextual information to the user stories (e.g. in what environment is the system used?, is the system used by individuals?).

creating the connection between the user story and the storyboard clarifies what type of user uses the system in a particular context of use (e.g. a specific component of a business process model). The visual depiction of related user stories and a scene of a COMuICSer storyboard reminds team members during the creation of user stories and in later stages of the development process of the different devices that are used in a particular context. Furthermore, we assume that a COMuICSer storyboard scene can be connected to several user stories, which implicitly groups user stories that are related to each other.

Figure 9.2 exemplifies how a user story is complemented with details of the context of use in the third scene of the storyboard shown in Figure 9.1. As this example shows, the functional requirement concerns a feature to open a personal folder. Based on the scene of the storyboard, a developer knows that this TV researcher needs to open the personal folder on a shared system in the meeting room. Furthermore, it concerns a large multi-touch display, which reminds the developer and UI designer to the fact that the application can be used by several personas simultaneously. Although rapid Contextual Design [HWW04] also suggests the use of storyboards, they do not describe the relation with user stories.

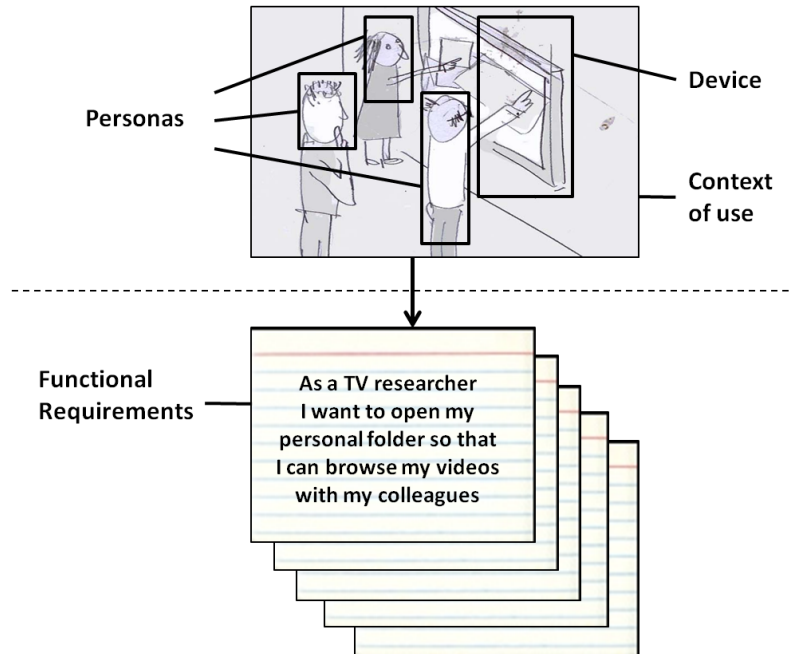


Figure 9.2: Example of the connection between a storyboard scene and a user story. The scene represents the context of use, including personas and device information, while the functional requirements are included in the user story.

9.5 Tool Support for Connected Storyboards

We propose an extension of our COMuICSer tool in order to realize the connection between storyboards and user stories which is introduced in section 9.4. In this section we will first describe existing agile planning tools that are used for managing user stories of a project. Following we will discuss how the COMuICSer tool is extended to support user stories.

9.5.1 Agile Planning Tools

Although paper cards and a pencil can be the basic tools for the creation of user stories, the digitization of these requirements can benefit the communication in project teams at several stages and different locations [MJ10, WMMO10]. A plethora of agile planning tools is currently available. Most of them support the creation and editing of user stories as well as managing and monitoring iterations [Coh]. Most of the tools are developed for a desktop setting, using a

keyboard and a mouse. Because the use of paper cards and pencil feels more natural, Hurlbutt et al. proposed a digital pen application to digitize paper notecards [HK06].

An inventory of agile planning tools and several guidelines for designers and users of this type of tools, was provided by Wang et al. [WMMO10]. Morgan et al. investigated support for agile planning meetings [MWK⁺07]. They distinguish two settings: distributed and collocated agile planning. Since most of the agile planning tools listed in [Coh] are intended for distributed planning, Ghanam et al. propose a tabletop application for collocated agile planning meetings [GWM08]. While this tool solely concentrates on agile planning and the creation of user stories, *Conversation Spaces* [MJ10], are explored to digitally collect all artifacts used during collocated creative requirements elicitation sessions. This concept may improve communication and collaboration during requirements elicitation.

Because of the availability of many existing tools for agile planning we do not intend to develop another agile planning tool. In order to reduce the extra effort for agile project teams, user stories created in our storyboarding tool, can be exported and consequently be imported in an agile planning tool the team is accustomed to. Our tool will support the relationships between storyboards and user stories and can operate as an interface between the UCD team and the agile team during and after the requirements elicitation, without forcing agile SE teams to switch to the COMuICSer tool.

The ART-SCENE [KMK09] tool guides project stakeholders in creating and generating scenarios and more complete requirements. This tool connects scenarios and requirements, which introduces a common language in requirements engineering at different locations. Extensions of ART-SCENE, support a composition in a Creative Requirements Innovation Space (CRIS) that connects image and text elements to requirements. In contrast to this tool, our tool visually represents scenarios, furthermore, the COMuICSer storyboard supports the connection between the requirements and other artifacts, created later in a design and development project.

9.5.2 User Stories in the COMuICSer Tool

We describe how the COMuICSer storyboarding tool presented in Chapter 5 and 7 was extended to support user stories. The tool can be used for the creation of user stories, that are connected to a COMuICSer storyboard. During the creation of the storyboard that involves a UCD team, and optionally an agile SE team or users, the COMuICSer tool can be used to connect the

scenario to storyboard scenes, and to add annotations (e.g. personas) to the storyboard.

In a follow-up session, preferably with both the UCD and the agile team, the user stories can be created in COMuICSer. For each user story that is created, a team member can drag a blank user story from the pile of index cards to a storyboard scene. Following, a new user story is added to the pane that is linked to the storyboard scene (Figure 9.3, A), and has a unique ID, which is helpful for planning and communication purposes in the agile SE team. The connection of a scene to a user story is not necessarily a unique connection. Since a functional requirement can be part of various contexts of use, a user story can for instance be connected to several different scenes.

A new user story contains some input fields in order to specify the user story (Figure 9.3, B). These fields are formatted according to the template that is often used to include user roles in user stories (Section 9.4). Automatically, all personas are extracted from the personas available in the storyboard scene, and can be selected from a combobox. Other fields are available to add a function, and a business value. Furthermore, a priority can be added to the user story. By representing the user story next to a storyboard scene, the context of use and non-functional requirements can easily be considered while formulating the user stories.

Once all user stories for a particular storyboard are created, agile planning can take place in the agile SE team. Since we decided not to develop another agile planning tool, and many of these agile planning tools contain a feature for exporting and importing user stories, our tool includes a feature to export all user stories (Figure 9.3, C). Since the formats for saving user stories can differ, this first version of our tool, exports the user stories to a XML file, which is for instance compatible with the agile planning tool ScrumDo [Scr]. Nevertheless, an export for any other agile planning tool can be included in our tool as well.

The agile planning tool ScrumDo allows agile teams to export the user stories any time. The information exported from ScrumDo, can be imported again into the COMuICSer storyboarding tool to keep the connection to the storyboard up to date, and to inform the UCD team about the project status (e.g. *todo*, *doing*, *review* and *done*). The status of user stories can for instance provide interesting information for a UCD team when this team is preparing usability tests. For instance, when all user stories in one scene of the COMuICSer storyboard are implemented, the UCD team can consider to conduct a usability test for the situation depicted by that storyboard scene.



Figure 9.3: User story creation in the COMuCSer tool, and the coupling to an agile planning tool such as Scrumdo [Scr].

9.6 First User Study

The extension of our COMuICSer tool was evaluated in an informal user study that involved one of the proxy managers that participated in the semi-structured interview we discussed in Section 9.3. We observed the proxy manager while she created the storyboard and user stories using the COMuICSer tool. Consequently, we interviewed the proxy manager with respect to the creation of user stories that are connected to a storyboard. Some of the questions asked during this interview are listed in Appendix A, Section A.4. Afterwards, we frequently contacted her to ask about the use of the storyboard throughout the project. The aim of this informal study was to obtain some feedback of practitioners in agile SE regarding the use of COMuICSer storyboards combined with user stories. We were interested in their opinion with respect to the time needed to create storyboards, the suitability of the storyboarding notation for their projects and the connections between the storyboard and other artifacts. Furthermore, we wanted to evaluate the features of the tool, described in Section 9.5.

This proxy manager decided to introduce storyboards to the agile SE team, that is responsible for the development of a system for several administrative services affiliated to a governmental institute. This system can be represented in a business process model: different users located at several offices need to pass particular information to each other using a system.

The aim of the storyboard is to clarify the business process model and the context of use of the system that needs to be developed. The proxy manager decides to prepare the storyboard as part of the high-level analysis (Section 9.3). Following, the storyboard will be used in workshops and meetings with the team and the customer in order to provide an overview of the use of the system. Before the storyboard is created, several raw models, such as a business process model and a preliminary interaction diagram are already specified.

9.6.1 Creating the Storyboard

Before the proxy manager starts storyboarding, she formulates the personas that may be involved in the case. When imagining names for the personas, she starts sketching and considers names that easily can be linked to the job role of this persona, as the job role will specify the characteristics of the front end and user interface of the system. While doing this, she already steps through all situations in the scenario of use, which results in several different cases

(e.g. submitting an application using paper documents, submitting an application using a webform, informing the customer about the acceptance of an application and informing the customer about the rejection of an application). The first sketches can be considered as a storyboard (Figure 9.4): personas, devices and documents are already depicted in these sketches.

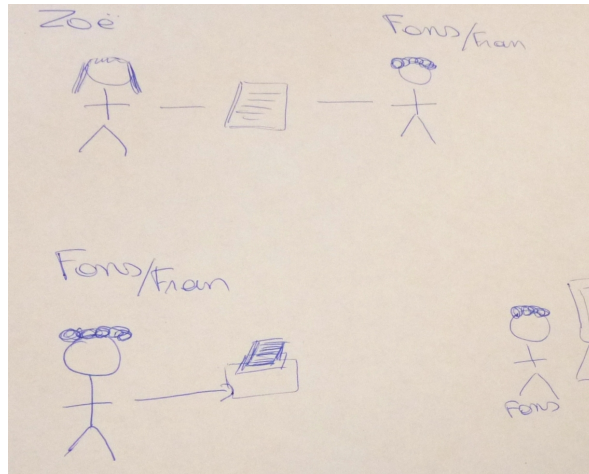


Figure 9.4: Extract of the sketches that were drawn before the storyboard for the agile team was created.

The proxy manager judges that the first sketches are not clear enough for a COMuICSer storyboard and decides to draw new scenes for the storyboard. First, one scene is drawn and uploaded into the tool. Because of limited drawing skills and in order to save time, the proxy manager prefers sketched scenes, supplemented by the faces of people that are provided by the COMuICSer tool, which make it easier to differentiate the characters in the storyboard. Furthermore, she considers to reuse parts of scenes as much as possible.

Once the first scene is finished, the proxy manager explored the main features provided by the COMuICSer tool and has a clear idea about the tool. She is convinced that the use of the sketches and the availability of images in the tool allows everyone to create storyboards. Next, all sketches of the storyboard are drawn before uploading them into the COMuICSer tool. Figure 9.5 shows four scenes of the storyboard that was created in this study.

Since only desktop computers are used by the users of this system, devices are less important, but annotating personas is important to depict the different situations in the business process. Because this storyboard describes several

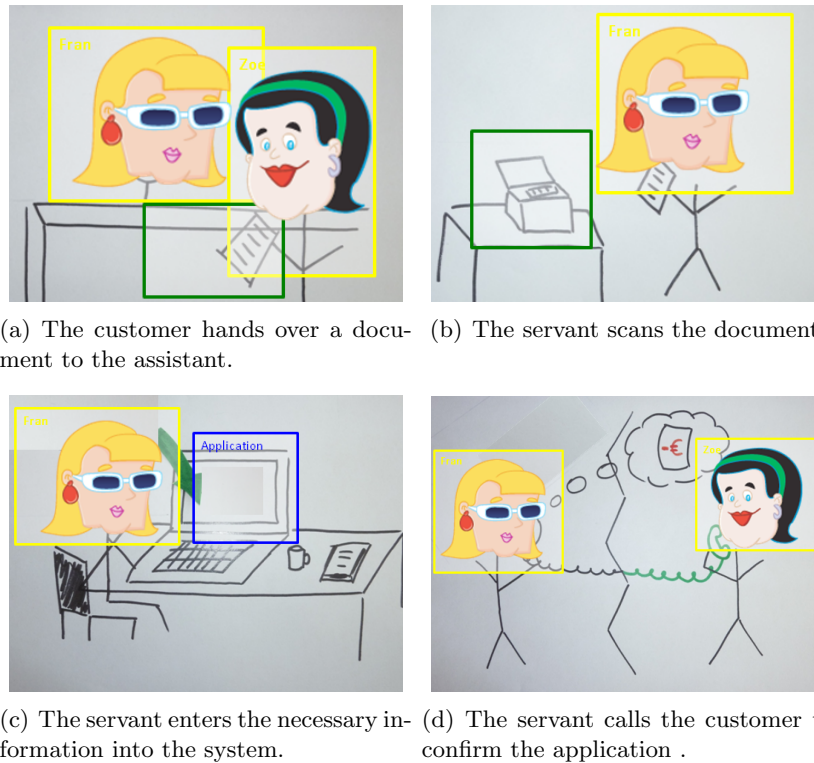


Figure 9.5: Four scenes, extracted from the whole storyboard that was created during this study. Only limited descriptions and details are shown because of anonymity reasons.

environments and services of a business process, the proxy manager decides to identify particular offices by drawing the logo of a particular service on the wall on the background of a scene. The logos allow the reader of the storyboard to recognize the different environments the system is used in. These logos are hidden in Figure 9.5 because of anonymity reasons. Additionally, the proxy manager uses annotations to highlight the different applications that need to be developed by the agile SE team. These annotations clarify which end user is using a particular application of the system. Besides annotating the scenes, the proxy manager also labels transitions and adds timing information. Finally, she creates a PowerPoint slideshow that contains the storyboard and can be used in workshops and meetings.

This storyboard will be used to create the backlog together with the development team and depicts four different cases. These four cases were visualized in parallel storyboard sequences. The proxy manager created the storyboard of the four best case scenarios. When creating user stories, she always starts with imagining the easiest cases, and following she considers worst case scenarios: *“It might be interesting to extend the storyboard with cases that are applicable for the worst case scenario. These different cases should also be mentioned in the user stories, mainly to keep them in mind for acceptance testing after development.”* Consequently, the storyboard can be extended by visualizing the worst case scenarios. However, one needs to consider if it is worth the effort to create different storyboards for small exceptions in a scenario.

9.6.2 Specifying User Stories

Based on the storyboard, the proxy manager starts creating the epic user stories. For each persona, she skims the storyboard and considers which epic user stories need to be defined in each scene that includes that persona. She appreciates the automatically generated list of personas in a user story form: *“This forces team members to think from the perspective of a user.”* An example of an epic user story created in this session is:

```
As a file manager of service X,  
I want to consult an application in detail,  
so that I can see the application information  
and possibly a decision.
```

During the creation of the epic user stories, the proxy manager encounters similar problems as in her earlier experiences when creating user stories. The storyboard as well as the user stories are formulated from a user’s perspective, which implies that the back-end services and systems are not shown in the storyboard and thus are not included in the user stories. In order to keep in mind the different services and applications that need to be available in a system, the proxy manager suggests to add an extra field in the user story form to add particular remarks and reminders that are already known in this phase of the project. When managing these user stories, which are exported from the COMuICSer tool in MS Excel, the proxy manager can take into account these remarks and reminders.

After the creation of the user stories, the proxy manager exports the user stories and concludes that she would use the COMuICSer tool to prepare the

storyboard and user stories as part of the high-level analysis, because the storyboard can be very helpful to visually clarify the use of the future system. She estimates that the creation of the storyboard would take half a day to one day, depending on the system that needs to be developed, which is a reasonable amount of time within the two weeks that are reserved for the preparation of the high-level analysis.

9.6.3 Using the Storyboard

The proxy manager confirms that the creation of the storyboard helped her to specify the different cases in which the system is to be used. Furthermore, she explained that she almost immediately would understand the project, if this storyboard was provided to her right after the first meeting with the customer: *“If a storyboard about the system would be available, then it would be much easier for me to understand the project and what we need to develop.”*

The PowerPoint slideshow, which was generated in the COMuICSer tool storyboard and contains the storyboard, was reused for meetings with the team. At a certain moment in the project, a new software engineer joined the team, and the storyboard was used to clarify the project and the status of the project to this new team member. This person declared that the storyboard was a clear, visual introduction to the system that quickly confirmed his idea regarding the system and its context of use. In particular, this new team member appreciates the representation of all systems and applications: *“The storyboard’s clear visual representation of the interaction between all involved applications did help me gain a good understanding of the whole ‘picture’.”*

9.6.4 Discussion

This preliminary user study showed that the creation of storyboards can be helpful for the elicitation of requirements in the form of user stories because of the following arguments. The storyboard created in this study represents several situations in a business process. The possibility to create user stories in the COMuICSer tool is appreciated. Because the visual depiction of personas is available in several scenes of the storyboard, suitable user stories can be formulated for each persona in each storyboard scene. Although relating user stories to a storyboard during the requirements elicitation is an improvement, one problem in formulating user stories persists. Specifying requirements for the back-end system is very difficult because neither the user stories nor the storyboard support this type of information.

As we expected, the proxy manager limits the use of our COMuICSer tool to prepare the project. The COMuICSer tool was mainly used to create the storyboard and connect storyboard scenes to user stories. However, by saving the storyboard in a PowerPoint slideshow and exporting the user stories, these artifacts could be reused throughout the project. When a UCD team would be involved, the agile SE team can limit the use of the COMuICSer tool to the creation of user stories that are connected to the storyboard.

This first user study showed that connecting storyboards and user stories in the COMuICSer tool is promising for agile SE teams because it forces team members to consider the contextual information contained by the storyboard when they create or consult user stories. Although this first study of use was only conducted in the field of agile SE and did not involve a UCD team, we may conclude that agile SE teams are likely to benefit from using connected storyboards, created in the COMuICSer tool. In combination with results of Chapter 6, we can conclude that both an agile SE team and a UCD team may benefit from creating and using storyboards. The involvement of a UCD team, should reduce the efforts of the agile SE team to create a storyboard, as we described in Section 9.5.

The company that participated in our study, takes an approach that to some extent is alternative with respect to general agile SE practices. This company prefers to create more documents in the preparation phase than is strictly recommended for agile SE, because these documents facilitate a general understanding within the team, during the development. Although these documents are only intended for guiding the development, their approach is different than in agile SE in general. Earlier research showed that most agile practitioners limit the documentation to user stories [CR08]. However, a study of Martin et al. showed that the company that was involved in our study, is not the only company that creates additional artifacts in the preparation phase [MBN04]. In general, Cao et al. [CR08] report that practitioners are aware of the risks of relying on the high-quality interaction that is accompanying the limited documentation of user stories. When this interaction or communication within the team breaks down, the user stories contain too little information regarding the requirements for the project. Consequently, we do not think that the alternative approach of the company that participated in our study, had a great influence on their opinion regarding the COMuICSer storyboards that are connected to user stories. Nevertheless, further evaluations of our approach in other agile SE teams is recommended.

9.7 Conclusion

Based on a focused literature study and an interview with agile SE practitioners, we introduced an approach to connect user stories, a typical notation for requirements elicitation in agile SE, to COMuICSer storyboards. Despite earlier studies that integrated UCD practices into agile SE [PN08, Sy07], there was no technique or notation proposed to enhance communication in the team and to combine functional and non-functional requirements. Based on the results of our first studies, we decided to implement accompanying tool support. In this tool, user stories can be linked to storyboard scenes during the elicitation of these user stories, which stimulates the team members to consider the correct context of use for the user stories.

In an informal first user study, the approach to combine COMuICSer storyboards and user stories, as well as our tool, were evaluated. As shown by the study, the visual representation of a storyboard in the COMuICSer tool allows teams to consider multiple solutions, to clarify complex interactions and to investigate several viewpoints when deciding about functional requirements. Furthermore, both the storyboard and user stories are useful at later stages of agile development projects.

In this chapter we answered the last research question introduced in Chapter 5, RQ 4: *How can storyboards be connected with software engineering artifacts?* Since our approach connects COMuICSer storyboards to user stories, functional requirements to contextual information and non-functional requirements, this approach will also be helpful to bridge the gap between UCD and agile SE. Both UCD and agile SE practitioners can keep using artifacts they are accustomed to, but they also have the possibility to keep track of the status of a project, using the storyboard which is connected to updated user stories.

Part III

Reflections and Conclusions

Chapter 10

Interpreting Storyboards

10.1 Introduction

In Part II of this dissertation we presented COMuICSer, a notation and tool for storyboarding in multi-disciplinary teams. COMuICSer can be used as a communication tool in user-centered software engineering (UCSE) processes in order to obtain a common understanding within the team. Furthermore, the structure and the information contained by a COMuICSer storyboard are helpful for later stages in UCSE processes. We proposed several opportunities to connect storyboards with other artifacts created in UCSE processes. Many of the techniques we proposed, were evaluated by a user study or experiment.

In order to evaluate our approach in a larger realistic UCSE project, a longitudinal study that involves our storyboarding techniques in the project is necessary. Because this type of study was not feasible within the scope of the different parts of this PhD, we conducted user studies to assess the use and creation of storyboards in multi-disciplinary teams (Chapter 6), to investigate how the techniques used in comics can be incorporated into a storyboard (Chapter 7) and to study how storyboards and user stories can be combined in agile software engineering (Chapter 9). However, instead of conducting a longitudinal user study, we evaluated the use and interpretation of COMuICSer storyboards in a more general user study than the studies presented in Part II of this dissertation. With this study, we aimed to answer questions such as:

- Do COMuICSer storyboards result in a common understanding regard-

ing a system's context of use within a multidisciplinary team?

- Can a COMuICSer storyboard be considered as a central document, used by a multi-disciplinary team that is involved in a UCSE project?

Truong et al. [THA06] and Sellen et al. [SML⁺09] conducted similar studies in order to investigate how people interpret and understand storyboards. Although these studies considered storyboards that may fit in our COMuICSer approach, the storyboards do not contain annotations that are useful for later stages in UCSE. Furthermore, we are not only interested in the way storyboards are interpreted, we also want to know how the storyboards are used and translated into other artifacts. Our study consists of three stages: (1) collecting realistic storyboards, (2) interpreting the storyboards and (3) discussing the results of the interpretation stage with the authors. Our approach taken for this study, as well as the results and lessons learned are described in this chapter.

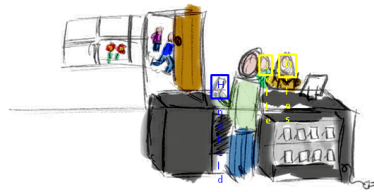
10.2 Collecting Storyboards

The interpretation of a storyboard may depend on the type of application context depicted by the storyboard and the style of the storyboard. Consequently, we collected five different storyboards. Five HCI researchers, three female and two male, were asked to create a realistic storyboard that depicted their applied research project. Their ages ranged from 26 to 33, with an average age of 30 years and all of them had at least a limited level of experience in storyboarding for one year or more. Three authors had a backgrounds in computer science, one author studied software engineering and one author had a non-technical background. There were no restrictions with respect to the style and the number of scenes of the storyboard. Each storyboard concerned a different application context. Furthermore, the number of scenes, the number of personas involved and the intended platforms varied. Table 10.1 shows the characteristics of each storyboard that was collected.

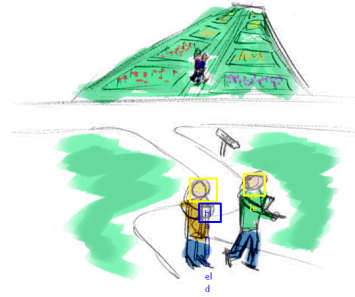
Each storyboard was digitized and annotated using our COMuICSer tool. Following, the storyboards were exported to PowerPoint slidesets, which were evaluated together with the authors and corrected if necessary. At the moment of our study, the cases' statuses ranged from conceptual to extensively evaluated during field tests. The following sections will provide a description of the cases that are depicted in the storyboards, and the styles that are used.

Storyboard	Application context that is depicted	Number of scenes	Number of personas	Platform(s)	Style of scenes
A	Location-aware application for museums	5	2	Handheld device	Sketches
B	Application to generate a tour in a museum and to follow the guided tour	6	3	Desktop pc and handheld device	Photographs with iconic characters
C	Serious game used for training sessions with multiple sclerosis patients	8	2	Desktop pc with haptic device	Photographs
D	Alarm and help system for people suffering from dementia	6	2	Smartphone	Photographs
E	System to query a social network	6	1	Smartphones	Sketches

Table 10.1: An overview of the storyboards that were collected for our study.



(a) Providing handhelds.



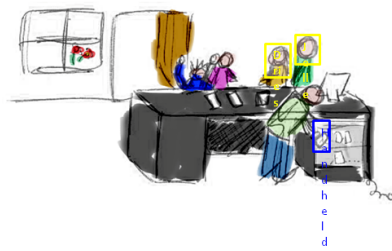
(b) Looking for a point-of-interest.



(c) Information about a building or event.



(d) Interactive moment at a location.



(e) Returning handhelds.

Figure 10.1: Storyboard A - scenes and their titles: Location-aware Application for Museums.

10.2.1 Storyboard A: Location-aware Application for Museums

A system for handheld devices for museums is considered in storyboard A (Figure 10.1). The end users of the system are students of the last grades of secondary school that visit one or more museums. The two personas in the storyboard have to share one handheld device during the outdoor tour. The goal of the system is to guide the students through several locations in an interesting and entertaining way. Storyboard A was created in the early stages of the process for this case. At the moment that this study was conducted, an interactive prototype of the system was developed and evaluated during a field test.

Storyboard A is created using five sketches. Each scene is accompanied by a short textual description that takes two to four lines. Furthermore, personas and devices are annotated in the scenes. The two descriptions of personas that were provided with the storyboard are limited to an age, a name and the study of the students. The handheld devices are tagged in each scene. However, neither the scenes nor the accompanying descriptions of the scenes explain in detail the minimal specifications of the handheld device.

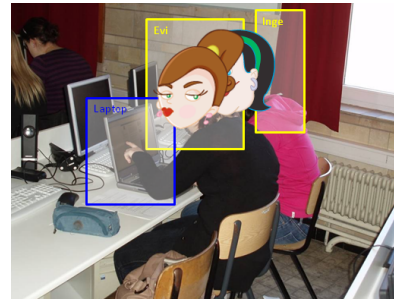
10.2.2 Storyboard B: Customized Museum Tour

Although storyboard B (Figure 10.2) focuses on the same application domain as storyboard A, both storyboards differ in several respects. Storyboard B concerns a system that can be used by students of secondary schools to create museum tours, based on information they consulted online. The tour is constructed gradually in the classroom. Following, the tour is uploaded online and later on it is downloaded on the digital devices available in a museum in order to provide tours for the students that are visiting the museum. The end users of the museum tour are classmates of the students who created the tour and another classmate that is guided by the tour in the museum. At the moment we conducted this study, an interactive prototype to build a tour was developed and evaluated in several field tests that involved several classes.

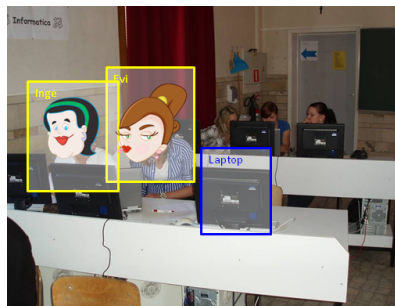
The storyboard for this system consists of six photographs taken during the field tests of the system. The end users shown by the photographs were anonymized by adding iconic images of the actors to the storyboard scenes. The personas and devices were tagged in the storyboard. Three personas accompanied the storyboard, two students that were collaborating for the creation of the tour, and one student who used that tour in the museum. The personas were very limited and only contained a name and an iconic



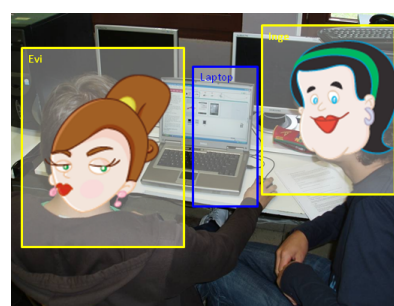
(a) Context.



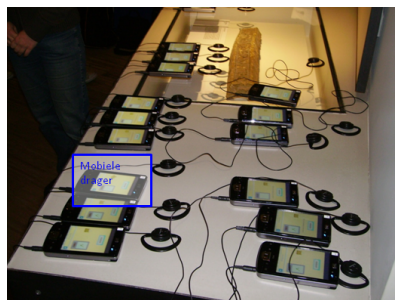
(b) Looking up the key figure of the museum.



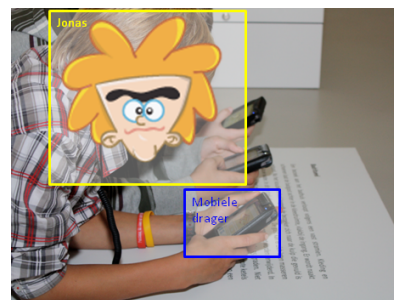
(c) Choosing figure and museum.



(d) Building the tour.



(e) Downloading the tour to a mobile device.



(f) Following the tour in the museum.

Figure 10.2: Storyboard B - scenes and their titles: Customized Museum Tour.

representation of a face. Two types of devices were depicted by the storyboard: a laptop to create the tour in a classroom and mobile devices to use the tour in a museum. The storyboard scenes were accompanied by descriptions that are two to seven lines long.

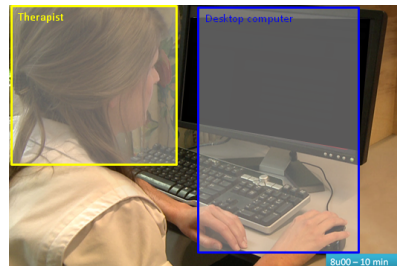
10.2.3 Storyboard C: Serious Game for MS Patients

Storyboard C (Figure 10.3) depicts the configuration and use of a serious game for patients suffering from multiple sclerosis (MS). The end users of the system are a therapist who schedules the game for an MS patient, configures several parameters and guides the patient to the start of the game. The patient plays this game and navigates through the game using a haptic device that trains their muscles of the upper limbs. After the game, statistics are available to measure the progress made by the patient and a social game is played together with the therapist. At the moment of our study, an interactive prototype of this system was developed, and extensively evaluated in several field tests in a clinic for MS patients.

Storyboard C was composed of eight photographs of field tests of the system. In contrast to the typical process of creating and using storyboards, these storyboards were created after the development of an interactive prototype. Creating storyboards later in a UCSE process is interesting for presenting the project and test results to stakeholders and to prepare next iterations in the project. The two personas involved in the storyboard concerned a patient and a therapist and were not elaborated. However, the personas were tagged in the storyboard scenes. Furthermore, the relevant devices of the system were annotated. These devices concern a desktop computer for the therapist, a large display and a haptic device that allows the patient to do the training by means of playing a game. Each scene was explained in a textual description of maximum two lines. Finally, time information was shown in the scenes in order to indicate how long a particular scene takes place.

10.2.4 Storyboard D: Alarm and Help System for People Suffering from Dementia

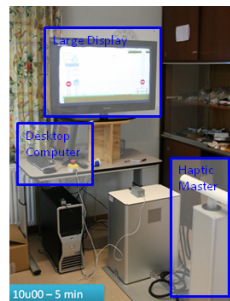
Similar to storyboard C, storyboard D (Figure 10.3) concentrates on the domain of health care. The system presented in storyboard D consists of sensing technologies that are integrated in the natural environment of a person with dementia. Whenever something goes wrong in the procedure of everyday tasks performed by the person with dementia, the system senses that the patient is possibly confronted with a problem. Consequently, the system sends an alarm



(a) Configuring game and schedule.



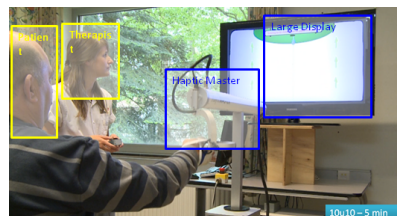
(b) The patient arrives.



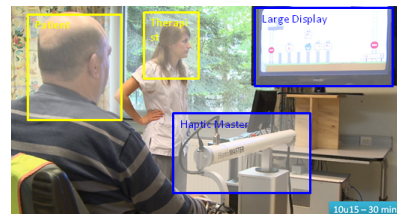
(c) Positioning the patient for the game.



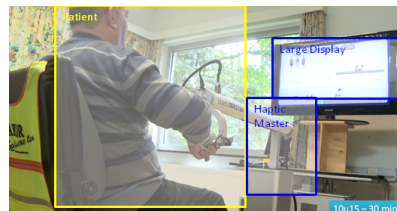
(d) Setting parameters for the patient in this game.



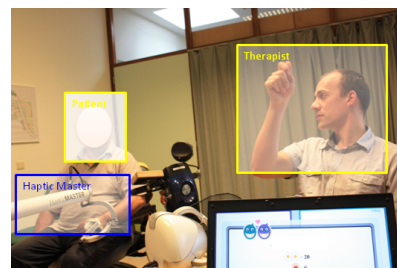
(e) Evaluation exercise.



(f) Start of the game.



(g) Playing the game.



(h) Social game.

Figure 10.3: Storyboard C - scenes and their titles: Serious Game for MS Patients. The face of the patient was anonymized because of privacy issues.

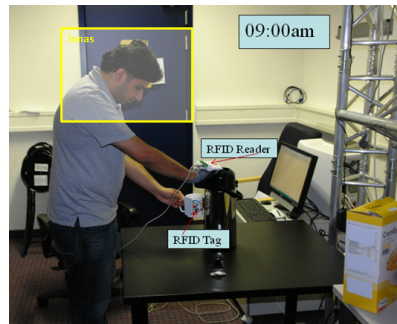
message to a relative. Using an application on their smartphone, relatives can provide help to the person with dementia. At the moment of our study, the system was in a conceptual phase, further design and implementation were planned for the near future.

The storyboard for this system consists of six photographs. The majority of the storyboard scenes depict a team member in a lab environment. Because of privacy reasons, the author opted to incorporate photographs of a team member and a lab environment in the storyboard rather than using photographs of a person with dementia and the rooms in their house. This storyboard contains personas which shortly describe the skills and goals of the end users of the system: a person with dementia and a relative who takes care of this person. These personas were annotated in the scenes. The author carefully labeled all technology available to sense what the person with dementia is doing. Furthermore, the device for which a UI design was needed, a smartphone, was annotated in the storyboard. The first and the last scene were explained in a description of two lines, the other scenes only were accompanied by a title.

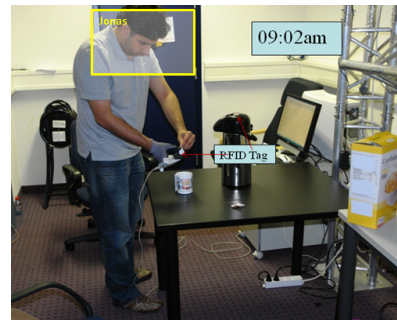
10.2.5 Storyboard E: Querying through Social Networks

Storyboard E (Figure 10.3) depicts the context of use of a system that allows people to query a social network. One of the cases for this system is the situation in which a PhD student is attending a summer school in a foreign country. This person gets off the train too soon and is lost. Because this person does not understand the language spoken on the current location, he decides to query his social network. His smartphone detects its location. Accompanied by a photograph of this person's surroundings, this detailed location information is sent to the social network. The destination and time constraints of this person are sent to the social network as well. Consequently, the contacts in the social network are invited to propose a solution to the person who is lost. In this storyboard, a colleague of his advisor is in the neighborhood, and arranges to share a taxi to arrive at the location of the summer school in time. At the moment of our study, an interactive prototype of this system is available and is evaluated in a user test.

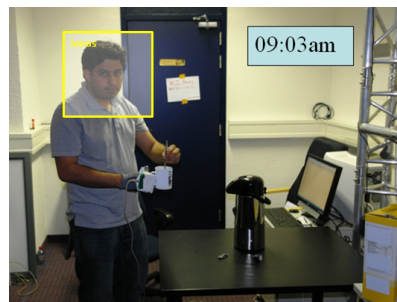
This storyboard contains six sketches depicting the situation of the person who is lost. This person is annotated in the storyboard as a persona, that consists of a name and the sketched representation of the person. Besides this persona, the smartphones are also annotated. Some scenes also specify the type of information that vocally can be entered into the system in order to save time. The descriptions accompanying the scenes consist of one to



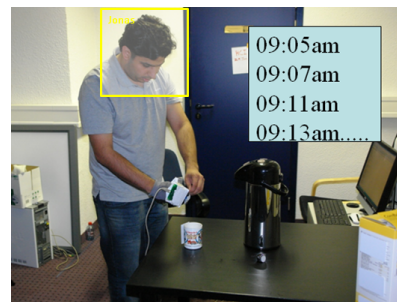
(a) Jonas puts coffee in cup.



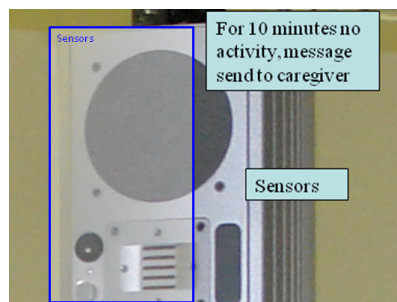
(b) Add creamer in a mug.



(c) Mixing creamer .



(d) Jonas waiting for long time to figure out what to do with mug.

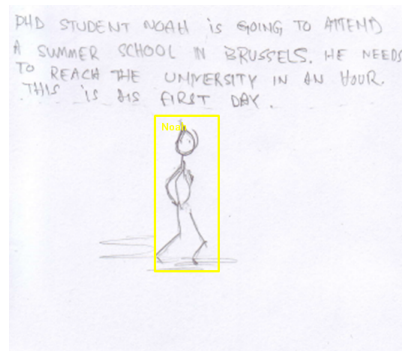


(e) Message send to Mia for assistance.

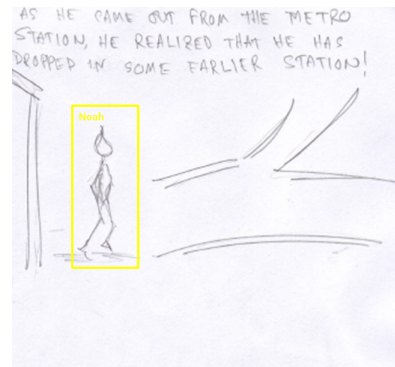


(f) Mia gets an alert and will provide assistance to Jonas.

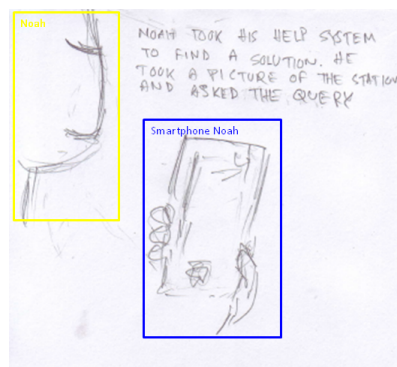
Figure 10.4: Storyboard D - scenes and their titles: Alarm and Help System for People Suffering from Dementia.



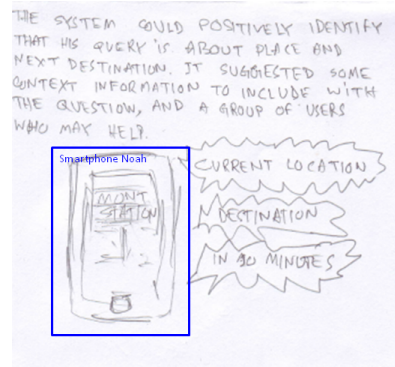
(a) First day of summer school.



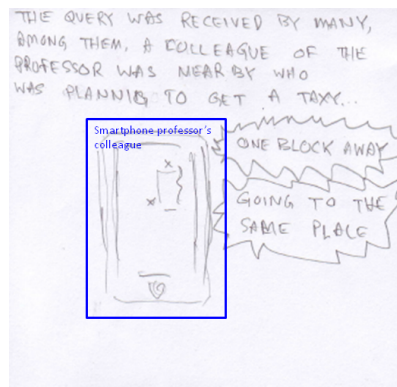
(b) Lost.



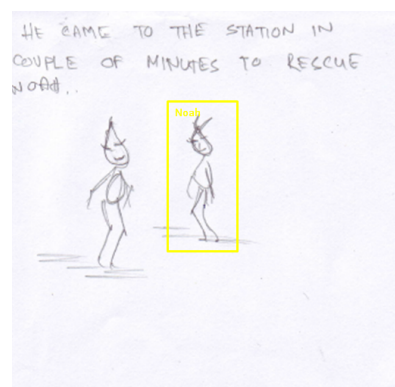
(c) Photo and query.



(d) Query.



(e) Answer.



(f) Solution.

Figure 10.5: Storyboard E - scenes and their titles: Querying through Social Networks.

three lines of text. None of these descriptions of the scenes specify the speech technology that is incorporated in the system.

The following sections will describe how these five different storyboards were interpreted in a user study.

10.3 UI Designs Informed and Inspired by Storyboards

The aim of this study was to gain insights in the way storyboards are interpreted, the thinking behind people's interpretation of storyboards and the approach people take to translate the storyboards into UI designs at later stages. This was stimulated by asking participants to create UI designs based on their interpretation of the storyboards we collected. We preferred not to limit this part of the study to just asking the participants to describe their interpretation of storyboard in their own words as Truong et al. [THA06] did in their study concerning storyboards. Because the storyboards we collected contain short textual descriptions of scenes, they could easily be copied or reformulated in the participants' explanation of the storyboard. By asking the participants to create UI designs, we could obtain information regarding the way information contained by a storyboard is interpreted and translated into other artifacts.

In order to check the correctness of the participants' interpretation of the storyboards, the participants were asked to discuss the resulting UI designs together with the author of the storyboard. The following sections present a detailed description of the participants and the setup and procedure for this user study.

10.3.1 Participants

Ten participants, three female and seven male, were asked to create a low-fidelity prototype in order to concretize their interpretation of one of the storyboards. Their ages ranged from 22 to 32, with an average age of 28. Eight participants had a background in computer science, the other two participants had backgrounds in graphic and interaction design. Four of them were occupied as a developer or project manager in different companies, three participants were HCI researchers in academics, two participants were students of our Master Computer Science-HCI and one participant was a graphic designer with experience in academics as well as industry. The participants of this part of the study were not involved in the creation of the storyboards in order to

avoid that their participation in two parts of the study would influence the results.

All participants but three (one HCI researcher and two students) had no experience in storyboarding. The participants' experience in UI design ranged from no experience at all to more than five years of experience in UI design. Two participants had no experience at all in UI design. Four participants had a few months to more than one year of experience in UI design and create UI designs a few times a year or less. Four other participants had more than five years of experience in UI design and create UI designs monthly or weekly.

10.3.2 Setup and Procedure

The setup and procedure were fine-tuned in a pilot study which involved a representative participant other than the participants that were invited for the study. Each participant of the study individually took part in a UI design session and was instructed to create UI designs based on one of the storyboards we collected. For each storyboard that was collected, two UI design sessions were conducted. When recruiting the participants, we randomly assigned a storyboard to the participant. However, we assured that they had no knowledge about the case that was presented in the storyboard they had to interpret.

In the beginning of each session, a briefing document was handed on to the participant. This briefing document shortly explained what a storyboard is and what level of detail was expected for the UI designs that had to be created. Considering the limited time to create the UI designs, it was allowed to sketch rather conceptual designs. However, we asked the participants to consider the information contained by the storyboard as much as possible for the resulting UI designs. Once a session was started, the PowerPoint slideshow containing the storyboard was provided to the participant on paper and on a laptop. During the UI design session, the participants could freely use the paper and digital storyboards and the materials available on the table, including pencils, colored pens, A4 sheets, a notebook, post-it notes, scissors, glue and a ruler.

In order to understand the thinking behind a participant's interpretation, we asked the participants to think aloud. An observer took notes throughout each session. Furthermore, all sessions were recorded using a video camera in order to analyze them later on. Each UI design session, including reading or interpreting the storyboard and creating the UI designs took maximum 45 minutes. If the participants finished in less than 45 minutes, the UI design session stopped earlier.

After the UI designs were created, the participants were asked to fill out a questionnaire asking about their experience in UI design and storyboarding, their findings of the UI design session and their satisfaction with the UI designs they just created. Following, the author of the storyboard was called to participate in a discussion regarding the UI designs. The authors of the storyboard were instructed to assess the correctness of the UI designs and the participant's interpretation of the storyboard. Since most of the storyboards concerned a system that was already prototyped, the authors of the storyboard had a clear view on the system in order to evaluate the UI designs. Each discussion took approximately 10 minutes.

After the discussion, the participant was asked to answer the last part of our questionnaire containing questions regarding possible misconceptions and their satisfaction of the UI designs after the discussion. The authors of the storyboard were asked to judge the UI designs, identify possible misconceptions and improvements in communicating the context of use of a system through a storyboard in a short interview. The questionnaire and the questions asked to the authors in the interview are available in Appendix A, Section A.5.

10.4 Observations and Results

In the following sections, we will describe the results of the questionnaire and our observations. First, the results of the UI design sessions and the discussions with the authors will be discussed in general. Following, we will present some results that specifically can be related to the five different storyboards.

10.4.1 Creating UI Designs

Once a participant received a storyboard, she started reading it from the beginning to the end. One participant preferred to use the digital storyboard, two participants used both the paper and the digital version of the storyboard and seven participants used the paper version to read the storyboard. The time needed to *read and interpret the storyboard* ranged from 01:16 minutes to 07:35 minutes, with an average time of 03:51 minutes. Table 10.2 shows more detailed information regarding the time needed to read the storyboard and other characteristics.

For all participants but one, the preference for the storyboard version (paper vs. digital) was unchanged for the creation of the UI designs. One of the participants that used the paper version while reading and interpreting the storyboard, switched to the digital version for the creation of the UI designs.

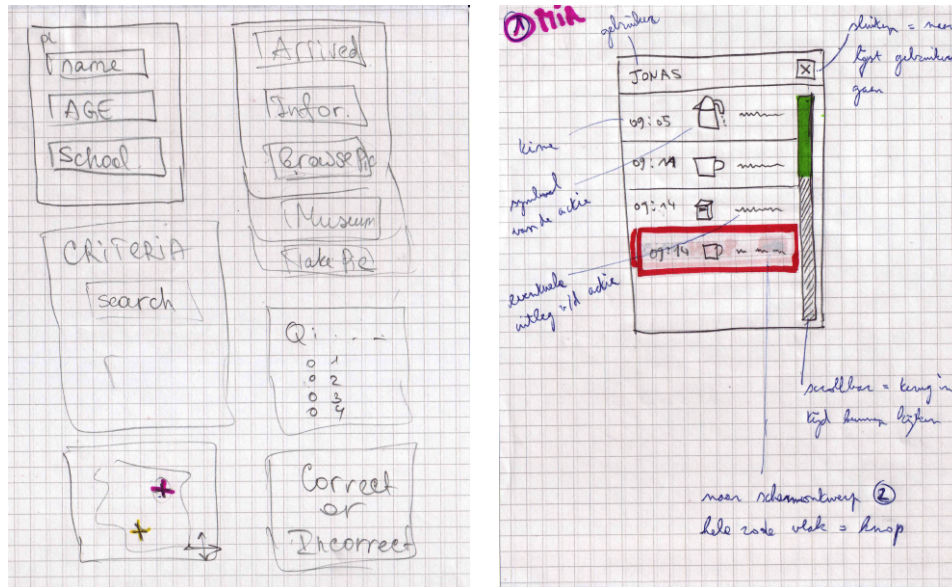
Seven participants only used paper (notebook or A4 sheets), a pencil and an eraser for sketching the designs. Three participants extended this set with colored pens, and one of them also used scissors and glue. The time needed to *create the UI designs* ranged from 13:26 minutes to 40:10 minutes, with an average time of 29:41 minutes.

Storyboard	Time to read storyboard		Time to create UI designs		Number of UI screens	
	μ	σ	μ	σ	μ	σ
A	3'11"	1'46"	20'46"	5'54"	9,5	2,12
B	5'16"	2'14"	33'25"	6'24"	9,5	0,71
C	3'09"	0'56"	39'54"	0'23"	13	5,66
D	5'59"	2'16"	35'22"	0'43"	6	2,83
E	1'41"	0'35"	18'56"	7'47"	9	1,41
All	3'51"	2'05"	29'41"	9'36"	9,4	3,72

Table 10.2: An overview of the observations for each storyboard in average (μ) and standard deviation (σ).

The *approaches* taken for accommodating the design of the UI to the storyboard differed. Most participants simply browsed or scrolled the storyboard scenes sequentially while creating the UI designs. However, four participants spread the paper storyboard scenes on the table in the right order and consulted this overview while sketching the designs. One of these participants first eliminated the scenes that did not require a UI design, and following spread the remaining scenes on the table. Another observation made during the design session was that six participants iterated on the design while running through the storyboard. Usually this resulted in adding features to UI designs that they already created and explaining the designs by adding arrows and annotations.

As expected when considering the limited time for a UI design session, all *UI designs* were sketchy. Figure 10.6 shows two examples of sketches that resulted from the UI design sessions. The colored pens were used to emphasize a particular part of the UI e.g. to highlight a message to the user or to show the difference between two locations or graphs. The scissors and glue were used by one participant to overlay a mistake in the design. The number of screens created in the UI designs varied from 4 to 17, with an average number of 9,4 screens. The difference of the number of screens in the two sets of UI



(a) The most sketchy UI design, created for storyboard A

(b) The most detailed UI design, created for storyboard D

Figure 10.6: Two UI designs, created by the participants of this user study.

designs created for one storyboard highly varied with differences from 1 to 8 screens.

Four participants *annotated* their UI designs to explain particular features and five participants used arrows between the UI design screens to clarify the navigation in the UI. The relationship between the UI designs and the storyboard was rarely indicated explicitly in the UI designs. Three participants labeled the designs with the persona that was expected to use that particular design and one participant labeled their UI designs with the titles of the storyboard scenes.

The most important results of the *questionnaires* are shown in Table 10.3. Despite the limited level of experience in storyboarding and UI design of some participants, all but two participants felt comfortable with the task they had to accomplish in this UI design session. Not all participants considered the storyboards as understandable. Three participants considered the storyboard as rather difficult to understand, two participants were neutral about this, and five participants rated the understandability of the storyboards at least

Story-board	How comfortable did you feel with the job?		How easy was understanding the storyboard?		How easy was creating the UI designs, starting from the storyboard?		How useful are storyboards to inform UI designers about a system's context of use?	
	μ	σ	μ	σ	μ	σ	μ	σ
A	3,5	0,71	3,5	0,71	2,5	0,71	4	0
B	4	0	3	1,41	3	1,41	4,5	2,12
C	3	1,41	3	1,41	2	0	4	0
D	4	0	2,5	0,71	3	1,41	3,5	2,12
E	4	0	4,5	0,71	3,5	0,71	4,5	0,71
All	3,7	0,67	3,3	1,06	2,8	0,92	4,1	0,88

Table 10.3: An overview of the responses to the questionnaire for each storyboard in average (μ) and standard deviation (σ). Participants were asked to answer to each question using a 5-point Likert scale.

rather understandable. Possibly, the diverse ratings of the participants may be caused by the missing information in the storyboards, which is described later in this chapter.

Five participants rated the creation of the UI designs based on the storyboard as a rather difficult task, two participants were neutral about the difficulty and three participants considered the creation of UI designs to be rather easy. However, all participants but one rated storyboards as rather useful to very useful to inform UI designers about the context of use of a system. All participants correctly interpreted the storyboards. The extent to which their understanding of the storyboards corresponded with the authors' interpretation, is described in the following section.

10.4.2 Discussions with the Authors

During the discussions with the authors, seven participants spontaneously *used the storyboard* to explain their design decisions. Three participants did not use the storyboard to clarify their designs and in one of the latter situations, one author explicitly referred to the storyboard to ask about particular design decisions. Six participants judged the storyboard to be a useful to very useful tool to discuss UI designs.

The discussions with the authors showed that all participants succeeded to correctly *understand and interpret the broad context of use* that was considered for a system. For six participants this approving character of the discussions was reflected in their satisfaction of the UI designs they created. These six participants rated their satisfaction of the UI designs higher after the discussion than before and at least they were rather satisfied after the discussion. Three participants did not change their satisfaction and one participant was less satisfied after the discussion. This participant explained that missing information in the storyboard influenced their UI designs, and that the UI designs would have been different if he knew this missing information before. More detailed results regarding the satisfaction of the participants are shown in Table 10.4. Furthermore, three participants explicitly remarked that the involvement of the author of a storyboard is desired before any UI designs are created.

In the UI designs, the participants carefully considered the *information regarding devices and personas* that was available in the storyboards. As was concluded in all discussions that involved the authors of the storyboards, the participants interpreted the storyboards correctly. However, in all resulting UI designs there were some missing parts or misinterpretations. The following section describes storyboard-specific results of our study, including most of the missing parts or misconceptions related to the information contained by the storyboards.

10.4.3 Storyboard-specific Results

Some observations were very specific for particular storyboards or the domains that the storyboards were situated in. As most of the authors emphasized in the short interview after discussing the UI designs, they did not consider the differences in interpretations as misconceptions, but rather as shortcomings in the storyboard that caused misinterpretations. The specific differences in interpretations and the authors' suggestions to avoid them are described in this section.

Story-board	How satisfied are you with the resulting UI designs? (1=not satisfied at all, 5=very satisfied)			
	Before the discussion		After the discussion	
	μ	σ	μ	σ
A	3,5	0,71	3,5	0,71
B	4	0	3	1,41
C	3	1,41	3	1,41
D	4	0	2,5	0,71
E	4	0	4,5	0,71
All	3,7	0,67	3,3	1,06

Table 10.4: An overview of the participants' satisfaction regarding the UI designs that were created in average (μ) and standard deviation (σ). Participants were asked to answer to each question using a 5-point Likert scale before and after the discussion with the author.

The aspects that were not clearly interpreted in both UI design sessions of storyboard A concerned the fact that the *locations in the tour had to be searched in a predefined order*. Both participants assumed that the locations had to be searched by the end users in a random order. Furthermore, one participant did not clearly understand that the end users were visiting the museums together with their classmates and teacher. One participant decided to draw UI designs for a *small smartphone screen*, while the other participant drew designs for a larger screen of the size of a *tablet PC's screen*.

Similarly to the sessions of storyboard A, one of the participants that created UI designs for storyboard B was not aware that *the locations were predefined or not*. Furthermore, this participant did not know whether the questions that were needed for the mobile tour, were predefined.

The authors of storyboard A and B both concluded that despite these small differences in interpretations, the participants correctly interpreted the storyboards and the broad context of use of the system. They suggested that this "loss of information" can be avoided by adding more details to the storyboard scenes and their textual descriptions.

In storyboard C, the participants of both UI design sessions were *not aware of the exact constraints of people having multiple sclerosis*. Consequently, the

participants had difficulties to imagine what kind of configurations and games were expected for this storyboard. Furthermore, the *participants did not realize that the haptic device that is used, allows its users to move in three dimensions*, and that the correct positioning of a patient is very important for this system.

Similar observations are resulting from the UI design sessions for storyboard D. The *participants did not have a clear understanding of the difficulties that people with dementia are facing*. Consequently, they were not able to clearly understand the patient's problem that was depicted in the storyboard. Despite the labels of sensing technologies and the annotation of the smartphone in the storyboard, both *participants had to make assumptions about the capabilities of these technologies*.

Both authors of storyboard C and D suggested to avoid the aforementioned misinterpretations by providing more elaborate personas of people having a disease and better descriptions and specifications of the technologies that are used.

For storyboard E, both participants of the UI design sessions succeeded to create UI designs that corresponded to the broad idea behind the storyboard. Nevertheless, the *participants were not aware of the capabilities of the smartphones* that were used. The depiction of *vocal input* of information for the query was not interpreted correctly by the participants. However, in the discussion, the participants and the author admitted that visually representing auditive information in a storyboard is rather difficult. Furthermore, the *participants were not aware that the location of a persona automatically could be detected by the GPS receiver* that was part of the smartphone.

The author of storyboard E suggested to add more information about the multi-modal interaction and the capabilities of the smartphone to the storyboard of this system. Furthermore, it is interesting to consider a visual representation for audio input in a system.

10.5 Discussion and Lessons Learned

In the following, we discuss the results and describe our lessons learned.

10.5.1 Interpreting the Storyboards

All participants broadly understood the whole storyboard they had to interpret and successfully created UI designs that were relevant for the context of use presented by the storyboard. Both the results of the UI design sessions and

the discussions with the authors showed that the annotations of the personas and devices were clearly noticed and carefully considered by the participants. In most situations, the misinterpretations or missing parts in the UI designs were not caused by misunderstanding the storyboard, including the persona and device annotations, but by incomplete descriptions of the storyboard, personas and devices.

10.5.2 Translating the Storyboards into UI Designs

The participants of our study rather used paper storyboards than digital storyboards. Some of the participants preferred paper storyboards over digital storyboards because they could spread out the sheets containing the scenes on the table to have an overview of the storyboard. Other participants browsed through the storyboards scene by scene to create the UI designs that were relevant for particular scenes. Usually, the UI designs were related to a particular scene in the storyboard. One participant explicitly annotated each UI design with the title of the storyboard scene it was intended for, while eight other participants referred to the storyboard during the discussion of the UI designs with the author of the storyboard. Consequently, we can conclude that most participants implicitly or explicitly connected the storyboard to their UI designs. These results are interesting arguments to consider a multi-touch tabletop application that supports storyboards and their connections with other artifacts throughout an entire user-centered project. Most participants of our study did not explicitly relate the storyboard to the UI designs, but during the discussions with the authors, referring to the storyboard proved to be useful to discuss and understand design decisions. Consequently, a multi-touch tool should stimulate team members to include as much connections between artifacts as possible.

10.5.3 Misinterpretations

When considering the misinterpretations in the storyboards, most of these misinterpretations are caused by missing information in the storyboard or personas. Although the use of well-described personas has already proven its benefits, this study showed the need for elaborate personas that accompany a storyboard. Certainly for health care applications, clear and correct personas that describe the constrained skills and the goals of patients are indispensable. We think that elaborate personas and a storyboard are a powerful combination to obtain a common understanding in a multi-disciplinary team.

For people in a team who are responsible for the design and development, it is important that devices and other technology used by a system, are clearly described in a storyboard. We observed that all participants of our study carefully considered the use of a particular device or technology that was annotated in the storyboard. However, only the name of the device did not specify the detailed technical capabilities of a system. For mobile or handheld devices it is important to describe specifications such as the screen size, the availability of a GPS receiver and speech input. When sensing technologies are part of the system, it is important to provide information of what the system can sense and deduce. Furthermore, non-traditional systems such as haptic devices should also be introduced and specified more carefully in a storyboard.

10.5.4 Preferences for Team Meetings

In a large scale applied UCSE process in a company, there is likely a stage between the creation of the storyboard and the creation of UI designs in which other structured models are created. These structured models are also helpful to guide the UI design. As already stated by some participants, in a UCSE project, usually a meeting or discussion with the author of the storyboard can be planned before the UI design takes place. In the most ideal situation, the person who is responsible for the UI designs can be involved during the creation of the storyboard. In such meetings, the storyboard can be built, and the UI designer can request to clarify or complete missing information in the storyboard. However, this study allowed us to observe how a storyboard is interpreted by people that were not involved in this type of meetings. Furthermore, some insights were gained regarding the kind of information that is valuable when storyboards are used at later stages of UCSE processes such as the creation of structured models or UI designs.

10.5.5 Opportunities for Further Research

In this study, we focused on the interpretation and understandability of storyboards. Because the task for the participants consisted of the creation of UI designs, it would be interesting to draw some conclusions regarding this task, based on storyboards. However, we are aware that care has to be taken in generalizing these results for UI design in general. We could observe how a storyboard was used to create UI designs, and we can conclude that the storyboards allowed the participants to consider the context of use that was included in the storyboard during the creation of the UI designs. Nevertheless, other studies are needed to investigate to what extent this context of use

is communicated by the storyboard. For instance, it would be interesting to compare the creation of UI designs when one group of the participants are provided with storyboards and the other group has no storyboards that can support the UI design.

We are aware of the limitations of this study. Although we collected five storyboards to obtain diverse storyboards other cases depicted in a storyboard, or styles used to create storyboard scenes may influence the results of this type of study. Furthermore, the most ideal setup to evaluate COMuICSer storyboards would be its use in a whole UCSE project that involves a multi-disciplinary team.

10.6 Conclusion

In this chapter, we presented a user study to evaluate the notation and the use of annotations in COMuICSer storyboards, presented in Part II of this dissertation. Our study focused on the interpretation and understanding of COMuICSer storyboards. We collected five realistic storyboards, and asked the participants of our study to interpret these storyboards and translate them into UI designs. Finally, the resulting UI designs were discussed with the authors of the storyboards, in order to identify changes in the interpretations of the participants and the authors.

The fact that all ten participants correctly interpreted the storyboards, while most of the participants had no experience in storyboarding, confirms that COMuICSer storyboards are suitable to obtain a common understanding of a system's context of use. The participants of the UI design sessions and the authors of the storyboards had backgrounds in HCI and beyond, including technical as well as non-technical disciplines, which shows that the COMuICSer storyboards allowed them to obtain a similar understanding regarding a system's context of use, and the requirements for the UI design. The annotations proved to be very helpful to show the participants the different personas and devices they needed to consider as part of the context of use of the system. Since the majority of participants referred to the storyboard when they discussed the UI designs with the author of the storyboard, we can conclude that the COMuICSer storyboard has the potential to be a central document in the early as well as later stages of UCSE processes. Consequently, it is helpful to connect a COMuICSer storyboard explicitly to other artifacts created in a UCSE project.

This user study investigated the use and interpretation of COMuICSer storyboards as evaluation of the understandability of storyboards. In Chapter 11

we reflect on other aspects of COMuICSer. Our conclusions regarding this PhD and opportunities for future work are presented in Chapter 12.

Chapter 11

Reflections

11.1 Introduction

Practitioners face several difficulties when they apply user-centered software engineering (UCSE). In Part I of this dissertation we proposed the MuiCSer process framework which helped us to identify the challenges in multi-disciplinary UCSE projects. Part II presented the COMuICSer storyboarding notation and accompanying tool as an approach to overcome these challenges in UCSE.

Our research regarding COMuICSer storyboarding covered several aspects of UCSE. In this chapter we will reflect on the work presented in this dissertation. We critically assess the strengths and weaknesses of COMuICSer and describe the scope of our work as well as opportunities for further research. First, we reflect on the notations of COMuICSer storyboards. Following, our research regarding COMuICSer is considered with respect to the MuiCSer process framework in order to assess to what extent COMuICSer is considered for UCSE processes. Finally, we assess the scope of the COMuICSer tool.

11.2 COMuICSer Storyboards

In this section we assess the notation used for COMuICSer storyboards. Our considerations will cover several aspects of storyboarding.

11.2.1 Structuring Scenarios and Annotating Personas

COMuICSer storyboards were introduced in Chapter 5 as a visual notation with potentials to bridge the early stages of UCSE processes. A storyboard can be obtained by structuring and annotating the scenario and the personas that are created based on the user needs analysis. We assume that the scenario and personas provide very useful information about a system's context of use but can be used more efficiently in another, visual representation. The studies presented in Chapter 6 involve scenarios and personas, which were easily adopted by the participants of our study. Additionally, the user study presented in Chapter 10 showed that the connection of storyboards and personas was easily understood and carefully considered for the creation of UI designs.

In the chapters that followed our introduction to COMuICSer (Chapter 5) and the user studies regarding storyboarding in multi-disciplinary teams (Chapter 6), we built on the assumption that personas and a scenario were available at the moment a storyboard is created. However, we think a storyboard can also be created independently. When the a storyboard is created without the availability of personas or a scenario, it is recommended to carefully consider the end user needs (e.g. based on a report that resulted from a user needs analysis). Furthermore, in that case the creation of the storyboard will likely take more time. In contrast to the situation in which the team thinks out the concept of a system and its context of use for the creation of a scenario and personas, the team has to carry out these tasks during the creation of the storyboard.

The connection of storyboards, scenarios and personas may imply that the need for extensive reports presenting the results of a user needs analysis is decreasing. Because the storyboard contains important information regarding a system's context of use, some information in the reports is redundant. Reducing the effort needed to document the results of a user needs analysis and concretizing these results by creating a storyboard may increase the efficiency of the communication within a multi-disciplinary team. Further research is needed for this, because carelessly reducing documentation in a UCSE process may cause a loss of information.

11.2.2 COMuICSer Storyboards for Multi-disciplinary Teams

Since COMuICSer storyboards are introduced as a communication tool for multi-disciplinary teams that are involved in a UCSE process, we also considered the roles involved in these teams. The ISO 13407 standard [Int99] suggests the involvement of several roles in a team that practices human-

centered design. Interviews with practitioners of UCSE showed that mostly HCI specialists, UI or visual designers, systems analysts and stakeholders (i.e. end users, purchasers and application domain specialists) are involved in a typical project. Consequently, when multi-disciplinary teams were considered in this PhD we concentrated on these roles.

By considering several roles in the user studies presented in this dissertation, we investigated the creation and use of storyboards by different roles involved in UCSE teams. The involvement of several roles and backgrounds is only one aspect when storyboarding is investigated. Since storyboards are used as a communication tool, the collaboration within a multi-disciplinary team also was considered in some studies. Table 11.1 provides an overview of the number of participants with particular roles that were considered in these studies. Furthermore, this table specifies which user studies investigated the collaboration within a multi-disciplinary team.

Study	Roles				Collaborative aspects
	HCI specialist	Designer	Systems Analyst	Stakeholder	
Storyboarding in Multi-disciplinary Teams: First User Study (Chapter 6, Section 6.2)	3	1	3		
Storyboarding in Multi-disciplinary Teams: Observational Study (Chapter 6, Section 6.3)	3	3	3	3	✓
The Visual Storyboarding Language: User Experiment (Chapter 7, Section 7.3)	7			4	✓
Connecting Storyboards and Agile Practices: Interviews + first user study (Chapter 9)			4		
Interpreting Storyboards: User study (Chapter 10)	10	1	4		✓

Table 11.1: The the number of participants with particular roles that were involved in and the presence of collaborative aspects in the studies conducted within the context of the storyboarding research presented in this dissertation.

In three of the five studies several roles were involved and we considered the collaborative aspects. In the first user study regarding storyboarding in multi-disciplinary teams we did not investigate the collaboration between the different roles. However, most of the following studies observed the collaborative aspects. Recruiting different roles in the context of all aspects investigated in this dissertation was not always feasible. For the interviews and the first user study regarding the use of storyboards in agile software engineering only project managers were involved with the background of systems analysts. We are aware that limiting the number of roles in this study also limits the opportunities to draw conclusions for multi-disciplinary teams. Nevertheless, by involving the systems analysts who are less familiar with storyboards and need to make more changes to their own practices than practitioners in user-centered design (UCD), the interviews and the first user study provided interesting results that confirm the usefulness of storyboards in agile SE and the relevance of broader studies in this respect.

We are aware that these studies are limited to only one activity within a UCSE process. In order to fully study and evaluate storyboarding in multi-disciplinary teams, a longitudinal study is needed that involves different roles and allows to observe the collaboration, the communication and the synergy within the multi-disciplinary team while it uses COMuICSer storyboards. Unfortunately, conducting this type of longitudinal study was not feasible within the scope of this PhD.

11.2.3 The Transformation to Formal Artifacts

One of the benefits of storyboards for UCSE is their applicability to create formal artifacts based on informal design knowledge contained by e.g. a scenario and personas. In Chapter 8 we presented several techniques to use storyboards for the creation of formal artifacts including task models, context models and prototypes.

The user study presented in Chapter 10 investigated the understandability of storyboards as well as their translation to UI designs. This study showed us that personas and device annotations are carefully considered for the creation of low-fidelity prototypes by the participants of this user study. In order to investigate what kind of information is used in practice to translate a storyboard into formal models such as task and context models, similar studies are recommended. This type of study was not within the scope of this PhD, but can show to what extent particular information is extracted from a storyboard to create structured interaction models. Similar to our user study (Chapter 10),

several storyboards need to be considered because the results of the study may depend on the case and the style of the storyboard.

One of the limitations of storyboards with respect to structured interaction models is that a storyboard usually considers one possible situation in a system's context of use. While some models support different situations and exceptions for a particular system, a storyboard is rarely used to depict all possible cases. When creating structured interaction models based on a storyboard, the different cases that are not depicted by the storyboard should be considered carefully.

11.2.4 Including Non-functional Requirements

A storyboard can be seen as a requirements document that can combine functional as well as non-functional requirements. Particular information of the storyboard can be used for the creation of formal artifacts, as presented in Chapter 8. In Chapter 9 we described how storyboards can be connected to the user stories used in agile software engineering approaches to collect and represent the requirements from a user's point of view. The first user study presented in Chapter 9 showed the use of storyboards in an agile SE project. One of the limitations, discovered during this study, is the lack of support to depict the back-end services and systems. Further investigations may study the use of additional scenes or models that represent the back-end of a system.

COMuICSer storyboards are suitable to depict non-functional requirements. In order to assess the types of non-functional requirements that can be specified by storyboards, we use the FURPS model, introduced by Hewlett-Packard and considered as a model for functional as well as non-functional aspects of software [CPL09]. The FURPS model represents several software quality attributes, corresponding to the letters of its name. *Functionality* can be depicted by a storyboard. In particular, a set of features and capabilities of software could be implicitly or explicitly included in the scenes of a storyboard. The main attribute of *Usability* that can be included in a storyboard is situated in human factors and may concern for instance several considerations with respect to ergonomics. For *Supportability* the configurability or installability of software systems can also be depicted by a storyboard. The attributes described above, can explicitly be included in the visual representation of a storyboard.

In several cases, the level of *Reliability* (e.g. recoverability and accuracy) and *Performance* (e.g. speed and efficiency, response time) can be deduced implicitly from the storyboard scenes and the context of use. Although these

attributes can be partially depicted in the visual part of a storyboard, COMuICSer provides several features to refer to attributes that are not included in the scene. Annotations and the textual description can be used to refer to particular attributes. Since COMuICSer storyboards can be connected to user stories (Chapter 9), which also are able to include particular requirements that correspond to the FURPS attributes (e.g. *Supportability*: testability, maintainability and portability). Further research, that incorporates COMuICSer storyboards in a complete UCSE project, has to reveal to what extent COMuICSer can include requirements that correspond to the FURPS attributes or attributes related to another framework that considers non-functional requirements.

11.3 COMuICSer Storyboarding and the MuiCSer Process Framework

In this section we assess the extent to which COMuICSer covers the MuiCSer process framework. Furthermore, we describe opportunities for extending the coverage of MuiCSer by using COMuICSer storyboards.

11.3.1 Mapping COMuICSer Storyboards to the MuiCSer Process Framework

As each chapter of Part II covers a certain aspect of storyboarding, we mapped these chapters on the MuiCSer process framework in order to visualize to what extent COMuICSer covers the MuiCSer process framework. Figure 11.1 shows this mapping which visualizes that most of our storyboarding research focuses on the early stages of MuiCSer, which was one of the major aims in our storyboarding research. The following sections describe these mappings and explain how the COMuICSer tool can cover several stages in UCSE.

The definition of COMuICSer storyboarding and the description of the accompanying tool described in Chapter 5 mainly cover the creation of the storyboard based on a scenario and personas, which are created in the first stage of our MuiCSer process framework. Since storyboards provide an alternative notation for representing *Requirements and User Needs*, we tend to situate them in the first stage of the MuiCSer process framework. However, the notation includes facilities to structure these requirements and user needs, so it also covers part of the transition from the first stage to the second stage of MuiCSer, which results in the *Structured Interaction Models*. Furthermore, the possibility to create UI designs based on a COMuICSer storyboard and its

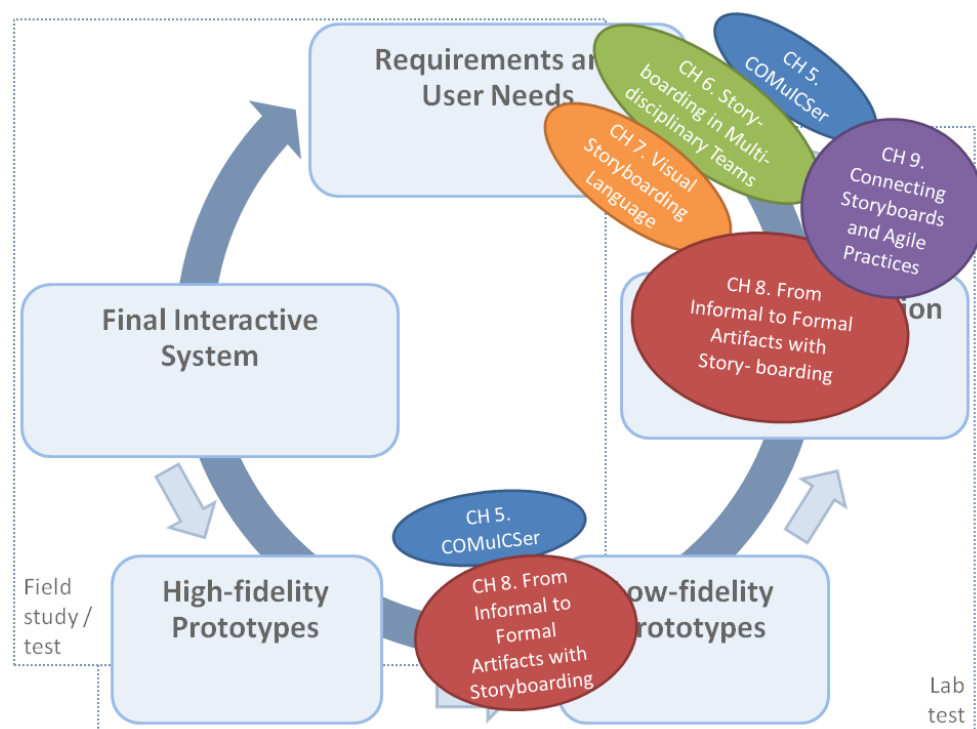


Figure 11.1: An overview of the extent to which our COMuICSer storyboarding research covers the MuiCSer process framework.

annotations, situates COMuICSer between the low-fidelity and high-fidelity prototyping stage. The blue ovals in Figure 11.1 illustrate these mappings.

The studies regarding storyboarding in multi-disciplinary teams, as presented in Chapter 6 can be situated at the end and after the stage *Requirements and User Needs* in MuiCSer. Mainly the involvement of several roles in the user studies showed that storyboards allow for a transition to *Structured Interaction Models*. This is illustrated by the green oval in Figure 11.1.

Likewise, the visual language of storyboarding which was discussed in Chapter 7 can be situated in the MuiCSer process framework between the first and second stage because the COMuICSer notation and accompanying tool support allow inclusion of a comprehensible visualization while providing features in order to structure the *Requirements and User Needs* (Figure 11.1, orange oval).

By considering storyboards to facilitate the transition from informal to formal artifacts, which is presented in Chapter 8, we also consider the transition between the first and second stage of the MuiCSer process framework. Nevertheless, this research relates also to the *Structured Interaction Models* stage and consequently, we situate this work closer to the second stage of MuiCSer, as shown by the red oval in Figure 11.1. Furthermore, this chapter presents how COMuICSer storyboards can be used for prototyping, which resulted in situating part of this work between the *Low-fidelity Prototypes* and the *High-fidelity Prototypes* (Figure 11.1, red ovals).

The last chapter in Part II of this dissertation combines storyboards and agile practices (Chapter 9). Since we focused on the agile practices that are common to specify requirements, this work can also be situated between the first and second stage of MuiCSer (Figure 11.1, purple ovals).

This mapping of our storyboarding research mainly concentrates between the stage where the Requirements and User Needs are elicited, and the stage that results in Structured Interaction Models. Although we had the intention to investigate storyboarding for the early stages of UCSE, we do not exclude the use of these storyboards throughout an entire UCSE project. Storyboarding can be beneficial for later stages of MuiCSer as well. The use of the storyboard can for instance be helpful for evaluating new artifacts created in a MuiCSer process, which will be explained in the following section.

11.3.2 Using COMuICSer Storyboards in Usability Evaluations

Our research regarding storyboarding mainly concentrated on the creation and use of storyboards to facilitate the creation of other artifacts in UCSE processes. By applying storyboards in a UCSE project, we experienced that storyboards can also be beneficial for the evaluation of artifacts.

The process shown in the center of Figure 11.2, which is derived from the MuiCSer process framework, is applied within the context of the AMASS++ project [MBTM08] and was employed to construct novel video information retrieval visualizations for the TV broadcasting domain [HMC09, HML⁺11]. Our storyboard, which was also introduced in Chapter 8, was based on a

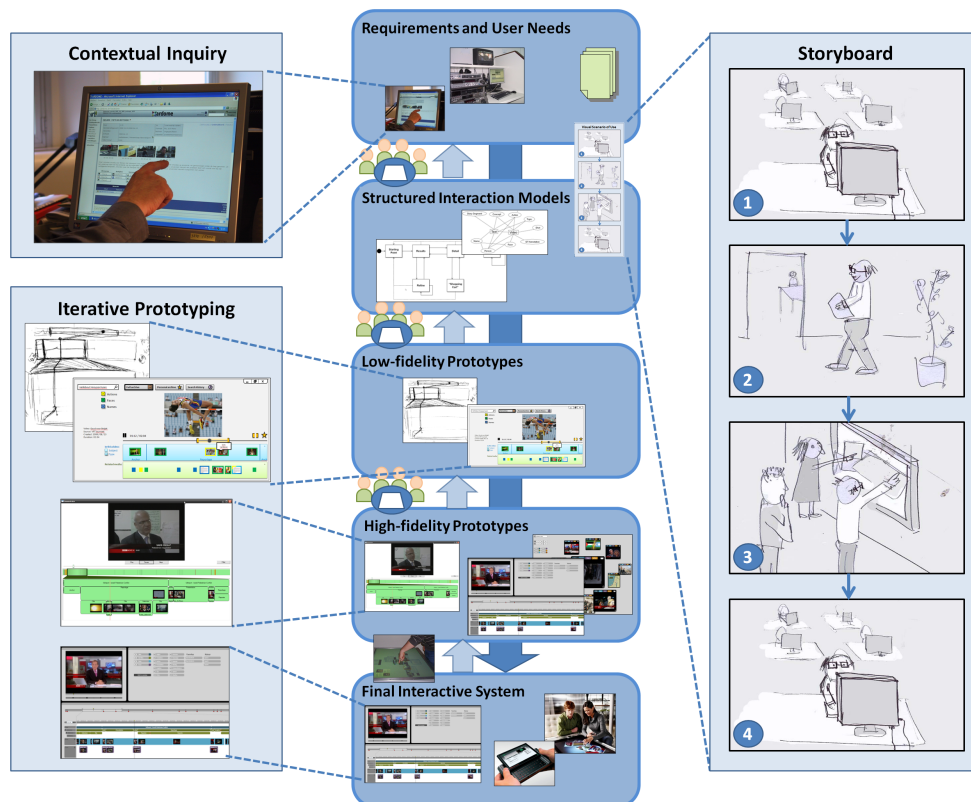


Figure 11.2: The user-centred process that was adopted for the development of novel video information retrieval visualizations for the TV broadcasting domain.

contextual inquiry (Figure 11.2, top left). The storyboard (Figure 11.2, right) exemplifies how the future application can be used for searching archives, browsing an archive video and adding video fragments to a favorites folder.

Several prototypes for different platforms were created in consecutive iterations (Figure 11.2, bottom left). Since it was not possible to conduct field tests of the different prototypes because of practical reasons (e.g. no access to the archive the professionals are using in their everyday job), the storyboard was used in order to inform the participants of the usability tests about the scenario of use, depicted by the storyboard. After the usability test, we presented the storyboard to the subjects and asked in the questionnaire whether a similar application on the multi-touch table was useful in the context of use presented by this storyboard. We observed that all subjects had a clear understanding of the storyboard and could formulate a critical reaction regarding the use of our prototypes in the context depicted by the storyboard.

Similarly, it would be interesting to investigate if the use of storyboards also would be useful in the situation presented in case of the mobile game for children (Chapter 3) In this case, the first prototypes were evaluated in a lab. Several techniques were discussed in this chapter to simulate the locations that were part of the game and to clarify that the final game had to be used while walking in the mining museum or the nature resort. By presenting a storyboard that showed the children walking on the site, this awareness could also be evoked.

Based on the experiences of the AMASS++ project, we see opportunities in future research regarding storyboarding for usability evaluations. Particularly the incorporation of storyboard visualizations in tools that facilitate the evaluation of prototypes and UI designs is an interesting aspect in this context. Denim [NLHL03], Damask [LL08b] and GRIP-it [VSH⁺11] are examples of tools that allow stakeholders and end users to interactively evaluate early prototypes. Combining the visualization of a storyboard with features to evaluate prototypes or UI designs in one tool, may allow end users to consider the context of use while evaluating a prototype. The ActivityDesigner tool [LL08a] provides features for Wizard of Oz testing of UI designs in the tool, while keeping track of the location in the activity model that is related to the UI designs. Although initially, this activity model has a visual representation in this tool, the tool mainly shows textual labels of the activity model when UI designs are evaluated. Since a COMuICSer storyboard can be connected to a UI design created in tools such as Gummy (Chapter 5) and Jelly (Chapter 8), a further extension of these tools that facilitates interactive usability evaluations and Wizard of Oz evaluations of the UI design may

be helpful. While keeping pace with the sequential flow of the COMuICSer storyboard, facilitators or participants are reminded of the context of use.

The extension of these UI design tools with logging features may allow team members to interpret test results (logs) in the correct context, shown by a storyboard scene. Some existing prototyping tools [CCDBR07, dSCDR09, VSH⁺11] already support logging particular actions in usability tests. By providing suitable visualizations of all collected logs in combination with the storyboard (e.g. in a tool like our COMuICSer tool), the multi-disciplinary team may be provided with interesting information for a next iteration in the development of the UI. Allowing team members to add additional test results (e.g. observations) in the visualization may provide a complete overview of all test results related to the context of use. This may be helpful to explain in which context the UI of a system needs to be improved to team members that were not involved in the evaluation.

11.3.3 Assessing Artifacts by Means of COMuICSer Storyboards

Another aspect of storyboarding in UCSE projects is the use of storyboards for the verification of artifacts. If a storyboard is correctly and consistently representing the context of use of a system, it may facilitate the verification of several artifacts' correctness and consistency with the user needs and requirements. Similarly to the last part of our study presented in Chapter 10, a storyboard can be used informally as a tool to assess and discuss the correctness of UI designs. In this study, the authors of the storyboard were involved in this verification. The storyboards were mainly used to argument the systems' context of use with respect to the UI designs that were created.

Formal methods such as cognitive walkthroughs and expert reviews allow team members or external experts to assess prototypes. In these methods the evaluation typically takes place from a user's perspective. Besides using personas as an extension of these methods to increase the awareness of end user needs [PA06], storyboards can be used to explain the context in which a system is used. Earlier work already investigated cognitive walkthroughs supported by videos in order to include a system's context of use [GMKC05]. When creating videos is too time consuming or too difficult, storyboards can be a suitable alternative. Certainly when the storyboard is already available from the first stages, using the storyboard as accompanying document for cognitive walkthroughs may be more efficient.

In analogy to assessing the correctness and consistency of UI designs or

prototypes by using storyboards, the evaluation of other artifacts could also be supported by storyboards. The correctness of formal models such as task models could also be verified within a multi-disciplinary team, in a manner that is comparable to the approach that was taken in the case studies presented in Chapter 3. There, verification of artifacts was conducted by using personas and a scenario. Further studies among which the longitudinal use of storyboards in UCSE projects are needed to investigate to what extent a storyboard can accompany (or replace) personas and a scenario for the verification of artifacts.

11.4 COMuICSer Tool Support

Our research regarding COMuICSer storyboards was not limited to the storyboarding notation and its applicability to MuICSer processes, but also involved the development of a proof of concept storyboarding tool that supports COMuICSer. This tool allows multi-disciplinary teams to create, use and reuse storyboards. The current version of this COMuICSer tool is intended to be used by individuals that are part of a multi-disciplinary team. The creation of a storyboard as well as its reuse is supported by the COMuICSer tool. We describe how the tool can be used in UCSE projects. Furthermore, we assess the COMuICSer tool with respect to the Cognitive Dimensions Framework and present ideas for broader COMuICSer tool support.

11.4.1 Using the COMuICSer Tool in UCSE Projects

A COMuICSer storyboard can be created by one team member individually or in a collaborative meeting. Based on our studies described in Chapter 6, it is likely that an HCI specialist of the team will take the lead in the creation of the storyboard. Sketches can be prepared on paper, which appears to feel more natural than many other digital tools. Once the sketches are created, they can be loaded in the tool. Following, additional information (e.g. annotations and transitions) and objects (e.g. iconic characters) can be included according to the techniques of comics that are applicable to storyboarding, as presented in Chapter 7. During the creation of the storyboard preferably several roles of a multi-disciplinary team are involved in order to obtain input and feedback from different viewpoints.

By creating and digitizing a storyboard in the COMuICSer tool, the storyboard contains information that can guide the structured interaction analysis and (low- and high-fidelity) prototyping that are carried out by systems an-

alysts and UI designers. In Chapter 8 we presented several approaches by which we are able to extract certain information from a COMuICSer storyboard, created in the COMuICSer tool.

When the project involves an agile development team, this team should collaborate from the moment that the COMuICSer storyboard is created or earlier. By involving all team members in a workshop which is intended to create the user stories for the future system using the COMuICSer tool, the connection between non-functional and functional requirements can be maintained in several stages of a UCSE project. In Chapter 9 we described how our tool supports the creation of user stories and how this connection between a storyboard and user stories gives a perspective on the status of a project.

11.4.2 Cognitive Dimensions of the COMuICSer Tool

The ideal situation to evaluate the COMuICSer tool, is by using the tool in a UCSE project during a longitudinal study. Since this study was not within the scope of this PhD, we were limited to evaluating the tool during the user studies presented in Chapters 6 and 9. In order to assess the tool from yet another point of view, we used the Cognitive Dimensions Framework, which was introduced by Green et al. [GP96].

Because the short-term evaluation of new visual languages and programming environments is very difficult, cognitive dimensions were introduced. The framework focuses on activities that are carried out within a visual programming environment, rather than the final environment as such. More recently, the limitations of the cognitive dimensions framework were described by Moody [Moo09]. Olsen’s evaluation framework for user interface systems [Ols07] and Moody’s “The physics of notations” are alternative frameworks to assess notations and tools. However, we opted not to evaluate COMuICSer by using these frameworks because they are developed to evaluate UI design systems and modeling notations respectively.

The COMuICSer storyboarding notation is highly visual and our accompanying tool provides an environment to specify COMuICSer storyboards. Although COMuICSer is not a programming language, some of the dimensions of the cognitive dimensions such as *closeness of mapping*, *hidden dependencies*, *viscosity* and *visibility and juxtaposability* are relevant for the evaluation of COMuICSer and its tool. The remainder of this section presents an overview of our assessment of COMuICSer with respect to the cognitive dimensions framework.

Abstraction gradient can be assessed as abstraction-hating because frag-

ments cannot be encapsulated within the COMuICSer notation. This has positive as well as negative consequences. The notation is easy to learn, but providing an overview of a large storyboard by means of abstractions is more difficult.

One of the most important dimensions for the COMuICSer notation is *closeness of mapping*, which can be rated very high because the problem world can almost directly be mapped to the visual language used in COMuICSer.

With respect to *consistency*, we can conclude that mainly the sequence of scenes and annotations have to be learned by novices. Once they understood this idea, it is very easy to preserve consistency within the storyboard.

The notation used for COMuICSer is rather *terse* for the structure of subsequent scenes. Usually the number of scenes used to depict a system's context of use is rather limited. When a storyboard consists of a large number of scenes, our COMuICSer tool provides a mapview, to maintain the overview of the storyboard. Regarding the content of the storyboard scenes and annotations, COMuICSer can be considered as *diffuse*. One storyboard scene can contain a wide range of drawings and annotations.

The syntax used for COMuICSer is limited, which also influences the *error-proneness*. The likelihood of introducing syntactical errors is small. With respect to the creation of annotations, the user can make mistakes when for instance a device annotation is created instead of a persona annotation. In the current proof of concept tool, this mistake can be corrected by deleting the device annotation and creating a new persona annotation. When a scene is connected to an incorrect sequence of the scenario, this can also be adjusted by changing the description of the scene.

COMuICSer does not force its users to make *hard mental operations*. The highly visual notation of COMuICSer limits the number of hard mental operations that have to be made and does not force its users to add annotations to keep track of what is happening in the storyboard. If necessary, annotations can be made in our COMuICSer storyboarding tool as part of the notation. These annotations can have a predefined structure for personas and devices, but free annotations can be added as well.

The most important cognitive dimension for the COMuICSer tool is *hidden dependencies*. Although the sequential scenes show all dependencies within the storyboard, storyboards may contain several hidden dependencies with respect to other artifacts and annotations. Annotations that refer to one particular persona or device can be contained by several different scenes, but their dependencies are not shown in a COMuICSer storyboard, particularly when this storyboard only is available on paper. Similarly, a storyboard scene can be

connected to a sequence of a narrative scenario, which is not always completely visualized in paper artifacts. Because exactly these hidden dependencies can provide interesting information for a UCSE project (e.g. to show in what scenes a particular persona occurs), it is useful to have a digital environment that supports the creation and use of COMuICSer storyboards. When details of a particular artifact change, cross-references allow that most of these dependencies automatically change accordingly.

In our research we presume that the creation of a COMuICSer storyboard starts from a scenario and personas. Nevertheless, *premature commitment* is limited. For instance, when a scenario is available, sequences of the scenario can be selected every time a new scene is created, but adjusting this connection with the scenario can happen at any time when creating or editing a storyboard. Furthermore, both the COMuICSer notation and tool support the creation of storyboards when no scenario and personas are available.

No *progressive evaluation* of storyboards is provided by the COMuICSer tool because this is not relevant for COMuICSer storyboards. For a future version of the COMuICSer tool, the completeness of annotations may be evaluated to some extent. For instance, when a particular number of personas is included in the tool, and one or more personas not linked to any annotation in the storyboard, the tool could remind its user that these personas are not annotated in the storyboard. Another possibility is using face recognition techniques to recognize the faces of personas and check the correctness of the annotations in the storyboard. However this progressive evaluation is only limited because COMuICSer concerns a highly free and visual notation used for COMuICSer.

The *role-expressiveness* of components of COMuICSer is rather high. The role of each isolated scene, description, label or annotation can easily be seen. In order to distinguish the difference between annotations, we use colors for each type of annotation. Labeling each type of annotation is also a possibility but when one scene contains many annotations, this may result in a very cluttered storyboard scene.

A *secondary notation and escape from formalism* is provided by COMuICSer in the form of annotations. Besides specifying predefined annotations for personas and devices, free annotations can also be added to a COMuICSer storyboard. Nevertheless, the form of annotations (i.e. using a rectangle to tag a particular part of a scene) cannot be changed for this.

In terms of *viscosity* we may conclude that rearranging scenes and other parts of a COMuICSer storyboard is possible in the tool, which is a benefit with respect to storyboarding on paper. However, because the current CO-

MuICSer tool concerns a proof-of-concept tool, it is recommended to provide more support for rearrangements and changes with respect to viscosity.

The *visibility and juxtaposability* of COMuICSer storyboards is supported in several ways. In the tool, the overview of a storyboard can be consulted by zooming out, while zooming in allows a user to see more details of scenes. The mapview provides an overview of the storyboard when the details of one or more scenes are shown. The current version of our tool does not support the possibility to show every pair of scenes simultaneously. Consequently, as we observed in the user study of Chapter 10, a storyboarding tool for a large multi-touch display that supports the free arrangement of slides, would be helpful. This tool would combine the benefits of using pencil and paper, and digitizing COMuICSer storyboards. The next section describes our preliminary ideas about a multi-touch storyboarding tool, based on several aspects that were investigated in this dissertation.

11.4.3 Multi-touch Storyboarding

The current version of our COMuICSer tool allows asynchronous collaboration, and supports several features that can facilitate transitions to other artifacts created in UCSE. However, the tool is mainly intended for individual use. Collocated collaboration using the COMuICSer tool is possible if one person of the team takes the lead during the creation of the storyboard. However, the creative aspects of storyboarding in collocated teams should be supported by a storyboarding tool that allows fully collocated collaboration. Current technologies allow teams to gather around digital tabletops to collaborate, so we think that a multi-touch tabletop system could be a suitable platform for this type of tool.

From the perspective of agile development and requirements engineering, several tools are available for interactive tabletop systems. Ghanam et al. [GWM08] proposed a digital tool for agile planning meetings in which user stories are created or discussed, while Maiden and Jones [MJ10] propose examples for interactive table systems that support communication and collaboration about functional and non-functional requirements. However, both of the aforementioned systems do not explicitly support storyboarding. Storify [AM11] is a tool that is being developed to allow designers to create storyboards. The Storify tool intends to be used in two modes, an individual mode and a collaborative mode. For the latter, a tabletop system is considered. Although this system supports storyboarding in order to take into account the user experience of a future system, it does not consider the different roles of a

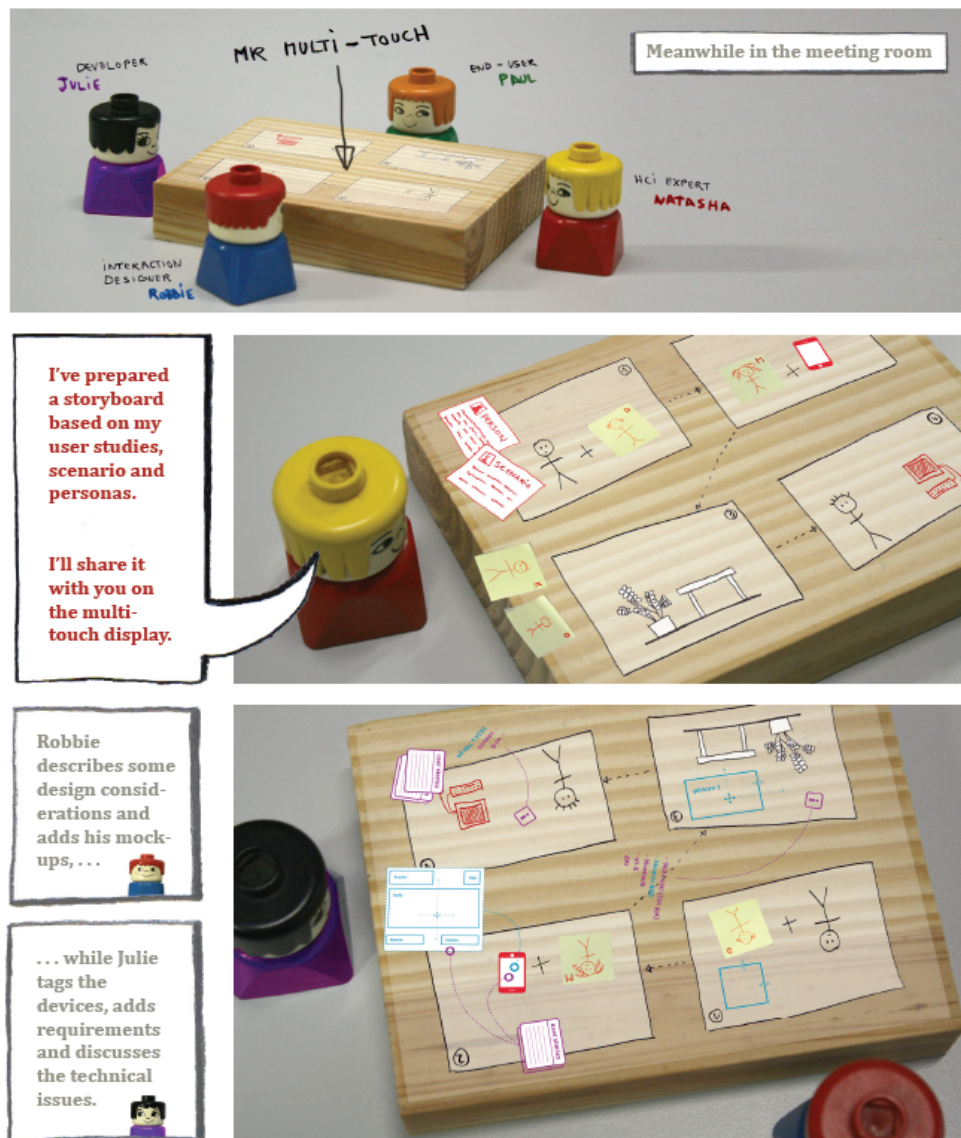


Figure 11.3: First part of a storyboard presenting how a multi-disciplinary team creates / discusses a COMuICSer storyboard in a meeting. The tool presented in this storyboard supports collocated storyboarding workshops.

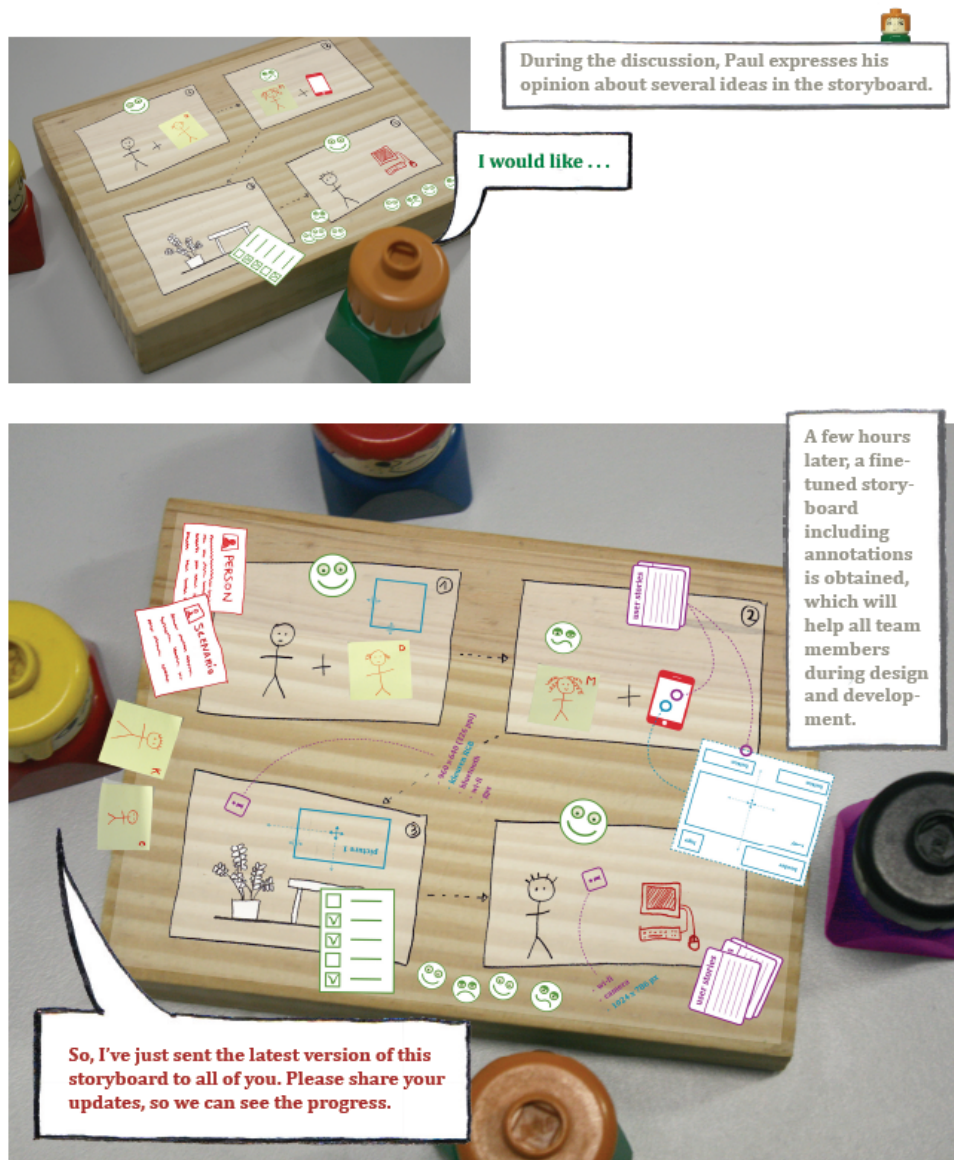


Figure 11.4: Second part of a storyboard presenting how a multi-disciplinary team creates / discusses a COMuICSer storyboard in a meeting. The colors shown on the tabletop display, indicate the different contributions of the different members of a multi-disciplinary team.

multi-disciplinary team.

In order to support collocated collaboration that allows the involvement and contribution of all team members of a multi-disciplinary team, the results of our observational study presented in Chapter 6 have to be considered carefully. The intended tool should allow for differences in backgrounds involved, but support agreements. As in our study the representation, style and viewpoints differed greatly and the team members of a multi-disciplinary team are already accustomed to specific tools and devices, the aimed tool should not try to enforce one particular way of preparing a storyboard. Instead, the tabletop system should be able to import and possibly connect several kinds of artifacts. A concept for this type of tool is represented by the storyboard shown in Figure 11.3 and Figure 11.4 [HRLC11].

Supporting a combination of digital and paper artifacts, which recently has been investigated in a general context by Hartmann et al. [HMBW10], can be considered as well. Including personal devices that allow to share artifacts at the digital tabletop system can reduce the problem of limited screen space, because they can act as personal workspaces.

11.5 Conclusion

This chapter presented our reflections with respect to the COMuICSer storyboarding approach and tool. By assessing the COMuICSer notation and tool as well as the mappings of COMuICSer with respect to the MuiCSer process framework, we could identify the strengths and weaknesses of our research. Chapter 12 presents an overview of the possible directions for future work and the conclusions of this PhD.

Chapter 12

Conclusions and Future Work

To conclude this dissertation, we present a summary of this PhD and an overview of possible directions for future work and our scientific contributions and publications.

12.1 Summary

In this dissertation we investigated several processes and techniques in order to support the user-centered design and development of interactive systems. Since the involvement of multi-disciplinary teams is of major importance in user-centered approaches, all our work concentrates on teams that include different disciplinary backgrounds.

The first part of this dissertation introduced our process framework for multi-disciplinary user-centered software engineering (MuiCSer). The framework first of all was intended to derive user-centered processes for applied research. Furthermore, this framework was used to investigate current user-centered software engineering (UCSE) approaches.

A literature survey of MuiCSer processes, artifacts and tools for UCSE led to an identification of weaknesses and difficulties of UCSE. A complementary study of UCSE in practice that involved our experiences of applied MuiCSer projects and interviews with UCSE practitioners, added findings and confirmed our assumptions based on the literature survey. A list of the weaknesses and difficulties is shown in the first column of Table 12.1.

We introduced COMuICSer storyboarding and accompanying tool support

Weaknesses and Difficulties in UCSE (Chapter 2, 3 and 4)	Research Challenges in Storyboarding (Chapter 5)	Research Questions (Chapter 5)
Notations and tools that support the collaboration in multi-disciplinary UCSE teams are lacking	<i>RC A:</i> Storyboarding for multi-disciplinary teams	<i>RQ 1a:</i> Are storyboards and accompanying tool support useful for UCSE practitioners? (Chapter 6)
		<i>RQ 1b:</i> How are storyboards created in a multi-disciplinary team? (Chapter 6)
Notations and tools that support the transition from informal artifacts into formal models are lacking	<i>RC B:</i> A visual storyboarding language	<i>RQ 2:</i> What aspects of a visual language can contribute to storyboards in UCSE? (Chapter 7)
		<i>RQ 3:</i> How can storyboards be used for the transformation from informal to formal artifacts? (Chapter 8)
Notations that incorporate all user needs and requirements are lacking	<i>RC D:</i> Storyboarding to connect UCD and SE	<i>RQ 4:</i> How can storyboards be connected with software engineering artifacts? (Chapter 9)

Table 12.1: Overview of problems we identified in UCSE and the research challenges and questions for storyboarding that can be related to these weaknesses and difficulties.

as a possible approach to answer the problems in UCSE. Based on our definition of COMuICSer, we listed several research challenges that can be related to the weaknesses and difficulties we identified in UCSE. Consequently, these challenges were concretized by formulating more detailed research questions. The second and third column of Table 12.1 show these research challenges and research questions. Each research question was investigated in detail in this dissertation.

In two user studies we further investigated questions regarding storyboarding in multi-disciplinary teams. A first user study provided insights regarding the usage of storyboards by UCSE practitioners. Furthermore, an observational study revealed how multi-disciplinary teams create storyboards collaboratively. Both studies showed the relevance of storyboarding in multi-disciplinary teams. On the one hand, storyboarding is beneficial to obtain a general understanding within the team. On the other hand, storyboards are considered as helpful tools for the communication in the team.

Techniques used for expressing messages in comics were investigated in order to obtain insights of the aspects of a visual that can be considered for COMuICSer storyboarding. Additionally, a user experiment was conducted to evaluate our findings in the comparison of comics and storyboarding. Based on both our comparison and the user experiment, we presented an extension of our COMuICSer tool that supports several principles of comics. The features of our tool related to comics, facilitate UCSE practitioners to provoke empathy with the actors of the storyboard and to provide information that is helpful for later stages in a UCSE process.

With the later stages in UCSE in mind, we studied how storyboards can support transitions from informal to formal artifacts. The creation of a storyboard meta-model allowed us to specify the information that can be extracted from a storyboard in order to obtain formal artifacts. We introduced mechanisms to map storyboards to task or context models and tool support to use storyboards for the design of high-fidelity prototypes. Both approaches make use of storyboards in order to consider contextual information of the future interactive system as much as possible.

Besides studying the relationship between storyboards and formal models of UCSE, we also investigated how storyboards can be connected with artifacts that are used in software engineering. We investigated the connection of COMuICSer storyboards with user stories, used in agile software engineering. This technique was inspired by a focused literature study and semi-structured interviews with agile SE practitioners. Furthermore, we extended our proof of concept tool for COMuICSer storyboarding with features to create user stories

that are related to the scenes of a storyboard. This extension of our tool was evaluated in a first user study.

In the third and last part of this dissertation we assessed the COMuICSer approach and tool.

12.2 Answers to the Research Questions

In Part II of this dissertation we investigated storyboarding in order to answer the research questions that were introduced in Chapter 5. Below, we summarize the answers to these questions based on the research described in this dissertation.

12.2.1 RQ 1a: Are Storyboards and Accompanying Tool Support Useful for UCSE Practitioners?

Our first user study described in Chapter 6 showed that UCD practitioners are interested in the use of storyboards in multi-disciplinary teams [HLC09]. The participants of our study confirmed that the storyboard could be used by team members having different backgrounds for several design, development and evaluation activities. Furthermore, they agreed that digitizing the storyboard in a tool and adding annotations would be useful in user-centered approaches. According to the participants, their work could benefit from storyboards because the notation provides rather explicit information regarding users' activities which can be used for later stages in UCSE.

During the study, we observed that tool support would be useful for several types of storyboarding activities, ranging from creating storyboards to sharing and presenting them. Supporting these activities in a storyboarding tool, allows members of a multi-disciplinary team to edit, archive and reuse storyboards during an entire UCSE process. Although this was a first study that informed and consulted UCD practitioners about storyboarding, further studies presented in this dissertation confirmed these findings. However, the usefulness of COMuICSer storyboards and the accompanying tool can be further evaluated in a longitudinal study.

12.2.2 RQ 1b: How are Storyboards Created in a Multi-disciplinary Team?

The observational study that was conducted in order to investigate this research question, confirmed that a highly creative activity such as storyboarding

is not easily generalizable. However, as we presented in Chapter 6, we observed that the different roles and disciplines involved in a multi-disciplinary team lead to different contributions to a storyboard. During the preparation phase, we observed several team members preparing complementary artifacts to be used in or connected to the storyboard that was created afterwards. Usually HCI specialists took the lead in the creation of the storyboard. All the other team members were actively involved in the discussion that took place before and during the creation of the storyboard, but they did not all participate in the creation of the storyboard.

Multi-disciplinary teams may structure storyboards in different ways. In our study for example, two storyboards consisted of scenes that were ordered chronologically, and one storyboard was structured spatially. Furthermore, we observed that the teams used images that were provided in the beginning of the study. We are aware that observing three teams that create storyboards in an observational study will not show us all possible approaches taken by multi-disciplinary teams for the creation of storyboards. However, this study provided some interesting insights that can be used for the informed design of a collaborative storyboarding tool as presented in Chapter 11. Based on these findings, the tool can be implemented and evaluated by multi-disciplinary teams in future work, to search for more insights in the way storyboards can be created in multi-disciplinary teams.

12.2.3 RQ 2: What Aspects of a Visual Language can Contribute to Storyboards in UCSE?

We investigated how the principles of the language of comics could be incorporated in the storyboarding notation [HMLC10]. This study, which was described in Chapter 7, showed that the vocabulary used in comics is similar to the vocabulary of storyboards. The use of facial expressions and body language that is typical in comics, is also applicable to storyboards and may help team members to empathize with characters or personas. The other techniques used in comics that combine iconic characters and realistic backgrounds also allows team members to identify and empathize with end users. By differentiating characters, the different personas that are involved in a system's scenario of use are emphasized in the storyboard. Besides the elements that are part of the scenes of a storyboard, the transitions between storyboard scenes may also provide information to team members. For instance, the time that passes between two scenes can be represented by these transitions.

We incorporated some of the techniques of comics in our COMuICSer tool.

By featuring several techniques that originate from comics, team members can easily label the transitions between storyboards and the resulting storyboards are more likely to contain elements that cause team members to carefully consider the context of use and empathize with end users.

12.2.4 RQ 3: How can Storyboards be Used for the Transformation from Informal to Formal Artifacts?

By creating a storyboard meta-model, we structured the information contained by a storyboard [LHO⁺10, HdBM⁺11]. This meta-model was used as a first step towards the transformation based on COMuICSer storyboards. We presented two model transformations that incorporate information of a COMuICSer storyboard in a task model and a context model. Furthermore, by supporting the use of COMuICSer storyboards during the creation of high-fidelity prototypes, a system's user requirements and contexts of use can be considered during the creation of UI designs [HdBM⁺11].

In Chapter 8, we focused our investigation on transformations to some formal artifacts that contain informal information that is part of storyboards. Based on these results, we believe that this approach can be extended as other formal artifacts may incorporate information contained by storyboards.

12.2.5 RQ 4: How can Storyboards be Connected with Software Engineering Artifacts?

In Chapter 9, we investigated the connection of storyboards and artifacts used in agile software engineering. Based on an interview with practitioners in agile software engineering, we proposed a technique that connects user stories with storyboards. This connection was also included in an extension of our COMuICSer tool and forces team members to consider a correct context of use and the personas of a storyboard when the user stories of a system are created.

Connecting artifacts of UCD with artifacts used in agile software engineering may encourage multi-disciplinary teams to consider contextual information and non-functional requirements in combination with functional requirements. We are aware that user stories are one type of artifact that is used in software engineering. In order to complete the answer to this research question, future research may investigate the connection of storyboards and other artifacts used in software engineering.

Although all research questions were carefully investigated in this dissertation, some questions were not completely answered. Consequently, in the

next section we propose possible directions for future work that can complete the answers to our research questions. Furthermore, this future work can contribute to extended support for storyboarding in multi-disciplinary teams.

12.3 Future Work

In Chapter 11 and Section 12.2 we discussed the scope of our work and introduced some possible directions for future research. In this section we present an overview of these opportunities for future work.

12.3.1 Extended Tool Support for COMuICSer Storyboards

Several directions for future research related to this PhD lead to extended tool support for COMuICSer storyboards. By using storyboards for the evaluation of prototypes, usability testing in a lab environment can be supported by a visual description that informs the participants about the context of use in which the prototypes will be used. The ActivityDesigner tool [LL08a] already supports features to conduct a usability test while keeping track of an activity model created in the first stages of a project. However, we believe that a visual notation as used in COMuICSer may be more easily understood by the people involved during the test.

Linking the test results that were logged during the usability tests with the specifications that depict the context in which the test results were collected is another feature that may facilitate the evaluation of user interfaces. By providing this information in a suitable visualizations, other team members can consult the test results related to the context in which they were collected. Existing prototyping tools already support logging during usability tests [CCDBR07, dSCDR09, VSH⁺11], but visualizations that combine the test results with storyboards may increase the understanding of team members of problems in the UI and may provide them with contextual information that may help them to find possible solutions.

As the creation of storyboards preferably takes place in a collocated collaborative team meeting, suitable tool support is desirable. An interesting platform for this type of tool is a large multi-touch display that allows to explore storyboards in a manner that is comparable with using paper artifacts. The size of the display should allow freedom to arrange and explore storyboards and related artifacts. The multi-touch features of the platform are useful to support the collaboration during the meeting.

Since storyboards should contain more details and requirements based on several perspectives, the contributions of the team members having different roles should also be supported and even stimulated in the tool. Further investigations of the typical contributions to the storyboard of the roles involved in a multi-disciplinary team, will provide more information with respect to possible collaborative patterns that could be facilitated by a multi-touch storyboarding tool. The Storify tool [AM11] is available for collaborative storyboarding, but in this tool the different roles involved in a multi-disciplinary team are not considered individually.

12.3.2 Longitudinal User Studies

Although we evaluated several parts of our work in user studies, a thorough use of COMuICSer storyboards in longitudinal UCSE projects from beginning to end will show to what extent the storyboarding techniques presented in this dissertation are beneficial for the collaboration within a multi-disciplinary UCSE team. Furthermore, we think that conducting this type of study in the context of a MuiCSer project will provide us with more details regarding remaining challenges for storyboarding in UCSE. Research questions that may be addressed are:

- Can COMuICSer storyboards replace part of the information contained by documents that typically are created in the first stage of a MuiCSer process?
- How will the different roles in a multi-disciplinary team contribute to and use the storyboard and what are the effects on the communication and collaboration within the team?
- What types of non-functional requirements can be contained by storyboards and what requirements depicted by the storyboard are exactly considered in the creation of other artifacts?
- Can storyboards contribute to the verification of artifacts created during a MuiCSer process? If so, what kind of information, annotation, component of a storyboard can highlight the essential elements of an artifact?

Because we are aware of the fact that the type of project, the team involved and other factors may influence the answers to these questions, it is recommended to study storyboarding in several longitudinal studies with varying parameters.

12.4 Scientific Contributions and Publications

The research presented in this dissertation was published in several articles and a book chapter. The following overview presents the most important publications which directly contributed to this dissertation.

[HCdBL08] Mieke Haesen, Karin Coninx, Jan Van den Bergh and Kris Luyten. MuiCSer: A Process Framework for Multi-disciplinary User-Centered Software Engineering Processes. *In Proceedings of 2nd Conference on Human-Centred Software Engineering (HCSE 2008), Pisa, Italy, September 2008.*

This paper describes the MuiCSer process framework and our research regarding existing UCSE approaches and UCSE in practice.

[HLC09] Mieke Haesen, Kris Luyten and Karin Coninx. Get your Requirements Straight: Storyboarding Revisited. *In Proceedings of Interact 2009, Uppsala, Sweden, August 2009.*

This paper describes COMuICSer storyboarding and our first proof of concept tool to facilitate storyboarding.

[HMLC10] Mieke Haesen, Jan Meskens, Kris Luyten, Karin Coninx. Draw Me a Storyboard: Incorporating Principles and Techniques of Comics to Ease Communication and Artefact Creation in User-Centred Design. *In the 24th BCS Conference on Human Computer Interaction (HCI2010), Dundee, UK, September 2010.*

This paper describes our study regarding the visual comics language and the incorporation of principles of comics in our COMuICSer storyboarding approach and accompanying tool support.

[HdBM⁺11] Mieke Haesen, Jan Van den Bergh, Jan Meskens, Kris Luyten, Sylvain Degrandart, Serge Demeyer, Karin Coninx. Using Storyboards to Integrate Models and Informal Design Knowledge. *Model-Driven Development of Advanced User Interface, Studies in Computational Intelligence, Vol. 340, 1st Edition, 2011.*

This paper describes our work regarding the use of storyboards to support the transition from informal to formal artifacts.

Besides these publications, several other articles and a book chapter, related to the work presented in this dissertation were published in the past four years. The following list presents these publications.

[**HLC⁺08**] Mieke Haesen , Kris Luyten , Karin Coninx , Jan Van den Bergh and Chris Raymaekers. MuiCSer: A Multi-Disciplinary User-Centered Software Engineering Process to Increase the Overall User Experience. *In Proceedings of 10th International Conference on Enterprise Information Systems (ICEIS 2008), Barcelona, Spain, June 2008.*

[**VdBHL⁺08**] Jan Van den Bergh, Mieke Haesen, Kris Luyten, Sofie Note-laers and Karin Coninx. Toward Multi-disciplinary Model-Based (Re)Design of Sustainable User Interfaces. *In Proceedings of Design, Specification and Verification of Interactive Systems (DSV-IS 2008), Kingston, Canada, 2008.*

[**MHC08**] Mieke Haesen, Chris Raymaekers and Karin Coninx. Evaluating a location-based mobile game in early stages of the development. *In Proceedings of 6th International Conference on Methods and Techniques in Behavioral Research (Measuring Behavior 2008), Maastricht, The Netherlands, August 2008.*

[**MHLC09**] Jan Meskens, Mieke Haesen, Kris Luyten and Karin Coninx. User-Centered Adaptation of User Interfaces for Heterogeneous Environments. *Advances in Semantic Media Adaptation and Personalization, volume 2, March 2009.*

[**HDBCR09**] Mieke Haesen, Joan De Boeck, Karin Coninx and Chris Raymaekers. An Interactive Coal Mine Museum Visit: Prototyping the User Experience. *In Proceedings of 2nd conference on Human System Interaction (HSI 2009), Catania, Italy , May 2009.*

[**HMLC09**] Mieke Haesen, Jan Meskens, Kris Luyten and Karin Coninx. Supporting Multidisciplinary Teams and Early Design Stages Using Storyboards. *In Proceedings of 13th International Conference on Human-Computer Interaction, San Diego, CA, USA, July 2009.*

[**HMC09**] Mieke Haesen, Jan Meskens and Karin Coninx. Visualising Digital Video Libraries for TV Broadcasting Industry: A User-Centred Approach. *In Proceedings of 3rd Workshop on Human-Computer Interaction and Information Retrieval (HCIR 2009), Washington DC, USA, October 2009.*

[**LHO⁺10**] Kris Luyten, Mieke Haesen, Dawid Ostrowski, Karin Coninx, Sylvain Degrandart, Serge Demeyer. On Stories, Models and Notations: Storyboard Creation as an Entry Point for Model-Based Interface Development with UsiXML. *In User Interface eXtensible Markup Language Workshop at The ACM SIGCHI Symposium on Engineering Interactive Computing Systems 2010 (EICS 2010), Berlin, Germany, June 2010.*

[**HML⁺11**] Mieke Haesen, Jan Meskens, Kris Luyten, Karin Coninx, Jan Hendrik Becker, Tinne Tuytelaars, Gert-Jan Poulisse, Phi The Pham, Marie-Francine Moens. Finding a Needle in a Haystack: an Interactive Video Archive Explorer for Professional Video Searchers. *Multimedia Tools and Applications - Special Issue on Automated Information Extraction in Media, 2011.*

[**HRLC11**] Mieke Haesen, Karel Robert, Kris Luyten, Karin Coninx. Multi-touch Storyboarding Tool Concept for User-Centred Design Teams. *Create 2011 Symposium, London, UK, June 2011.*

Appendices

Appendix A

Documents of User Studies and Interviews

In this appendix the documents that were used for the user studies and semi-structured interviews are presented. These documents include questions for semi-structured interviews, personas, scenarios and questionnaires.

A.1 Semi-structured Interviews with UCSE Practitioners

The questions that were asked during the semi-structured interviews with UCSE practitioners in Chapter 3, Section 3.3:

- What models and artifacts do you use in a UCSE process?
- What is the background of a typical designer?
- What tools do you use in a UCSE project?
- How is input from the customer and end users collected? What notation and process are used for this?
- What roles suggested by ISO 13407 does your team include ?
 - end user?
 - purchaser, manager of user?
 - application domain specialist, business analyst?

- systems analyst, systems engineer, programmer?
 - marketer, salesperson?
 - user interface designer, visual designer?
 - human factors and ergonomics expert, human-computer interaction specialist?
 - technical author, trainer and support personnel?
- How is feedback of end users and customers obtained? Do you only conduct usability tests, or are other techniques used for this?

A.2 First User Study

The documents that were used for the first user study described in Chapter 6 Section 6.2.

A.2.1 Personas

Mary, 49 years, owner of a bistro (customer) (Figure A.1) Mary owns a bistro for about two years. Since she bought the bistro, her company started growing and now she employs five cooks and fourteen waiters. Although she trusts her personnel, she is always in the bistro to keep an eye on everything and to solve problems. Managing this bistro is a nice but busy job.



Figure A.1: Photograph of Mary, persona for an owner of a bistro, used for the first user study presented in Chapter 6 Section 6.2.

Currently she noticed that the growth of her company causes some difficulties in organizing and setting tasks, so she decided to buy a new computer system. She wants that the waiters can take orders using a PDA, and then the order can be sent to the kitchen where dishes can be prepared. The bill should be generated and she wants to control the food stock using a touch screen of a traditional screen size.

Mary wants to provide a nice work climate to her employees and thinks this system should fit into the company. Using the new system should not cause extra tension, but it should ease activities for the personnel. She heard of user-centered design and thinks a personalized system for her bistro can improve efficiency and work climate.

Ann, 39 years, Usability engineer / Project manager, in a relationship (Figure A.2) Ann is working for a company that designs and develops systems in a user-centered way. For six years, she works for her current employer and since one year she manages projects herself. She is experienced in user-centered design approaches and was responsible for the UI design of more than 20 projects. Although her company is specialized in user-centered design, there are still some difficulties in multidisciplinary cooperation with her colleagues.



Figure A.2: Photograph of Ann, persona for a usability engineer, used for the first user study presented in Chapter 6 Section 6.2.

As a usability engineer, she is convinced of the benefits of a multidisciplinary cooperation, so during a project she tries to communicate as much as possible with her colleagues. She noticed that each discipline has its own needs to communicate and depending on the project and the team members, she creates artifacts that seem the most suitable. She likes experimenting with new tools and techniques and can use the experience of the last six years to

know what type of artifact should be used in a particular situation.

Besides communicating with colleagues, communication with end users and customers is also very important. At the beginning of a project, Ann organizes a meeting with the customer, and if possible, she carries out a contextual inquiry to get a realistic idea of the user requirements. Processing and communicating these user requirements is the most crucial part during the development cycle.

Because of her interest for user-centered design, she visits some conferences to learn about the the most recent studies and approaches in this field. She likes her job because she can meet several people and can cooperate with colleagues that each provide complementary contributions to a project.

Thomas, 35 years, Developer, married, two children (Figure A.3)

Thomas is working for a company that designs and develops systems in a user-centered way. He is a developer since he graduated, and for the last years he has learned a lot about user-centered design. It took some time to get used to the input of usability engineers. Sometimes the requirements specifications were very vague and the first developed prototypes raised questions and arguments with colleagues. This caused a lot of tension and Thomas had to work overtime almost every day.



Figure A.3: Photograph of Thomas, persona for a developer, used for the first user study presented in Chapter 6 Section 6.2.

Together with his colleagues, Thomas organized some workshops to discuss the needs of all team members and looked for suitable tools that could be

used by the entire team. All colleagues agreed that user-centered design is not determined by tools, but by techniques and knowledge. However, tools that support several stages of user-centered design can contribute to a good traceability and visibility during a project.

The colleagues decided to involve Thomas during brainstorm sessions. Now, Thomas participates in meetings at several stages of the user-centered design process and he can discuss some technical issues before the UI design has started. He likes this new approach because his colleagues understand his needs and the technical difficulties are considered before it is too expensive to change UI designs. Actually, attending these meetings is an investment in time that finally saves total time for a project. Nowadays, Thomas can go home before dinnertime and can spend some time with his wife and children.

Michael, 23, Graphic Designer, in a relationship (Figure A.4) Michael is a graphic designer and works for a company that designs and develops systems in a user-centered way. He likes to contribute to UI designs and wants to add a personal touch to each product his company develops. All components of a UI design need to be original and adapted to the domain they are used in.



Figure A.4: Photograph of Michael, persona for a graphic designer, used for the first user study presented in Chapter 6 Section 6.2.

Sometimes, Michael regrets that he has to take into account the design style of the customer or a company logo. But usually he can find a compromise that suits the customer. After all, customers ask for a unique design, and they need his creativity.

When Michael is thinking about a design, he does not want to be disturbed. He uses several tools and off course he cannot miss pencil and paper during

design. Usually the UI designs are prepared by a colleague, but when adapting the UI designs after a first evaluation, Michael cooperates with the UI designer. By doing this, Michael can consider things that are necessary for the graphic design of UI components.

A.2.2 Scenario

A new project has started and concerns the development of a touch screen application for a bistro. All employees need to use this application to take orders, prepare dishes and make the bill. Mary is the owner of the bistro and contacted the company Ann is working for, to develop the touch screen application she needs.

Ann organizes a meeting with Mary and lists the global requirements of the system. The system should increase efficiency and decrease mistakes. Ann understands the needs of the bistro, but since each organization has different approaches and needs, she decides to carry out a contextual inquiry at the bistro. During three days, Ann goes to the bistro and observes and interviews all employees to learn the approach of taking orders, handing these orders over to the cooks, making bills and keeping an inventory of the food stock. Especially the cooperation between all employees will influence the user needs for the new system.

As soon as Ann gets back to the office, she starts reading her notes and writing down her first ideas for the system. That results in two personas, a scenario and a storyboard. Once these artifacts are created, she calls Mary to plan a meeting together with her colleagues. The personas and scenario are created in a word processing tool, while the storyboard is created on paper, followed by adding the personas, scenario and storyboard in the Gummy tool. Besides these artifacts, Ann also creates some sample content for the future system.

During the meeting, Ann introduces the personas to Mary, Thomas and Michael. Following, she starts discussing the storyboard in the Gummy tool by projecting it on a large screen. First the course of the storyboard is described scene by scene. Sometimes Mary comes up with some extra information regarding special situations in the bistro. It also happens that Thomas comes up with some technical remarks. For each comment, Ann adds an annotation to the corresponding scene. Next, the sample content is discussed with Mary. At the end of the meeting, all attendees agree that the storyboard should be refined, so Ann and Michael plan to do an iteration of the storyboard.

Once the storyboard is adjusted, Ann discusses the changes with Thomas

and sends a digital file containing the storyboards to Mary. Within an hour, Mary calls Ann. She has shown the storyboard to her employees, and they agree with the way the future application should be used. They cannot wait to start using this system because the current approach in the bistro causes mistakes and a lot of tension. Ann and Mary decide that the UI design can start.

Together with Michael, Ann starts designing the UI. While Ann creates some first prototypes of the UI, Michael picks color schemes and starts designing graphical components for the UI. Concurrently, Thomas starts transforming the storyboard into a flow diagram that can be used to guide prototyping and developing the application. He uses the metadata of the storyboard and the sample content to see what data needs to be available on each device. As soon as this diagram is finished, he discusses it with Ann. Together, they adjust a few things in the diagram and decide to start designing and developing based on this chart.

Ann continues designing the UI and iterates on this design after some meetings with Michael and Mary. As soon as the designs can show how the system can be used, Ann goes to the bistro and evaluates the design together with the employees. The storyboarding tool, and the metadata containing persona information helps Ann to show the suitable designs for each employee. Since two employees are on vacation, Ann promises Mary to e-mail her a file containing the UI designs. This file plays the scenes of the storyboard together with the UI designs.

After another iteration, Ann decides to start the high-fidelity prototyping. Michael transforms Ann's UI designs into more detailed prototypes, while Thomas starts developing the database and the application logic for the touch screen application. As soon as the first results are available, Thomas starts integrating the designs into the code and develops a testable prototype.

Ann goes back to the bistro and asks some employees to test the prototype. Similar to the first evaluation of the UI designs, she uses the storyboard to make sure all roles at the bistro participated in the test. Based on a list with remarks, Michael and Thomas adjust the prototype and continue designing and developing the touch screen application. The tasks of testing and developing are done in a few iterations until the final product is delivered to Mary.

A.2.3 Questionnaire

Pre-test

Gender: Female / Male **Age:** years
Education: **Current job:**

Do you have any experience being part of a multi-disciplinary team? Yes / No

For how long did / do you work in a multi-disciplinary team?

- ☐ Few months
- ☐ More than a year
- ☐ More than 5 years

Was a multidisciplinary team beneficial for the project(s)?

.....
.....
.....

What were the difficulties?

.....
.....
.....

Figure A.5: Questionnaire used for the first user study presented in Chapter 6
Section 6.2

Creating a storyboard

How difficult / easy was it to add scenes to the storyboard?

Very easy	Easy	Neutral	Difficult	Very difficult

How difficult / easy was it to add metadata (personas, devices, annotations) to the storyboard?

Very easy	Easy	Neutral	Difficult	Very difficult

Was it clear how personas and devices and annotations were linked to the storyboard?

Very clear	Clear	Neutral	Not clear	Not clear at all

How difficult / easy was it to go on with the design for the user interface of the laptop?

Very easy	Easy	Neutral	Difficult	Very difficult

Rank the following tools that can be used for storyboarding (1= most suitable; 3= least suitable):

- Pencil and paper
- Powerpoint
- The tool used in this experiment
- Other:

Explain your choice:

.....

.....

.....

Storyboard walkthrough

What persona was assigned to you?

Do you think your job is similar to the job of another persona?

- ☐ Mary, 49 years, Owner of a bistro (customer)
- ☐ Ann, 39, Usability engineer / Project manager
- ☐ Thomas, 35, Developer
- ☐ Michael, 23, Graphic Designer

What do we need to change in the persona we used in the walkthrough (you only need to consider the job or responsibilities of the persona)?

.....

.....

.....

How easy / difficult was it to understand the approach described by the storyboards?

Very easy	Easy	Neutral	Difficult	Very difficult

Did the metadata contribute to your understanding of the storyboard?

- ☐ Yes, the personas were helpful to understand the responsibilities of each team member
- ☐ Yes, the devices were helpful me to understand the different needs in the team
- ☐ Yes, the personas and devices were helpful to understand the needs
- ☐ Yes,
- ☐ No, the metadata was confusing
- ☐ No,

Are you familiar with the use of storyboards?

yes / no

When do / would you consider the use of storyboards in a project?

.....
.....
.....

Do you know another kind of document / artefact to discuss the use of a context aware application?

.....

Rank the following documents / artefacts to discuss the use of a context aware application? (1= most suitable; 3= least suitable)

(other document / artefact)
Narrative scenario
Storyboard

Thank you for your participation!

A.3 Observational Study

The documents that were used for the observational study presented in Chapter 6, Section 6.3.

A.3.1 Personas

Bob, 45 years, banker, father (Figure A.6) Bob is 45 years old, works as a banker, is married to Mary and is a father of two children: Kate and Benjamin. His working days are very long, and in the evenings he often has to do some work or has appointments with customers. Together with his wife and children, he lives in a nice house, which was constructed a few years ago.

Bob likes gadgets and new technologies, and was the first one in his circle of friends that was using a smartphone. Since that moment, he can check his e-mails on any location and he likes the fact that he has a browser, a route planner and a camera in his pocket. Recently, he also uses his smartphone to control the home automation system. At home, it is easy to control the heating, lighting and music from any place. He loves playing with the settings when he is relaxing in the living room. He also uses the control application on his smartphone at work. As a banker, he continuously checks how he can save money. Because of economic and environmental reasons, he also tries to do savings in expenses for energy and electricity. The home automation system helps him to follow up possible energy savings.



Figure A.6: Photograph of Bob, persona for a banker, father, used for the observational study presented in Chapter 6, Section 6.3.

Mary, 43 years, teacher, mother (Figure A.7) Mary is 43 years old, works as a teacher, is married to Bob and is a mother of two children: Kate and Benjamin. She teaches the languages English and German to high school

students. She loves her job, and the fact that she can spend long vacations together with the children. Since a few years, they are living in a newly constructed house with a big garden, and she loves cooking and gardening.

Although she has a laptop for teaching activities, Mary is not very eager to learn new technologies and limits the use of computers to the necessary computer tasks she has to do for teaching and keeping in touch with a few friends. Every time the family buys new household appliances like a coffee machine, or a washing machine, it takes her several days to learn how to use the new system. Since a few months, a new home automation system is installed in their house. Mary loves the idea that some things can be controlled automatically, Bob installed the application on her laptop and because of its ease of use, Mary can easily control settings using her laptop. However, usually, she controls the settings at home, using the central displays that are available on each floor of their house.



Figure A.7: Photograph of Mary, persona for a teacher, mother, used for the observational study presented in Chapter 6, Section 6.3.

Kate, 14 years, student (Figure A.8) Kate is a 14 year old student. She is very interested in mathematics and her hobbies are reading and chess. Because her mother says it is important to do some sports, she tries jogging a few times a week, but she does not like that at all. She rather likes to read a book or to play chess or computer games. She is also interested in her father's smartphone and the different applications that are available for this device. She would love to have one of her own. Although she begged several times to have her own smartphone, her parents do not allow her to have one. Since her parents bought a home automation system, she likes to play with the settings of it. She loves the way that the system adapts to several user profiles and activities. For each activity at home, she has programmed settings in her profile.



Figure A.8: Photograph of Kate, student, used for the observational study presented in Chapter 6, Section 6.3.

Benjamin, 10 years, student (Figure A.9) Benjamin is 10 years old, he likes playing outside. Soccer is his favorite game, which he loves to play together with his friends. Two days a week, he practices soccer at the local soccer team. Sometimes, he helps his mother gardening and cooking. A few weeks ago, he surprised his parents and his sister with homemade desserts. He is really proud that he can prepare these recipes on his own. His sister is usually teasing him, and is interested in the opposite things. However, lately she showed him some cool computer games and configured the home automation system so that he can easily select his favorite lighting settings and TV show, by selecting only one profile. Benjamin is worried about the future of the environment. The theme of a current project at school is the environment, and Benjamin tries to contribute by applying some guidelines at home.

A.3.2 Scenario

Recently, Bob and Mary decided to install a home automation system. They decided to do that in order to control heating and lighting easily and to save some costs on energy consumption. This system allows them to control settings and to adapt these settings according to their own profiles. In the past, it often happened that the children left the house without turning off the lights, or that the programmed heating system was heating the house, while no one was at home. Using the two displays installed in the house, settings of the system



Figure A.9: Photograph of Benjamin, student, used for the observational study presented in Chapter 6, Section 6.3.

can be controlled. One display is available in the living room, and another one in the hallway on the first floor. Today, Kate wants to read a book, while her mother and her brother are outside. In the living room, she loads her personal profile, and automatically all lights are switched off, the light near the sofa is turned on and her favorite pop music starts playing. Thirty minutes later, her mother and Benjamin come in. Benjamin loads his personal user profile using the home automation display in the living room. The system recognizes two profiles now. Based on the profiles Kate programmed before, the light above the sofa stays on, while the pop music stops and the TV starts playing Benjamin's favorite TV show.

Mary begins to prepare dinner. Five minutes later, the phone is ringing. Mary picks up the phone. It is Bob, who wants to notify Mary that he is stuck in traffic on his way home. Tonight he will have to work in the home office, and he already programmed the heating for that room using his smartphone. He also mentions that the family does not have to wait for him for dinner. One hour later, Bob arrives at home. Together with Benjamin, he explores the home automation system using the central display in the living room. He teaches Benjamin how to interpret the statistics regarding the energy savings. Benjamin is impressed that the system can record this information and is already thinking how the efficiency can be improved.

Later that evening, when the children are in bed, Bob shuts down his computer in the home office. He joins Mary, who is reading a book in the living room. Bob's smartphone reminds him that he left the home office, but the heating in this room is still on. Bob accepts the system's suggestion to switch off the heating in the home office. Bob and Mary discuss their day, and then they go to sleep. As programmed in the system, the heating is turned

off automatically, and the lights are switched off by one press on a button of Bob's smartphone, when he gets into bed. ?

A.3.3 Questionnaire

Questionnaire

Team:

Role:.....

This questionnaire can be answered in English or in Dutch.

ABOUT YOU

1. **Gender:** Female / Male
2. **Age:** years
3. **Education:**
4. **Current job:**
5. **What role matches best with your current job:** (you can select multiple answers)
 - ☐ HCI specialist
 - ☐ Designer
 - ☐ Systems analyst / Programmer
 - ☐ Stakeholder (e.g. purchaser, application domain specialist)
 - ☐ None of the above

6. How comfortable did you feel with the role that was assigned to you?

Not comfortable at all	Not comfortable	Neutral	Comfortable	Very comfortable

EXPERIENCE

7. **Do you have any experience being part of a multi-disciplinary team?** Yes / No
 If you answered yes, for how long did / do you work in a multi-disciplinary team?
 - ☐ A few months
 - ☐ More than 1 year
 - ☐ More than 5 years
8. **Do you have any experience in creating this type of storyboards?** Yes / No
9. **Do you have any experience in using this type of storyboards?** Yes / No

Figure A.10: Questionnaire used for the observational study presented in Chapter 6, Section 6.3.

THE STORYBOARDING WORKSHOP

10. Were there any missing tools during the storyboarding workshop?

If yes, what tools should be added?

- ☐ Yes, _____
- ☐ No

11. How easy was it to create the storyboard, starting from the scenario and personas?

Very difficult	Difficult	Neutral	Easy	Very easy

12. Did you understand what your immediate goals were and what you needed to do to achieve them?

- ☐ Yes
- ☐ No

13. How satisfied are you with the resulting storyboard?

Not satisfied at all	Not satisfied	Neutral	Satisfied	Very satisfied

Why? _____

14. How would you estimate your influence on the storyboard?

.....% of the storyboard is based on my ideas.

15. How would you estimate your direct contribution to the storyboard?

.....% of the storyboard contains my sketches, artifacts, etc.

16. How satisfied are you with the extent to which you could contribute to the storyboard?

Not satisfied at all	Not satisfied	Neutral	Satisfied	Very satisfied

17. How satisfied are you with the extent to which the team used your contributions?

Not satisfied at all	Not satisfied	Neutral	Satisfied	Very satisfied

18. Did you work on ideas in private (e.g. on an isolated piece of paper) before sharing them?

- ☐ Yes
☐ No

Why? _____

19. Were you continually aware of what the others were doing throughout the workshop?

- ☐ Yes
☐ No

20. Did you understand the symbolism and sequence of the final storyboard and the related information that was visible on the table?

- ☐ Yes
☐ No

21. Did being part of a multidisciplinary team influence the ideas contained by the storyboard?

- ☐ Yes
☐ No

Why? _____

22. Given the role you had during the workshop, what aspects of the storyboard would be useful for your following tasks during the project?

A.4 Interview and User Study with Agile SE Practitioners

A.4.1 Semi-structured Interview

The questions that were asked during the semi-structured interview with Agile SE practitioners presented in Chapter 9, Section 9.3:

- How are requirements elicited and specified in your company?
- Do you use user stories as a notation for requirements?
- What is the format you use for user stories?
- How is the UI specified in a project?
- How is the context of use considered in a project?
- What tools are used for specifying the requirements?

A.4.2 First User Study

The questions that were asked during the first user study in an agile team presented in Chapter 9, Section 9.6:

- Was the time needed for the creation of the storyboard reasonable?
- Was the notation used for the storyboards offering enough flexibility to depict a case?
- Can other artifacts you created before the storyboard be connected to this resulting storyboard?
- Was the storyboarding tool helpful when preparing a storyboard?
- Was the storyboarding tool easy to use?
- What features were interesting in the tool?
- What features were interesting in the tool?

A.5 Interpreting Storyboards

Documents used for the user experiment presented in Chapter 10.

Storyboard:

Participant no.:

Questionnaire

This questionnaire can be answered in English or in Dutch.

ABOUT YOU

1. **Gender:** Female / Male
2. **Age:** years
3. **Education:**
4. **Current occupation:** in Industry / Academics
5. **Do you have any experience in UI design?** Yes / No
 If you answered yes, for how long did / do you create UI designs?
 - ☐ A few months
 - ☐ More than 1 year
 - ☐ More than 5 years

If you answered yes, how often do you create UI designs?

Less than once a year	A few times a year	Once a month	Once a week	Every day

6. **Do you have any experience in using this type of storyboards?** Yes / No
 If you answered yes, for what kind of activities did you use storyboards?

7. **Do you have any experience being part of a multi-disciplinary team?** Yes / No
 If you answered yes, for how long did / do you work in a multi-disciplinary team?
 - ☐ A few months
 - ☐ More than 1 year
 - ☐ More than 5 years

Figure A.11: Questionnaire used for the observational study presented in Chapter 6, Section 6.3.

UI DESIGN SESSION

8. How comfortable did you feel with the job you had to carry out?

Not comfortable at all	Rather not comfortable	Neutral	Rather comfortable	Very comfortable

9. How familiar are you with creating UI designs for this platform / device?

Not familiar at all	Rather not familiar	Neutral	Rather familiar	Very familiar

10. How easy was it to understand the storyboard?

Very difficult	Difficult	Neutral	Easy	Very easy

11. How easy was it to create the UI designs, starting from the storyboard?

Very difficult	Difficult	Neutral	Easy	Very easy

12. What parts of the storyboard do you consider to be the most valuable for the UI designs?

.....

.....

.....

13. How satisfied are you with the resulting UI designs?

Not satisfied at all	Rather not satisfied	Neutral	Rather satisfied	Very satisfied

14. How useful are storyboards to inform UI designers about the context of use of a system?

Not useful at all	Rather not useful	Neutral	Rather useful	Very useful

DISCUSSION WITH THE AUTHOR

15. How useful was the storyboard to explain your design decisions of the first session?

Not useful at all	Rather not useful	Neutral	Rather useful	Very useful

16. Was there any missing information in the storyboard? Yes / No

If you answered yes, what information was missing?

.....

.....

.....

17. Did the discussion reveal any misconceptions regarding the storyboard? Yes / No

If you answered yes, what kind of misconceptions were introduced?

.....

.....

.....

Did these misconceptions influence your UI design?

.....

.....

.....

18. Did you change your satisfaction of the resulting UI designs change after the discussion?

Yes / No

If you answered yes, how satisfied are you with the resulting UI designs now?

Not satisfied at all	Rather not satisfied	Neutral	Rather satisfied	Very satisfied

What influenced your opinion?

.....

.....

.....

19. Do you have any general remarks/comments regarding the study?

.....

.....

.....

Thank you very much for participating in this study!

Storyboard:

Participant no.:

Post-discussion interview: author

Was there a particular approach you followed to verify the UI designs according to the storyboard?
What approach?

Where there any misconceptions in the designer's interpretation of the storyboard?

How did you discover these misconceptions?

Could the misconceptions be avoided by providing more/other annotations, more/other information? How?

Did the UI designs include any interesting suggestions that were not considered in your project?

Appendix B

Samenvatting (Dutch summary)

De laatste decennia is het gebruik van interactieve systemen sterk gestegen. De beschikbaarheid van verschillende technologieën maakt het mogelijk om deze systemen overal en in verschillende omstandigheden te gebruiken. Tegenwoordig is de groep van gebruikers van interactieve systemen zeer breed en eindgebruikers verwachten ook steeds meer positieve gebruikerservaringen (bv. gebruiksvriendelijkheid en toegankelijkheid) van deze systemen. Bijgevolg is het aangewezen om vanaf het begin van het ontwerp en de ontwikkeling van software toepassingen rekening te houden met de gebruikersnoden.

Gebruikersgerichte methoden voor het ontwerp en de ontwikkeling van interactieve systemen, zoals user-centered design (UCD), blijken zeer geschikt te zijn om grondig rekening te houden met gebruikersnoden en de context waarin een toepassing zal worden gebruikt. Bij deze methoden worden meestal meerdere disciplines betrokken, waardoor complementaire standpunten aan bod komen in het ontwerp en de ontwikkeling van een toepassing. Desalniettemin gaat de samenwerking binnen zogenaamde multi-disciplinaire teams gepaard met verschillende moeilijkheden tijdens het ontwerp- en ontwikkelingsproces.

In dit doctoraat werd de combinatie van UCD en software engineering bestudeerd. Hierbij heb ik mij vooral geconcentreerd op de betrokkenheid van multi-disciplinaire teams, omdat zij onmisbaar zijn wanneer er grondig rekening moet worden gehouden met de gebruikersnoden en de context van het gebruik van een toepassing. MuiCSer¹, een proces raamwerk voor multi-disciplinaire gebruikersgerichte software engineering, werd voorgesteld

¹MuiCSer wordt uitgesproken als “mixer”.

om meer specifieke user-centered software engineering (UCSE) processen te definiëren, die kunnen worden gebruikt voor toegepaste projecten. Daarnaast hebben we MuiCSer aangewend om het gebruik van ontwerp- en ontwikkelingsomgevingen en artefacten in UCSE evenals het gebruik van UCSE processen in praktische omstandigheden te bestuderen. Dit onderzoek resulteerde in de identificatie van een aantal tekortkomingen en moeilijkheden van UCSE zoals het tekort aan notaties en ontwerp- of ontwikkelingsomgevingen die de samenwerking in multi-disciplinaire teams en de transitie van informele artefacten naar formele modellen ondersteunen. Daarnaast toonde dit onderzoek een gebrek aan notaties die alle gebruikersnoden en vereisten kunnen bevatten.

De COMuICSer ² storyboarding werkwijze en ontwerpomgeving werden voorgesteld om de tekortkomingen en moeilijkheden in UCSE te minimaliseren. Vooreerst werden er twee gebruikersstudies uitgevoerd om de creatie en het gebruik van storyboards in multi-disciplinaire teams te bestuderen. De resultaten van deze studies lagen aan de basis van verder onderzoek betreffende COMuICSer dat betrekking had tot het opnemen van technieken gebruikt in de creatie van stripverhalen in de COMuICSer werkwijze en ontwerpomgeving.

COMuICSer werd niet enkel als communicatiemiddel beschouwd. Daarnaast werd het gebruik van COMuICSer bestudeerd voor de transitie van informele artefacten naar formele modellen. COMuICSer storyboards werden bovendien ook verbonden met user stories, gebruikt in agiele software engineering zodat niet-functionele vereisten en contextuele informatie konden worden opgenomen in de vereisten voor een te ontwikkelen systeem.

De zeer visuele notatie gebruikt in COMuICSer voor het afbeelden van vereisten voor interactieve systemen blijkt zeer geschikt voor de communicatie binnen een multi-disciplinair team. Daarnaast kunnen COMuICSer storyboards worden beschouwd als centraal document in UCSE processen. De storyboarding technieken die werden onderzocht in dit doctoraat werden geëvalueerd tijdens verscheidene gebruikersstudies. In een reflectie op COMuICSer storyboarding vanuit verschillende perspectieven werden de sterktes en zwaktes van COMuICSer beschreven en werden verschillende opportuniteiten voor toekomstig onderzoek en ontwikkeling gesuggereerd.

²COMuICSer wordt uitgesproken als “comics-er”.

Appendix C

Glossary

Abstract Interaction Object (AIO): Interface objects that mainly describe the behavior of a user interface, without specifying their presentation.

Abstract UI: A user interface specification that shows the interaction and the information in an abstract way. An abstract UI consists of *AIOs*.

Activity Designer: A tool that supports an activity-based prototyping process. One of the first steps supported by the tool, is the creation of scenes, based on everyday observations. These scenes contain textual labels describing actions and situations, as well as an accompanying visual representation [LL08a].

Activity Diagram: A visual representation of a flow of activities and actions. An activity diagram includes notations to specify choice, iteration and concurrency.

Agile Software Engineering: A branch of software engineering that concentrates on iterative and incremental development. The focus is on close collaboration with the customer and the team.

CanonSketch: A tool that supports creating, designing and editing *Canonical Abstract Prototypes (CAPs)* [CN07a].

Canonical Abstract Prototyping (CAP): A form of abstract user interface design. The CAP notation facilitates several aspects in design, such as comparison and specification of design patterns [Con03].

CAP3: A graphical abstract user interface modeling language that extends *CAP* and specifies interaction design patterns. Its name, CAP3, refers to the three different ways of interpreting *CAP* [VdBLC11].

Concrete UI: A user interface that contains details with respect to its appearance and behavior.

ConcurTaskTree (CTT): A hierarchical task modeling notation that includes temporal operators and specifies the types of tasks [MPS02].

CTTE: A tool that supports creating and editing task models by using the ConcurTaskTree (CTT) notation [MPS02].

Damask: A prototyping tool that facilitates the design of web user interfaces for multiple platforms by sketching the user interface. The use of design patterns allows the specification of high-level concepts [LL08b].

Domain Model: A visual representation of a domain of interest. The domain model can be compared with a conceptual model and is used to specify the vocabulary and most important aspects of a particular domain.

Final UI: A final, interactive user interface that is linked to the full system and application logic.

GrafiXML: A graphical multi-platform user interface design tool. The graphically designed user interface can be saved in the *UsiXML* notation [MV08].

Gummy: A graphical user interface design tool, developed in our research lab, which supports the graphical UI design of multi-device and context-aware user interfaces [MVL08].

Hierarchical Task Analysis (HTA): A task modeling notation that splits a task into subtasks, in order to obtain a hierarchy of tasks, subtasks, sub-subtasks etc.

High-fidelity Prototype: A prototype that looks like the *final UI*, and contains much but not all detail and functionality.

IntuiKit: A programming environment that supports multimodal interaction design [CSV⁺04].

Jelly: A multi-device user interface design tool, developed in our research lab. The resulting designs can be used in a model-based engineering process without forcing the designers to change their working practices considerably [MLC10].

Low-fidelity Prototype: A preliminary prototype that is often sketchy and needs to be completed and fine-tuned before its implementation can take place.

Persona: Fictitious descriptions of end users. By using specific and concrete representations of these users, members of multi-disciplinary UCSE teams are more likely to empathize with the end users [PA06].

Scenario: Stories about people and their activities, describing goals suggested by the appearance and behavior of the system [Car00].

SketchiXML: A sketch-based design tool for user interfaces. The graphically designed user interface can be saved in the *UsiXML* notation [CKV07].

System Architecture: A formal representation of a system's behavior and structure.

System Interaction Model: A model that describes the interactions between the system and the user.

Structured Interaction Model: A graphical representation of interaction that takes several forms. In the context of the MuiCSer process framework, presented in this dissertation, structured interaction models include task models, context models, activity diagrams etc.

Task Model: A description of how activities can be carried out in order to reach a certain goal. Several notations are used for the creation of task models, such as *CCTT* and *HTA*.

TaskSketch: A tool that supports creating and editing use-case models that can be transformed into *system architectures* and *activity diagrams* [CN07a].

Teresa: A tool that supports the design of interactive systems at several abstraction levels. Since Teresa is transformation-based, it generates *concrete UIs* for several platforms [MPS04].

Traceability: A characteristic that specifies to what extent requirements for an interactive application can be linked to the needs presented by stakeholders and to other artifacts created in a design and development process.

Usage Centered Design: An approach that combines user-centered and agile methods. In contrast to user-centered design, usage-centered design focuses on the usage of a system rather than the users [CL02].

User Interface Markup Language (UIML): An XML language that supports the specification of user interfaces in an abstract form.

Use Case: A notation that specifies the interaction between a user and a system.

User Story: A description of the requirements of a system that concentrates on the system's functionality from a user's point of view [Coh04]. User stories are used as a planning tool for *agile software engineering* practices.

UsiXML: A device and platform independent XML markup language that describes UIs.

Visibility: The extent to which the status of a software design and development project and the artifacts created in the project are visible to all members and stakeholders of the team.

Vista Environment: A prototyping tool that examines links between design artifacts. Vista supports hyperlinked task hierarchies, task-oriented specifications, software architectures and code documents [BGW98].

Z: A formal specification language, based on the standard mathematical notation, that can be used to model software systems.

Bibliography

- [AAB06] Jonathan Arnowitz, Michael Arent, and Nevin Berger. *Effective Prototyping for Software Makers (The Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.
- [All83] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.
- [AM11] Berke Atasoy and Jean-Bernard Martens. Storify: a tool to assist design teams in envisioning and discussing user experience. In *CHI Extended Abstracts*, pages 2263–2268, 2011.
- [Axu] Axure Wireframes, Prototypes & Specifications. <http://www.axure.com>.
- [BA04] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional, 2004.
- [BBvB⁺01] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development, 2001. <http://www.agilemanifesto.org>.
- [BGW98] Judy Brown, T. C. Nicholas Graham, and Timothy Wright. The vista environment for the coevolutionary design of user interfaces. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 376–383, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

- [BH98] Hugh Beyer and Karen Holtzblatt. *Contextual design: defining customer-centered systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [BLB08] Judith Brown, Gitte Lindgaard, and Robert Biddle. Stories, Sketches, and Lists: Developers and Interaction Designers Interacting Through Artefacts. In *Proceedings of Agile 2008*, pages 39 – 50, 2008.
- [BR06] Judy Bowen and Steve Reeves. Formal refinement of informal gui design artefacts. In *ASWEC '06: Proceedings of the Australian Software Engineering Conference*, pages 221–230, 2006.
- [But96] Keith A. Butler. Usability engineering turns 10. *interactions*, 3:58–75, January 1996.
- [Bux07] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, first edition, March 2007.
- [CA07] Betty H. C. Cheng and Joanne M. Atlee. Research directions in requirements engineering. In *2007 Future of Software Engineering, FOSE '07*, pages 285–303, Washington, DC, USA, 2007. IEEE Computer Society.
- [Car00] John M. Carroll. *Making use : scenario-based design of human-computer interactions*. MIT Press, Cambridge, Mass. [u.a.], 2000.
- [CCDBR07] Karin Coninx, Erwin Cuppens, Joan De Boeck, and Chris Raymaekers. Integrating support for usability evaluation into high level interaction descriptions with nimmit. In *Proceedings of the 13th international conference on Interactive systems: Design, specification, and verification, DSVIS'06*, pages 95–108, Berlin, Heidelberg, 2007. Springer-Verlag.
- [CCT⁺02] Gaelle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Nathalie Souchon, Laurent Bouillon, Murielle Florins, and Jean Vanderdonckt. Plasticity of user interfaces: A revised reference framework. In *Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design*, pages 127–134. IN-FOREC Publishing House Bucharest, 2002.
- [Cel] Celtx. <http://celtx.com>.
- [CHB05] Karin Coninx, Mieke Haesen, and Jan Bierhoff. VIP-lab: A virtual lab for ICT experience prototyping. In *Proc. Measuring Behavior 2005*, pages 585–586, 2005.

- [Cil11] Sal Cilella. Did you ever know that you're my hero?: the power of storytelling. *interactions*, 18:62–66, January 2011.
- [CKV07] Adrien Coyette, Suzanne Kieffer, and Jean Vanderdonckt. Multifidelity prototyping of user interfaces. In *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction*, INTERACT'07, pages 150–164, Berlin, Heidelberg, 2007. Springer-Verlag.
- [CL02] Larry L. Constantine and Lucy A. D. Lockwood. Usage-centered engineering for web applications. *IEEE Softw.*, 19:42–50, March 2002.
- [CLC04] Tim Clerckx, Kris Luyten, and Karin Coninx. The mapping problem back and forth: customizing dynamic models while preserving consistency. In *Task Models and Diagrams for User Interface Design*, pages 33–42, 2004.
- [CN07a] Pedro Campos and Nuno Jardim Nunes. Practitioner tools and workstyles for user-interface design. *IEEE Software*, 24(1):73–80, 2007.
- [CN07b] Pedro F. Campos and Nuno Jardim Nunes. Towards useful and usable interaction design tools: Canonsketch. *Interacting with Computers*, 19(5-6):597–613, 2007.
- [Coh] Mike Cohn. User Stories Website. <http://www.userstories.com/products>.
- [Coh04] Mike Cohn. *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional, 2004.
- [Con95] Larry L. Constantine. Essential modeling: use cases for user interfaces. *interactions*, 2:34–46, April 1995.
- [Con96] Larry L. Constantine. Usage-centered software engineering: New models, methods and metrics. *Software Engineering: Education and Practice, International Conference on*, 0:2, 1996.
- [Con03] Larry L. Constantine. Canonical abstract prototypes for abstract visual and interaction. In *DSV-IS*, pages 1–15, 2003.
- [CPL09] Lawrence Chung and Julio Cesar Prado Leite. Conceptual modeling: Foundations and applications. chapter On Non-Functional Requirements in Software Engineering, pages 363–379. Springer-Verlag, Berlin, Heidelberg, 2009.
- [CR08] Lan Cao and Balasubramaniam Ramesh. Agile requirements engineering practices: An empirical study. *IEEE Softw.*, 25:60–67, January 2008.

- [CSV⁺04] Stéphane Chatty, Stéphane Sire, Jean-Luc Vinot, Patrick Lecoanet, Alexandre Lemort, and Christophe Mertz. Revisiting visual interface programming: creating gui tools for designers and programmers. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 267–276, New York, NY, USA, 2004. ACM.
- [DJA93] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of oz studies: why and how. In *Proceedings of the 1st international conference on Intelligent user interfaces*, IUI '93, pages 193–200, New York, NY, USA, 1993. ACM.
- [DLDZ07] Scott Davidoff, Min Kyung Lee, Anind K. Dey, and John Zimmerman. Rapidly exploring application design through speed dating. In *Proceedings of the 9th international conference on Ubiquitous computing*, UbiComp '07, pages 429–446, Berlin, Heidelberg, 2007. Springer-Verlag.
- [dSCDR09] Marco de Sá, Lu's Carrio, Lu's Duarte, and Tiago Reis. Supporting the design of mobile interactive artefacts. *Advances in Engineering Software*, 40(12):1279 – 1286, 2009. Designing, modelling and implementing interactive systems.
- [DSL06] Steven Dow, T. Scott Saponas, Yang Li, and James A. Landay. External representations in ubiquitous computing design and the implications for design tools. In *DIS '06: Proceedings of the 6th conference on Designing Interactive systems*, pages 241–250, New York, NY, USA, 2006. ACM.
- [Ecl] Eclipse Process Framework. <http://www.eclipse.org/epf>.
- [Eva03] Evans. *Domain-Driven Design: Tackling Complexity In the Heart of Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
- [Fli] FlickrNet. Api library. <http://www.codeplex.com/FlickrNet>.
- [Gct] Scott McCloud Google Chrome team. Google chrome. <http://www.google.com/googlebooks/chrome>.
- [GGB⁺03] Jan Gulliksen, Bengt Göransson, Inger Boivie, Stefan Blomkvist, Jenny Persson, and Åsa Cajander. Key principles for user-centred systems design. *Behaviour & IT*, 22(6):397–409, 2003.
- [GMKC05] Silvia Gabrielli, Valeria Mirabella, Stephen Kimani, and Tiziana Catarci. Supporting cognitive walkthrough with video data: a mobile learning evaluation study. In *Proceedings of the 7th international*

- conference on Human computer interaction with mobile devices & services*, MobileHCI '05, pages 77–82, New York, NY, USA, 2005. ACM.
- [GP96] T. R. G. Green and M. Petre. Usability analysis of visual programming environments: a ‘cognitive dimensions’ framework. *JOURNAL OF VISUAL LANGUAGES AND COMPUTING*, 7:131–174, 1996.
- [Gre02] Saul Greenberg. Working through task-centered system design. In *The Handbook of Task Analysis for Human-Computer Interaction.*, pages 49–66. Lawrence Erlbaum Associates., 2002.
- [GWM08] Yaser Ghanam, Xin Wang, and Frank Maurer. Utilizing digital tabletops in collocated agile planning meetings. In *Proceedings of the Agile 2008*, pages 51–62, Washington, DC, USA, 2008. IEEE Computer Society.
- [GWW10] Krzysztof Z. Gajos, Daniel S. Weld, and Jacob O. Wobbrock. Automatically generating personalized user interfaces with supple. *Artificial Intelligence*, 174:910–950, 2010.
- [HA08] James Helms and Marc Abrams. Retrospective on ui description languages, based on eight years’ experience with the user interface markup language (uiml). *International Journal of Web Engineering and Technology (IJWET)*, 4(2), 2008.
- [HCB⁺06] Mieke Haesen, Karin Coninx, Ilse Bakx, David Geerts, and E. Van Evenstraat A. Ict experience prototyping in practice: Artefact transitions. In *Proc. of Improving the Relationship between Research and Practice in Interaction Design, workshop of NordiCHI 2006*, 2006.
- [HCdBL08] Mieke Haesen, Karin Coninx, Jan Van den Bergh, and Kris Luyten. Muicser: A process framework for multi-disciplinary user-centred software engineering processes. In *TAMODIA/HCSE*, pages 150–165, 2008.
- [HDBCR09] Mieke Haesen, Joan De Boeck, Karin Coninx, and Chris Raymaekers. An interactive coal mine museum visit: prototyping the user experience. In *Proceedings of the 2nd conference on Human System Interactions*, HSI’09, pages 543–550, Piscataway, NJ, USA, 2009. IEEE Press.
- [HdBM⁺11] Mieke Haesen, Jan Van den Bergh, Jan Meskens, Kris Luyten, Sylvain Degrandart, Serge Demeyer, and Karin Coninx. Using storyboards to integrate models and informal design knowledge. In

- Model-Driven Development of Advanced User Interfaces*, pages 87–106. Springer Verlag, London, 2011. ISBN=978-3-642-14561-2.
- [HK06] Tom Hurlbutt and Scott R. Klemmer. Bridging the gap: fluidly connecting paper notecards with digital representations for story/task-based planning. In *CHI '06 extended abstracts on Human factors in computing systems*, CHI '06, pages 887–892, New York, NY, USA, 2006. ACM.
- [HLC⁺08] Mieke Haesen, Kris Luyten, Karin Coninx, Jan Van den Bergh, and Chris Raymaekers. Muicser: A multi-disciplinary user-centered software engineering process to increase the overall user experience. In *ICEIS (5)*, pages 371–374, 2008.
- [HLC09] Mieke Haesen, Kris Luyten, and Karin Coninx. Get your requirements straight: Storyboarding revisited. In *INTERACT (2)*, pages 546–549, 2009.
- [HMBW10] Björn Hartmann, Meredith Ringel Morris, Hrvoje Benko, and Andrew D. Wilson. Pictionaire: supporting collaborative design work by integrating physical and digital artifacts. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, CSCW '10, pages 421–424, New York, NY, USA, 2010. ACM.
- [HMC09] Mieke Haesen, Jan Meskens, and Karin Coninx. Visualising digital video libraries for tv broadcasting industry: A user-centred approach. In *3rd Workshop on Human-Computer Interaction and Information Retrieval (HCIR)*, October 2009.
- [HML⁺11] Mieke Haesen, Jan Meskens, Kris Luyten, Jan Hendrik Becker, Tinne Tuytelaars, Gert-Jan Poulisse, Phi The Pham, and Marie-Francine Moens. Finding a needle in a haystack: an interactive video archive explorer for professional video searchers. *Multimedia Tools and Applications - Special Issue on Automated Information Extraction in Media*, 2011.
- [HMLC09] Mieke Haesen, Jan Meskens, Kris Luyten, and Karin Coninx. Supporting multidisciplinary teams and early design stages using storyboards. In *HCI (1)*, pages 616–623, 2009.
- [HMLC10] Mieke Haesen, Jan Meskens, Kris Luyten, and Karin Coninx. Draw me a storyboard: Incorporating principles and techniques of comics to ease communication and artefact creation in user-centred design. In *24th BCS Conference on Human Computer Interaction*, HCI2010. British Computer Society, 2010.

- [HRLC11] Mieke Haesen, Karel Robert, Kris Luyten, and Karin Coninx. Multi-touch storyboarding tool concept for user-centered design teams. In *Create Symposium*, June 2011.
- [HWW04] Karen Holtzblatt, Jessamyn B. Wendell, and Shelley Wood. *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design (Interactive Technologies)*. Morgan Kaufmann, December 2004.
- [Imm] Immersive Education. Kar2ouche. <http://www.immersiveeducation.eu/index.php/kar2ouchepg>.
- [Int99] International Standards Organization, Geneva, Swiss. *ISO 13407. Human Centred Design Process for Interactive Systems*, 1999.
- [JA07] Maria Johansson and Mattias Arvola. A case study of how user interface sketches, scenarios and computer prototypes structure stakeholder meetings. In *BCS-HCI '07: Proceedings of the 21st British CHI Group Annual Conference on HCI 2007*, pages 177–184, Swinton, UK, UK, 2007. British Computer Society.
- [JH98] Stan Jarzabek and Riri Huang. The case for user-centered case tools. *Commun. ACM*, 41:93–99, August 1998.
- [JLN10] Frédéric Jourde, Yann Laurillau, and Laurence Nigay. Comm notation for specifying collaborative and multimodal interactive systems. In *EICS*, pages 125–134. ACM, 2010.
- [Kar97] John Karat. Evolving the scope of user-centered design. *Commun. ACM*, 40:33–38, July 1997.
- [KJ07] Niina Kantola and Timo Jokela. Svsb: simple and visual storyboards: developing a visualisation method for depicting user scenarios. In *Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces, OZCHI '07*, pages 49–56, New York, NY, USA, 2007. ACM.
- [KMK09] Inger Kristine Karlsen, Neil Maiden, and Andruid Kerne. Inventing requirements with creativity support tools. In *Proceedings of the 15th International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ '09*, pages 162–174, Berlin, Heidelberg, 2009. Springer-Verlag.
- [L. 03] Larry L. Constantine. Canonical abstract prototypes for abstract visual and interaction design. In *Proc. DSV-IS 2003*, volume 2844 of *LNCS*, pages 1–15. Springer, 2003.

- [LDT⁺06] Gitte Lindgaard, Richard Dillon, Patricia Trbovich, Rachel White, Gary Fernandes, Sonny Lundahl, and Anu Pinnamaneni. User needs analysis and requirements engineering: Theory and practice. *Interact. Comput.*, 18(1):47–70, 2006.
- [LHO⁺10] Kris Luyten, Mieke Haesen, Dawid Ostrowski, Karin Coninx, Sylvain Degrandt, and Serge Demeyer. On stories, models and notations: Storyboard creation as an entry point for model-based interface development with usixml. In *User Interface eXtensible Markup Language Workshop at The ACM SIGCHI Symposium on Engineering Interactive Computing Systems 2010 (EICS 2010)*, 2010.
- [LL08a] Yang Li and James A. Landay. Activity-based prototyping of ubi-comp applications for long-lived, everyday human activities. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI 2008*, pages 1303–1312, 2008.
- [LL08b] James Lin and James A. Landay. Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces. In *Proc. CHI '08*, pages 1313–1322. ACM, April 2008.
- [LM96] James A. Landay and Brad A. Myers. Sketching storyboards to illustrate interface behaviors. In *Conference companion on Human factors in computing systems: common ground, CHI '96*, pages 193–194, New York, NY, USA, 1996. ACM.
- [LMVC08] Kris Luyten, Jan Meskens, Jo Vermeulen, and Karin Coninx. Meta-gui-builders: Generating domain-specific interface builders for multi-device user interface creation. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, New York, NY, USA, 2008. ACM.
- [LVM⁺04] Quentin Limbourg, Jean Vanderdonckt, Benjamin Michotte, Laurent Bouillon, and Víctor López-Jaquero. Usixml: A language supporting multi-path development of user interfaces. In *EHCI/DS-VIS*, pages 200–220, 2004.
- [Mar02] Panos Markopoulos. Modelling user tasks with the unified modelling language. In *In Proc. of Workshop TUPIS2000 in UML2000 International Conference*, 2002.
- [May99] Deborah J. Mayhew. *The usability engineering lifecycle: a practitioner's handbook for user interface design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

- [MBN04] Angela Martin, Robert Biddle, and James Noble. The xp customer role in practice: Three studies. In *Proceedings of the Agile Development Conference*, pages 42–54, Washington, DC, USA, 2004. IEEE Computer Society.
- [MBTM08] Scott Martens, Jan Hendrik Walter Becker, Tinne Tuytelaars, and Marie-Francine Moens. Multimodal data collection in the AMASS++project. In *Multimodal Corpora: from Models of Natural Interaction to Systems and Applications*, pages 60–63. European language resources association, 2008.
- [McC] Scott McCloud. Scott mccloud - journal. <http://scottmccloud.com/>.
- [McC93] Scott McCloud. *Understanding Comics: The Invisible Art*. Tundra Publishing Ltd, 1993.
- [Mcc06] Scott Mccloud. *Making Comics: Storytelling Secrets of Comics, Manga and Graphic Novels*. Harper Paperbacks, September 2006.
- [MGM03] Heather L. McQuaid, Aradhana Goel, and Mickey McManus. When you can't talk to customers: using storyboards and narratives to elicit empathy for users. In *Proceedings of the 2003 international conference on Designing pleasurable products and interfaces*, DPPI '03, pages 120–125, New York, NY, USA, 2003. ACM.
- [MGR07] Thomas Memmel, Fredrik Gundelsweiler, and Harald Reiterer. Agile human-centered software engineering. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1*, BCS-HCI '07, pages 167–175, Swinton, UK, UK, 2007. British Computer Society.
- [MHC08] Chris Raymaekers Mieke Haesen and Karin Coninx. Evaluating a location-based mobile game in early stages of the development. In *Proceedings of the 6th International Conference on Methods and Techniques in Behavioral Research*, Measuring Behavior 2008, pages 113–114, 2008.
- [MHLC09] Jan Meskens, Mieke Haesen, Kris Luyten, and Karin Coninx. User-centered adaptation of user interfaces for heterogeneous environments. In Phivos Mylonas Marios C Angelides and Manolis Wallace, editors, *Advances in Semantic Media Adaptation and Personalization*, volume 2, pages 43–65. Auerbach Publications, 2009.
- [MHP00] Brad Myers, Scott E. Hudson, and Randy Pausch. Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.*, 7(1):3–28, 2000.

- [MJ10] Neil Maiden and Sara Jones. Agile requirements can we have our cake and eat it too? *IEEE Softw.*, 27:87–88, May 2010.
- [MJK⁺10] Neil Maiden, Sara Jones, Kristine Karlsen, Roger Neill, Konstantinos Zachos, and Alastair Milne. Requirements engineering as creative problem solving: A research agenda for idea finding. *Requirements Engineering, IEEE International Conference on*, 0:57–66, 2010.
- [MLC10] Jan Meskens, Kris Luyten, and Karin Coninx. Jelly: A multi-device design environment for managing consistency across devices. In *Proc. AVT'10*, pages 289–298. ACM, 2010.
- [MLD⁺07] Neema Moraveji, Jason Li, Jiarong Ding, Patrick O’Kelley, and Suze Woolf. Comicboarding: using comics as proxies for participatory design with children. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI ’07, pages 1371–1374, New York, NY, USA, 2007. ACM.
- [MM05] Paul McInerney and Frank Maurer. Ucd in agile projects: dream team or odd couple? *interactions*, 12:19–23, November 2005.
- [Moo09] Daniel L. Moody. The “physics” of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35:756–779, 2009.
- [MPN⁺08] Brad A. Myers, Sun Young Park, Yoko Nakano, Greg Mueller, and Andrew Ko. How designers design and program interactive behaviors. In *VL/HCC*, pages 177–184, 2008.
- [MPS02] Giulio Mori, Fabio Paternò, and Carmen Santoro. CTTE: support for developing and analyzing task models for interactive system design. *IEEE Transactions on Software Engineering*, 28(8):797–813, 2002.
- [MPS04] Giulio Mori, Fabio Paternò, and Carmen Santoro. Design and development of multidevice user interfaces through multiple logical descriptions. *IEEE Transactions on Software Engineering*, 30(8):507–520, August 2004.
- [Mul91] Michael J. Muller. Pictive – an exploration in participatory design. In *CHI ’91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 225–231, New York, NY, USA, 1991. ACM Press.
- [MV08] Benjamin Michotte and Jean Vanderdonckt. A multi-target user interface builder based on usixml. In *Proc. of ICAS2008*, Los Alamitos, USA, 2008. IEEE Computer Society Press.

- [M VLC08] Jan Meskens, Jo Vermeulen, Kris Luyten, and Karin Coninx. Gummy for multi-platform user interface designs: shape multiply fix use me. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 233–240, New York, NY, USA, 2008. ACM.
- [MWK⁺07] Robert Morgan, Jagoda Walny, Henning Kolenda, Estaban Ginez, and Frank Maurer. Using horizontal displays for distributed and collocated agile planning. In *Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming*, XP'07, pages 38–45, Berlin, Heidelberg, 2007. Springer-Verlag.
- [NE00] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 35–46, New York, NY, USA, 2000. ACM.
- [NeC00] Nuno Jardim Nunes and João Falcão e Cunha. Towards a uml profile for interaction design: the wisdom approach. In Andy Evans, Stuart Kent, and Bran Selic, editors, *UML 2000 - The Unified Modeling Language. Advancing the Standard.*, volume 1939 of *LNCS*, pages 101–116. Springer, October 2000.
- [NHA07] Jeffrey Nichols, Duen Horng Chau, and Brad A. Myers. Demonstrating the viability of automatically generated user interfaces. In *Proc. CHI'07*. ACM, 2007.
- [NL00] Mark W. Newman and James A. L. Sitemaps, storyboards, and specifications: A sketch of web site design practice. In *in DIS 2000 Designing Interactive Systems*, pages 263–274. ACM Press, 2000.
- [NLHL03] Mark W. Newman, James Lin, Jason I. Hong, and James A. Landay. Denim: An informal web site design tool inspired by observations of practice. *Human-Computer Interaction*, 18(3):259–324, 2003.
- [NNC05] Leonel Nobrega, Nuno Jardim Nunes, and Helder Coelho. Mapping concurtasktrees into uml 2.0. In *Proc. DSV-IS 05*, volume 3941 of *LNCS*, pages 237–248, 2005.
- [NNC06] Leonel Nóbrega, Nuno Jardim Nunes, and Helder Coelho. The meta sketch editor. In *CADUI*, pages 201–214, 2006.
- [Nor88] Donald A. Norman. *The psychology of everyday things / Donald A. Norman*. Basic Books, New York :, 1988.
- [Nor05] Donald A. Norman. Human-centered design considered harmful. *interactions*, 12:14–19, July 2005.

- [Nun03] Nuno Jardim Nunes. Representing user-interface patterns in uml. In *OOIS*, pages 142–163, 2003.
- [OKZ⁺10] Fatih Kursat Ozenc, Miso Kim, John Zimmerman, Stephen Oney, and Brad Myers. How to support designers in getting hold of the immaterial material of software. In *Proceedings of the 28th international conference on Human factors in computing systems, CHI '10*, pages 2513–2522, New York, NY, USA, 2010. ACM.
- [Ols07] Dan R. Olsen, Jr. Evaluating user interface systems research. In *Proceedings of the 20th annual ACM symposium on User interface software and technology, UIST '07*, pages 251–258, New York, NY, USA, 2007. ACM.
- [PA06] John Pruitt and Tamara Adlin. *The Persona Lifecycle : Keeping People in Mind Throughout Product Design*. Morgan Kaufmann, 2006.
- [Pat99] Fabio Paterno. *Model-Based Design and Evaluation of Interactive Applications*. Springer-Verlag, London, UK, 1st edition, 1999.
- [Pat00] Fabio Paterno. Model-based design of interactive applications. *Intelligence*, 11:26–38, December 2000.
- [PGS06] David Pinelle, Carl Gutwin, and Sriram Subramanian. Designing digital tables for highly integrated collaboration. Technical Report HCI-TR-06-02, University of Saskatchewan, 2006.
- [Pla] Plasq. Comic life. <http://plasq.com/products/comiclife/>.
- [PN08] Volker Paelke and Karsten Nebe. Integrating agile methods for mixed reality design space exploration. In *Proceedings of the 7th ACM conference on Designing interactive systems, DIS '08*, pages 240–249, New York, NY, USA, 2008. ACM.
- [PRS02] Jenny Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2002.
- [PS05] Sonja Pedell and Wally Smith. Relating context to interface: an evaluation of picture scenarios. In *Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future, OZCHI '05*, pages 1–4, Narrabundah, Australia, Australia, 2005. Computer-Human Interaction Special Interest Group (CHISIG) of Australia.
- [Roa08] Dan Roam. *Back of the Napkin: Solving Problems and Selling Ideas with Pictures*. Portfolio, New York, NY, 2008.

- [RPM95] D. Redmond-Pyle and A. Moore. *Graphical User Interface Design and Evaluation*. Prentice Hall, London, 1995.
- [Scr] LLC ScrumDo. Get Scrum Done with ScrumDo. <http://www.scrumdo.com>.
- [SF] Oliver Spindler and Thomas Fadrus. Grimace project. <http://grimace-project.net/>.
- [SM10] Malcolm Sutherland and Neil Maiden. Storyboarding requirements. *IEEE Softw.*, 27:9–11, November 2010.
- [SML⁺09] Katherine M. Sellen, Micheal A. Massimi, Danielle M. Lottridge, Khai N. Truong, and Sean A. Bittle. The people-prototype problem: understanding the interaction between prototype format and user group. In *Proceedings of the 27th international conference on Human factors in computing systems, CHI '09*, pages 635–638, New York, NY, USA, 2009. ACM.
- [Sne] Snels Design. Storyboard tool for interaction design. http://www.snels-design.com/nl/algemeen/gallerij/industrieel_ontwerpen/storyboard_tool1.
- [SR04] Helen Sharp and Hugh Robinson. An ethnographic study of xp practice. *Empirical Softw. Engg.*, 9:353–375, December 2004.
- [Sut05] Allistair Sutcliffe. Convergence or competition between software engineering and human computer interaction. In Ahmed Seffah, Jan Gulliksen, and Michel C. Desmarais, editors, *Human-Centered Software Engineering: Integrating Usability in the Software Development Lifecycle*, volume 8 of *Human-Computer Interaction Series*, pages 71–84. Springer Netherlands, 2005.
- [Sy07] Desire Sy. Adapting usability investigations for agile user-centered design. *Journal of Usability Studies*, 2007.
- [TC99] David Thevenin and Jolle Coutaz. Plasticity of user interfaces: Framework and research agenda. In *Proc. Interact99, Edinburgh, A. Sasse & C. Johnson Eds, IFIP IOS Press Publ*, pages 110–117, 1999.
- [THA06] Khai N. Truong, Gillian R. Hayes, and Gregory D. Abowd. Storyboarding: an empirical determination of best practices and effective guidelines. In *Proceedings of DIS '06*, pages 12–21. ACM, 2006.
- [Usi] UsiXML Website. <http://www.usixml.org>.

- [UW08] Jim Ungar and Jeff White. Agile user centered design: enter the design studio - a case study. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI '08, pages 2167–2178, New York, NY, USA, 2008. ACM.
- [Van96] Jean Vanderdonckt, editor. *Computer-Aided Design of User Interfaces I, Proceedings of the Second International Workshop on Computer-Aided Design of User Interfaces, June 5-7, 1996, Namur, Belgium*. Presses Universitaires de Namur, 1996.
- [VC05] Jan Van den Bergh and Karin Coninx. Towards Modeling Context-Sensitive Interactive Applications: the Context-Sensitive User Interface Profile (CUP). In *SoftVis '05: Proc. the 2005 ACM symposium on Software visualization*, pages 87–94, New York, NY, USA, 2005. ACM Press.
- [VDBBM⁺07] Jan Van Den Bergh, Bert Bruynooghe, Jan Moons, Steven Huypens, Koen Handekyn, and Karin Coninx. Model-driven creation of staged participatory multimedia events on tv. In *Proceedings of the 5th European conference on Interactive TV: a shared experience*, EuroITV'07, pages 21–30, Berlin, Heidelberg, 2007. Springer-Verlag.
- [VdBHL⁺08] Jan Van den Bergh, Mieke Haesen, Kris Luyten, Sofie Notelaers, and Karin Coninx. Toward multi-disciplinary model-based (re)design of sustainable user interfaces. In T. Graham and Philippe Palanque, editors, *Interactive Systems. Design, Specification, and Verification*, volume 5136 of *Lecture Notes in Computer Science*, pages 161–166. Springer Berlin / Heidelberg, 2008.
- [VdBLC11] Jan Van den Bergh, Kris Luyten, and Karin Coninx. Cap3: context-sensitive abstract user interface specification. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '11, pages 31–40, New York, NY, USA, 2011. ACM.
- [vdL06] Corrie van der Lelie. The value of storyboards in the product design process. *Personal Ubiquitous Computing*, 10(2-3):159–162, 2006.
- [VLC09] Geert Vanderhulst, Kris Luyten, and Karin Coninx. Photo-based User Interfaces: Picture it, Tag it, Use it. In *Proceedings of the 4th International Workshop on Ontology Content (OnToContent'09)*, pages 610–615. Springer, 2009.
- [VMSC02] Karel Vredenburg, Ji-Ye Mao, Paul W. Smith, and Tom Carey. A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, CHI '02, pages 471–478, New York, NY, USA, 2002. ACM.

- [VSH⁺11] Jan Van den Bergh, Deepak Sahni, Mieke Haesen, Kris Luyten, and Karin Coninx. Grip: get better results from interactive prototypes. In *EICS*, pages 143–148, 2011.
- [VVC⁺07] Jo Vermeulen, Yves Vandriessche, Tim Clerckx, Kris Luyten, and Karin Coninx. Service-interaction descriptions: Augmenting services with user interface models. In *Proc. EIS'07*. Springer, March 2007.
- [WBC⁺09] Shahtab Wahid, Stacy M. Branham, Lauren Cairco, D. Scott McCrickard, and Steve Harrison. Picking up artifacts: Storyboarding as a gateway to reuse. In *INTERACT (2)*, pages 528–541, 2009.
- [WBMH10] Shahtab Wahid, Stacy M. Branham, D. Scott McCrickard, and Steve Harrison. Investigating the relationship between imagery and rationale in design. In *Proceedings of DIS '10*, pages 75–84. ACM, 2010.
- [WMMO10] Xin Wang, Frank Maurer, Robert Morgan, and Josyleuda Oliveira. Tools for supporting distributed agile project planning. In *Agility Across Time and Space - Implementing Agile Methods in Global Software Projects*. Springer Verlag, London, 2010. ISBN=978-3-642-12441-9.
- [Wor] WordNet.Net. <http://opensource.ebswift.com/WordNet.Net/>.