



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Research Commons

<http://waikato.researchgateway.ac.nz/>

Research Commons at the University of Waikato

Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from the thesis.

Department of Computer Science



Hamilton, New Zealand

Specifying Single-user and Collaborative Profiles for Alerting Systems

Doris Jung

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy at the University of Waikato

December 2009

© Doris Jung 2009

For Dad

Who could not live to see this completed work

And

For Mum

With all my Love

Abstract

The 21st century is the age of information overload. Often, humans are incapable of processing all of the information that surrounds them and determining its relevance. The impact of overlooking crucial information ranges from annoying to fatal.

Alerting systems help users deal with this vast amount of information by employing a push-based rather than a pull-based approach to information delivery. In this way, users receive the information they require at the appropriate moment. Users specify their alerting needs in a profile that is subscribed to the alerting system. The alerting system is continuously fed with data, and filters this data against all subscribed profiles. Whenever incoming data matches a profile, the subscriber is alerted.

Although alerting systems solve the problem of information overload, the potential of these systems has not been fully put into practice. Alerting systems are either realised as dedicated systems that, at best, offer a set of possible profiles to choose from or, at worst, offer a preset profile for one purpose only. Alternatively, they are application frameworks that offer no support for the average user; that is, the specification of profiles is realised using a programming interface. Collaboration between users when specifying profiles is not supported.

This thesis verifies the described situation by considering the example application domain of health care. Within this context, a requirements analysis was undertaken involving a patient-based online survey and interviews with health care providers. This analysis revealed the utility of alerting systems but a need for support for profile specification by end-users. It also identified the need for such a system to support the collaborative nature of health care.

The shortcomings of alerting systems identified for the health-care area also exist in other domains. Hence, a variety of application areas will benefit from providing universal solutions to eliminate these shortcomings.

Based on these findings, this thesis proposes the graphical profile specification language GPDL and an interactive single-user software tool that supports its use (GPDL-UI). The thesis introduces a novel collaborative alerting model for Information Systems. A collaborative extension of GPDL is implemented in the software tool CoastEd, an editor for the graphical specification of collaborative profiles. The developed languages and software tools target average users who have no expertise in specifying profiles involving logics and temporal constraints.

The efficacy of the proposed languages and software were evaluated through three user studies. The first study examined interpretation and specification with GPDL. Based on the results of this first study, the single-user system GPDL-UI was designed and implemented and then evaluated in a second study. In turn, the lessons learned from the implementation and user studies for the single-user system influenced the development of the collaborative approach CoastEd; this editor was evaluated in the third study.

The studies have shown that GPDL and GPDL-UI are suitable means for average users to effectively specify profiles in single-user alerting systems. High levels of accuracy were reached for specification and interpretation in both studies. GPDL-UI turned out to be a usable and effective software tool. The collaborative approach and CoastEd succeed in conveying the idea of collaborative profile specification to average users. Most types of collaborative profiles were successfully specified by users. For the initiator of the collaborative profile specification process, two types of profiles call for further research.

Overall, the approach, languages and software tools developed are shown to be effective and merit future research in that area.

Acknowledgements

Undertaking my PhD was quite a journey—both literally and figuratively. With my two huge suitcases in tow, I began by travelling to the other side of the world, to a country I had never seen before. This should be indicative of my journey throughout my PhD years. I had thought it was about doing the research, but I was wrong. Financial worries, visa worries, relationship issues, the passing away of several family members, illness of close ones; whenever I thought I had tackled a problem, a new one crept over the horizon; or fell like a bomb. Occasionally I realized that really what I was supposed to do was complete this PhD—not fight some daemons of life. But the daemons made that hard. This makes me appreciate even more the help that I received along the way: I would like to take this opportunity to thank you all!

I wish to thank my chief supervisor, Steve Jones, who gave advice when it was needed and who stayed calm and focussed when I did not. You helped me picking up my PhD pieces after my dad had passed away. This is most appreciated. You also taught me to prioritise. Thank you!

I wish to thank Geoff Holmes for his honest and unbiased advice throughout my PhD studies, even before you were on my supervisory panel, you contributed to my surviving this PhD life.

I wish to thank Annika Hinze and the Department of Computer Science for giving me the opportunity to undertake my PhD at the University of Waikato. This includes thanks for the departmental scholarship and the opportunity to work as a tutor.

Sven Bittner played a major part: You were like a rock in the sea. From providing plenty of constructive criticism for me to wrestle with—leading to great results—to the mere fact that I could always count on you. Thank you for everything!

Most of all I wish to thank my dad. He insisted on me continuing my PhD work on the other side of the world, sacrificing his last opportunity to see me and relieving me of the decision of what to do. Thank you Papa!

I also wish to thank my mum. Thank you mum, for the support via many many telephone conversations. Thank you for being such a strong person—listening to my worries and not complaining that I am on the other side of the world.

Thanks to my sister for lending me an ear when everyone else was soundly asleep!!

I wish to thank Brook Novak for the encouragement and for the pleasant company while I was finishing my research; your sunrises of inspiration often brightened up my day. And thanks for the treats on the home straight!

Thanks to Judy Bowen for sharing an office with me and also along with Anette Kira, Gabi Schmidberger and Stefan Mutter answering my endless questions and being supportive friends.

Thanks to all my proof-readers: Judy Bowen, Anette Kira, Sven Bittner, Brook Novak, Michael Walmsley, Susan Melville and her friend Elisabeth.

I wish to thank all my survey, interview and study participants, and TSG and the secretaries.

Thanks to Aimee, Heather, Susan, Janine, Nick, Claudia, Gayle, Geoff, Bryan, Claire, Jono, Emma, Malcolm, Max, Anu, Andrea, Abby and Alastair and all my other lovely friends. Thanks for so deeply believing in me when I was in doubt.

I am thankful for the awesome New Zealand ocean, the only place that could bring my mind at peace in the last three months.

This PhD journey has changed my life, it was not only a journey from Germany to New Zealand, doctoral candidate to Doctor of Philosophy, but from being one person to becoming another.

Contents

1	INTRODUCTION	1
1.1	The Alerting Approach	1
1.2	Limitations of Alerting Systems	3
1.2.1	Requirements in Practice	4
1.3	Research Questions.....	5
1.3.1	Usability and Adaptability (Research Question 1)	6
1.3.2	Collaboration (Research Question 2).....	7
1.4	Structure of this Thesis	8
2	END-USER REQUIREMENTS FOR ALERTING SYSTEMS	11
2.1	Related Work: Alerting in Health Care	12
2.1.1	Push-based versus Pull-based Approach	12
2.1.2	Clinical Information Systems	13
2.1.3	Clinical Workflow Systems	16
2.1.4	Decision Support	17
2.1.5	Conclusions	17
2.2	Survey about Alerting in Health Care	18
2.2.1	Survey	18
2.2.2	Results of the Survey Analysis	22
2.2.3	Conclusions	30
2.3	Interviews about Alerting in Health Care	30
2.3.1	Participants	31
2.3.2	Study Method	31
2.3.3	Questions	32

2.3.4	Findings	32
2.3.5	Conclusion	36
2.4	Findings of the Requirements Analysis.....	38
2.4.1	Alerting is Beneficial for Health Care	38
2.4.2	Patients Need Alerting	38
2.4.3	Flexibility Wanted for Alerts	39
2.4.4	Doctors Want Control	39
2.4.5	Better Usability Required	39
2.4.6	Collaboration Required.....	39
2.5	From Requirements to Research Questions	40
2.6	Summary.....	41
3	THE GPD L LANGUAGE FOR THE SPECIFICATION OF SINGLE-USER PROFILES.....	43
3.1	Event Models: Overview and Taxonomy.....	44
3.1.1	Distinguishing Property 1: Granularity of Events.....	44
3.1.2	Distinguishing Property 2: Structure of Events	46
3.1.3	Distinguishing Property 3: Accuracy of Events.....	48
3.2	Profile Definition Languages: Overview and Taxonomy	49
3.2.1	Primitive Profiles	49
3.2.2	Connecting Events: Composite Profiles	52
3.3	Choosing an Event Model and PDL Characteristics as Basis for GPD L.....	56
3.3.1	Event Model.....	57
3.3.2	Profile Definition Language	58
3.3.3	Temporal Concepts	60
3.4	The Profile Definition Language Underlying GPD L	60
3.4.1	Summary of Requirements	60
3.4.2	Specification of the Profile Definition Language	61
3.5	Existing Graphical Definition Languages.....	64
3.5.1	Query Languages offering Boolean Logic.....	65
3.5.2	Representation of Hierarchical Tree Structures	68
3.5.3	Temporal Representation	73
3.5.4	Discussion.....	76
3.6	The Graphical Profile Definition Language GPD L	78

3.6.1	Condition	78
3.6.2	Operators	79
3.6.3	Profile Parameters and Grouping.....	83
3.6.4	Examples with Various Concepts.....	86
3.7	Mapping GPDL to PDL	88
3.7.1	Challenges when Mapping GPDL to PDL	88
3.7.2	Transformation	90
3.8	Summary	111
4	USER STUDY OF THE GPDL LANGUAGE FOR THE SPECIFICATION OF SINGLE-USER PROFILES.....	113
4.1	Goals of the Study.....	113
4.2	Experimental Design	114
4.2.1	Method.....	114
4.2.2	Tasks.....	114
4.2.3	Procedure.....	115
4.2.4	Data Captured.....	117
4.2.5	Participants	117
4.2.6	Pilot Study	118
4.3	Quantitative Findings.....	121
4.3.1	Task Completion Time	121
4.3.2	Accuracy.....	122
4.4	Qualitative Findings	125
4.4.1	Likert Responses.....	125
4.4.2	Observations	126
4.4.3	Participants' Comments.....	131
4.5	Discussion	133
4.5.1	Accurate Profile Specification and Interpretation	133
4.5.2	Subjective Uptake of GPDL	134
4.5.3	Target Group.....	134
4.5.4	Teaching GPDL.....	135
4.5.5	Completion Time.....	135
4.6	Summary	135

5	THE EDITOR GPDL-UI FOR THE SPECIFICATION OF SINGLE-USER PROFILES	137
5.1	GPDL-UI: The GPDL Interface	137
5.1.1	Overview	137
5.1.2	Layout	139
5.1.3	Interaction Mechanisms	142
5.1.4	Profile Specification Process	145
5.2	Profile Manipulation.....	158
5.2.1	Taxonomy of Edit Operations.....	159
5.2.2	Presentation of Edit Operations	161
5.2.3	Deletion Logic	166
5.3	Automatic Grouping.....	166
5.3.1	Reasons for Grouping	167
5.3.2	Grouping Rules	168
5.4	Summary.....	175
6	USER STUDY OF THE EDITOR GPDL-UI FOR THE SPECIFICATION OF SINGLE-USER PROFILES	177
6.1	Goals of the Study	177
6.2	Experimental Design.....	178
6.2.1	Method	178
6.2.2	Tasks	178
6.2.3	Procedure	179
6.2.4	Data Captured	180
6.2.5	Participants.....	180
6.2.6	Pilot Study.....	182
6.3	Quantitative Findings	186
6.3.1	Accuracy	186
6.4	Qualitative Findings	193
6.4.1	Likert Responses.....	193
6.4.2	Observations	196
6.4.3	Participants' Comments.....	200
6.5	Discussion	203

6.5.1	Presentation of Timeframe Concept.....	204
6.5.2	Navigation	205
6.5.3	Learning GPDL and the Interface.....	205
6.5.4	Temporal versus Logical <i>Conjunction</i>	206
6.5.5	Layout and Interaction Mechanisms.....	207
6.5.6	Target Group.....	208
6.6	Summary	208
7	COLLABORATIVE PROFILES AND THEIR SPECIFICATION.....	211
7.1	Motivating Examples.....	211
7.1.1	Examples from Health Care	212
7.1.2	Other Application Areas	215
7.1.3	Implication.....	217
7.2	Related Approaches.....	217
7.2.1	CSCW.....	218
7.2.2	Recommender Systems.....	222
7.2.3	User Profiler Systems	224
7.3	The Concepts of Collaborative Profiles	224
7.3.1	General Idea.....	224
7.3.2	Specification of Underlying Concepts.....	227
7.3.3	Integration into Conventional Alerting Systems	233
7.3.4	Application Variant for Collaborative Profiles.....	234
7.4	The Collaborative Graphical Profile Definition Language CGPDL.....	234
7.4.1	Colour	235
7.4.2	Placeholders.....	235
7.5	CoastEd: The CGPDL Interface	238
7.5.1	Description.....	238
7.5.2	Specification Options	241
7.5.3	Specification Process.....	244
7.6	Summary	256
8	USER STUDY OF THE EDITOR COASTED FOR THE SPECIFICATION OF COLLABORATIVE PROFILES.....	257

8.1	Goals of the Study	257
8.2	Experimental Design	258
8.2.1	Method	258
8.2.2	Tasks	258
8.2.3	Procedure	259
8.2.4	Data Capture	261
8.2.5	Participants.....	261
8.2.6	Pilot Study.....	263
8.3	Quantitative Findings	269
8.3.1	Accuracy	270
8.4	Qualitative Findings	274
8.4.1	Likert Responses.....	274
8.4.2	Observations	282
8.4.3	Participants' Comments.....	289
8.5	Discussion	292
8.5.1	Collaboration Concept	294
8.5.2	Collaborative Profile Levels	295
8.5.3	Interaction Mechanisms.....	297
8.5.4	Language Concepts.....	298
8.5.5	Presentation.....	299
8.5.6	Test Situation	300
8.6	Summary	302
9	CONCLUSION	303
9.1	Summary	304
9.2	Contributions	305
9.3	Cross-Study Observations	307
9.4	Future Work	312
9.5	Epilogue	315
	REFERENCES	317
	APPENDIX A. PATIENT ONLINE-SURVEY	327

APPENDIX B. HEALTH CARE PROVIDER INTERVIEWS 371

APPENDIX C. GRAPHICAL PROFILE DEFINITION LANGUAGE STUDY: PARTICIPANT
WORKBOOK 385

APPENDIX D. SINGLE-USER PROFILE SOFTWARE INTERFACE STUDY: PARTICIPANT
WORKBOOK 425

APPENDIX E. COLLABORATIVE PROFILE SOFTWARE INTERFACE STUDY: PARTICIPANT
WORKBOOK 457

List of Figures

Figure 1-1: Overview of an alerting system.....	2
Figure 1-2: Overview of an alerting system using a collaborative profile.....	4
Figure 2-1: Example partial rule in PLAN.....	16
Figure 2-2: Further statuses of patient-participants.	20
Figure 2-3: Nationalities of participants.	20
Figure 2-4: Conditions of patients.	21
Figure 2-5: Which alerts do patients want?.....	23
Figure 2-6: Preference of devices.	25
Figure 2-7: In- and output.	26
Figure 3-1: EBNF of the underlying PDL.....	63
Figure 3-2: Young and Shneiderman’s filter/flow representation.	67
Figure 3-3: Example transformation of a Venn diagram into the InfoCrystal view. ...	68
Figure 3-4: Indented label representation.	69
Figure 3-5: Node-link diagram.....	69
Figure 3-6: Tree view and corresponding tree-map view of a file system.....	70
Figure 3-7: Example of the circular presentation in Sunburst.	71
Figure 3-8: Example of a Cone tree.	71
Figure 3-9: Example of the representation of a hyperbolic tree.....	72
Figure 3-10: Napoleon’s march.	74
Figure 3-11: Example of the perspective wall.	74
Figure 3-12: Example of LifeLines.....	75
Figure 3-13: Condition box.....	79
Figure 3-14: <i>Simple conjunction</i> expressed by the <i>all</i> operator.....	80
Figure 3-15: <i>Composite conjunction</i> expressed by the <i>all</i> operator.....	80
Figure 3-16: <i>Disjunction</i> expressed by the <i>some</i> operator.....	81
Figure 3-17: <i>Composite negation</i> expressed by the <i>no-occurrence</i> operator.	82
Figure 3-18: <i>Selection</i> expressed by the <i>repetition</i> operator.....	82
Figure 3-19: <i>Sequence</i> operator.	82
Figure 3-20: Absolute timeframe.....	84

Figure 3-21: Repetitive timeframe.	84
Figure 3-22: Example of the sizing of condition boxes (attributes, operator and values omitted).....	85
Figure 3-23: Grouping boxes ensuring unambiguous profiles.	86
Figure 3-24: Example profile in GPD (described in Example 18).....	87
Figure 3-25: Example profile in GPD (described in Example 19).....	87
Figure 3-26: Schema (event types) used for examples.....	90
Figure 3-27: Partial EBNF of the internal GPD representation.	91
Figure 4-1: Query experience.	118
Figure 4-2: Query frequency.	118
Figure 4-3: Original <i>repetition</i> operator.	119
Figure 4-4: <i>Sequence</i> operator.....	120
Figure 4-5: New <i>repetition</i> operator.....	121
Figure 4-6: Overview of success rates for separate tasks. “In” denotes interpretation tasks; “Sn” denotes specification tasks.....	124
Figure 5-1: Empty interface of GPD-UI.	138
Figure 5-2: Node exchange profile manipulation operations.	162
Figure 5-3: Profile manipulation operations for the <i>all</i> operator.....	163
Figure 5-4: Example of ambiguous profile.....	168
Figure 5-5: EBNF of the extension to the internal GPD representation.	168
Figure 5-6: Grouped profiles using the rules for the <i>all</i> operator (Example 29): Rule 1 (left), Rule 2 (middle) and Rule 3 (right).	169
Figure 5-7: Grouped profiles using the rules for the <i>some</i> operator (Example 30): Rule 1 (left), Rule 2 (middle) and Rule 3 (right).....	171
Figure 5-8: Grouped profiles using the rules for the <i>repetition</i> operator (Example 31): Rule 1 (left), Rule 2 (right).....	172
Figure 5-9: Grouped profiles using the rules for the <i>sequence</i> operator (Example 32): Rule 1 (left), Rule 3 (right).....	174
Figure 5-10: Grouped profiles using the Rule 2 for the <i>sequence</i> operator (Example 32).....	174
Figure 5-11: Grouped profiles using the rules for the <i>no-occurrence</i> operator (Example 33): Rule 1 (left), Rule 2 (right).....	174
Figure 6-1: Computer frequency.	181
Figure 6-2: Computer experience.....	181
Figure 6-3: Query frequency.	181

Figure 6-4: Query experience.....	181
Figure 6-5: Alerting frequency.....	181
Figure 6-6: Alerting experience.	181
Figure 6-7: Absolute timeframe.....	185
Figure 6-8: Repetitive timeframe.....	185
Figure 6-9: Overview of the success rates for each task level in percent and absolute, conditions: I=Interpretation, S=Specification, levels: S=Simple, M=Medium, A=Advanced, P=Professional.	187
Figure 6-10: Overview of the overall success rates for each task level in percent and absolute.	188
Figure 6-11: Overview of success rates for individual interpretation tasks.....	189
Figure 6-12: Overview of success rates for individual specification tasks.	190
Figure 6-13: Overview of the influence of the task order on success rates, IS=first interpretation then specification, SI=first specification then interpretation.....	191
Figure 6-14: Overview of the influence of the study duration on success rate.....	192
Figure 6-15: Overview of the timeframe edit window. Left: Absolute timeframe tab. Right: Repetitive timeframe tab.....	198
Figure 7-1: Collaborative profile definition described in Example 34.	212
Figure 7-2: Collaborative profile definition described in Example 35.	213
Figure 7-3: Collaborative profile definition described in Example 36.	214
Figure 7-4: Collaborative profile definition described in Example 37.	215
Figure 7-5: Collaborative profile within an alerting system.	225
Figure 7-6: Partial EBNF of an internal representation of collaborative profiles.....	232
Figure 7-7: Integration of collaborative profiles into alerting systems.....	233
Figure 7-8: Vague condition in CGPDL.....	235
Figure 7-9: Vague duration in CGPDL.....	236
Figure 7-10: Vague <i>repetition</i> in CGPDL.....	236
Figure 7-11: Vague timeframe in CGPDL.....	237
Figure 7-12: Vague subtree in CGPDL.....	237
Figure 7-13: Vague operator in CGPDL.....	238
Figure 7-14: Empty interface in CoastEd.....	239
Figure 7-15: Example of the job tray in CoastEd.	240
Figure 8-1: Computer experience.....	262
Figure 8-2: Query experience.....	262

Figure 8-3: Query frequency	262
Figure 8-4: Alerting experience.....	263
Figure 8-5: Alerting frequency	263
Figure 8-6: Duration pop-up window.....	264
Figure 8-7: Edit duration window, pilot study.	267
Figure 8-8: Edit duration window, full study.	267
Figure 8-9: Vague operator versus vague subtree realisation, pilot study.....	267
Figure 8-10: Vague operator versus vague subtree realisation, improvements based on pilot study findings.	268
Figure 8-11: Assign timeframe window, pilot study.....	269
Figure 8-12: Assign timeframe window, full study.....	269
Figure 8-13: Success rates for owner tasks levels (V=vague).....	271
Figure 8-14: Success rates for provider tasks levels (V=vague).	271
Figure 8-15: Influence of gender on success rate.....	272
Figure 8-16: Influence of intuitiveness on success rate.....	272
Figure 8-17: Influence of ease on success rate.	273
Figure 8-18: Influence of satisfaction on success rate.....	273
Figure 8-19: Influence of physical activity on success rate.....	273
Figure 8-20: Correlation between perceived subjective success and actual success rate.	274
Figure 8-21: Influence of mother tongue on intuitiveness.	277
Figure 8-22: Influence of mother tongue on satisfaction.	277
Figure 8-23: Influence of mother tongue on mental activity.....	278
Figure 8-24: Influence of mother tongue on frustration.....	278
Figure 8-25: Influence of gender on level of physical activity.....	278
Figure 8-26: Influence of gender on frustration.	278
Figure 8-27: Influence of profession on intuitiveness.....	279
Figure 8-28: Influence of profession on satisfaction.	279
Figure 8-29: Influence of profession on mental activity.	279
Figure 8-30: Influence of profession on physical activity.....	279
Figure 8-31: Influence of query experience on mental activity.....	280
Figure 8-32: Influence of query experience on physical activity.	280
Figure 8-33: Influence of query experience on subjective success.	280

Figure 8-34: Influence of query frequency on mental and physical activity. 281

Figure 8-35: Influence of computer experience on mental and physical activity. 281

Figure 8-36: Influence of having participated in a previous study on intuitiveness, ease and satisfaction. 282

Figure 8-37: Influence of having participated in a previous study on frustration and subjective success. 283

List of Tables

Table 3.1: Mapping of operators from GPDL to PDL.....	83
Table 4.1: Profile levels.....	115
Table 4.2: Task completion times, (N=12).....	121
Table 4.3: Task completion times computer scientists versus non-computer scientists.	122
Table 4.4: Success rates of task completion for specification and interpretation together.....	123
Table 4.5: Success rates specification tasks versus interpretation tasks, (N=12).	123
Table 4.6: Results for intuitiveness, (N=12).....	125
Table 4.7: Results for ease, (N=12).....	125
Table 4.8: Results for satisfaction, (N=12).....	126
Table 5.1: Bottom-up profile specification.....	146
Table 5.2: Top-down profile specification.....	153
Table 5.3: Overloading of the buttons for condition box and operators. Buttons are shown in the columns, current workspace component in the rows.....	165
Table 6.1: Results for intuitiveness, (N=18).....	193
Table 6.2: Results for ease, (N=18).....	193
Table 6.3: Results for satisfaction, (N=18).....	194
Table 6.4: Mental activity, (N=18).....	194
Table 6.5: Physical activity, (N=18).....	194
Table 6.6: Frustration, (N=18).....	195
Table 6.7: Subjective success, (N=18).....	195
Table 7.1: Available specification options for selected language concepts.....	244
Table 7.2: Specification of a collaborative profile (condition job, duration job and timeframe job) for the owner.....	245
Table 7.3: Specification of a collaborative profile (operator job) for the owner.....	249

Table 7.4: Specification of a collaborative profile (condition job, duration job and timeframe job) for the provider.	251
Table 7.5: Specification of a collaborative profile (operator job) for the provider. ...	255
Table 8.1: Individual and task level success rates.	270
Table 8.2: Results for intuitiveness, (N=14).	275
Table 8.3: Results for ease, (N=14).	275
Table 8.4: Results for satisfaction, (N=14).	275
Table 8.5: Results for mental activity, (N=14).	276
Table 8.6: Results for physical activity, (N=14).	276
Table 8.7: Result for frustration, (N=14).	276
Table 8.8: Results for subjective success, (N=14).	277

1 Introduction

This thesis investigates how end-users can effectively specify both single-user profiles and collaborative profiles in alerting systems. Specifically, it proposes a graphical profile specification language, GPDL, and its realisation in a single-user software application, GPDL-UI. It introduces a new collaborative alerting model and realises this as a prototypical implementation, CoastEd (Collaborative Alerting System Editor). CoastEd can be employed by users to collaboratively specify their alerting needs. The language used by this collaborative software application is a collaborative extension of the single-user version introduced in this thesis.

1.1 The Alerting Approach

Alerting systems¹ are systems that alert their users according to their specifications. These specifications are referred to as profiles and generally describe information needs. Profiles are registered with the alerting system by persons, devices or systems—components acting as so-called subscribers. Figure 1-1 shows such subscribers, as well as their profiles that have been registered with the alerting system. Profiles are defined with the aid of a profile definition language (PDL).

Information is sent to the alerting system by publishers. Information typically describes a change of the state of an object, such as a sensor or other technical device. This information in alerting systems is referred to as an event; the left-hand side of

¹ These systems are also referred to as event notification systems or publish/subscribe systems.

Figure 1-1 shows these events and their publishers. An event is marked by a time-stamp stating the occurrence time.

The alerting system is situated between its publishers and subscribers. It is responsible for filtering the events that are sent to the system against the registered profiles. If the information need described by any profile is fulfilled by incoming events, the respective subscriber is alerted. These alerts can be in various formats, such as a text message or an e-mail. The conceptual assumptions about events and profiles in an alerting system are referred to as an alerting model.

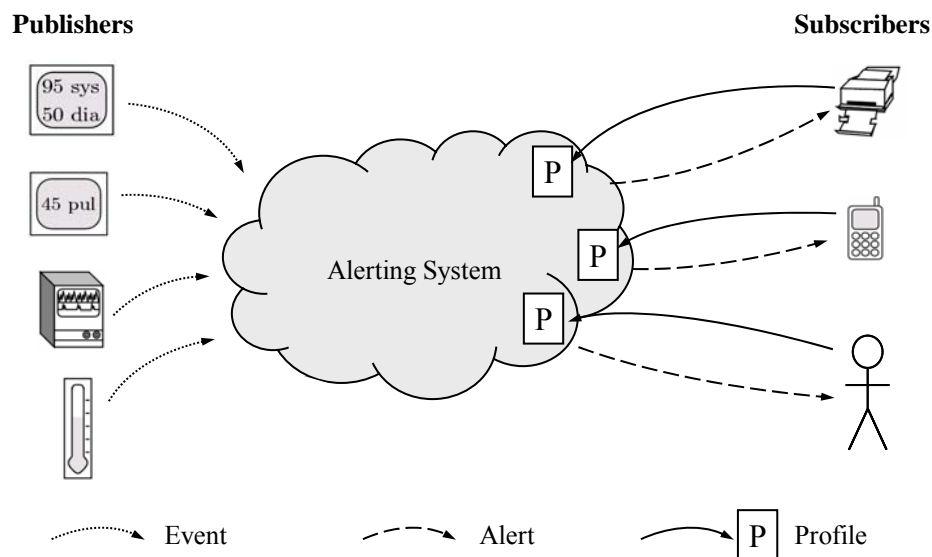


Figure 1-1: Overview of an alerting system.

Current research in the alerting area focuses on concepts underlying alerting systems. This includes how to ensure an efficient and scalable filtering process [1, 2], how to distribute events and profiles in a network to ensure a scalable distributed alerting system [3, 4], how to support alerting systems in mobile environments [5, 6] and how to provide alerting mechanisms within an architectural middleware component [7, 8].

1.2 Limitations of Alerting Systems

Currently, alerting systems are not designed for the average end-user. In order to use these systems, people have to understand the complete set of underlying concepts. For example they need to have an understanding of formal logic and time-based constraints. Otherwise it is likely that they will have great difficulty in using the specification mechanisms required for the definition of their alerting needs within profiles. Therefore a level of abstraction is needed that hides the underlying concepts that average users are unfamiliar with.

However, current alerting systems work with research-focused textual interfaces or, in some instances, are only accessible via programming interfaces. To date, no research has been undertaken that targets the provision of alerting mechanisms to a wider, non-technical audience.

Moreover, alerting systems are not designed to be used in highly collaborative work environments. They provide no support for integrating the expert knowledge of several users into the definition of a single profile. This impedes the uptake of alerting systems and their employability, particularly for specialised professions that require collaboration such as health care

Figure 1-2 shows the idea of several users collaborating in the process of profile definition: Several users share their knowledge and collaboratively specify and refine the profile. The requirement to collaboratively specify profiles is shown in the following example:

Example 1 *In health care, it is essential that various health care providers jointly work on the treatment of patients. For example, the scheduling of operations usually depends on the results provided by several parties and the consideration of factors being judged by others. Automating the scheduling process requires the collaboration between these different parties. This can be realised in a collaborative profile, involving the expert knowledge of all concerned parties to describe when an operation should be scheduled and to ensure that all factors have been taken into account.*

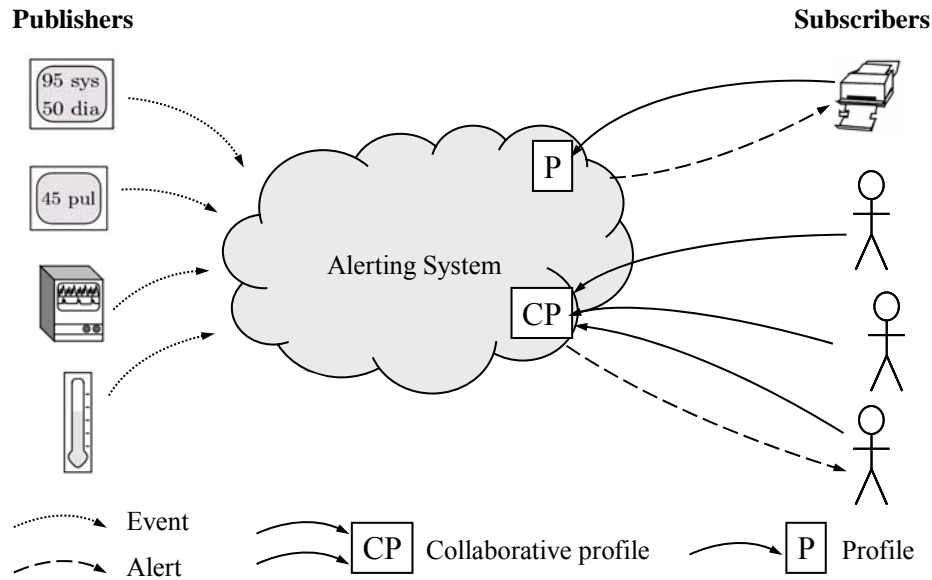


Figure 1-2: Overview of an alerting system using a collaborative profile.

1.2.1 Requirements in Practice

In the health care domain, the usefulness of alerting systems is evident.

For example, repeated oversights (however small) can be problematic in the treatment of patients with chronic conditions. Patients with chronic or long-term conditions have issues they have to deal with on a recurring basis. Examples of these are to remember to take their medication, get new medication from the pharmacy, to measure some physiological parameters or to document these measurements and other information such as pain patterns. It is onerous for them to keep track of all of these details alongside their daily routine. Thus, the compliance of these patients is not always optimal and could be improved by appropriate support, that is, by using an alerting system.

Due to their hectic work environment, health care providers are incapable of following challenging abstract mathematical processes to interact with their information systems, containing, amongst others, alerting systems. Thus, effective and usable interfaces for alerting systems are required by both patients and health care providers.

Moreover, the requirement of collaboration is inherent in health care, as it comprises the typical characteristics of a highly collaborative work environment: In the treatment of patients, it is essential for the contributing health care providers (such as general practitioner (GP), specialist and laboratory) to jointly work on their patients' treatment regimes. This avoids repeat examinations and thereby saves time and valuable resources. In addition, collaboration ensures that relevant results found by one party are considered by other parties as well. It is vital to exploit the knowledge of all parties providing treatment in order to optimise the outcome of the treatment. Patients themselves can and should contribute to this process, as they are the best experts on their own lives.

Similar patterns and requirements can be found in various other application areas. Within this thesis, the health care area is used as an exemplar application.

1.3 Research Questions

The origin of this investigation stems from first-hand experience of the health care profession and the current state-of-the-art in alerting systems research. The motivation was to improve the uptake of alerting systems for the treatment of patients with chronic conditions. In order to find crucial factors influencing this and to identify the focal point of the thesis, the requirements for alerting systems in health care had to be determined.

To target this, first-cut requirements of alerting for health care were analysed. These requirements were verified and extended by conducting an online survey with patients and interviews with health care providers. In addition, they were placed in perspective with a related work analysis.

This work addressed two major research questions. The first question works towards the claim that the usability and adaptability of alerting systems can be more effective than they are to date.

Research Question 1: *Does the graphical profile definition language GDDL and the software application developed for its use enable average users to correctly and effectively specify their alerting needs for subscription with alerting systems?*

The second question targets the support of collaboration in the profile specification process. It aims at substantiating the claim that alerting systems are suitable to support collaboration between different knowledge sources. As the realisation of the collaborative profile specification approach is based on the single-user, that is, the conventional non-collaborative approach, the first research question had to be answered successfully as a prerequisite for the research of the second question.

Research Question 2: *Are the concept of collaborative profiles and the collaborative alerting model suitable means to satisfy the requirement of collaboration in alerting? Does the representation of collaborative profiles found in this thesis allow domain experts to correctly and effectively express their alerting needs?*

Each research question raises several detailed research questions that are described in the following subsections.

1.3.1 Usability and Adaptability (Research Question 1)

The results of the survey, interviews and related work analysis suggested some major problems regarding the usability and adaptability of alerting systems. These problems are challenged by Research Question 1. This thesis claims that alerting systems can be made usable and in doing so the adaptability of the system can be increased: Through a more effective and correct specification of profiles, users are able to adapt the system to their individual alerting needs.

Currently, alerting systems use textual languages that are based on Boolean logic and other concepts that are based on formal notations. These notations are hard to use for the average user. Hence, Research Question 1 raises the following question:

- What is an effective representation for a profile definition language that enables users to specify their own profiles correctly and thereby to adapt the alerting system to their own needs?

In order to answer this question a graphical profile definition language called GPDL was developed. This language was evaluated regarding its usability and amended according to the results of that evaluation.

For the specification of profiles it is not sufficient to have a language, it is also necessary to have an interface that uses this language. This yields more questions raised by Research Question 1:

- What is an effective interface that can be used for the correct specification of profiles using the profile definition language developed in the previous step? What interaction mechanisms help to correctly and effectively specify profiles?

The research targeting this question suggested that the editor GPDL-UI can be employed for profile specification using GPDL.

Moreover, the usability of the editor was studied to substantiate the claim that the editor offers an interface that can be used for the effective and correct specification of the average user's alerting needs.

1.3.2 Collaboration (Research Question 2)

If Research Question 1 can be corroborated, the concept of collaboration can be addressed. To date, collaboration is neither integrated into alerting systems nor into its underlying alerting model. This is an insight that was confirmed by the related work analysis. However, it is a required concept in many areas in health care as has been shown by health care provider interviews. Hence, Research Question 2 raises the following detailed questions:

- Do current alerting models support collaboration in the process of profile specification? What concepts are required to support such an approach?

In the course of this research, a novel concept had to be developed—the collaborative alerting model. Along these lines, this thesis suggests a new concept, the concept of collaborative profiles. This is a means for jointly working parties to collaboratively

define their alerting needs. The realisation and representation of this concept is one of the main foci of this work.

The interviews undertaken to research the requirements for alerting systems for health care targeted this new concept in an exploratory way. They gave a first impression that this concept is going to be useful for health care staff in better supporting them in their daily routine of treating patients.

In addition to the underlying theoretical concepts, a practical representation of the collaborative concept was required. Accordingly, Research Question 2 raised two further questions:

- What is an effective representation for a profile definition language which enables users to specify profiles collaboratively?
- What is an effective interface that can be used to correctly specify profiles collaboratively using the collaborative profile definition language that was developed? What interaction mechanisms help to correctly and effectively specify collaborative profiles?

To answer these questions, the collaborative profile specification language CGPDL and the editor CoastEd that supports the specification mechanisms of that language were developed and evaluated. They serve to show that the claims made regarding the second research question hold.

1.4 Structure of this Thesis

This thesis is structured as follows: Chapter 2 presents a requirements analysis that supports the claim that more effective and usable alerting systems are needed. This analysis is based upon the example application area of health care; it contains an analysis of related work regarding alerting in this area as well as the results from a survey and interviews that were conducted.

Chapter 3 provides the basis to answer Research Question 1: After analysing related work in the general alerting area and areas relevant to the graphical visualisation of profiles, the chapter proposes the Graphical Profile Definition Language, GPDL. GPDL is a novel graphical approach to specifying profiles within alerting systems. The practical evaluation of GPDL in a user study is then presented in Chapter 4, prompting minor refinements to GPDL.

Chapter 5 deals with the topic of applying GPDL in an interface for profile specification. It presents the interface GPDL-UI, and details its interaction mechanisms as well as the approach of GPDL-UI to ease the amount of work required for specifying profiles.

Chapter 6 addresses Research Question 1: The chapter presents a user study of GPDL-UI, draws conclusions from the results, and from the feedback and recommendations of the study participants, which influenced the design of the software developed as further research.

Chapter 7 and Chapter 8 focus on Research Question 2: Chapter 7 proposes the novel collaborative alerting model, involving the important concept of collaborative profiles. As this is pioneering work in the alerting area, the chapter briefly analyses related work from the Computer Supported Cooperative Work area (CSCW) to build on the expertise and the issues identified within this community. The concepts of collaborative profiles are integrated into the graphical language CGPDL, an extension of GPDL. Chapter 7 concludes by presenting CoastEd, an editor for collaborative profiles.

Chapter 8 evaluates CoastEd and the general collaborative alerting approach in a user study. The results of that study show the effectiveness of the approaches from Chapter 7 to correctly specify profiles, answering the second research question.

Finally, Chapter 9 concludes by discussing the overall results from this thesis and suggesting possible avenues for future work.

2 End-user Requirements for Alerting Systems

This chapter presents an analysis of the requirements of alerting systems from the perspective of the users of such systems. The goal of this chapter is to provide a foundation for the research undertaken. Based on the results of the requirements analysis, the research questions presented in the previous chapter were developed.

The requirements analysis was undertaken in the area of health care, serving as a recurring example application throughout the thesis. The requirements analysis contains the following parts:

First-cut requirements [9] were identified from first hand experience in health care and a related work analysis. This part of the analysis is presented in Section 2.1. In a next step, these requirements were verified and extended with the help of an online survey [10]. This survey and its results are detailed in Section 2.2. Finally, a series of interviews with health care providers [11] were undertaken; Section 2.3 presents the results of this last step.

The requirements of alerting found in the health care area are not exclusive to this particular application domain. The overall findings of the analysis can be abstracted to general requirements of alerting systems from a user's perspective, as presented in Section 2.4. Finally, to conclude this chapter, Section 2.5 relates the findings of the analysis to the research questions of the thesis.

2.1 Related Work: Alerting in Health Care

First hand experience in the health care sector led to the idea that alerting systems can be beneficial for solving problems in health care. The kind of problems that come to mind are those issues that naturally arise in the hectic workflow in a clinic: disregard of crucial information—as there is no time in a physicians’ work to use a pull-based approach to seek for information; disorganisation due to changes of priority about how critical one patient’s condition is in comparison to another patient’s condition, or data loss due to insufficient back up arrangements. These kinds of problems could be alleviated by using reminders or automatic notifications, subsumed under the notion of alerts applied in the alerting area. Examples for alerts include the following:

- Alerts for patients with chronic conditions that support the management of recurrent tasks that are inherent to the treatment regime of chronic conditions
- Alerts for clinicians about critical parameters about patients at the emergency ward
- Alerts about incoming lab results
- Alerts about adverse drug effects
- Alerts to nurses about upcoming tests for patients

Based on this idea, the current state of the art of alerting systems in health care is analysed in the following subsections to see in how far such systems are already used and what problems their deployment might have introduced.

2.1.1 Push-based versus Pull-based Approach

Purves and Robinson [12] envisage a change from the “information seeker-finders (hunter-gatherers)” to a new generation of “tool utilizers-appliers (farmers)”. This strengthens the assumption underlying this thesis that alerting systems provide solutions to existing problems; for example, typical health care problems can be solved by

utilising the push-based approach of alerting systems (farmers) rather than the pull-based approach of traditional information systems (hunter-gatherers).

Wagner et al. [13] have found that push-based systems have proven valuable in the medical area when they compared the use of e-mail (pull-based) versus pager (push-based). Medical staff preferred using their pager rather than e-mail in order to attain information. A problem they identified is, however, that users have different preferences on what information they want to receive at what point in time. Therefore the authors concluded that user profiles would be helpful.

2.1.2 Clinical Information Systems

Looking at clinical information systems it can be seen that a number of alerting systems have been used during the last decade. Examples for such systems include the following:

Iordache et al. [14] have researched an alerting system for ambulatory and hospitalised patients that deals with laboratory results. Hripcsak et al. [15] have developed the Columbia-Presbyterian Medical Center clinical event monitor which is an alerting system that supports clinical staff by making them aware of medical events such as medical errors. Shabot et al. [16] report a wireless clinical alerting service for physiologic, laboratory and medication data; while the paper does not report an extensive evaluation of the benefit of the system, the authors report that their first impression of the usefulness of the system was confirmed.

Some research gives an evaluation of systems rather than just a description.

Staes et al. [17] cover the effect of computerized alerts on the quality of outpatient laboratory monitoring for transplant patients. Their findings show that these alerts improve the laboratory monitoring of these patients. Alerts led to more efficient, complete and timely management of laboratory information. Kheterpal et al. [18] researched how alerts impact on procedure documentation compliance and professional fee reimbursement. They point out that documentation is neglected due to the complexity of the tertiary care process. Their findings show that electronic alerts are suit-

able to improve compliance dramatically. Kuperman et al. [19] undertook a study showing that the use of alerting systems potentially results in an improvement of patients' outcomes due to faster reaction of physicians to laboratory results and events involving medication-related issues.

Some research gives information about the approaches that are applied to specify profiles.

Chen et al. [20] have developed a real-time clinical alerting system for intensive care units. The system employs a textual language for the definition of profiles by directly mapping the logical expression of a profile to a textual representation, such as:

```
HeartRate > 54 & ArtSys < 60 & FiO2 > 60 for 4 hour
```

Another system is a real-time alerting service for laboratory data developed by Poon et al. [21]. The system works with a fixed-contents language. Wagner et al. [22] report about CLEM, a clinical event monitor that works with predefined profiles; that is, there is no emphasis on the process of profile definition. Hoch et al. [23] deal with country-wide computer alerts that were sent to community physicians in order to improve potassium testing in patients receiving diuretics. There was only one kind of profile, which was predefined. The level of testing increased positively through the alerts. The time delay until testing was done after the reminder had been sent also decreased over the time of the study.

Other research covers findings concerning the usability of alerting systems:

Fung et al. [24] report about a nation-wide study in the US that analysed the impression of Veterans Health Administration primary care physicians regarding the usefulness and usability of alerts. Each facility had predefined alerts that primary care physicians could choose from. However, they could not define their own kind of alerts. In detail, their results showed the following medians (on a scale ranging from 0 to 100 with 0="strongly disagree" and 100="strongly agree"): design/interface 52.8, easy to use most computerised clinical reminders 50.0, easy to learn how to use com-

puterised clinical reminders 66.7, expected functions and capabilities are available 33.3, formats easy to use 50.0, not surprised by actions of some computerised clinical reminders 50.0, information on computerised clinical reminder screen is presented pleasantly 50.0 [24]. The findings of the study showed that the usability of the interface leaves room for improvement and the authors point out that more usability testing is required.

Saleem et al. [25] researched how different factors influence the uptake or rejection of computerised clinical alerts. Their findings entailed a number of interesting observations, amongst them some that concern the usability of alerts:

“Results: Optimally using the CR system for its intended purpose was impeded by (1) lack of coordination between nurses and providers; (2) using the reminders while not with the patient, impairing data acquisition and/or implementation of recommended actions; (3) workload; (4) lack of CR flexibility; and (5) poor interface usability. Facilitators included (1) limiting the number of reminders at a site; (2) strategic location of the computer workstations; (3) integration of reminders into workflow; and (4) the ability to document system problems and receive prompt administrator feedback.”

The first impeding factor “lack of coordination between nurses and providers” hints at the fact that the analysed clinical alerting system did not support collaboration between health care staff.

The fourth factor “lack of CR flexibility” corresponds to findings described for other systems analysed in this section. The majority of systems have predefined alerts that will pop up no matter if wanted or not. Users have no opportunity to effectively specify what information they want to be alerted about in what context. This would require a language that is easy to understand by both health care providers and patients.

Impeding factor five, the “poor interface usability”, shows that health care providers are discouraged from using a clinical alerting system due to an insufficient interface. Other systems do not even offer the option to specify individual alerts that accommodate the alerting needs of users in a flexible way. This also implies that their developers have not researched usability issues that might occur with such an interface.

2.1.3 Clinical Workflow Systems

In clinical workflow systems, the push-based alerting approach also has been applied effectively. Information has to be delivered without medical staff requesting it, as their work is too demanding to constantly reflect on all potentially required emergency actions. Any patient can show unexpected reactions; so it is not feasible to use a pull-based approach to request information; using this approach dramatically increases the workload of health care staff.

The following systems were analysed for their suitability to the push-based approach:

Wu and Dube [26] have developed PLAN, a framework and specification language with an event-condition-action mechanism for clinical test request protocols. The textual language that is offered for the specification of event-action rules is targeted at clinical staff and does not allow for the use of the system by patients. An example partial rule that can be specified by PLAN is shown in Figure 2-1.

```

Rule-Name: protocol-rule-2 ;
Designer-Name: Dr AB;Design-date:29/5/2000;
Design-Time:1417;
Rule-Type: protocol; Rule-Status: active;
Begin-Static-Rule
Zero-Time:Check-in-Day +
On day {today} IF (K > 5.1OR K < 3.4OR Delta > 4) and
  ( (UR>10 and Day<9) or UR>20 or Delta > 4) and
  ((Day<9 and CREAT>110) or CREAT>210 or Delta > 25
Do order-tests{U&E} on days {today + 1}
End-Static-Rule;

```

Figure 2-1: Example partial rule in PLAN².

A clinical alerting system for ambulatory care is described by Zheng et al. [27]. It works with predefined, dedicated alerts and no possibility to define individualised alerts is offered.

² The example is taken from

[26] B. Wu and K. Dube, "PLAN: a Framework and Specification Language with an Event-Condition-Action (ECA) Mechanism for Clinical Test Request Protocols," in *34th Hawaii International Conference on System Science (HICSS-34)*. Maui, USA, 2001.

2.1.4 Decision Support

Galanter et al. [28] conducted a study examining automated decision support for prescribing medication. Their findings showed that alerts were effective in decreasing the ordering and administration of drugs that were contraindicated due to renal insufficiency.

Steele et al. [29] report about a trial that looked at health care providers in an outpatient primary care clinic. The authors found that the health care providers in question adhered to alerts and used this information to improve their patient care. The trial worked with predefined alerts. Their purpose was to prevent medication errors and adverse drug events.

2.1.5 Conclusions

To conclude this related work analysis, its main findings can be summarised as follows.

- Alerts appear to be beneficial for the improvement of patients' treatment.
- The majority of alerting systems target health care providers rather than integrating patients in their own treatment and thereby fostering their responsibility for their treatment.
- The majority of alerting systems only allow fixed alerts. Sometimes users can select from a pre-defined set of alerts; nevertheless, they cannot explicitly specify the alerts they would like to use and what kind of context should influence triggering that alert.
- For the few cases where users can choose which alerts and reminders they would like to use, poor usability of the interfaces involved has been observed. Researchers have not reported details about this; however, they indicated that further usability evaluation would be required.
- Related work reports that health care providers are not sufficiently coordinated.

Based on these findings, first-cut requirements for alerting in health care were analysed. The analysis targeted supporting patients, doctors, nurses and other health care providers with a mobile alerting system to support patients with chronic conditions. Details about this analysis are published elsewhere [9].

The first-cut requirements analysis formed the foundation for an online survey that is the focus of the following section.

2.2 Survey about Alerting in Health Care

The goal of the online survey presented in this section was to test and verify the requirements for alerting systems in health care. As the survey targets this broad area, it gave insights into issues that go beyond the scope of this thesis. Nevertheless, a summary of the most important results is given in this section to allow for a full overview of the requirements of the envisaged alerting systems in health care.

This section is structured as follows: Section 2.2.1 gives background information about the online survey and its participants. The analysis of the survey results is undertaken in Section 2.2.2. Finally, Section 2.2.3 draws conclusions from the survey.

2.2.1 Survey

The survey was conducted during the last two weeks of February 2005. It was directed towards patients, doctors, nurses and computer scientists employed at IT departments of hospitals. A summary of the results can also be found in [10].

Methods of the Survey. The survey worked with a mixed approach since it followed two objectives:

1. Verification of those requirements that had already been found [9]
2. More thorough exploration of the topic to identify requirements that had not yet been identified

Hence, quantitative methods were used that were complemented by a number of qualitative questions. For this mixed approach, the concurrent nested strategy as suggested by Creswell [30] was chosen. Quantitative methods guided the survey to promote the participants' understanding of the issues involved before any qualitative questions were asked.

Participants. In the online survey a total of 73 questionnaires were filled in by the participants. Three of them were obvious outliers which were removed from the result set³.

Status. Sixty of the remaining 70 participants were patients, with 32 of them also indicating a second status. The distribution of these statuses is given in Figure 2-2. About half of the patients had no other status; 13 were also computer scientists and therefore well versed with technical devices. There were four patients that also were nurses; their responses did not appear to be significantly different from those of any other patient. Among the participants who gave another status, positions were named such as dietician, biologist and family member of a patient.

The other 10 questionnaires were filled in by one doctor, two nurses and seven computer scientists. Since the number of completed questionnaires for doctors, nurses and computer scientists was very small, it was decided to only analyse the results of the patient questionnaires.

Personal Background. Thirty-five of the patients who filled in one of the 60 questionnaires gave their gender as male and 23 as female. Two people did not specify their gender. The participants were between 18 and 79 years old with a mean of 40 years and a standard deviation of 15.65.

³ Two of them were identified as outliers because they were duplicates of two other results (the participants accidentally submitted their questionnaires twice). The third outlier was due to a spammer, who announced themselves having submitted a faked questionnaire.

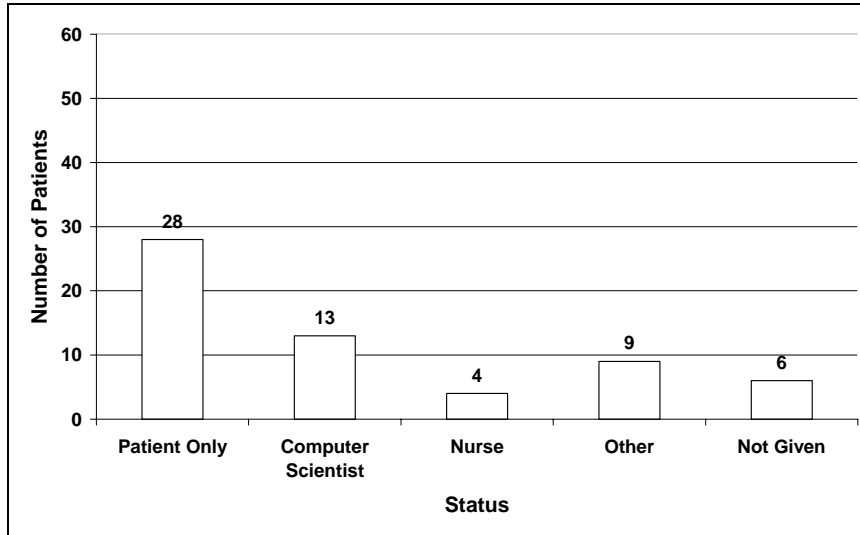


Figure 2-2: Further statuses of patient-participants.

Their spread of nationalities is depicted in Figure 2-3. The two major groups were Americans and Germans with the Americans constituting the bigger group. These groups were followed by Canadians and New Zealanders. Six people did not specify their nationality. This roughly reflects the distribution of nationalities of participants in the targeted online communities.

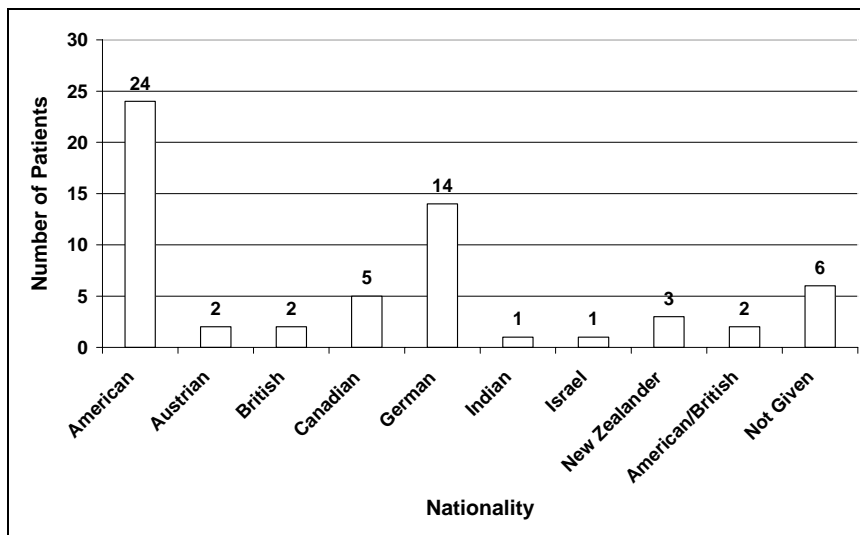


Figure 2-3: Nationalities of participants.

Conditions. An uneven spread of conditions was found among the participants. The group with the highest number of participants were people suffering from inflammatory bowel diseases (IBD—for example Crohn's disease and ulcerative colitis). Glaucoma patients formed the second strongest group. This was followed by the group of diabetes patients. The actual distribution of conditions is depicted in Figure 2-4.

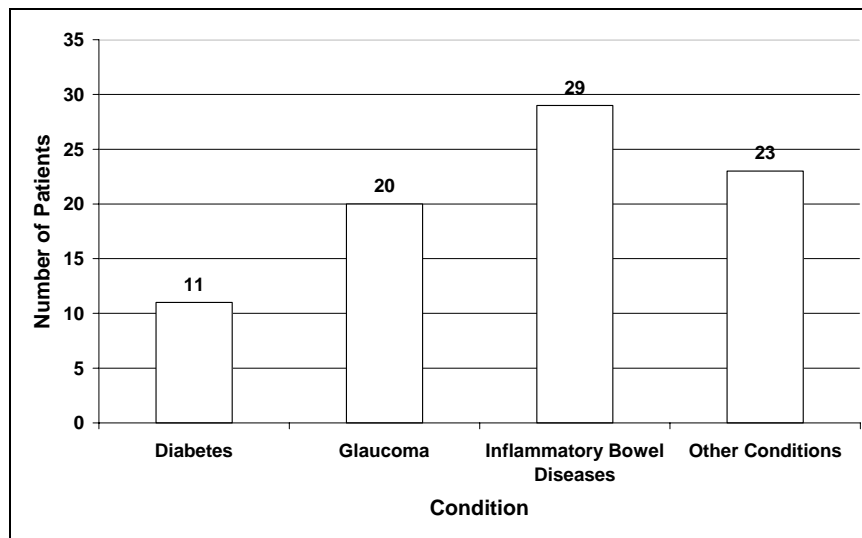


Figure 2-4: Conditions of patients.

Computer-literacy. The participants of the survey were all computer-literate. On a scale from 1 to 5, they judged their own experience in using computers and other technical items with a mean of 4.09 and a standard deviation of 0.98.

The median of their hours of computer use was 30 hours per week. The participants have been using computers for 14 years as the median.

Entry into the Field. Information about the survey was posted to online communities of support groups for patients. This was done via a number of English and some German speaking mailing lists and newsgroups covering various conditions. The conditions that were addressed are diabetes, glaucoma, AIDS, hypertension, polycythaemia, leukaemia, arthritis and IBD. In order to guarantee the appropriateness of the posting to the list, the announcement was sent to the moderators of the mailing lists. As a consequence, for some diseases the announcement was not forwarded to

the list, for example for several lists supporting AIDS patients. For two conditions the announcement was posted through actual patients, who were well known to the list members by their postings to the respective lists. The conditions affected by this are IBD and glaucoma.

Additionally, several doctors, nurses and computer scientists working in clinics were contacted directly via e-mail and asked for their participation.

Data Sources and Analysis. An online survey collected the data for the analysis within a period of two weeks. The results of the questionnaires were gathered anonymously. The quantitative analysis followed Moore and McCabe [31]. The qualitative analysis followed the grounded theory approach, as described by Creswell [32].

2.2.2 Results of the Survey Analysis

The survey results and their analysis are presented in the following subsections. First the quantitative results are discussed. This is followed by a qualitative analysis.

Quantitative Results. The quantitative questions that were asked were based on the first-cut analysis of the requirements for a mobile alerting system to support patients with chronic conditions. The intention of the questions was to gain an insight on the validity of this analysis.

Desired Information. The participants were asked which information they would like to be stored and available for them personally within the system. They were very interested in storing information on their current medications and adverse effects of their medications. Furthermore, they favoured the storage of possible interactions of their medications with other medications and their parameters measured by themselves or clinical staff. However, they were indifferent about the storage of their personal data in combination with the information mentioned before. Several explanations can be envisaged for this: It could be due to the fact that their personal data is something they do not need to be alerted of; it could be due to data security reasons (even though patients were asked to abstract from this when answering this question);

or it could be due to the fact that the participants did not consider the need for an association between their medical and personal data in order to be used successfully for their treatment.

Types of Alerting. The survey asked what issues patients would like being alerted about. The replies to this clearly showed that overall alerting is something which is desired by computer-literate patients. Only 5% were not interested in receiving alerts (cf. Figure 2-5). Nevertheless, these patients were still interested in a support system for the management of their condition that they could use for storing and querying of information.

Figure 2-5 shows that more than two thirds of the participants are interested in alerts about new educational material and in reminder alerts, such as being reminded of doctor appointments, getting a new prescription or to take their medicine at the correct time. Only about one third of the participants wanted to be reminded to take the correct amount of medicine, or the correct type of medicine. This can be explained by the fact that this information is not so easily forgotten: patients rather feel the need to be reminded that they should take their medication; not how much of it, or when to take it.

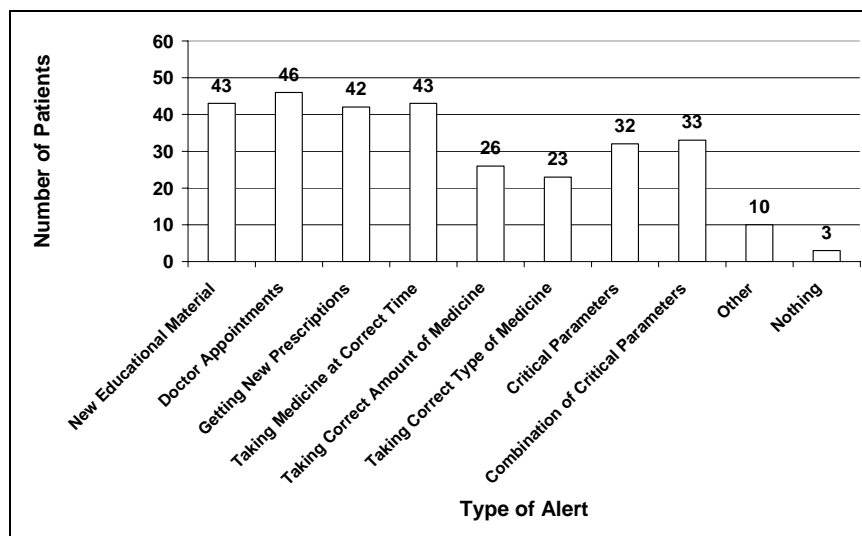


Figure 2-5: Which alerts do patients want?

About half of the participants were interested in alerts concerning their health-related parameters. These personalised alerts are triggered according to definitions doctors have specified, for example when a patient's blood sugar values are too high.

Alerting Signal. The way patients want their attention to be drawn to any new kind of information was evaluated. Participants could choose between audio, visual and vibration signals. They also had the option to select between the modes “home”, “night” and “business”. Furthermore, the selection of the signal had to be made depending on the medical priority of the alert.

As expected, for a high medical priority, when out on business, patients wanted a vibration signal. When at home or during the night, they prefer an audio signal to indicate a high priority alert.

For a medium medical priority, patients still prefer to be alerted with a vibration signal when they are out on business. At home and at night they favour visual signals for medium priority alerts. This most likely correlates with the fact that patients do not want to be disturbed at night.

For all low priority alerts, visual signals were desired by the participants.

The overall percentage of patients indicating a visual signal was relatively stable independently of the medical priority and the time of the day. Probably patients have chosen visual signals, since these signals might imply the utilisation of e-mails as the medium, which allows for easier documentation. Also, visual signals are less interruptive than audio signals.

Alerting Medium. Patients were asked in what way they would like to receive condition-related information. Their replies showed similar tendencies for their preferences at home and at night. For all times of the day, patients favoured receiving e-mails in an equal proportion. A possible explanation for this might be that patients want to archive alerting messages. However, they hardly ever wanted e-mail as the sole medium of alerting. This was the case only for a low medical priority. Both

printouts and automatic entries into electronic health records (EHR) are also preferred in relatively equal proportions independently of the medical priority. It appears that once patients have opted for an automatic documentation of their condition, they prefer to stay with this decision. As expected, patients were less interested in voice messages when being out on business. For a high medical priority the favoured alerting medium was always text messages.

Alerting Device. The participants were asked which devices they would like to use to get informed about condition-related issues. They generally favoured devices such as their home computer, the phone, a mobile phone or a mobile device (for example integrated into a watch). The exact distribution of devices is given in Figure 2-6.

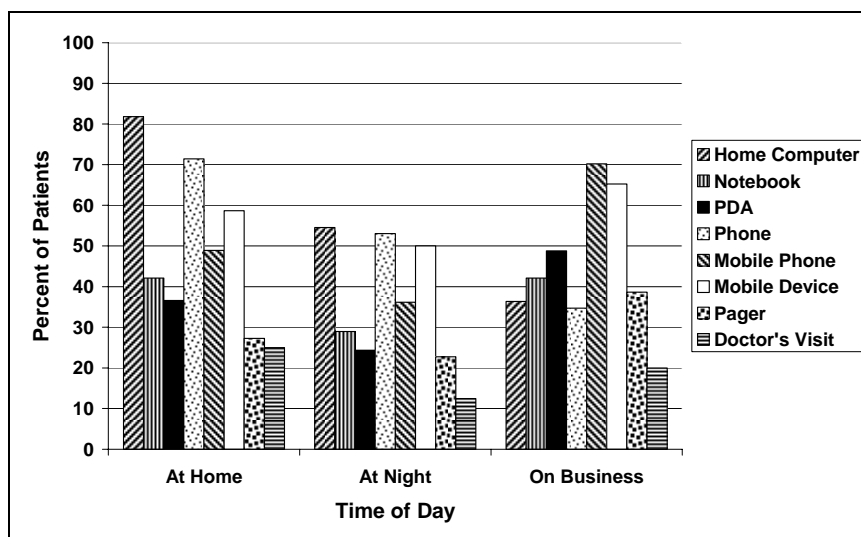


Figure 2-6: Preference of devices⁴.

It was striking that when being out on business patients preferred using their personal mobile phone or mobile device, and did not like using office devices provided by their employers. The assumption is that patients do not want to give their employer the ability to find out about or to track their condition. Another reason could be the desire for a strict separation of personal and professional life.

⁴ The participants had the choice to tick several boxes.

The interest in using a home computer remained relatively stable independently of the time of the day. This is most likely due to the fact that patients would like to use their home computer to document their condition themselves. In the qualitative part of the survey this assumption was corroborated by the fact that a high number of patients wanted the possibility to plug in the mobile device into their home computer via USB.

Input and Output. Figure 2-7 shows the patients' preferred forms of output. Most participants preferred a visual output which corresponds to their current way of working.

However, several patients have special needs and preferred a voice feedback. Among these participants approximately 78% either had glaucoma or diabetes. Fifty percent of all glaucoma patients, approximately 36% of all diabetes patients and approximately 14% of all IBD patients wanted the possibility to use voice feedback. This clearly suggests the need for a voice feedback due to bad eyesight.

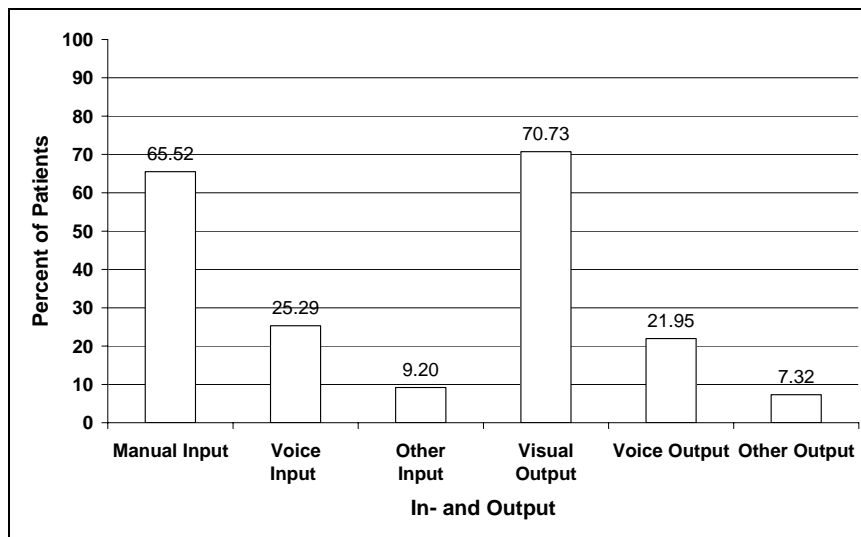


Figure 2-7: In- and output.

The preferences for giving input into the system were less obviously determined by the patients' conditions than the answers for the preferred output. An overview of the answers is shown in Figure 2-7. The tendency which seems to emerge though is that

patients with vision problems also seem to want voice input. However, they want voice input to a lesser extent than they want to have voice output. Voice input is also favoured by patients who have problems with movements such as patients with MS or rheumatoid arthritis. The majority of the participants who wished for another input had mistaken this question for a question asking about what devices they would like to use and answered things such as palm device, website, computer or PDA.

Level of Functionalities. The patients were asked as to their needs regarding the desired level of functionality the system should offer. It was possible to select several options. None of these selections was clearly favoured by patients: Almost 40% wanted to have basic functionalities and around 54% wished for advanced functionalities. Another 66% were looking for default functionalities.

Another question asked concerned the trade-off between a basic level of functionality with an easy-to-use interface in comparison to a sophisticated level of functionality with an interface one has to learn. Here the participants opted for the exact middle. This indicates that patients are willing to put time into understanding the interface offered by the system.

Confidentiality. The participants were overall extremely concerned when asked for their concern about general issues of confidentiality and data security. The participants had to tick one value on a scale from 1 to 5 for the issues of data integrity, authenticity, non-repudiation and confidentiality, respectively. These choices resulted in a median of 5 for each of the three first categories. For the matter of confidentiality the median was 4.

The quantitative questions in the survey aimed at the verification of the first-cut requirements analysis and the determination of the relevance of each requirement. Additionally, the survey contained qualitative questions to identify further requirements for alerting in health care.

Qualitative Results. The grounded theory approach [32] that was chosen for the evaluation consists of three phases: the open coding phase (categorisation of given answers), the axial coding phase (relating categories to identified central phenomena) and the selective coding phase (building a “story” around the categories—within the background of our study the development of context and research questions).

The open coding phase was started by segmenting the issues that were covered by the participants into single information units. Afterwards, they were grouped into five categories: usability, technical requirements, design (aesthetics), stored information and alerting. Two categories could be identified as central phenomena for the axial coding: alerting and stored information.

The relationships between these central phenomena were determined and the influences of the remaining three categories on these central phenomena were discovered.

Finally, in the last phase it was observed that the participants stated the need to store a great amount of data without specifying the way of accessing it. Since they excluded to be alerted about them, it was deduced that a category “querying” was needed. In the end phenomena and categories were related to each other.

Results in the Categories. From the evaluation of the questions regarding the category alerting functions, it became obvious that patients’ needs are extremely diverse. The context of patients has to be taken into account for the alerting functions. It covers areas such as the conditions of patients, the current location of patients, their mood, general disposition, educational background and the time of the day. To be able to realise context-dependent alerting, personalised information has to be stored. Another reason for the need of personalised information storage is an issue which, even though not directly stated by the participants, could be drawn from the fact that patients wanted a lot of information to be stored but not to be alerted about it. Many of the patients wanted to have an electronic journal to jot down their symptoms, dietary needs or pain patterns. Also healthy lifestyle advice and lost/found information (for example owner details and acoustic help to find a lost device) are information to

query for when needed rather than to be alerted about. However, the mode of access (querying versus alerting) might vary depending on a patient's condition: A patient with IBD might want to be alerted about dietary information; for glaucoma patients such an alert is extremely unlikely and they would prefer to query the system for this kind of information.

When asking for further requirements for the mobile alerting system, issues mentioned repeatedly were concerns regarding the category of the usability of the system. In general, patients favoured an easy-to-use system. They were extremely concerned about the size of the device carrying the system. They all wanted a pocket-sized device. Any smaller size would yield to a negative trade-off between the ability to easily take along the device and an easy-to-use and easy-to-see interface.

Some participants also gave comments on the signal that should be used. The participants who commented on this issue said that they would favour a vibration signal. The participants expressed a range of positions concerning the in- and output of the device carrying the system: Some clearly wanted a visual display whilst others mentioned vision problems and would therefore prefer an audio in- and output. In contrast, others with poor eyesight still wanted a visual display but required this to be easily readable.

The participants also expressed some concerns regarding the category of technical requirements for such a mobile alerting system. Their statements covered topics such as low energy consumption, problems in cases of no reception, the ability to employ the system internationally and the possibility to connect the mobile device via USB to a home computer. A further issue which appeared to be important for some participants was the category of aesthetics of the device. Nevertheless, one participant mentioned the exact opposite, namely, that the design does not matter at all.

2.2.3 Conclusions

This section presented an international online-survey regarding the requirements of a patient-based mobile alerting system for the support of patients with chronic conditions.

With respect to this thesis, the main results of the survey evaluation can be summarised as follows.

1. Overall, patients with chronic conditions strongly support the idea of having the opportunity to use alerting to support them in their treatment-regime.
2. Patients want flexible alerts, that is, they want to be able to personalise them and adapt them to their individual circumstances.
3. The usability of the proposed mobile alerting system is a highly relevant issue due to the special needs of patients with chronic conditions.

The participants of this survey were mostly patients, even though it explicitly targeted patients, health care staff and computer scientists working in health care. To also consider the requirements of the most important target group apart from patients, health care providers, it was decided to undertake selected interviews with doctors and nurses as the final step of the requirements analysis.

2.3 Interviews about Alerting in Health Care

The interviews described in this section elicited the views of health care staff about supporting their work with alerting systems. They also gave the researcher the opportunity to get first-hand insights into the topic of collaboration in health care.

The interviews were conducted with doctors and nurses in New Zealand and in Germany. The interviews in Germany were carried out over a one week period in June 2005. Most of them took place outside of the health care staff's work premises. The interviews in New Zealand were carried out at a university health care centre over a

one week period in January 2006. Three doctors and two nurses were interviewed. Details on the results of these interviews can also be found in [11].

2.3.1 Participants

The survey captured age, gender, nationality, experience with technology as well as the position and field of practice of the participant. Nevertheless, not all of these were relevant for the evaluation. Status, gender and nationality were found to be significant. The participants were recruited through direct contact. An attempt to recruit participants via notices on notice-boards failed.

The participants ranged in age from 31 years to 61 years (mean 46 years). Three of them were female, two male. Three were German and two from New Zealand of which one was of South African origin. Two of the doctors were specialists for haematology/oncology and ENT. The other one was working in general practice. One nurse was a specialist nurse for heart disease, the other working in general practice.

The interviewees rated their experience in computer use on a 5-point scale with five representing the highest experience. They selected values between three and five (mean 3.6). They reported computer use between 5 and 45 hours a week (mean 21 hours). They also said that they had been using computers for a time period between 7 and 15 years (mean ~11 years).

2.3.2 Study Method

Two of the interviews were conducted outside of the participants work premises and thereby gained a more informal character than the other interviews. Nevertheless, the direction of the results did not seem to be different than those from the more formal interviews, which took place in the participants' individual offices.

The participants were briefly introduced to the idea of a mobile alerting system and collaborative profiles. This was done using a figure of such a system to illustrate its use. Afterwards, they were shown figures of some examples for collaborative profiles. The main part of the interview followed, asking the participants for their opin-

ion and their expectations regarding the concept of collaborative profiles and the usefulness of mobile alerting systems. Finally, they were asked to think about examples of their daily work for which employing collaborative alerting would be beneficial.

During the interview only the interviewee and the researcher were present, with the researcher taking notes of the participant's replies.

2.3.3 Questions

The researcher used four questions to lead the participants through the interview with:

1. To what extent do you consider mobile alerting systems to be useful for chronic patients and their health care providers?
2. To what degree is the concept of collaborative profiles useful for chronic patients and their health care providers?
3. What do you expect from collaborative profiles?
4. Can you give me examples from your daily work in which a mobile alerting system with collaborative profiles could support you?

Nevertheless, in several interviews it was not necessary to ask all of these questions as the participants elaborated on the topics covered by these questions out of their own account after the initial introduction to the topic.

2.3.4 Findings

All participants have taken up a positive stance on the suggested mobile alerting system and on the concept of collaborative profiles.

The interviews looked at the usefulness of a mobile alerting system for patients with chronic conditions, and at the usefulness of collaborative profiles in combination with this system. The applicability of collaborative profiles to the interviewees' daily work was investigated and the expectations of the participants regarding collaborative profiles were analysed.

Usefulness of Mobile Alerting System. All participants agreed that the idea of a system that alerts patients and health care staff of important condition-related issues is very feasible and that there is a need for this kind of system. However, they varied in their opinion for which kind of scenarios they would like to employ such a system.

The nurses that were interviewed were most happy about the prospect of such assistance, whereas doctors were also concerned about the consequences of potential errors made by the users of such a system. One doctor therefore preferred using the system only for simple applications. One of the nurses in particular appreciated the idea for application in monitoring tasks, because contrary to humans the alerting system does not tire out. It would be especially good for treatment support for nurses in intensive care units with patients having diseases such as chronic heart conditions. Another nurse stated that a mobile alerting system would be extremely suitable for supporting patients being cared for in their homes, who otherwise would forget their medication when they are on their own. She also put forward that it would be very helpful if the stored parameters would automatically be transferred to the patients' health records. She said that then health care staff could see whether patients were within the parameters—whether they would still be healthy or not—and that this way the health care provider could provide follow-up care if required.

One issue that arose was the fact that such a system lacks the intuition experienced personnel have acquired. Moreover, they wanted to be assured that the system sends reminders or alerts rather than to function as a decision help taking over decisions from health care staff.

All participants of the study agreed that if the system would be used correctly, that is, if patients are compliant and data is input correctly by health care staff and patients, it would be very helpful.

Usefulness of Collaborative Profiles. Most interviewees highly favoured the idea of supporting collaboration between patients and different parties of health care providers with the help of collaborative profiles.

All of the participants were open to the use of the concept for simple cases. One conservative doctor would use it for more complicated scenarios only if all participating parties knew about the general processes and workflow of all the participating parties. Otherwise, he seemed to be worried about losing control over the situation.

However, all other participants were positively inclined towards the idea without too many concerns when it is being used for general applications. For example they stated that it was excellent to support collaboration because currently in the New Zealand health care system, person A would not know what person B was doing and vice versa. They assessed that this concept would be helpful, because for people working as some sort of health professional any multidisciplinary involvement would be really good. It would also be excellent for avoiding unnecessary jobs and be useful for supporting communication processes—which are highly required for patients with chronic conditions. Another good use would be the incorporation of laboratory results into decisions over a patient's medication regime. They appreciated the system's usefulness for supporting the work in a nursing home, presuming it was ensured that it will be kept track of who is manipulating alerts. This is to avoid settings being erroneously changed.

All of this put together leads to the general impression that the idea of collaborative profiles is very good as its application can save time and money. This way, extra resources would be available to look after patients.

Expectations of Health Care Staff regarding Collaborative Profiles. It turned out that the participants, even though they highly appreciated the ideas of this research, had trouble relating information about their daily work directly to the idea of collaborative profiles. Some of their replies concerned the general idea of a mobile

alerting service rather than the support of collaboration. However, the first two of their statements did relate to collaboration:

Communication. In particular, the nurses that participated in the interviews were extremely interested in a support of communication in all directions. Also one of the doctors pointed out the interest in communicating with other doctors.

Liability. This issue was the most controversial amongst the participants. German health care staff seemed to take on a different stance than health care staff from New Zealand. Germans were deeply concerned about liability problems in the collaboration, whereas New Zealanders worried less about this matter.

Germans strongly insisted upon the fact that it has to be ensured that liability issues are resolved. When several people are participating in the specification of alerting needs, it has to be clearly defined who is responsible for what.

New Zealanders agreed upon the fact that “it would probably be helpful to have someone that was definitely in direct control of it, and so that you’d know you know people being done properly. [Meaning patients are looked after properly.]” Though, it would not matter who this person was as long as someone did have the control over the whole process. They also reported that it would gradually develop who is responsible for the definition of alerts. They stated that “It is always like this when you follow the holistic approach” and suggested that one could determine who is responsible within a certain liaison and let the patient approve of that person.

Adaptability. Several of the participants also emphasised the importance of adaptability of the system and its profiles. The system should be adaptable to the various needs of their individual patients. However, they welcomed the idea to offer default profiles so that health care providers have a basis from which they can start elaborating. It should be possible to modify these default profiles.

Other Expectations. Other issues that arose covered concerns about handling the system and its impact on treatment. One doctor wanted to make sure that essential

information can not be eliminated without their knowledge. The importance of the data being entered correctly was pointed out by one of the nurses. In addition, an automatic transfer of the data to a patient's EHR would be essential because this offers the possibility to check what is going on in the treatment of patients. Moreover, the alerting signals should be easy to recognise and easy to remember, for example a melody would be suitable, one of the nurses remarked. One doctor called attention to the requirement to support patients in developing their independence regarding the management of their condition, rather than being driven into lethargy concerning their condition management.

Further Application Scenarios for Collaborative Profiles. Clarifying the question of further application scenarios for the concept of collaborative profiles turned out to be the most challenging. The participants confused applications for the mobile alerting system with single-user profiles with the suggested collaborative approach.

However, some remarks were related to the collaborative concept. On the one hand, they suggested using the system for organisational matters in a clinic, for example in order to coordinate x-ray examinations with nurses and patients. On the other hand, they suggested using it to support collaboration between different medical establishments. In the example they gave, they were referring to a number of patients who were living out of town. The nurse said it would be handy to be linked up to the patients and their General Practitioners (GPs) and to be able to access their data. This would improve the continuity of care.

2.3.5 Conclusion

Four lessons could be identified from this research. The results of the interviews led to the first three lessons; the research around the interviews led to the last lesson.

Lesson 1. Health care providers appreciate the idea of an alerting system for patients with chronic conditions—provided an appropriate use of the system is guaranteed.

Lesson 2. The participants positively accepted the concept of collaborative profiles. They have an idea of what they consider most important for the realisation of this concept: A multi-directional communication, a well-defined approach for liability issues and the possibility to adapt the system to their patients' individual needs were the most important features for them.

Lesson 3. Even though health care providers are affirmative of the idea of collaborative profiles, they have difficulties to immerse themselves into the concept of collaborative profiles and cannot properly relate it to their work in a short interview. Thus, there is a need to realise the concept and to evaluate it in a longitudinal study rather than to merely discuss it.

Lesson 4. The suggested concepts are not only applicable to the area of health care but also to other domains which require a collaboration of several people. This occurs for example in e-commerce, facility management or tourism. It therefore was concluded to continue researching on an abstract level in order to gain universally applicable results.

Relating the interviews to the scope of this thesis, this section yields the following findings:

- Health care providers find the idea of being supported by an alerting system beneficial for their work.
- Doctors and especially nurses appreciate the idea of supporting patients with chronic conditions in their treatment through a mobile alerting system.
- Health care staff also want to be able to choose what kind of alerts they employ, that is, flexibility in the specification process is required.
- It is extremely important for some of the doctors to retain control over the treatment process and decision process at all times.

- Health care staff acknowledge that a mobile alerting system has to suffice certain usability criteria in order to be effectively employed.
- Health care staff support the idea of collaborative profiles.

The following section summarises the findings of all three parts of the requirements analysis presented before and relates them with each other.

2.4 Findings of the Requirements Analysis

Taking into account all findings from the previous three sections, the following issues were identified as being relevant to this thesis:

2.4.1 Alerting is Beneficial for Health Care

Alerting systems have been shown to be beneficial for improving numerous treatment-related issues. Examples include health care providers' faster reaction to laboratory results, or decreasing the number of drug prescriptions that might lead to adverse effects in patients. Thus, it is worthwhile to further explore and consider the application of alerting systems in health care.

2.4.2 Patients Need Alerting

It could be noted that the majority of existing alerting systems target health care providers—mostly clinicians.

Shifting the point of view from the support of clinical staff to the support of patients with chronic conditions, it can be observed that such patients also have to manage tasks similar in complexity to those performed by clinicians. They have to keep track of their regime of medications and doctor appointments and have to check parameters such as blood sugar or blood pressure. Furthermore, depending on their condition they might have to scrutinize their dietary intake and to take appropriate actions in the event of any abnormalities. Despite these time- and concentration-consuming obliga-

tions, patients with chronic conditions have a life and thus managing their condition has to be accommodated into it.

Therefore, it is worthwhile to further research possibilities to make alerting systems suitable for both patients and health care staff.

2.4.3 Flexibility Wanted for Alerts

The majority of reminders and alerts that currently are offered to health care providers are predefined alerts. At best, users can choose a subset of alerts they want to use from an offered selection. However, depending on the current circumstances, for example time of day, treating physician, patient background or age of patients, different alerts are desired. This necessitates a greater flexibility in the profiles used for alerting.

2.4.4 Doctors Want Control

The most pressing issue identified for doctors was their need for staying in absolute control. They do not wish to carry the responsibility for the treatment of a patient if they are not entirely certain that they can fully influence that treatment.

2.4.5 Better Usability Required

Some studies that have been undertaken showed that one of the reasons for the failure of the adoption of alerting systems in some clinics is poor usability of the interface of alerting systems.

This problem is intensified by the fact that patients with chronic conditions often have special needs that make a badly designed interface even harder to use than it is for the average user.

2.4.6 Collaboration Required

A major problem of modern health care is the lack of collaboration between the treating parties. Just to name a few problems, this leads to ignoring crucial results, de-

layed treatment and repeat examinations. Obviously, patients have to be included in this collaboration.

2.5 From Requirements to Research Questions

With these findings and requirements in mind the research questions of this thesis were developed:

Alerting systems can be beneficial for the area of health care (Section 2.4.1). However some major issues could be identified that must be addressed in order to improve the successful uptake of alerting systems. These issues are reflected in the research questions of this thesis (Section 1.3).

Adaptability. Current alerting systems lack adaptability to varying contexts and circumstances:

- Current systems can not be easily employed for patients. They can only be used in the dedicated way they have been designed to support health care staff (Section 2.4.2).
- More flexible, personalised alerts are desired by current users of alerting systems (Section 2.4.3).
- Doctors feel the need to be in control of all possible alerting triggers at all times in order to guarantee an optimal care for their patients (Section 2.4.4).

Thus, more flexibility in the specification of alerts is required that can express whatever a user (patient or health care staff) is interested to be alerted about. This way, full control of the alerting triggers could be guaranteed.

This led to the following question, constituting one part of Research Question 1: What is an effective representation for a profile specification language which enables users to specify their own profiles and thereby to adapt the alerting system to their own needs?

Usability. Alerting systems lack the usability that is required by health care providers and patients (Section 2.4.5).

This leads to the question of what an effective interface is that can be used for the specification of profiles using the profile definition language envisaged before? This question is also incorporated into the formulation of Research Question 1.

Collaboration. Collaboration is an unavoidable concept in health care (Section 2.4.6). Nowadays, patients are treated by a team of health care providers rather than by a single doctor. These specialists, GPs, nurses, pharmacists, health insurance providers, physiotherapists etc. have to co-operate in order to provide the best possible care and in order to work economically.

The following questions were brought up through this need for collaboration, all of them triggering Research Question 2:

Do current alerting models support collaboration in the process of profile specification? What concepts are required to support such an approach?

What is an effective representation for a profile definition language which enables users to specify profiles collaboratively?

What is an effective interface that can be used to correctly specify profiles collaboratively using the collaborative profile definition language that was developed? What interaction mechanisms help to correctly and effectively specify collaborative profiles?

2.6 Summary

This chapter contained a detailed analysis of the requirements of alerting systems in health care. The intention is to provide the research in this thesis with a proper foundation, and to show the usefulness and requirement of the concepts and software that are developed throughout this thesis.

The requirements analysis contained three parts, reflected within the structure of this chapter: Section 2.1 described an analysis of related work, Section 2.2 described the results of an online survey and Section 2.3 described interviews that were conducted.

The most pressing requirements that have been identified within this analysis directly lead to the research questions of this thesis, as demonstrated in Section 2.4 and Section 2.5.

The requirements analysis focused on the application domain of health care. Nevertheless, the identified requirements of an effective representation for profile definition languages, and the support of collaboration in alerting in general and in profiles in particular do exist in other application domains as well. Providing substantial evidence (that is, evidence based on surveys, interviews etc.) for these requirements in a variety of areas apart from health care is clearly beyond the scope of this thesis. The amount of research on alerting (cf. Chapter 3) provides evidence for the general usefulness of alerting mechanisms. For the requirement of collaboration, Chapter 7 lists further areas with explicit examples for the requirement of collaboration.

3 The GPD Language for the Specification of Single-user Profiles

This chapter proposes a novel profile definition language (PDL) for defining single-user (that is, non-collaborative) profiles in alerting systems. The language—Graphical Profile Definition Language (GPD)—is graphical in nature, and the intent is that it eases the task of profile specification for users who may be expert in an application domain, but not in alerting systems.

Different communities have been researching alerting mechanisms [33]: Among them are the publish-subscribe, the active databases and the data streams community. They work with differing assumptions, naming conventions and concepts that evolve over time. The first two sections of this chapter organise these existing approaches to discover commonalities and differences between them. Section 3.1 focuses on concepts for events, Section 3.2 focuses on profiles.

The next two steps to GPD are undertaken in Section 3.3, identifying those of the previously organised concepts that are relevant as a basis for GPD, and Section 3.4, defining a non-graphical PDL that can be used as an intermediate layer between GPD and existing alerting systems.

After this preliminary work, Section 3.5 analyses work that is relevant to the development of GPD. Section 3.6 introduces GPD. Finally, Section 3.7 describes how GPD can be mapped to the underlying intermediate PDL (allowing the mapping to other appropriate alerting systems).

3.1 Event Models: Overview and Taxonomy

There are different possibilities of how events can be represented in an alerting system. These assumptions about events are referred to as event model [3]. In the scope of this research three distinguishing properties were identified regarding the body of existing work: the granularity of events, the structure of events and the accuracy of events. Depending on which values are used for these distinguishing properties, they follow a different event model.

3.1.1 Distinguishing Property 1: Granularity of Events

The granularity of an event is the level of detail that can be used to specify those parts of events that can be filtered by an alerting system. The following example illustrates this concept:

***Example 2** New material has arrived at a medical library. The material is always characterised by three attributes: It concerns a particular disease, it has been written for a particular audience (that is, it refers to a certain level of education) and it has been written in a particular year. The library uses an alerting system to inform users about material that is of interest to them. Depending on what level of granularity the underlying event model uses, the system would exploit either several pieces of information about the respective publication (disease, audience and publication year) or just the most distinguishing piece of information (for example the disease).*

It can be observed in the literature that granularity has become finer over time.

Coarse-grained Events. Coarse-grained events are used by the channel-based publish-subscribe approach and by the subject-based publish-subscribe approach.

Channel-based alerting is one of the older alerting approaches. Here, events can be published to a channel. Channel-based alerting supports the option for users to subscribe to a particular channel such as a network address. This could be an address of a multicast group. Examples for the channel-based approach are the early CORBA Event Service Specification [34] and the Java Distributed Event Specification [35].

The subject-based alerting approach is sometimes also referred to as topic-based alerting. It is an extension of the channel-based approach. Publishers tag the event with a subject string. These subjects are matched using string-matching mechanisms. Usually, users have to subscribe to each subject separately and need to explicitly publish their events for each subject using the unique name of that subject. A variant is to allow several subjects per event. Examples of subject-based systems are Information Bus by Oki and colleagues [36] and Jedi by Cugola and colleagues [37].

Fine-grained Events. Fine-grained events are used by various classes of alerting approaches: content-based publish-subscribe, alerting over data streams and continuous queries.

In content-based alerting, events contain several attributes as a description that can be used by subscribers to specify their information needs. In Example 2, events about medical information contain values for three attributes: one for the disease, one for the audience and one for the publication year. Typical systems representing content-based alerting include Siena developed by Carzaniga et al. [3], Elvin developed by Fitzpatrick et al. [38], the Gryphon project [1, 39], Hermes by Pietzuch and Bacon [40], Keryx developed by Brandt and Kristensen [41], the Le Subscribe project [2], the Padres project [42] and Rebeca by Mühl and colleagues [43].

Data streams are continuous streams of data items that pass streaming applications (for example financial tickers) in real-time. The general goal when monitoring data streams is to analyse recent data items. For monitoring the data, it is sliced into time-windows, for example the last 100,000 items or items from the last 10 minutes can be related in the analysis. Items usually consist of several attributes; therefore the data (events) can be classified to have a fine granularity. Examples for data stream research are [44] by Golab and Özsu, and [45] by Gray and Nutt.

Another approach is the use of continuous queries such as researched on by Liu et al. [46, 47] and by Chen et al. [48]. These are systems that have been developed to monitor changes in web documents, databases and general files. That means users will be

notified (alerted) if new data has been added or data has been updated. The items to be monitored can be described by several attributes. Therefore events in these systems, describing changes in data, are classified as fine-grained events.

3.1.2 Distinguishing Property 2: Structure of Events

Event specifications are either modelled as a flat structure, such as an unordered set of attribute-value pairs, or as a hierarchical structure, such as an XML document. Additionally, the structure could be open or restricted by some kind of schema.

Flat Structure. The most prominent examples for event models employing a flat structure are the original content-based publish-subscribe approaches, for example by Carzaniga et al. [3] and Mühl et al. [43], data streams as used by Golab and Özsu [44], and continuous queries as used by Liu et al. [46] and by Chen et al. [48]. They all possess the capability of modelling events as a specification that comprises what can be generalised as attribute-value pairs.

Other approaches that work with a flat event structure are channel-based publish-subscribe, such as [34], [35] introduced before, and subject-based publish-subscribe, such as Information Bus [36] and Jedi [37]. Both approaches specify events using only a single attribute such as a network address or subject string (see Section 3.1.1).

Hierarchical Structure. Events can also be described containing a hierarchical structure. A general model is event messages containing an XML document as description, for example, those described by Pereira et al. [49], Altinel and Franklin [50], Chand and Felber [51], Chen et al. [48], and Yang et al. [52]. The difference from a flat structure is that—besides the values of attributes or the content of elements—the hierarchy of elements might convey information relevant to subscribers.

Structural Restriction. Apart from the general distinction between a flat or a hierarchical structure, the structure of events can be constrained by other restrictions. This generally is helpful to coordinate publishers and subscribers. If subscribers have an idea of the structure of possible events, it is easier for them to specify profiles that

can be matched. Clearly, the specification of data types is part of the structural restriction.

The specific kind of restriction that is being used depends on the particular assumptions of an event model, as described in the following overview. Nevertheless, all of the presented approaches to structural restriction serve the purposes of coordinating subscribers and publishers and ensuring the use of a common semantics.

Type-based Restriction. The type-based approach can be seen as an extension or supplement to the content-based approach. This concept was introduced by Eugster [7]. The main characteristic is that profiles and events specify an event type. Incoming events are filtered by the alerting system according to their type. In practice, the type-based approach is usually used in combination with the content-based approach: The (flat) structure of an event is restricted by its event type. That is, an event type describes what attributes are contained in an event of this type. Such an event then contains values for exactly these attributes.

When introducing sub-typing, subscribers can receive alerts about events by specifying a general event type. This way, users do not need to subscribe separately to each type of event they are interested in.

Database-related Restriction. Research working with the concepts of databases, such as undertaken by Liu et al. [46], implicitly employs some structural restriction. Events are represented as some entity (or table) and thus need to fulfil the underlying database schema. This schema can be seen as describing the existing attributes in events.

Pre-defined Subjects. Subject-based approaches, such as [36, 37], allow tagging an event with a subject string. A restriction employed with this approach can be to only allow event tags from a predefined set of tags. A further extension is to organise predefined tags within a tree-structure with the goal of some tags being specialisations of other tags (similar to the sub-typing approach described before).

Hierarchical Restriction. When employing a hierarchical structure for events, it is possible to restrict the allowed attributes as well as the allowed hierarchical structure. When taking the general representation of an event as an XML document (as in approaches like [48-52] described before), these documents would need to fulfil a particular schema, for example described in XML Schema. Different types of events could then define different schemas.

3.1.3 Distinguishing Property 3: Accuracy of Events

The third property that distinguishes alerting approaches is the accuracy of events in the system: Events can be seen as having an exact and precise specification or as having an approximate or vague specification.

Exact Events. For exact events, all information that is represented by an event is unambiguously given. This approach is assumed by the majority of research. For example the content-based alerting approach in conjunction with event types assumes that an event contains exactly one value for each attribute of its type, or a hierarchical approach in conjunction with a schema assumes that an event contains a mandatory attribute or hierarchy element.

Vague Events. Events can also be seen as containing vague or approximate information.

Liu and Jacobsen suggest an alerting model supporting uncertainties [53] that is based on the exact content-based approach. They use probability theory and fuzzy set theory to support fuzziness. This allows users to specify approximate values in both profiles and events, such as the price of a medication is cheap, expensive or reasonable. What is meant by such approximate values and how they relate to other values is specified somewhere else in the system.

Other realisations for allowing vague events are that events do not need to specify values for all attributes of their type, or events can contain an interval instead of a

fixed value to represent uncertainty, or some parts of the hierarchy of elements are omitted.

If approximate information in events is allowed, it needs to be defined how approximate events are matched to profiles by the system.

3.2 Profile Definition Languages: Overview and Taxonomy

The previous section organised and structured the differing assumptions and notions for the definition of events within alerting systems. Also for the subscribers in these systems, the different approaches and notions of various communities can be subsumed to general ideas and concepts. This is the goal of this section, which is structured into the description of the two concepts of primitive profiles and composite profiles.

3.2.1 Primitive Profiles

Primitive profiles are the means for subscribers to restrict individual events to those of interest. The restrictions that can be applied by subscribers to constrain events depend on the event model that is being used by the alerting system. According to the previous section, information in events can be represented by what has been subsumed as attributes and by a potential hierarchy of these attributes. Hence, primitive profiles need a means to formulate restrictions on the attribute part and a means to formulate restrictions on the hierarchy part. Similarly to events, primitive profiles can represent either a vague or an exact specification of user interests. The following three subsections detail different approaches for these areas.

Restricting Attributes. Looked upon in an abstract way, events for all event models contain one or several attributes and their values. This is regardless of their granularity and their structure. Primitive profiles thus require and offer a means to restrict such attributes. These range from string matching for subject-based approaches, such as in the Jedi system [37], to the more general concept of predicates in other approaches, such as the Siena system [3].

This general concept of a predicate contains a reference to a particular attribute and further specifies an operator and an operand. An alerting system can then evaluate predicates on events, that means predicates are evaluated based on the values of the attribute they reference. With this evaluation as a basis, the system can decide whether a profile matches an event.

As events can contain several attributes, profiles can contain several predicates. Various systems, for example Gryphon [39], [1], Hermes [40], Le Subscribe [2], Padres [42], Rebeca [43] and Siena [3], only support predicates with an implicit conjunctive semantics, that is, for a matching profile all predicates need to match. In other systems, for example BoP [54], Elvin [38] and the system described by Campailla et al. [55], primitive profiles can arbitrarily combine several predicates using Boolean logic.

Restricting Hierarchies. When dealing with events that contain hierarchical information, profiles have to be able to pose restrictions on the hierarchical structure. Most work in this area is found for events representing XML documents.

XML-based Restrictions. Using XML-based query languages within profiles supports the specification of patterns that are matched against incoming events. These patterns typically contain an expression specifying a path in the hierarchy and may be extended by conditions (similarly to predicates) matching against attributes.

In recent years, a variety of XML-based query languages evolved in academia, including Lorel by Abiteboul et al. [56], XDuce by Hosoya and Pierce [57], XPathLog by May [58], XQL by Robie et al. [59], YAXQL by Moerkotte [60] and XML-GL by Ceri et al. [61]. There are several standards from the World Wide Web Consortium, including XML-QL [62], XPath [63] and XQuery [64].

Research on alerting systems can apply these languages in profile specifications: XPath is used by the systems WebFilter by Pereira et al. [49], XFilter by Altinel and Franklin [50] and XNet by Chand and Felber [51]. NiagaraCQ, by Chen et al. [48], applies XML-QL. XQuery [64] is used in the system described by Yang et al. [52].

The common goal of these approaches for alerting is to allow profiles to describe restrictions on events as XML documents, covering both restrictions on attributes and their hierarchical structure.

Other Restrictions. In subject-based alerting approaches that work with predefined trees of subjects, a hierarchical restriction is implied by the subject specified by the profile. A potential matching semantics is that profiles specifying a particular subject (parent in a tree) automatically describe their interest in all descendant subjects (descendants in a tree).

The same general idea is used in type-based alerting approaches, such as type-based publish-subscribe [7], allowing sub-typing among event types.

Accuracy. Similarly to the different options for the accuracy of events (Section 3.1.3), primitive profiles can be specified in an exact or in an approximate way. Different approaches can be seen to introduce concepts of vagueness.

Wildcards. One approach to introduce vagueness into profiles is the use of wildcards, offering various possibilities:

- A simple type of wildcard is profiles that do not restrict an attribute, having the semantics that all values are allowed for this attribute.
- Another type of wildcard can be seen by allowing subscribers to specify regular expressions within predicates. Examples for such an approach are Elvin [38] and Jedi [37].
- Subject-based alerting with a tree of subjects allows for the introduction of wildcards: Instead of explicitly naming a complete path in the tree of subjects in a profile, some parts of this subject tree might be left out. Events with any value in the subject specification being left out by the profile would then be considered to be matching.

- Generalising the previous idea of subject-based alerting leads to the option to allow some kind of wildcard in the hierarchical restriction of profiles. For example XPath [63] offers wildcards for the selection of unknown nodes.

Uncertainties. In [53] Liu and Jacobsen suggested an approach of modelling uncertainties in alerting systems (Section 3.1.3). To model vagueness, profiles can for example describe an interest in cheap, expensive or reasonable medication instead of providing a specific value. The approach is based on probability and fuzzy set theory. How vague specifications are resolved by the system is modelled independently of the actual profiles.

ApproXQL. With the query language ApproXQL and the PDL ApproXFilter, a vague XML-based alerting approach was developed by Hinze et al. [65]. The goal of this work is to allow subscribers to express imprecise profiles. The general approach of ApproXFilter is to alter profile restrictions (by renaming, skipping or inserting predicates or path restrictions into profiles) and thereby to base the matching of events on these altered profiles as well.

3.2.2 Connecting Events: Composite Profiles

It is also possible that the restriction of a profile covers several individual events, which cannot be expressed by a primitive profile. This extended kind of profile is referred to as a composite profile. Generally a composite profile can contain several primitive sub-profiles, describing the individual events of interest. There exist different means to combine primitive sub-profiles and thus to connect events. Additionally, several parameters can be used to specify how this connection should take place in detail. The following subsections introduce these concepts.

Means of Connecting Events. Composite profiles can connect events over their content or over their times of occurrence. In practice both options, content and time of occurrence, will be used together.

Content. Connecting events over their content means to formulate corresponding predicates that target different events. This is described in the following example:

Example 3 *Let us assume events of two types: One type for blood pressure readings (type `CardioReading`) and another type for ESR readings (type `EsrReading`). Both event types define various attributes describing the actual reading. The patient name is assumed to exist in both types, identifying who this reading belongs to. If a composite profile is used to describe some relationship between blood pressure and ESR readings from Mr Smith, it could connect both events as follows:*

```
CardioReading.Name = Mr Smith and  
EsrReading.Name = Mr Smith
```

More generally, to build this connection for any patient, the composite profile could contain:

```
CardioReading.Name = EsrReading.Name
```

Connecting events by directly relating values of attributes of different types with each other corresponds to the notion of SQL-like joins and is a direct extension of research inspired by the database area, such as [45].

The connection of events by specifying corresponding predicates (over their content) is directly applicable to any system without extending the notions of predicates for primitive profiles.

Time. Events can also be connected over their times of occurrence, that is, their timestamps. This introduces different operators for specifying composite profiles. They cover operators that are purely time-related (time operators); but they also cover operators based on Boolean logic (Boolean operators), potentially being extended by a time component (Boolean-time operators). Both time operators and Boolean-time operators can be specified to only be fulfilled if the events matching their operands occur within a specific duration.

- Time operators: It is possible to filter events according to their order of occurrence. This may mean that a composite profile uses the order of different events as a restriction (*sequence*), or it selects a certain occurrence of multiple occurrences of an event (*selection*), such as the first occurrence, last occurrence, n^{th} occurrence or a range of occurrences. The duration parameter of these operators specifies the time difference that is allowed between the events being connected.
- Boolean-time operators: Events can be combined using an extended version of the Boolean operators *conjunction* and *negation*. They are extended by the time-dimension. The *conjunction* of a composite profile (*composite conjunction*) only matches if all of its operands (that is, sub-profiles) match within the time specified by the duration parameter. The *negation* in a composite profile (*composite negation*) is matched if the event that is described by the sub-profile does not occur within the specified duration.
- Boolean operators: The *disjunction* in a composite profile (*composite disjunction*) is used analogously to the classic definition. It does not require a duration parameter as the *disjunction* matches as soon as *any* of its sub-profiles matches, that is, as soon as any event matching one of the sub-profiles occurs. Hence, the specification of a duration is futile as the matching of only one event is required to take place.

More technical definitions of composite operators for connecting events can be found in works by Zimmer and Unland [66], Hinze and Voisard [67] and Jung and Hinze [68].

Time and Content. The most flexible way to specify composite profiles is to employ a combination of content- and time-based connection constraints. This yields the greatest flexibility and use to the user. For example a profile could specify the following alerting need:

Example 4 *Alert if there is a bad blood pressure reading for Mr Smith and a bad ESR reading for Mr Smith within 10 minutes. The duration of 10 minutes describes a*

time-based connection requiring both events to occur in any order (composite conjunction). The predicates specifying that Mr Smith's readings should be monitored describe a content-based connection of event messages.

Parameters for Connecting Events. There are different parameters that describe the matching process of registered profiles and incoming events further. Different authors propose different parameters. The subsequent overview covers parameters introduced by Zimmer and Unland [66] and by Hinze and Voisard [67].

Consumption Mode. The consumption mode for a composite profile defines whether matching events should be disposed of or can be used again in the context of another connection of events. If disposed, there are two possibilities: One is to delete all events which occurred before the matching event. The other possibility is to only delete those events that have really taken part in the matched event. Another approach is to not dispose of any event.

Traversion Mode. The traversion mode describes the direction in which events are traversed for matching. This could be either with the flow of time or against it. Depending on this mode, for example the first or the last occurrence of an event that matches a sub-profile would be used for the connection described by the composite profile.

Concurrency Mode. The concurrency mode describes whether the events matching the sub-profiles of a composite profile can be interleaved by other events matching the profile, or whether all of these events have to occur before the matching of another composite profile is considered.

Coupling Mode. The coupling mode determines whether it is allowed that events matching the sub-profiles of a composite profile are interrupted by other events that are irrelevant for this composite profile or whether they may not be interrupted.

Duplicate Handling. Duplicate handling describes which event occurrences out of several duplicates are regarded for matching. Events are called duplicates if they are

of the same event type and specify the same values for their attributes. Event occurrences that are interrupted by events of the same type but with at least one different value are not referred to as duplicates.

These five parameters can be combined to two independent sets of parameters, one set including consumption mode, traversal mode, concurrency mode and coupling mode, and another set including consumption mode and duplicate handling.

Jung and Hinze compared these two sets of parameters and showed that they can be used to express the same semantics. For details refer to [68, 69].

Accuracy. For composite profiles, existing research assumes exact profile specifications.

Looking at the approach of collaborative profiles, suggested in this thesis in Chapter 7, from the perspective of accuracy reveals that collaborative profiles contain approximate information as an intermediate stage during the refinement process (exact profiles are eventually registered with the system). The possibilities to express vagueness introduced in this thesis cover primitive profiles and composite profiles, including content- and time-based concepts.

3.3 Choosing an Event Model and PDL Characteristics as Basis for GPD

Existing PDLs are not suitable for end-users. Some examples for textual PDLs from the healthcare area were given in Section 2.1. The general-purpose alerting approaches described in Section 3.1 and Section 3.2 report to apply research-focussed textual languages if considering a representation of profiles at all. Interfaces to these systems are mainly programming interfaces that describe a profile definition as a function call that contains the textual profile representation. Section 3.6 proposes

GPD that aims to be a tool for the average end-user to effectively specify their alerting needs.

This section gives insights into the choice of the event model underlying GPD. It also describes the characteristics of the PDL serving as a basis for GPD. Both the underlying event model and the PDL-characteristics need to be extendable to the concept of collaboration, being required for the second research question of this thesis: Collaborative profiles need to contain all those features that are offered by single-user profiles.

3.3.1 Event Model

Event models are characterised by three distinguishing properties, as introduced in Section 3.1. The following subsections choose those realisations of these properties that are most appropriate for the graphical language GPD.

Granularity. The granularity of the event model that underlies the implementation of an alerting system determines a range for the potential precision of the matching process.

For the application of collaborative profiles to be worthwhile it is favourable to use an event model that offers a high granularity and a PDL that can use the offered granularity. Otherwise the collaborative concept mostly would be obsolete as several users can only apply their expert knowledge if there is the possibility to base this knowledge on a fine-grained event model.

Coarse-grained models, as channel-based and subject-based alerting approaches, leave little room to transport rich information within events. Fine-grained models, however, allow the existence of several information units, such as attributes, within event messages. Further work will thus be based on a fine-grained approach.

Structure. The most common kind of structural concept for alerting systems is a flat structure of events. The use of a hierarchical event structure is more complicated than the flat approach and does not show advantages with respect to the research

goals of this thesis. It thus suffices to work on flat-structured events; an extension to a hierarchical structure could become part of future research.

A structural restriction serves as a means to coordinate publishers and subscribers (Section 3.1.2). For the chosen structure, a type-based event model provides the general concept to restrict the sets of attributes allowed in different kinds of events, that is, in events modelling different types of information.

The event model will thus be based on a flat structure of events, involving a type concept for structural restrictions (also referred to as event schema).

Accuracy. In the application scenario driving this research, health care, it is extremely relevant that information is well-defined. In health care staff interviews (cf. Section 2.3 or [11]), doctors pointed out that it is highly important for them that collaborative profiles are resolved in an exact way with no blurriness whatsoever.

This prerequisite for profiles implies the same semantics for events in the system: If an event is defined in a vague fashion, the system cannot ensure an exact matching process, even if profiles would be defined in an exact way. An accurate event model is thus the choice for this research.

3.3.2 Profile Definition Language

Next to an event model, GPD must be based on concepts regarding primitive and composite profiles. The following subsections identify those concepts that are most suitable for this research.

Primitive Profiles. The most common kind of concept used for the specification of profiles in alerting systems is predicates that are being combined by Boolean operators (Section 3.2.1). *Conjunction*, *disjunction* and *negation* are sufficient as these connectives can express all other Boolean operators. This model for primitive profiles is naturally applicable to the chosen event model.

Within predicates of profiles, attributes of events can be restricted by comparison operators. For the scope of this thesis, the operators greater than, greater than or equal, less than, less than or equal, equal to and not equal to suffice for the realisation of collaborative profiles.

Composite Profiles. Composite profiles describe the capability to express an interest in several events being connected (Section 3.2.2).

For the realisation of collaborative profiles, one does not necessarily require the possibility to express such a connection. However, there are applications which might benefit from collaborative profiles that need composite profiles. For example, in health care it is extremely likely that results from different examinations will have an influence on the notification or alert about a certain health state of a patient. As this is the application domain that was used as motivation for the research of this thesis, it would be beneficial to realise the possibility to connect events for collaborative profiles. Both content- and time-based connection concepts should be supported by GPD.

In conjunction with composite profiles, parameters that further refine the definition of profiles can be offered (cf. Section 3.2.2). It might sometimes be desirable for a user to have the choice between several parameters, for example, for the consumption mode or certain modes of duplicate handling. However, within the scope of the research goals, the free choice of these parameters for composite profiles by the user can be neglected.

Accuracy. As detailed in Section 3.3.1, collaborative profiles need to be resolved in an exact way and be subscribed to the alerting system. However on the way of specification, there is a state, where the profiles must be capable to store vague information that later on can be refined by other information providers. If a user defines a sub-profile that will later be refined by the expert knowledge of another user, a certain kind of wildcard could represent the missing knowledge before the sub-profiles

are resolved into the collaborative (exact) profile. Details about this are given in Chapter 7.

3.3.3 Temporal Concepts

It is impossible to realise collaborative profiles without some form of temporal concept. Events inherently require timestamps to allow the determination of an order of occurrence. This is in turn a requirement to connect events over the time dimension with the aid of composite profiles.

As profiles should only be valid for a particular period of time, profiles should be given the possibility to define a timeframe stating this time of validity. The inclusion of temporal concepts is thus required for this research.

3.4 The Profile Definition Language Underlying GPD

This section presents the specification of the PDL underlying GPD. The following subsection briefly summarises the requirements for this language, identified before.

3.4.1 Summary of Requirements

The properties of the underlying event model and PDL that were decided to be most relevant to GPD are as follows:

- Fine-grained granularity
- Flat and typed structure
- Accurate events and profiles (with an option to extend the language to be capable of holding vague information at a later stage when working on collaborative profiles)
- Capability of specifying primitive profiles and composite profiles, including the content- and time-based connection of events

Fulfilling these requirements, the event model generally follows the content-based alerting approach (Section 3.1). This means an event is defined by an event type, a set of attribute-value pairs and a timestamp. The event type specifies what attributes are defined within this event.

The following section details the PDL.

3.4.2 Specification of the Profile Definition Language

The PDL given in the following contains those language features considered relevant for the basis of an effective graphical PDL, and for the basis of a PDL supporting the concept of collaboration. The language given in the specification can be mapped to other PDLs, provided these languages include the applied concepts. Such a mapping can be used to apply the graphical and collaborative concepts, being introduced later on, within existing alerting systems.

In summary the underlying PDL comprises the following features:

- Primitive profiles
 - Formed out of predicates
 - Boolean operators to combine several predicates: *conjunction*, *disjunction*, *negation*
 - Targeting typed events
- Composite profiles
 - Composite profiles are formed out of other composite and/or primitive profiles and connection operators
 - Operators: *sequence* (with duration parameter), *conjunction* (with duration parameter), *disjunction*, *selection* (with duration parameter), *negation* (with duration parameter)

- Time concepts
 - Duration: Can be given in months, weeks, days, hours, minutes, and seconds
 - Timeframe: Can specify absolute dates; also can specify repetitive time intervals using months, days or times

Following is the grammar of the PDL. It is described in an EBNF-like notation, as defined in [70]:

```

compositeProfile : compositeExpr timeframe ;

compositeExpr : booleanExpr type+
  | '(' 'SEQUENCE' duration? compositeExpr
    compositeExpr+ ')'
  | '(' 'DISJUNCTION' compositeExpr
    compositeExpr+ ')'
  | '(' 'CONJUNCTION' duration? compositeExpr
    compositeExpr+ ')'
  | '(' 'SELECTION' '[' INTEGER ']' duration?
    compositeExpr ')'
  | '(' 'NEGATION' duration compositeExpr ')'
  | '(' compositeExpr ')' ;

booleanExpr : (
  '(' attribute ',' comparisonOperator ',' value ')'
  | '(' '|' booleanExpr booleanExpr+ ')'
  | '(' '&' booleanExpr booleanExpr+ ')'
  | '(' '!' booleanExpr ')'
  | '(' booleanExpr ')'
  ) ;

type : 'TYPE' STRING | IDENTIFIER | INTEGER | TEXT ;

duration : '{'(INTEGER 'M')? (INTEGER 'w')?
  (INTEGER 'd')? (INTEGER 'h')? (INTEGER 'm')?
  (INTEGER 's')? '}' ;

attribute : STRING | INTEGER | TEXT ;

comparisonOperator : '<' | '>' | '=' | '<=' |
  '>=' | '!=' ;

```

```

value :    STRING | INTEGER | TEXT ;

timeframe : ( '{' timeframeAbsolute '}' )?
              ( '{' timeframeRepetitiveMonth '}' )?
              ( '{' timeframeRepetitiveDay '}' )?
              ( '{' timeframeRepetitiveTime '}' )? ;

timeframeAbsolute : timeframeDate ','
                    timeframeDate ;

timeframeDate : (INTEGER '/' INTEGER '/' INTEGER) |
                 'UNDEFINED' ;

timeframeRepetitiveMonth : MONTH ',' MONTH ;

timeframeRepetitiveDay : DAY ',' DAY ;

timeframeRepetitiveTime : TIME ',' TIME ;

MONTH : 'JAN' | 'FEB' | 'MAR' | 'APR' | 'MAY' |
         'JUN' | 'JUL' | 'AUG' | 'SEP' | 'OCT' | 'NOV' |
         'DEC' ;

DAY : 'MON' | 'TUE' | 'WED' | 'THU' | 'FRI' |
        'SAT' | 'SUN' ;

TIME : INTEGER ':' INTEGER ':' INTEGER ;

INTEGER : NUMBER+ | (SIGN NUMBER+) ;

STRING : (NUMBER | LETTER)+ ;

TEXT : '\\"'~( '\\"' ) * "' ;

SIGN : ('+' | '-' ) ;

NUMBER : ('0'..'9') ;

LETTER : ('a'..'z' | 'A'..'Z') ;

```

Figure 3-1: EBNF of the underlying PDL.

An example follows of a profile given in the textual representation of the grammar described by these rules:

Example 5 Consider the following information need: The doctor wants to be alerted if a patient shows an irregular heart beat. He also wants to be alerted if the patient is jogging and has a systole higher than 200 and a diastole higher than 130. This information is captured in events of type Cardio Reading. Following the presented language, this profile is expressed as follows:

```
( |
  (&
    ("Systole", >, 200)
    ("Diastole", >, 130)
    ("Situation", =, "Jogging")
  )
  "Heart Beat", =, "Irregular"
) "Cardio Reading"
```

A profile specified with this PDL uses explicit parentheses, as known from the Cambridge Polish notation (used in Lisp)⁵. Profiles defined with this language thus can always be unambiguously resolved.

Having introduced the underlying PDL, the remainder of this chapter focuses on graphical PDLs.

3.5 Existing Graphical Definition Languages

The goal is to design a user interface that can be used by average users to correctly and effectively specify profiles in a graphical manner. As a foundation for this, an analysis of related work was undertaken.

To date, no graphical PDLs are used in the area of alerting (covering the different communities working on alerting concepts). This is due to the fact that some research focuses on service frameworks that offer application programming interfaces. The majority of research (in publish-subscribe, event-based systems, data streams and continuous queries) focuses on efficient matching, scalability and other middleware-

⁵ Polish notation is only parentheses-free for operators of fixed arity.

oriented topics. No research has been undertaken to offer an effective user interface for alerting.

Therefore related research areas were analysed. The problem at hand, that is, the representation of profiles in a graphical manner, can be subdivided into three areas that relate back to the major areas covered in the specification of the underlying PDL (Section 3.4.2):

- The representation of Boolean expressions: Primitive profiles use Boolean operators; Boolean operators also are a subset of the operators used for the combination of composite profiles
- The representation of hierarchical tree structures: Nested expressions—as found in composite profiles—commonly are represented as tree structures
- The representation of time concepts: Timeframe and duration are time concepts; the two non-Boolean operators (time operators) for composite profiles cover time-oriented concepts

Hence, research areas that target these problems were examined: Querying languages normally can be used to describe Boolean expressions. The representations of hierarchical tree structures as well as of data over time have been widely discussed in information visualisation research.

3.5.1 Query Languages offering Boolean Logic

The area of query languages was analysed as these languages, with their ability to express Boolean expressions, offer a subset of the requirements for a graphical PDL.

They target a problem that is inherent in using textual language queries. The *conjunction* and *disjunction* are logically represented by “and” and “or”. However as Michard reported, in natural language very often they are used the other way round [71], as shown in the following example:

Example 6 *People might say I want to know if there is a movie showing with the actress Liv Tyler and I want to know if there are new movies by Peter Jackson. However, by saying this they most likely do not wish to express a search query of the form:*

```
actress = "Liv Tyler"  $\wedge$  director = "Peter Jackson"
```

But they would want to know:

```
actress = "Liv Tyler"  $\vee$  director = "Peter Jackson"
```

As seen in this example, people might swap the *conjunction* and *disjunction*. Graphical representations tend to be less prone to a confusion of these two Boolean operators.

Some interesting approaches for graphical representations of Boolean queries are presented in the following subsections.

Graphical Filter/Flow Representation. Young and Shneiderman have used a graphical filter/flow representation for database queries [72]. Following from left-to-right they set the data-filtering process analogous to the flow of water. The researchers have used the metaphor of pipes and filters to represent the data flow and queries performed on that data. Each further part of the query expression narrows down the flow of water, that is, data. Thus, a user can immediately see the effect of the query they have formed. An example of their application is given in Figure 3-2. It resembles the complex query:

“Find the accountants or engineers from Georgia who are managed by Elizabeth, or the clerks from Georgia who make more than 30,000.” [72]

Venn Diagram Representation. Various research has covered the representation of Venn diagrams (set diagrams) for defining Boolean queries. Among them are the works by Davies and Willie [73], Halpin [74], Hertzum and Frøkjær [75], Jones, McInnes and Stavely [76], Katzeff [77], Michard [71] and Willie and Bruza [78].

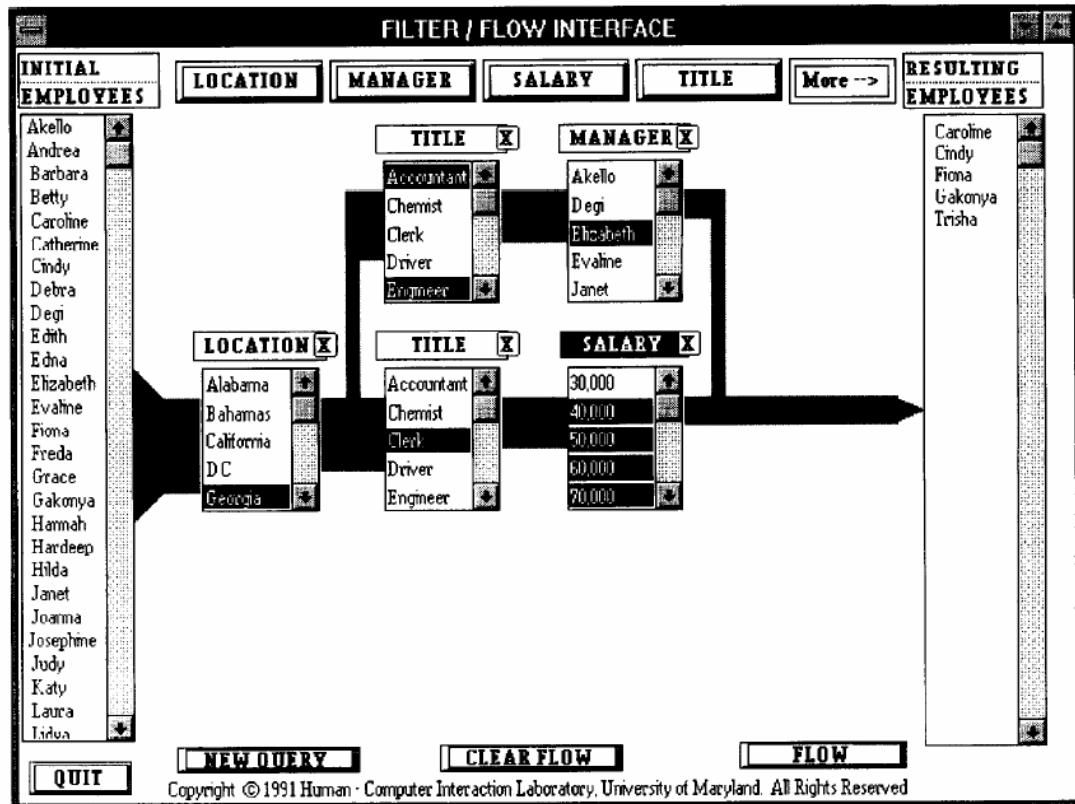


Figure 3-2: Young and Shneiderman's filter/flow representation⁶.

Most of these works [71, 73, 75, 77] include studies comparing the success rates of specifying queries using Venn diagrams and textual languages. Overall, the common result is that a graphical approach is more effective than using textual languages. Users would form less erroneous queries using a graphical approach and take less time to do so than using textual queries.

InfoCrystal. Another development was InfoCrystal by Anselm [79]. Figure 3-3 gives an example from [79], showing the transformation from a Venn diagram presentation into the InfoCrystal view.

⁶ The figure is taken from [72] D. Young and B. Shneiderman, "A graphical filter/flow representation of Boolean queries: a prototype implementation and evaluation," *Journal of the American Society for Information Science*, vol. 44, pp. 327-339, 1993.

The general idea of an InfoCrystal view is to represent all possible combinations of the input sets as icons in a compact representation, using various forms of visual coding such as shape, colour or orientation. Users can then select those icons that represent the combination of input sets they are interested in.

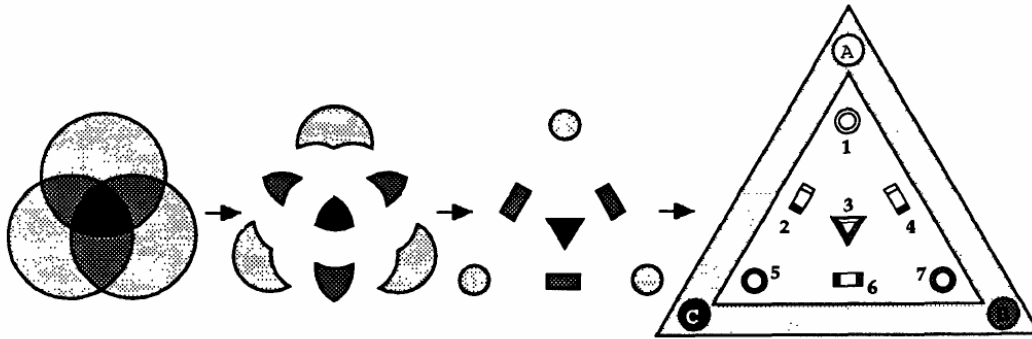


Figure 3-3: Example transformation of a Venn diagram into the InfoCrystal view⁷.

3.5.2 Representation of Hierarchical Tree Structures

This subsection shows an overview of possibilities for the representation of hierarchical structures as discussed in the research area of information visualisation.

Data that comes in the form of a tree hierarchy can be displayed in various ways. Some of these possibilities are described in the following subsections.

Indented Labels. This technique has been used for a long time. For example, Egan et al. [80] have used it for their SuperBook, and Chimera and Shneiderman [81] have used it in their research regarding the browsing of hierarchical tables of contents. The technique displays the root of the tree on the left-most position. The more a line is indented the further down is that particular node in the hierarchy. This is shown in Figure 3-4 for the following example:

⁷ The figure is taken from

[79] A. Spoerri, "InfoCrystal: a visual tool for information retrieval & management," in *Second International Conference on Information and Knowledge Management (CIKM 93)*. Washington, USA, 1993, pp. 11-20.

Example 7 A user wants to be notified about movies by Liv Tyler. She also wants to be notified about movies directed by Peter Jackson or Davis Yates, but only if the movie was released before 2007. Figure 3-4 shows this alerting need using indented labels.

```

    Actress = "Liv Tyler"
  ∨
    year < "2007"
  ∧
    director = "Peter Jackson"
  ∨
    director = "Davis Yates"

```

Figure 3-4: Indented label representation.

Node-link Diagram. One of the most commonly used representations for hierarchical structures in computer science is node-link diagrams. Nodes are connected by lines in order to express parent-child relationships. Often the root is shown at the top. Occasionally, other representations are used, for example the root is shown on the left or in the middle with the children spreading circularly from it. Figure 3-5 shows an example of a node-link diagram for the situation described in Example 7.

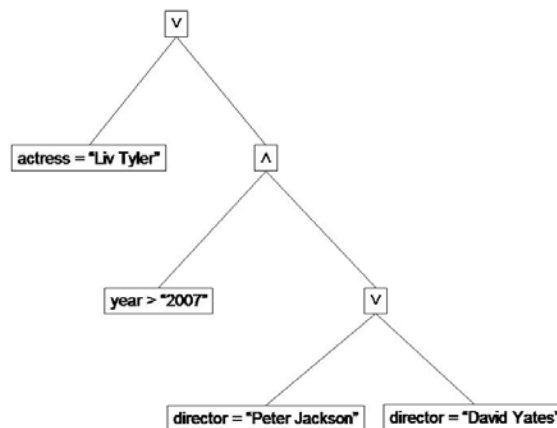


Figure 3-5: Node-link diagram.

Tree-map. First developed by Shneiderman and his colleagues [82], this approach uses the horizontal and the vertical dimension to represent the hierarchical structure

of the tree. A rectangle is filled by nested smaller rectangles recursively. The innermost rectangles represent the leaves of the tree.

Figure 3-6 shows the transformation of a tree view into a tree-map view. The example is taken from [82] and represents the visualisation of a file system. Both views shown represent the same part of the file system. The size of the rectangles in the tree-map view depends on the size of the file that is represented.

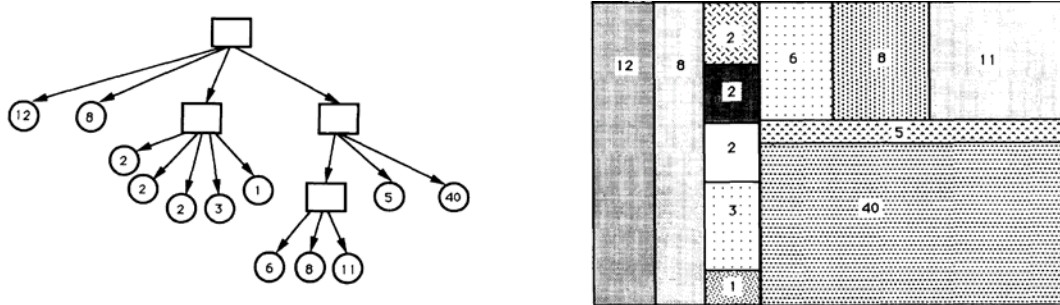


Figure 3-6: Tree view and corresponding tree-map view of a file system⁸.

A newer approach of the technique is called Sunburst which was developed by Stasko and colleagues [83]. It has as a basis a circular presentation with the root as the centre of the circle. This way, it is easier for users to grasp parent-child relationships than in the rectangular presentation of tree-maps. Figure 3-7 shows a view of a file system.

3-dimensional Cones. This representation as a Cone Tree could be seen as node-link diagrams translated to the third dimension. The children are placed circularly under their parents and linked by edges (cf. Figure 3-8). These edges and the circular arrangement of the children form a cone. The idea has been developed by Robertson et al. [84]. A horizontal version of the Cone Tree is the Cam Tree. This version has more space available for labelling the leaves.

⁸ The figure is taken from
 [82] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Transactions on Graphics (TOG)*, vol. 11, pp. 92-99, 1992.

Dynamic Pruning in a Tree-browser. In [85], Kumar et al. present a tool, Pruning with Dynamic Queries Tree-browser (PDQ Tree-browser), that works with two different kinds of views. Each of these views uses a node-link diagram that has its root on the left-hand side.

A user can form queries. Through these queries, one of the views is pruned out and only leaves the results of the query, while the other view retains the overview.

Hyperbolic Trees. Lamping, Rao and Pirolli [86] have developed a hyperbolic browser that applies a fisheye technique to smoothly blend the focus while keeping the actual context of hierarchical data. This smooth blending guides the user when navigating within the hierarchy.

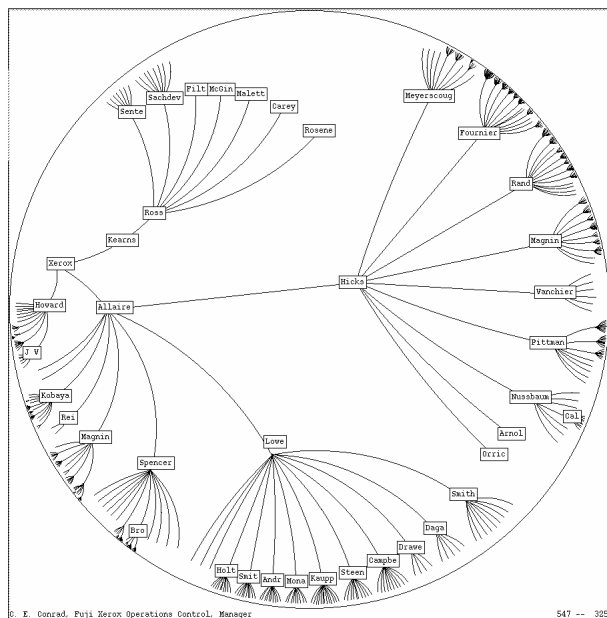


Figure 3-9: Example of the representation of a hyperbolic tree¹¹.

¹⁰ The figure is taken from [84] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone Trees: animated 3D visualizations of hierarchical information," in *SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology*. New Orleans, USA, 1991, pp. 189-194.

¹¹ The figure is taken from [86] J. Lamping, R. Rao, and P. Pirolli, "A focus+context technique based on hyperbolic geometry for visual-

The original hierarchical data is mapped onto a hyperbolic plane, which is then spread out into a 2-dimensional plane. Figure 3-9 shows an example of this representation.

3.5.3 Temporal Representation

This subsection gives an overview of approaches of visualising temporal data.

Timeline. The most commonly used concept to display data in a way that relates it to the dimension of time, is the timeline. Normally, exploiting the direction we read, the time flow is indicated by an arrow that points from left to right. Data items are placed above or below this arrow at the positions that indicate the point of occurrence of a data item. A common example for a timeline is the description of events in history.

Napoleon's March. A classic example of time presentation is Charles Joseph Minard's presentation of Napoleon's march from Tufte [87]. Minard created this chart in 1861. It gives details about the losses of soldiers, their movements, and the temperature of Napoleon's 1812 Russian campaign. The lighter broader band indicates the soldiers that marched to Moscow. The darker thinner band indicates the soldiers that were retreating. At the bottom the temperature during that retreat is depicted. The chart easily reveals the terrible losses that were experienced.

Perspective Wall. In [88], Mackinlay and colleagues present a perspective wall that can be used to visualise large amounts of linear data. Their technique offers the possibility to view details of relevant data while still displaying the context of this data. This is achieved by using a third dimension for displaying data that is currently out of focus but required to be displayed for determining the actual context. Figure 3-11 shows an example of the respective system.

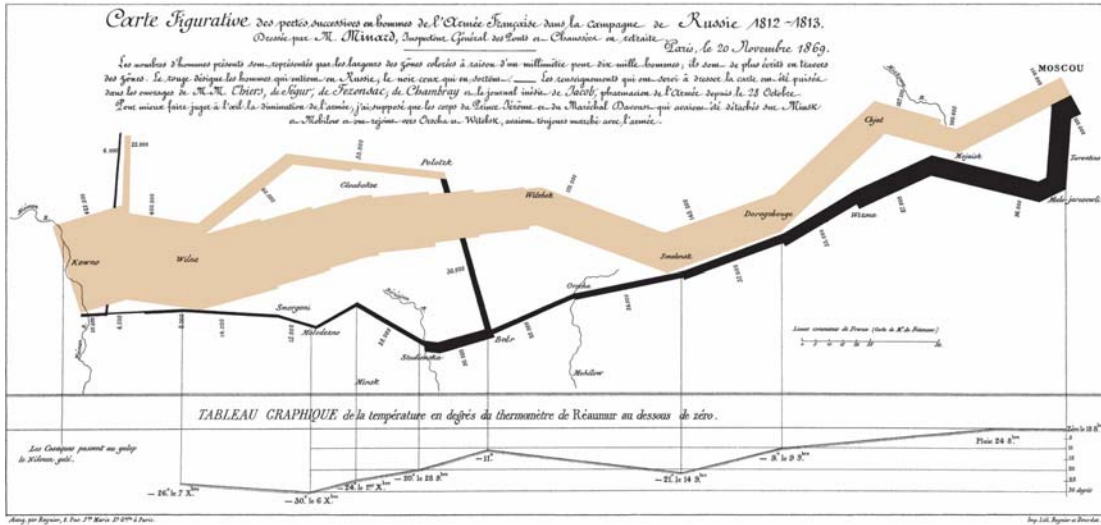


Figure 3-10: Napoleon's march¹².

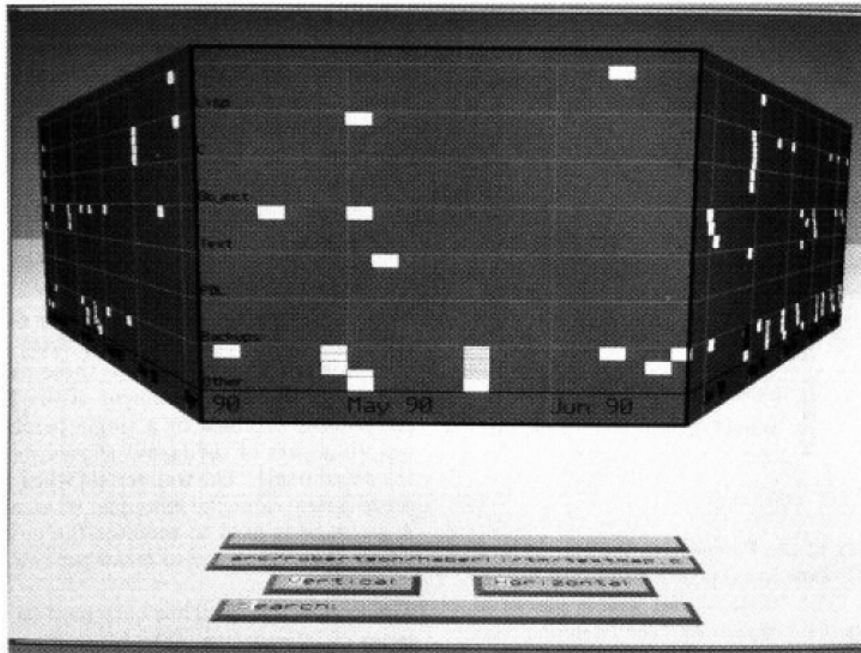


Figure 3-11: Example of the perspective wall¹³.

¹² The figure is taken from Wikipedia, <http://www.wikipedia.org/>

¹³ The figure is taken from [88] J. D. Mackinlay, G. G. Robertson, and S. K. Card, "The perspective wall: detail and context smoothly integrated," in *SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology*. New Orleans, United States, 1991, pp. 173-179.

LifeLines. Another way of visualising data that is organised over time is referred to as LifeLines and has been presented by Plaisant et al. [89].

LifeLines basically use the underlying concept of a timeline. However, the approach arranges multiple data items in a diagram and assigns icons to particularly important data items. This is done in order for the user to not miss out on this important data. The technique has been suggested for medical and court records. Figure 3-12 shows an example of the system.

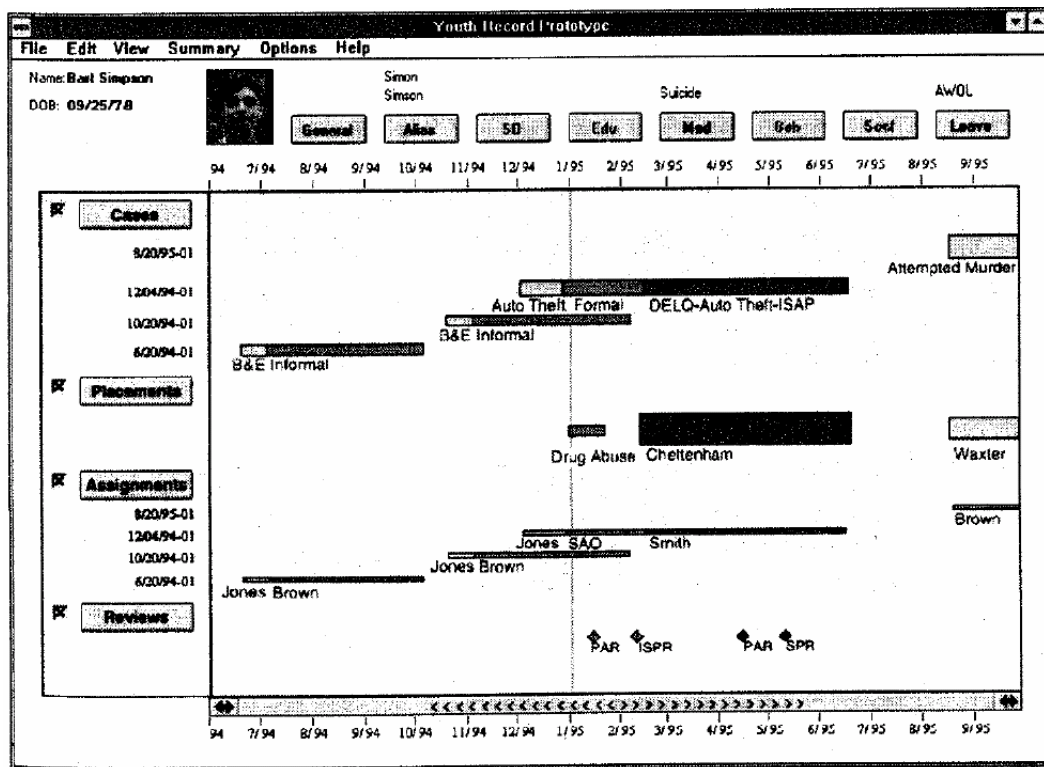


Figure 3-12: Example of LifeLines¹⁴.

¹⁴ The figure is taken from [89] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman, "LifeLines: visualizing personal histories," in *SIGCHI Conference on Human Factors in Computing Systems: Common Ground*. Vancouver, Canada, 1996, pp. 221-227.

Flow Charts. Flow charts yield an overview of a process of actions or of a program. This in its own way involves some flow of time: An action can only happen after the previous action has happened.

Activity Diagram. Activity diagrams are part of the UML standard [90]. This kind of diagram is used in workflow management. It is applied to express the sequential flow of work and offers options for alternative flows of action.

3.5.4 Discussion

The major concepts GPD has to represent are: Boolean logic, time concepts, nested expressions and contents of predicates.

Concepts to express Boolean logic such as Venn diagram representations [71, 73, 74, 77] or InfoCrystal [79] were found to be not suitable. The circular character of Venn diagrams impedes their combination over a time line. The presentation InfoCrystal uses appears very complex as it contains a high level of abstraction. The filter/flow [72] approach seemed more straightforward as it directly maps real-world concepts.

All of the analysed presentation techniques and diagrams for the concept of time involve a time arrow of some sort to represent the flow of time. None of them seems to have a major different way of presentation. Thus, GPD should use this commonly accepted paradigm.

Common hierarchical tree representations such as indented labels or node-link diagrams were considered to be not suitable since GPD targets average users—these presentations are more suitable for computer scientists. Tree-maps [82] appeared to be more suitable as they basically describe the same as “stacking” boxes into each other—something undertaken by most users at some point of their lives. However, while it offers an option for displaying nesting of expressions, this research aimed at improving the readability of this nesting; Shneiderman and colleagues used the tree-map visualisation for representing file systems. The sizes of the rectangles used in this representation signify the size of a file. GPD, however, uses size to support a

clearer readability at what level of the nested expression the user is located (cf. Section 3.6.3).

For the presentation of the contents of predicates it did not seem feasible to base GPD L on a circular presentation such as SunBurst [83] as this would waste a lot of space once different parts of profiles were combined. Also labelling would be more complicated. The rectangular presentation of the tree-map seemed more workable.

Generally, it was decided to restrain from any 3-dimensional representations as they are more demanding on cognitive processes and on visual perception. Therefore, for the presentation, it was considered to be useful to reserve one dimension for the Boolean representation and one dimension for the time representation.

Overall, the tree-map and the filter/flow approach were found to be inspiring representations. The advantages of tree-map are:

- It uses only two dimensions rather than three; therefore, its interpretation is less cognitively demanding than spatial approaches.
- It is capable of expressing nesting.
- It displays contents in a readable manner as they are displayed horizontally within the component, rather than as a label or in a non-horizontal orientation if displayed within the component.

The advantages of filter/flow are:

- It uses one dimension for the sequential order of evaluation and the other dimension for Boolean logic.
- It is a very clear, easy-to-understand, descriptive and non-abstract presentation of the sequential evaluation order in combination with Boolean logic.

The clear descriptive approach of the filter/flow model was very convincing and inspirational. Therefore it forms the basis of the representation in GPD L. Its sequential

evaluation order can be easily translated into the time-dimension required by alerting. It offers the potential for extension by further operators that are essential for expressive profiles for alerting systems.

The two approaches—filter/flow and tree-map—can be usefully combined as the good use of space made by a tree-map fits in with a presentation using a time-line.

3.6 The Graphical Profile Definition Language GPD

For users to more easily be able to specify profiles, GPD was developed. The main component of this language is a box containing a predicate—a condition. These boxes are placed in a two-dimensional space and their relative positioning expresses different operators.

In particular, the language relies on the use of the horizontal dimension for expressing time and on the use of the vertical dimension for expressing logical relationships.

The language uses descriptive names for the operators that imply their purpose.

GPD abstracts from the underlying event schema as it would be too complicated for the users having to deal with it directly. This means that a profile specified in GPD does not include information about event types. Additionally, there is no explicit distinction between composite and primitive profiles.

3.6.1 Condition

GPD uses a box in order to express a single information unit. In this box, a condition can be expressed by a triple of an attribute name, an attribute operator and an attribute value. A box represents the smallest information unit that can be expressed by a profile. Figure 3-13 in combination with the following example gives an example of the use of these attribute-operator-value boxes.

Example 8 A user wants to be alerted if the pulse is higher than 90 BPM. Figure 3-13 shows this user interest in GPD.

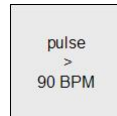


Figure 3-13: Condition box.

3.6.2 Operators

GPDL offers two kinds of operators, those that are based on logic and those that involve the flow of time. The *conjunction*, *disjunction* and *negation* traditionally are logical operators (Boolean operators, see Section 3.2.2). They are used in that capacity. The *sequence* of several events and the *selection* of a particular duplicate of events are clearly time-related concepts and used as such (time operators, see Section 3.2.2). On top of these concepts, however, GPDL follows other PDLs and offers the *conjunction* and *negation* as time-related concepts (Boolean-time operators, see Section 3.2.2). The *disjunction* is only used as a logical operator as a time-related use is pointless due to the definition of the *disjunction* per se.

If an operator is used as a time-related operator, it can be supplied with the duration parameter. This parameter describes a duration that is started when the first operand of the operator is matched and then counts down for the specified time. Once this time is over or if the operator has matched beforehand, the monitoring process ends and the entire expression matches.

For purely time-related operators, GPDL uses the horizontal dimension. Logical operators and those potentially serving as both types use the vertical dimension. The following subsections introduce all of these operators.

Conjunction. In order to express that all of several conditions have to hold for an alert to be sent, the condition boxes involved have to be placed adjacent beneath each other. Figure 3-14 demonstrates the usage of the *conjunction* in combination with this example:

Example 9 Alert if a patient's systole is higher than 140 mmHg and the diastole is higher than 100 mmHg.

If the same operator is used in combination with the duration parameter, we speak of a *composite conjunction* in terms of the underlying PDL. Whichever of the operands occurs first starts the countdown of the duration. If within the given duration, all of the other operands occur—regardless of order—the expression is matched. This is shown in Figure 3-15 in combination with the following example:

Example 10 Alert if a patient's systole is higher than 150 mmHg and the diastole is higher than 110 mmHg, both occurring within a 2 minute period.

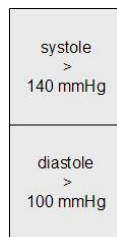


Figure 3-14: Simple conjunction expressed by the *all* operator.

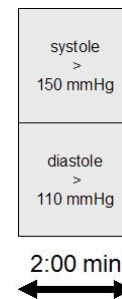


Figure 3-15: Composite conjunction expressed by the *all* operator.

The concept of *conjunction* is referred to as *all* operator within GPD.

Disjunction. The *disjunction* holds true once at least one of the operands holds true. GPD indicates this by placing the operands denoted by boxes underneath each other; however, these boxes are separated by a gap from each other. This is to contrast it from the *conjunction*. Figure 3-16 in combination with the following example shows the use of the *disjunction* operator in GPD, where it is referred to as *some* operator.

Example 11 Alert if a patient's pulse is higher than 90 BPM or his body temperature is above 38 °C or his cholesterol is above 230 mg/dl.

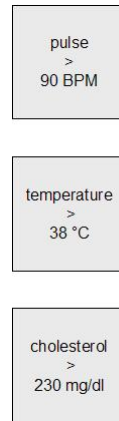


Figure 3-16: Disjunction expressed by the *some* operator.

Negation. There are two different forms of *negation* in GPDL. In order to express the simple logical *negation* it is possible to choose the *negation* operator “≠” for the attribute-operator-value triple in the condition. More generally, the inverse operator can be chosen to express the *negation*, for example “<” is the inverse of “≥”.

The other option is the *composite negation*. It is called *no-occurrence* and denoted by a crossed out box (or combination of boxes). Semantically this stands for an event that does not occur. The monitoring time is determined by the duration parameter. Once the monitoring duration has passed and the box crossed out has not been matched, an alert is sent to the subscriber. If the sub-profile was matched, the duration is started anew. Figure 3-17 presents the profile specification in GPDL of the following example:

Example 12 *Alert if there is no blood pressure reading for Mr. Smith for a day. Assumption: Patient Smith is supposed to send in information about his blood pressure on a daily basis.*

Selection. The *selection* operator selects the i^{th} duplicate of an event. As the operands spread over time, the horizontal dimension is used. The operands are denoted by boxes that are placed horizontally adjacent to each other. This operator is used in combination with the duration parameter. In GPDL it is referred to as *repetition*. Figure 3-18 shows the profile specification of the following example in GPDL:

Example 13 Alert if the pulse is higher than 100 BPM at least three times in 5 minutes.

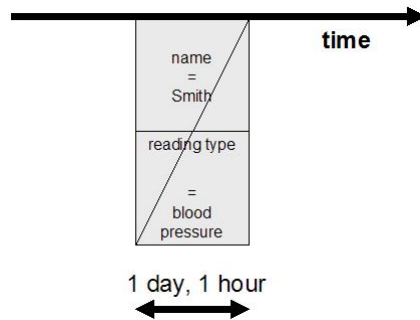


Figure 3-17: Composite negation expressed by the *no-occurrence* operator.

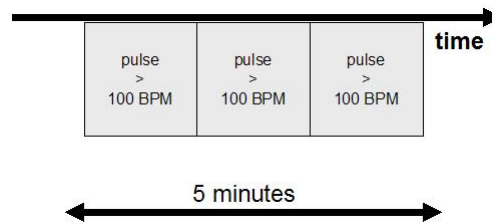


Figure 3-18: Selection expressed by the *repetition* operator.

Sequence. The *sequence* operator works analogously to the *selection* operator. However, it is used with different operands rather than a *repetition* of one and the same operand. The left-most operand has to occur first. It starts the duration counter. If from left to right the operands occur, the profile matches. The following example is shown in its GPD Language representation in Figure 3-19.

Example 14 Alert if the pulse is higher than 100 BPM and this is followed by a pulse lower than 50 BPM within maximally 1 minute.

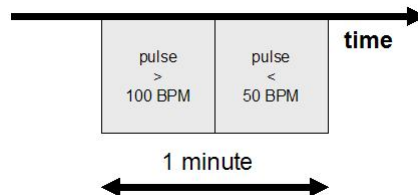


Figure 3-19: Sequence operator.

Table 3.1 gives an overview of all operators available in GPDL and their corresponding concepts in the underlying PDL.

Table 3.1: Mapping of operators from GPDL to PDL.

GPDL operator	PDL operator
<i>All</i>	<i>Primitive conjunction</i>
	<i>Composite conjunction</i>
<i>Some</i>	<i>Primitive disjunction</i>
	<i>Composite disjunction</i>
<i>No-occurrence</i>	<i>Composite negation</i>
\neq	<i>Primitive negation</i>
<i>Sequence</i>	<i>Sequence</i>
<i>Repetition</i>	<i>Selection</i>

3.6.3 Profile Parameters and Grouping

GPDL includes other concepts, as introduced in the following subsections.

Absolute Timeframe. The timeframe of a profile has an absolute start date, indicated by a green vertical line on the left-hand side of the profile. Analogously to that, it has an absolute end date, indicated by a red vertical line on the right-hand side of the profile. These two dates determine the time within which the alerting system monitors the incoming events and matches them against this particular profile. The specification of the following example in GPDL is shown in Figure 3-20.

Example 15 Alert if any rainfall has been registered. Check this only from June to August.

Repetitive Timeframe. The absolute timeframe of a profile can be extended by the repetitive timeframe. This extension indicates that while the system still monitors within the bigger absolute timeframe, it now interrupts this time span such that only within certain time spans during this bigger period, a matching of incoming events is undertaken. Figure 3-21 shows this concept in GPDL for the following example.

Example 16 Alert when there is any rain, weekdays between 8:30 and 8:35, starting from 1 January 1999 until further notice.

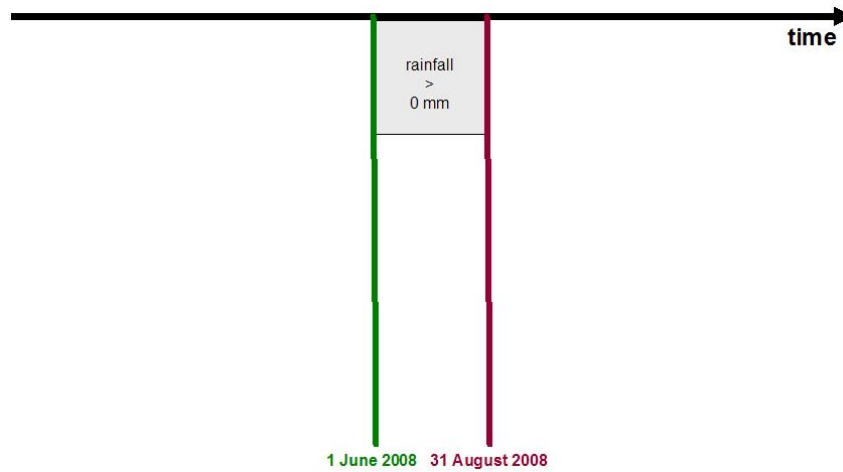


Figure 3-20: Absolute timeframe.

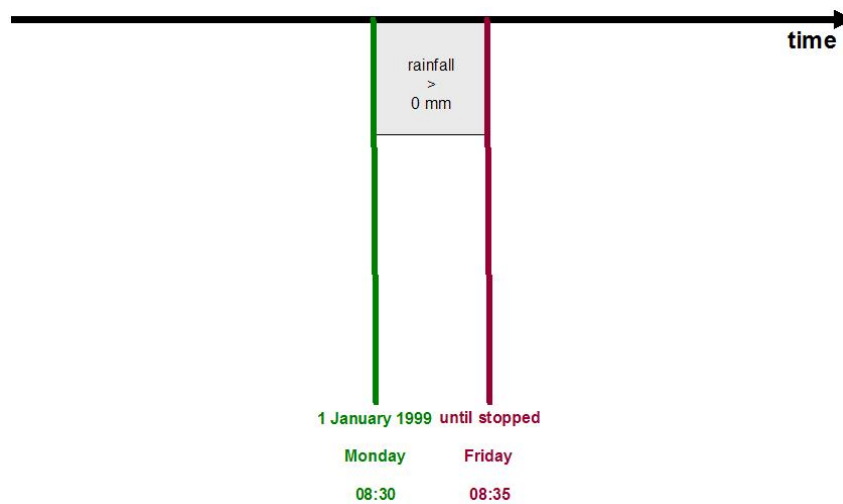


Figure 3-21: Repetitive timeframe.

Group. GPD uses the sizes of boxes to support an unambiguous readability of profiles. An example of differently-sized boxes is given in Figure 3-22. This sizing clarifies that the lowermost box is combined by an *all* with the complex upper boxes, which jointly have the same size. Similarly, the smallest two boxes are combined by an *all* with the box below of the same size. These three boxes together, in turn, form the second operand of the *sequence*.

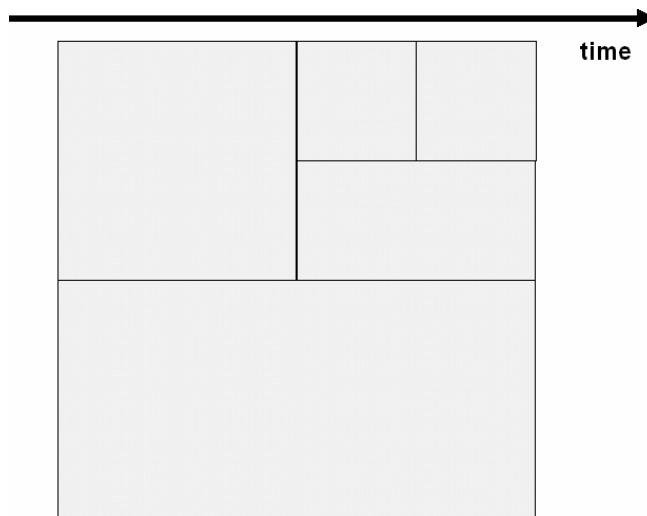


Figure 3-22: Example of the sizing of condition boxes (attributes, operator and values omitted).

However, under some circumstances there might not be a distinct representation of different profiles. In this case, GPDL supports grouping boxes to allow for the unambiguous readability of profiles, as shown in Figure 3-23 (being described in Example 17). Another use of grouping boxes is to clearly specify which boxes a duration arrow belongs to (for example as indicated in Figure 3-24). If considering grouping boxes on their own, that is, without the contents they group, grouping boxes can be imagined as high-level conditional units.

Example 17 *Alert if patient Miller has been treated for the third time by Dr Brown within a month. Check this from 1 January 2009 until 31 December 2010. This profile is shown in Figure 3-23.*

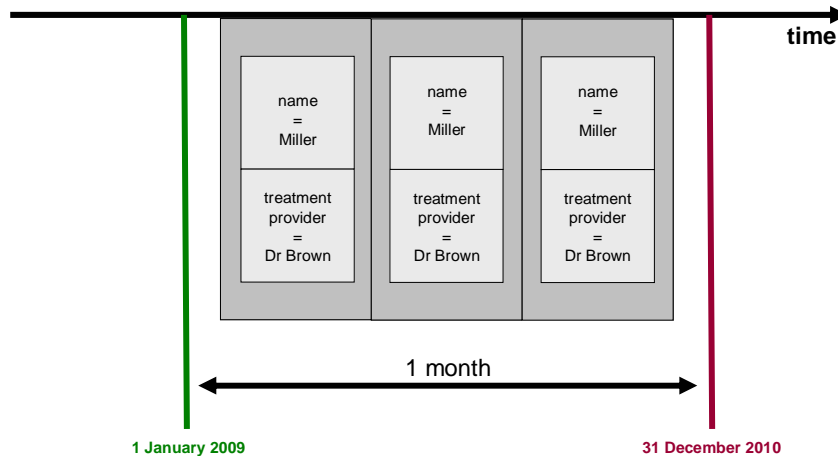


Figure 3-23: Grouping boxes ensuring unambiguous profiles.

3.6.4 Examples with Various Concepts

After the introduction of GPD's different language concepts, this subsection contains examples applying different concepts together in profiles.

Example 18 *A nurse wants to be alerted if something is wrong with the blood pressure readings of Mr Miller. She wants to be alerted if his pulse is above 100 BPM for at least three times within 5 minutes. Additionally, she wants to be alerted if there is a problem with his blood pressure sensor; that is, there is no blood pressure reading for Mr Miller for 2 minutes. She wants these alerts to start from 1 January 2009, but only during her break, being from 12 pm to 8am. A representation of this profile in GPD is shown in Figure 3-24.*

Example 19 *A nurse wants to monitor the condition of the unstable patient Mr Miller. She wants to ensure that there are no irregularities with the blood pressure sensor from Mr Miller within two days after he has received his medication A. From experience, the nurse knows that under normal circumstances it is very unusual if there is no reading (values are sent every minute) for a patient three times within one hour. The profile should be monitored starting on 1 January 2009. Figure 3-25 shows a GPD profile expressing this complex condition.*

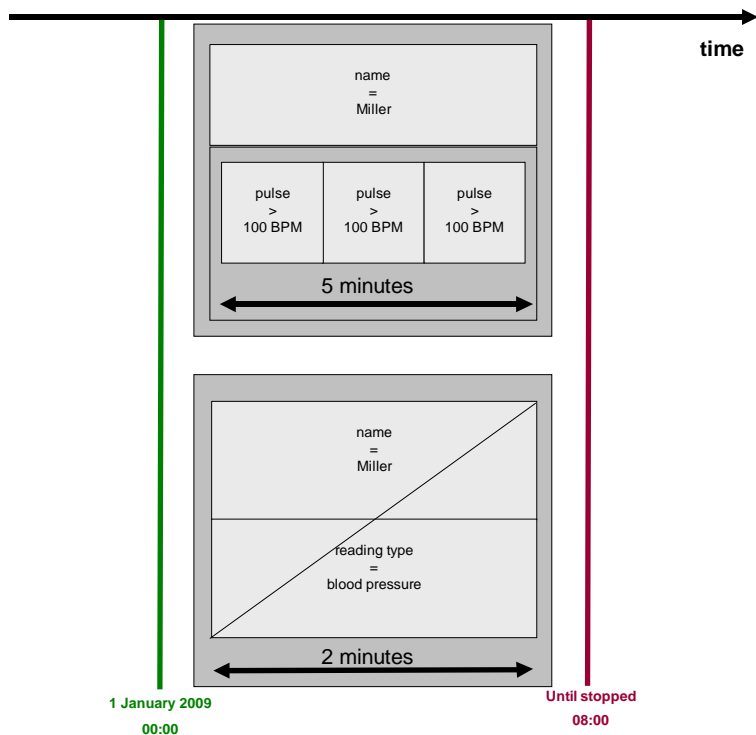


Figure 3-24: Example profile in GPDL (described in Example 18).

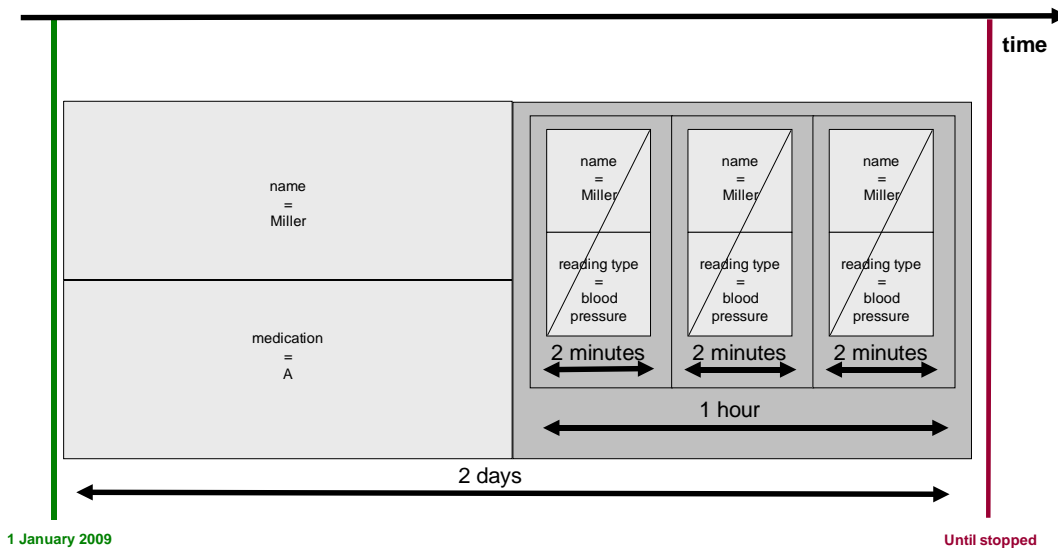


Figure 3-25: Example profile in GPDL (described in Example 19).

3.7 Mapping GPD to PDL

Having introduced GPD and an underlying PDL (see Section 3.4), it remains to show how a profile in GPD can be mapped to a profile in the underlying PDL. This is the purpose of this section.

Internally, a profile in GPD is mapped to a tree of objects that is a one-to-one representation of the profile. In a second step, this internal representation of GPD is transformed to the underlying PDL. This involves several transformation rules and interpretation steps.

3.7.1 Challenges when Mapping GPD to PDL

Due to the abstraction that has been given to a user of GPD and the simplifications that have been undertaken to support ease of use, it is challenging to determine the corresponding PDL profile for a given GPD profile.

The set of operators in GPD needs to be mapped to a larger set of operators in PDL (cf. Table 3.1). Also operators with more than two operands in GPD might need to be mapped to more than one operator in PDL. The mapping process needs to consider the underlying schema in combination with the semantics of the given conditions (represented by boxes in GPD). Examples for some intricate cases of the mapping process are given in Example 20.

Example 20 Imagine the following profile in GPD using abstract attribute names:

```
(ALL {1M}
  ("Attribute 1", =, "A")
  ("Attribute 2", =, "B")
  ("Attribute 3", =, "C")
)
```

A duration has been specified by this profile. Therefore this profile is a composite profile in the underlying PDL. The underlying schema needs to be consulted in order to know what attributes exist in what types: There is a difference between Attribute 1 and Attribute 2 existing in one type and Attribute 3 in another type, or Attribute 1 and

Attribute 2 existing in one type and Attribute 2 and Attribute 3 in another. In the first situation, the all operator might need to be mapped to a primitive conjunction and a composite conjunction. In the second situation, it might need to be mapped to two primitive conjunctions and one composite conjunction. Additionally, the condition on Attribute 2 might need to be duplicated in PDL in the second situation.

Now consider the following abstract profile in GPD L:

```
(ALL
  ("Attribute 1", "=", "A")
  ("Attribute 1", "=", "B")
  ("Attribute 2", "=", "C")
)
```

Might it be a good approach to map this all operator to a primitive conjunction, provided both attributes are used within the same type? This might not be the case, as the profile will never match: Attribute 1 cannot specify two different values within one event. It makes sense to map the all operator to a composite conjunction with sub-profiles involving primitive conjunctions.

As described before, a profile in GPD L abstracts from the underlying schema whereas a profile in PDL requires a schema. The mapping process thus needs to consider this schema when creating a profile in PDL. The schema shown in Figure 3-26 is used throughout the following section and examples. The event type is given first followed by the attributes available in this type.

Cardio Reading:

Patient Name, Reading Type, Systole, Diastole,
Pulse, Situation, Heart Beat, Sensor ID

Esr Reading:

Patient Name, Reading Type, Esr, Sensor ID

Temperature Reading:

Patient Name, Reading Type, Temperature, Sensor ID

Cholesterol Reading:

Patient Name, Reading Type, Cholesterol, Sensor ID

Oxygen Reading:

Patient Name, Reading Type, Oxygen Saturation, Sensor ID

Patient Information: Patient Name, Patient Position, Patient Condition

Reference Material:

Material Topic, Material Subtopic, Level of Expertise

Figure 3-26: Schema (event types) used for examples.

The process of mapping a GPD profile to a PDL profile is referred to as a transformation. It is described in the next subsection.

3.7.2 Transformation

The transformation works as a two-step process. First, the graphical GPD expression is converted to a tree of objects, similar to an abstract syntax tree of the profile. In a second step, this abstract syntax tree of the GPD profile is transformed into an abstract syntax tree of a corresponding PDL profile.

Grammar for GPD. Figure 3-27 shows a part of the grammar that produces the GPD syntax tree. It only states those nonterminal symbols that do not exist in the grammar for the underlying PDL (Figure 3-1). Nonterminal symbols not given in Figure 3-27 are equivalent to those given in Figure 3-1.

```

profile : profileExpr timeframe ;

profileExpr :
  condition
  | '(' 'SEQUENCE' duration? profileExpr
    profileExpr+ ')'
  | '(' 'SOME' profileExpr profileExpr+ ')'
  | '(' 'ALL' duration? profileExpr profileExpr+ ')'
  | '(' 'REPETITION' '[' INTEGER ']' duration?
    profileExpr ')'
  | '(' 'NOOCCURRENCE' duration profileExpr ')'
  | '(' profileExpr ')' ;

```

```

condition : attribute ',' comparisonOperator ','
             value ;

```

Figure 3-27: Partial EBNF of the internal GPD L representation.

Like the underlying PDL, the textual representation of GPD L uses Cambridge Polish notation. It is similar to the underlying PDL, but there is no distinction between operators for primitive and composite profiles.

Transformation of Syntax Trees. The second step of the transformation process transforms the abstract syntax tree of a profile in GPD L to an abstract syntax tree of a profile in PDL. This is the challenging part of the transformation. The following subsections present the rules for the transformation, starting with the simple cases.

As it will be detailed later, there are some GPD L expressions that cannot be uniquely transformed to a PDL expression. In this case, the result of the transformation is several PDL candidate profiles. Here a user intervention is required to uniquely resolve the GPD L profile. This approach was chosen as it is not permissible that the profile after its transformation expresses a different interest than before the transformation—specifying exact profiles was an explicit decision and one of the requirements (Section 3.3.2). These circumstances of several PDL candidate profiles are quite rare and potentially theoretical, but they can occur and thus need to be considered.

The following subsections present the transformation rules for the two temporal concepts duration and timeframe, for all operators in GPD L and for profiles themselves. Transformation rules can be applied in a recursive fashion. For example, to transform a profile involving an *all* with two operands and a duration, the two operands can be transformed independently and their result as well as the result of the transformation of the duration can be used in the transformation of the *all*. For completeness, all transformation rules are presented; they are detailed according to their complexity.

Duration. Duration specifications in GPD L profiles (used for the operators *sequence*, *selection*, *all* and *no-occurrence*) can be retained and mapped one-to-one to

PDL; they have exactly the same representation. For completeness this is shown in the following example:

Example 21 *A duration of one month, one day and one second has the following textual representation in GPD that is equivalent to its representation in PDL:*

```
GPD: {1M 1d 1s}
PDL: {1M 1d 1s}
```

Timeframe. Timeframes, the second temporal concept of GPD profiles, are equivalent to timeframes in PDL profiles. They are thus mapped directly from the GPD representation to their PDL representation. For completeness, the following example shows this direct mapping:

Example 22 *The timeframe of a profile alerting Tuesdays to Thursdays, in February to September, starting on 1 Jan 2009 has the following textual representation in GPD and PDL:*

```
GPD: {01/01/2009, UNDEFINED} {FEB, SEP} {TUE, THU}
PDL: {01/01/2009, UNDEFINED} {FEB, SEP} {TUE, THU}
```

For simplicity, the timeframe of profiles is omitted in later examples.

Condition. A condition in GPD is equivalent to a predicate in PDL: both contain an attribute, an operator and an operand. The transformation of a condition to a predicate is thus straightforward:

Example 23 *The restriction to patient Mr Smith is represented as follows in GPD as well as PDL:*

```
GPD: ("Patient Name", "=", "Mr Smith")
PDL: ("Patient Name", "=", "Mr Smith")
```

Repetition. The concept of *repetition* in GPD corresponds to the concept of *selection* in PDL. Both operators optionally specify a duration, whose mapping has been given before. This is shown in the following example:

Example 24 *If one wants to be alerted if the pulse is above the value of 100 for the third time within five minutes, the textual representation in GPD L and its transformation to PDL is as follows:*

```
GPD L: (REPETITION [3] {5m} ("Pulse", >, 100))
PDL: (SELECTION [3] {5m} ("Pulse", >, 100))
```

Sequence. The *sequence*, describing the occurrence of one event after another event, exists in both GPD L and PDL. The mapping is again straightforward, with the duration being transformed as described before.

Example 25 *The occurrence of a pulse above 100 followed by a pulse below 50 within one minute is expressed by the following textual GPD L and PDL expressions:*

```
GPD L: (SEQUENCE {1m} ("Pulse", >, 100) ("Pulse", <, 50))
PDL: (SEQUENCE {1m} ("Pulse", >, 100) ("Pulse", <, 50))
```

No-occurrence. The concept of the operator *no-occurrence* in GPD L is mapped to the Boolean-time operator *negation* in PDL, including mapping the duration as described before.

Example 26 *In order to be alerted if there is no cardio reading within one day and one hour, the following GPD L textual expression and its transformation to PDL would be used:*

```
GPD L: (NOOCCURRENCE {1d 1h}
        ("Reading Type", =, "Cardio"))
PDL: (NEGATION {1d 1h} ("Reading Type", =, "Cardio"))
```

Some. The concept of *some* in GPD L can be transformed to both the *disjunction* operator for primitive profiles and the *disjunction* operator for composite profiles. The decision as to what type of transformation is applied depends on the underlying schema; that is, the existing event types and attributes. The operands of the original *some* might be combined by different operators in the transformed PDL expression.

Furthermore, some operands of the original *some* might need to be distributed, that is, duplicated, over the resulting PDL expression.

The transformation rules described in the following aim at creating primitive profiles by combining as many operands of the original *some* as possible. If operands cannot be combined to a primitive profile, they are combined as a composite profile, while each primitive profile should contain as many operands as possible.

Let us assume a GPD profile with the following structure is given (operands are abbreviated by “ o_n ”):

```
SOME o1 ... on
```

There are three different cases, Case 1 to Case 3, that need to be distinguished; Case 1 contains two sub-cases, Case 1a and Case 1b. In the following, these cases as well as the transformation rules to be used are presented in a descriptive notation. The different cases are detailed with examples.

- Case 1 (*some*): All transformations of o_1 to o_n result in primitive profiles.
 - Case 1a (*some*): All attributes in the transformation result of o_1 to o_n share the same event type. Then the expression is interpreted as a *disjunction* in the sense of primitive profiles. Assuming the schema given in Figure 3-26, an example for this case follows:

```
GPD :
(SOME
  ("Material Topic", =, "Hypertension")
  ("Material Topic", =, "Blood Pressure")
  ("Level of Expertise", =, "High")
)

PDL :
(|
  ("Material Topic", =, "Hypertension")
  ("Material Topic", =, "Blood Pressure")
  ("Level of Expertise", =, "High")
)
```

- Case 1b (*some*): Not all attributes in the transformation result of o_1 to o_n share the same event type. Try to combine as many operands as possible using the *disjunction* in the sense of the primitive profile. That means to build all maximal k -combinations¹⁵ (with $k \leq n$) of the operands that share an event type and to combine them by the *primitive disjunction*. Then combine the results by the *composite disjunction*.

The following three examples describe this case. In the first example, the maximal k -combinations contain one operand each, as its attributes do not share an event type. The transformation thus results in a combination by a *composite disjunction*:

```

GPD L :
(SOME
  ("Pulse" , > , "90" )
  ("Temperature" , > , "38" )
  ("Cholesterol" , > , "230" )
)

PDL :
(DISJUNCTION
  ("Pulse" , > , "90" )
  ("Temperature" , > , "38" )
  ("Cholesterol" , > , "230" )
)

```

In the second example, the first two operands share an event type, whereas the third operand does not share an event type with any of the other two operands. The first two operands are thus combined by a *primitive disjunction*. This *disjunction* is then combined with the third operand by a *composite disjunction*:

```

GPD L :

```

¹⁵ All maximal k -combinations refers to all subsets of $\{o_1, \dots, o_n\}$ that share an event type and there exists no superset of $\{o_1, \dots, o_n\}$ that shares an event type.

```

(SOME
  ("Systole",>,"160")
  ("Diastole",>,"120")
  ("Cholesterol",>,"230")
)

PDL:
(DISJUNCTION
  (|
    ("Systole",>,"160")
    ("Diastole",>,"120")
  )
  ("Cholesterol",>,"230")
)

```

In the final example, the maximal k -combinations contain three and two operands. The transformation thus creates *primitive disjunctions* with three and two operands that are combined by a *composite disjunction*. This example also shows the distribution of operands:

```

GPD:
(SOME
  ("Systole",>,"160")
  ("Diastole",>,"120")
  ("Cholesterol",>,"230")
  ("Patient Name",="Mr Smith")
)

PDL:
(DISJUNCTION
  (|
    ("Systole",>,"160")
    ("Diastole",>,"120")
    ("Patient Name",="Mr Smith")
  )
  (|
    ("Cholesterol",>,"230")
    ("Patient Name",="Mr Smith")
  )
)

```

- Case 2 (*some*): Some transformations of \circ_1 to \circ_n result in primitive profiles and some transformations result in composite profiles. Apply Case 1 (*some*) for the

transformation of the operands that result in primitive profiles. Combine the results of this partial transformation with the remaining operands by a *composite disjunction*.

The following example shows the application of these rules to a profile:

```

GPDL:
(SOME
  ("Diastole", <, "50")
  ("Systole", <, "90")
  (SEQUENCE
    ("Cholesterol", >, "250")
    ("Pulse", >, "140")
  )
)

PDL:
(DISJUNCTION
  (|
    ("Diastole", <, "50")
    ("Systole", <, "90")
  )
  (SEQUENCE
    ("Cholesterol", >, "250")
    ("Pulse", >, "140")
  )
)

```

- Case 3 (*some*): All transformations of o_1 to o_n result in composite profiles. Then combine the transformation results of all operand by a *composite disjunction*. The following profile in GPD L leads to the application of this transformation rule:

```

GPDL:
(SOME
  (SEQUENCE
    ("Cholesterol", >, "250")
    ("Systole", <, "80")
  )
  (SEQUENCE
    ("Cholesterol", >, "250")
    ("Pulse", >, "140")
  )
)

```

```

    )
  )
PDL:
(DISJUNCTION
  (SEQUENCE
    ("Cholesterol", >, "250")
    ("Systole", <, "80")
  )
  (SEQUENCE
    ("Cholesterol", >, "250")
    ("Pulse", >, "140")
  )
)

```

All. The concept of *all* in GPD can be transformed to both the *conjunction* operator for primitive profiles and the *conjunction* operator for composite profiles. The transformation rules for the concept of *all* are more complex than the rules for the concept of *some*, presented before:

Firstly, semantic conflicts between predicates need to be considered. In case of semantic conflicts, it is not sufficient to merely consider the schema for the decision of combining sub-profiles with the means of primitive or composite profiles. A semantic conflict indicates that a composite combination of sub-profiles is required as a primitive combination would lead to a contradicting profile in PDL. For example, expressing with the GPD concept of *all* that the pulse should be above 60bpm as well as below 50bpm implies that a composite profile needs to be created in PDL.

Secondly, next to semantic conflicts, the distribution of sub-profiles for the concept of *all* is required to be handled in a more advanced way than is the case for the concept of *some*. There are fundamental differences between the primitive and composite concepts of *conjunction* and *disjunction*, respectively: For the concept of *some*, only one operand needs to match to lead to a matching profile. It is thus of no consequence whether these operands are combined in a primitive or a composite way. For the concept of *all*, however, this decision is vital as it determines whether the operands need

to hold on one event (primitive combination) or on several events (composite combination).

This increased complexity leads to more cases to be distinguished for the transformation: There are three cases for the situation that no duration is specified by the *all* operator in GPD L and three cases for the situation that there is a duration. The former cases are similar, even though slightly more complex, to the cases for the *some* operator. All cases are described in the following, using the same notation and structuring as before.

Let us assume the following type of GPD L profile is given:

$$\text{ALL } o_1 \dots o_n$$

The cases if no duration is specified are as follows:

- Case 1 (*all*, no duration): All transformations of o_1 to o_n result in primitive profiles.
 - Case 1a (*all*, no duration): All attributes in the transformation result of o_1 to o_n share the same event type and their combination with a *primitive conjunction* leads to a satisfiable PDL expression. Combine the operands by a *primitive conjunction*, as shown in the following example:

```

GPD L :
(ALL
  ("Systole", >, "140")
  ("Diastole", >, "100")
)

PDL :
(&
  ("Systole", >, "140")
  ("Diastole", >, "100")
)

```

- Case 1b (*all*, no duration): Not all attributes in the transformation result of o_1 to o_n share the same event type and their combination with a *primitive conjunction* leads to a satisfiable PDL expression. Try to combine as many operands as possible by using the *primitive conjunction*. That means build all maximal k -combinations¹⁶ (with $k \leq n$) of the operands and combine them by the *primitive conjunction*. Then combine the results by the *composite conjunction*. In this example, the two maximal k -combinations contain two elements each:

```

GPD:
(ALL
  ("Patient Name", "=", "Mr Smith")
  ("Systole", ">", "140")
  ("Systole ", "<", "90")
)

PDL:
(CONJUNCTION
  (&
    ("Patient Name", "=", "Mr Smith")
    ("Systole", ">", "140")
  )
  (&
    ("Patient Name", "=", "Mr Smith")
    ("Systole ", "<", "90")
  )
)

```

- Case 2 (*all*, no duration): Some transformations of o_1 to o_n result in primitive profiles and some transformations result in composite profiles. Let C be the set of composite profiles.

¹⁶ The concept of all maximal k -combinations differs from its equivalent for the *some* operator. It now refers to all subsets of $\{o_1, \dots, o_n\}$ that share an event type and whose combination by a primitive conjunction is satisfiable, and there exists no superset of $\{o_1, \dots, o_n\}$ that shares an event type and whose combination by a primitive conjunction is satisfiable.

Firstly, apply Case 1 (*all*, no duration) for the transformation of the operands that result in primitive profiles. Let S be the set of all primitive profiles after applying Case 1 (*all*, no duration).

Secondly, for all elements s in S: If s shares an event type with all primitive sub-profiles of all elements in C, the combination of s with a *primitive conjunction* leads to a satisfiable expression for all elements in C and no element in C contains a *composite negation*, then combine s with all primitive sub-profiles of all elements in C with a *primitive conjunction* (also referred to as distribution).

Thirdly, combine those elements in S that have not been combined before and all elements in C by a *composite conjunction* (provided there is more than one element).

The following example shows the application of this transformation rule. The inner *all* operators lead to four composite sub-profiles, specifying one predicate each (Case 1b (*all*, no duration)). Then, the *some* operator is transformed to a *composite disjunction* with two operands (Case 3, *some*). Finally, this transformation rule, Case 2 (*all*, no duration), is applied with C containing one element (*composite disjunction*) and S containing one element (predicate on patient name). This element in S can be combined with all primitive sub-profiles in C, leading to a combination with a *primitive conjunction*. The final combination by a *composite conjunction* is not required as just one sub-profile (*composite disjunction*) remains:

```
GDDL:
(ALL
  ("Patient Name", "=", "Mr Smith")
  (SOME
    (ALL
      ("Patient Condition", "=", "Cold")
      ("Esr", ">", 15)
    )
    (ALL
      ("Patient Condition", "!=", "Cold")
      ("Esr", ">", 10)
    )
  )
)
```



```

    )
  )
)
PDL:
(DISJUNCTION
  (CONJUNCTION
    (&
      ("Patient Name", "=", "Mr Smith")
      ("Patient Condition", "=", "Cold")
    )
    (&
      ("Patient Name", "=", "Mr Smith")
      ("Esr", ">", "15")
    )
  )
  (CONJUNCTION
    (&
      ("Patient Name", "=", "Mr Smith")
      ("Patient Condition", "!=", "Cold")
    )
    (&
      ("Patient Name", "=", "Mr Smith")
      ("Esr", ">", "10")
    )
  )
)
)

```

- Case 3 (*all*, no duration): All transformations of o_1 to o_n result in composite profiles. Combine the transformation results of all operands by a *composite conjunction*. This is shown in the following example:

```

GPDL:
(ALL
  (SEQUENCE
    ("Cholesterol", ">", "250")
    ("Pulse", ">", "140")
  )
  (SEQUENCE
    ("Systole", "<", "100")
    ("Systole", "<", "70")
  )
)
)
PDL:
(CONJUNCTION

```

```

    (SEQUENCE
      ("Cholesterol", >, "250")
      ("Pulse", >, "140")
    )
    (SEQUENCE
      ("Systole", <, "100")
      ("Systole", <, "70")
    )
  )
)

```

If the *all* operator in GPD L specifies a duration, there are other cases (Case 2, Case 3) to distinguish. They depend on the number of operands of the *all* operator, except of a universal case (Case 1).

- Case 1 (*all*, duration): If the transformations of o_1 to o_n result in exactly one composite sub-profile C that allows a duration but does not specify a duration (*sequence*, *repetition*, *conjunction*), and all primitive sub-profiles can be combined with all primitive operands of C by a *primitive conjunction* (that is, same type and satisfiable result), then distribute the primitive operands into C and apply the duration of the original *all* to C.

This universal case is shown in the following example:

```

GPD L:
(ALL {12M}
  ("Patient Name", =, "Mr Miller")
  (
    REPETITION [3] ("Cost in Month", <, 100)
  )
)

PDL:
(SELECTION [3] {12M}
  (&
    ("Patient Name", =, "Mr Miller")
    ("Cost in Month", <, 100)
  )
)

```

- Case 2 (*all*, duration): If the *all* operator specifies exactly two sub-profiles (operands), that is, if $n = 2$, combine them by a *composite conjunction*. This case is in-

dependent of the schema as an *all* operator with a duration in GPD can only be expressed by a *composite conjunction* in PDL, as shown in this example:

```
GPD:
(ALL {2m}
  ("Systole",>,"150") ("Diastole",>,"110")
)

PDL:
(CONJUNCTION {2m}
  ("Systole",>,"150") ("Diastole",>,"110")
)
```

- Case 3 (*all*, duration): The *all* operator specifies more than two sub-profiles ($n > 2$).
 - Case 3a (*all*, duration): All transformations of o_1 to o_n result in composite sub-profiles. Then combine the results of these transformations by a *composite conjunction*, as shown in the following:

```
GPD:
(ALL {30m}
  (SEQUENCE
    ("Cholesterol",>,"250")
    ("Pulse",>,"140")
  )
  (SEQUENCE
    ("Systole",<,"100")
    ("Systole",<,"70")
  )
  (SEQUENCE
    ("Pulse",>,"100")
    ("Heart Beat",="Irregular")
  )
)

PDL:
(CONJUNCTION {30m}
  (SEQUENCE
    ("Cholesterol",>,"250")
    ("Pulse",>,"140")
  )
  (SEQUENCE
```

```

        ("Systole", <, "100")
        ("Systole", <, "70")
    )
    (SEQUENCE
        ("Pulse", >, "100")
        ("Heart Beat", =, "Irregular")
    )
)

```

- Case 3b (*all*, duration): The transformations of \circ_1 to \circ_n result in primitive sub-profiles and potentially in composite sub-profiles as well. Let P and C be the set of primitive and composite sub-profiles, respectively.

The step described in the following might result in several candidate profiles. Its general goal is to create all combinations of primitive profiles that can be inferred from the processed profiles. The reason for the specialised handling in this case is that the specification of a duration for the *all* operator implies that the user is interested in the combination of different events. Hence, combining all primitive sub-profiles as much as possible might not be the semantics intended by the subscriber. The set of candidate profiles can be decreased if the schema contains information about what attributes are typically used to combine events by content (cf. Section 3.2.2). This extension has not been considered so far for the reason of finding general transformation rules. The required transformation works as follows:

Build all k -combinations (same type and satisfiable when combined by *conjunction*), with $0 < k < n$ if the cardinality of C is 0 and $0 < k < n+1$ otherwise, of the elements in P. Let us refer to the set of these k -combinations as K (each element of K is a set of primitive sub-profiles), describing all potential combinations of the primitive sub-profiles in P.

Now, let us create all combinations of elements in K that contain all elements of P , where the combined elements of K must not contain any subset relationships. Each of these combinations becomes part of a candidate profile that is built by combining the elements (sub-profiles) of the particular combination and the sub-profiles in C by a *composite conjunction* with the duration of the original GPDFL *all* operator.

Following is a detailed example of this procedure. Consider the following GPDFL profile:

```
(ALL {2m}
      ("Patient Name", "=", "Mr Smith")
      ("Systole", ">", "150")
      ("Diastole", ">", "110")
    )
```

For this profile, C is an empty set and P contains three predicates. Let us use the abbreviation P_1 for the predicate on the patient name, P_2 for the predicate on the systole and P_3 for the predicate on the diastole. The k -combinations of P lead to set K and are as follows (each line describes one element in K using the abbreviations):

```
{P1}
{P2}
{P3}
{P1, P2}
{P1, P3}
{P2, P3}
```

Out of these six elements in K , all combinations that contain all elements of P (that is, P_1 , P_2 and P_3) but no subset relationships are:

```
{P1} , {P2} , {P3}
{P1} , {P2, P3}
{P2} , {P1, P3}
{P3} , {P1, P2}
{P1, P2} , {P1, P3}
{P1, P2} , {P2, P3}
{P1, P3} , {P2, P3}
```

This means that the following candidate profiles exist:

```
(CONJUNCTION {2m}
  ("Patient Name", "=", "Mr Smith ")
  ("Systole", ">", "150")
  ("Diastole", ">", "110")
)

(CONJUNCTION {2m}
  ("Patient Name", "=", "Mr Smith ")
  (&
    ("Systole", ">", "150")
    ("Diastole", ">", "110")
  )
)

(CONJUNCTION {2m}
  ("Systole", ">", "150")
  (&
    ("Patient Name", "=", "Mr Smith ")
    ("Diastole", ">", "110")
  )
)

(CONJUNCTION {2m}
  ("Diastole", ">", "110")
  (&
    ("Patient Name", "=", "Mr Smith ")
    ("Systole", ">", "150")
  )
)

(CONJUNCTION {2m}
  (&
    ("Patient Name", "=", "Mr Smith ")
    ("Systole", ">", "150")
  )
  (&
    ("Patient Name", "=", "Mr Smith ")
    ("Diastole", ">", "110")
  )
)

(CONJUNCTION {2m}
  (&
```

```

        ("Patient Name",=,"Mr Smith ")
        ("Systole",>,"150")
    )
    (&
        ("Systole",>,"150")
        ("Diastole",>,"110")
    )
)

(CONJUNCTION {2m}
    (&
        ("Patient Name",=,"Mr Smith ")
        ("Diastole",>,"110")
    )
    (&
        ("Systole",>,"150")
        ("Diastole",>,"110")
    )
)

```

Profile Itself. The difference between the concepts of profiles in GPD and PDL is that in PDL event types become part of the profile specification. Whenever a profile is transformed, the transformation of the timestamp and the profile expression itself are performed independently. Hence, it only remains to determine the event types of all primitive sub-profiles or the primitive profile itself.

As the transformation rules for operators ensure that only those predicates that belong to the same event type are combined to a primitive profile, it remains to add event type specifications to primitive (sub)-profiles:

```

PDL without event types
(DISJUNCTION
    (|
        ("Diastole",<,"50")
        ("Systole",<,"90")
    )
    (SEQUENCE
        ("Cholesterol",>,"250")
        ("Pulse",>,"140")
    )
)

```

```

Final PDL with event types
(DISJUNCTION
  ( |
    ("Diastole", <, "50")
    ("Systole", <, "90")

  ) "Cardio Reading"
(SEQUENCE
  ("Cholesterol", >, "250")
  "Cholesterol Reading"
  ("Pulse", >, "140") "Cardio Reading"
)
)

```

Further Examples. This subsection gives some examples of transforming GPD L profiles while stating all transformation rules in a step-by-step manner. Again, the timeframes of profiles are omitted in these examples.

Example 27 *Let us consider the following GPD L profile:*

```

(ALL
  ("Patient Name", =, "Mr Smith")
  (SEQUENCE {1M}
    ("Patient Position", =, "Manager")
    (SOME
      ("Diastole", >, "130")
      ("Systole", >, "200")
    )
  )
)

```

On the deepest level of this profile, the some operator is transformed by applying Case 1a (some):

```

( |
  ("Diastole", >, "130")
  ("Systole", >, "200")
)

```

The next step is to transform the sequence:

```

(SEQUENCE {1M}

```



```

("Patient Position",=,"Manager")
(|
  ("Diastole",>,"130")
  ("Systole",>,"200")
)
)

```

Finally, the transformation processes the all operator by applying Case 2 (all, no duration), distributing the predicate on patient name. Also event types are added:

```

(SEQUENCE {1M}
  (&
    ("Patient Name",=,"Mr Smith")
    ("Patient Position",=,"Manager")
  ) "Patient Information"
  (&
    ("Patient Name",=,"Mr Smith")
    (|
      ("Diastole",>,"130")
      ("Systole",>,"200")
    )
  ) "Cardio Reading"
)

```

Example 28 Let us consider the following GPD profile:

```

(REPETITION [2] {1h}
  (ALL {1m}
    ("Patient Name",=,"Mr Smith")
    (SEQUENCE
      ("Pulse",>,100)
      ("Heart Beat",=,"Irregular")
    )
  )
)

```

On the deepest level of this profile, the sequence operator in GPD is directly mapped to the sequence operator in PDL. As the next step, the all operator is transformed by applying Case 1 (all, duration), which shifts the duration into the sequence and distributes the predicate on patient name:

```

(SEQUENCE {1m}

```

```

    (&
      ("Patient Name",=,"Mr Smith")
      ("Pulse",>,100)
    )
    (&
      ("Patient Name",=,"Mr Smith")
      ("Heart Beat",=,"Irregular")
    )
  )

```

Finally, the repetition in GPDL is mapped to the selection in PDL and the event types of primitive sub-profiles are added, leading to the following profile:

```

(SELECTION [2] {1h}
  (SEQUENCE {1m}
    (&
      ("Patient Name",=,"Mr Smith")
      ("Pulse",>,100)
    ) "Patient Information"
    (&
      ("Patient Name",=,"Mr Smith")
      ("Heart Beat",=,"Irregular")
    ) "Cardio Reading"
  )
)

```

3.8 Summary

This chapter proposed the graphical profile definition language GPDL for the specification of single-user profiles in alerting systems.

After analysing and structuring existing alerting concepts in Section 3.1 and Section 3.2, and after deciding on those alerting concepts required for this research in Section 3.3, a profile definition language underlying the graphical language GPDL was developed in Section 3.4.

Section 3.5 analysed related work that needs to be considered before defining a graphical profile definition language. Based on this analysis and the inspirations from related concepts, the graphical language GPDL was introduced in Section 3.6. To conclude this chapter, Section 3.7 showed how GPDL can be mapped to the underly-

ing PDL. After having defined GPD, the following chapter evaluates this language in a user study.

4 User Study of the GPD Language for the Specification of Single-user Profiles

A study was undertaken in order to evaluate GPD for the specification of single-user profiles in alerting systems. The study investigated users' accuracy in specifying and interpreting alerting needs using GPD and analysed their subjective experience of the language.

The chapter is structured as follows: Section 4.1 gives an insight into the goals of the study. This is followed by an overview of the experimental design in Section 4.2. Section 4.3 describes the quantitative findings of the study; Section 4.4 focuses on the qualitative findings. Concluding this chapter, Section 4.5 discusses the overall findings of the user study.

4.1 Goals of the Study

The goals of the study were to determine:

- How accurately users can specify profiles using GPD
- How accurately users can interpret profiles expressed in GPD
- Users' subjective experience of GPD for profile specification
- Users' subjective experience of GPD for profile interpretation

4.2 Experimental Design

4.2.1 Method

The study was a paper-based observational laboratory experiment. A controlled within-subjects study design using randomisation was employed. For the entire study participants were asked to think aloud. No time constraints were set, as the goals of the study were concerning the accuracy of the users' handling of the language as well as their subjective experience with it rather than to aim for an efficiency analysis.

4.2.2 Tasks

The study analysed two different conditions; one condition was a set of specification tasks while the other was a set of interpretation tasks.

Condition 1 (Specification Tasks). Each specification task contained a description of an alerting need in English. The participants had to specify the corresponding profile by drawing it on paper using GPD, to which they had been introduced.

Condition 2 (Interpretation Tasks). For the interpretation tasks, users had to give English language descriptions of diagrams showing profile specifications. These specifications were given in GPD. The English language descriptions had to be written down on paper.

Half of the participants undertook the specification tasks first; the other half undertook the interpretation tasks first.

The tasks within each condition varied in complexity. There were four levels of complexity.

Profile Levels. The four levels that were formed are simple, medium, advanced and professional profiles. Each level contains the components of the underlying levels plus additional components as shown in Table 4.1.

- Simple profiles are those that consist of either a single condition only or several conditions combined by *all* or *some*.
- Medium profiles are those that additionally use the *repetition*, *sequence* or *no-occurrence* operator to combine conditions or those that contain a duration (for example x and y happening within 1 hour).
- Advanced profiles additionally contain a timeframe. This timeframe expresses the absolute validity interval of a profile (for example 2007-2008) and can be modified to express time-stretches, that is, repeating intervals of time (for example every night).
- Professional profiles also nest profiles.

Table 4.1: Profile levels.

Level	Additional components
Simple	Condition <i>All</i> <i>Some</i>
Medium	<i>Repetition</i> <i>No-occurrence</i> <i>Sequence</i> Duration (for complex profiles, for example x and y happening within 1 hour)
Advanced	Timeframe (absolute validity of profile, for example 2007-2008) Repetitive timeframe (for example every night)
Professional	Nested profiles

The tasks in the evaluation represent all four levels. All of the underlying concepts represented in the underlying PDL describing a typical single-user profile definition language (see Section 3.4.2) were included in the evaluation tasks.

4.2.3 Procedure

A copy of the Bill of Rights was given to each participant. Two copies of the Research Consent Form were read and signed by the researcher and the participant. One

copy was retained by the researcher, the other given to the participant. At the beginning of each study session the researcher verbally explained these documents, with particular reference to the participant's right to withdraw from the study at any point without explanation.

Each participant was guided through a tutorial that introduced them to the specification and interpretation of profiles. This was followed by a training phase. Subsequently, an evaluation started. The evaluation was concluded by a short interview and a questionnaire collecting background information.

Tutorial. The tutorial started off with an example of the domain data that was used in the study. Weather data observation was chosen as an example domain in this study. After the introduction, the participants were shown 13 examples of graphical profile specifications and a natural language scenario describing each profile specification. The researcher talked the subjects through these examples and gave as much information as required. The examples belong to all four profile levels and increase in difficulty and complexity. Every participant was shown the same tutorial.

Training. The training phase entailed similar tasks as presented in the tutorial, that is, the subjects were presented with tasks for each of the two conditions. There were eight tasks in each condition. All participants were given the same tasks in the same order. The tasks were presented with increasing difficulty and complexity to enable the subjects to learn GPD step by step. During the training the participants were asked to think aloud.

Evaluation. The evaluation consisted of one set of tasks for each of the two conditions. The order of these sets and the tasks within them was controlled. Each set contained 11 tasks. The tasks in each set belonged to levels of varying difficulty and complexity. All participants were presented with the same tasks. During the evaluation the participants were asked to think aloud.

Conclusion. The study was concluded with a questionnaire collecting information about the background of the participant and a short interview about the experience of the participant with GPD.L.

4.2.4 Data Captured

For each individual specification and interpretation task it was recorded whether the subject's response was correct or incorrect. The time to complete each task was recorded for all subjects.

During all phases, the researcher recorded observations made. In the evaluation phase this was kept to a minimum as to not influence the participants in their completion of the given tasks.

Post-task questionnaires and discussions were used to capture participants' subjective responses to the language. The questionnaires contained a section giving the participants the opportunity to comment on GPD.L.

4.2.5 Participants

The pilot study was conducted with three subjects and the full study with 12 subjects. Neither was geared towards a particular user group. However, the subjects were chosen to have some level of formal education and represent both computer scientists and non-computer scientists.

The participants' ages lay between 21 years and 46 years; the mean was ~31.2 years and the standard deviation ~7.3 years. Five of the participants were male and seven female. Five were computer scientists and seven non-computer scientists that held positions such as training and development facilitator or planner for resource centers. Half of the participants had English as a first language, whereas the other half did not.

The participants rated their query experience on a five-point scale. When asked for clarification, the researcher pointed out that query experience would entail database

queries or things like queries on Google if they were more than merely typing in a single search term. They selected values covering the entire range of the scale (see Figure 4-1).

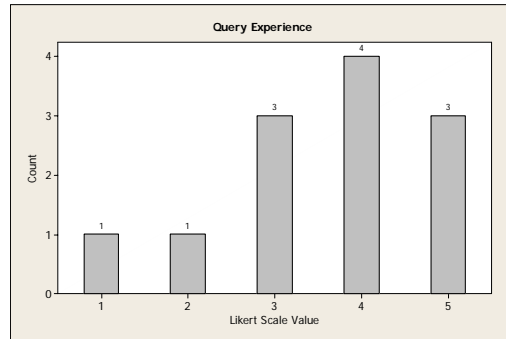


Figure 4-1: Query experience.

They gave their query frequency on a seven-point scale. Most participants rated their use of queries as very high (see Figure 4-2).

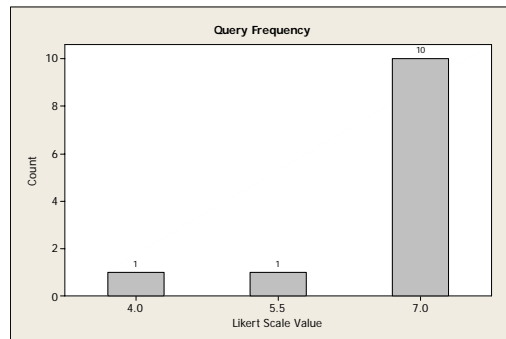


Figure 4-2: Query frequency.

For all scales the higher values represented a better experience or higher frequency.

4.2.6 Pilot Study

In the pilot study, the participants took between 55 and 95 minutes to complete the whole study. As the participants were all postgraduate students from computer science with one being specialised in formal logic, the estimated duration for other subjects was assumed to be longer. Therefore, the study had to be shortened in order not to take too long for subjects outside of computer science and academia. Moreover, the

pilot study revealed ambiguous phrasing for a couple of task descriptions and some English language mistakes. Both issues were corrected for the full study.

The major issue that had arisen was a superfluous operator in GPDL that could be expressed with the help of another already existing operator. The presentation of two concepts, the *repetition* of an event and the *sequence* of several events, works similarly. The *repetition* operator selects the x^{th} event of a recurrent event. That means that the profile specification for each of the events is identical. The *sequence* is fulfilled when events matching its operands occur in an order that is predetermined by the profile.

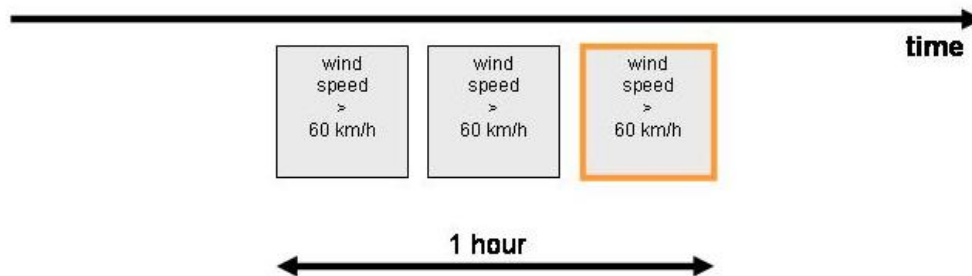


Figure 4-3: Original *repetition* operator.

Figure 4-3 describes the *repetition* in the pilot study with the example of wind speed monitoring. If a sensor sends three readings of a wind speed greater than 60 km/h within maximally 1 hour then the alerting system sends an alert. This is represented by horizontally placed condition boxes involving a gap. The profile is fulfilled once the reading for the last condition box, marked by an orange frame, has been registered by the system.

This representation of the *repetition* is similar to the representation the *sequence* operator uses: Figure 4-4 shows the *sequence* using the example of wind direction and wind speed monitoring. If the alerting system receives readings of the wind coming from the South and this is followed by a wind speed greater than or equal to 100km/h the alerting system sends an alert.

As for the *repetition* representation, the *sequence* representation also uses horizontally placed condition boxes. Analogously here, the profile also is fulfilled once the reading for the last condition box has been registered by the system. However, it does not use an orange frame for marking this.

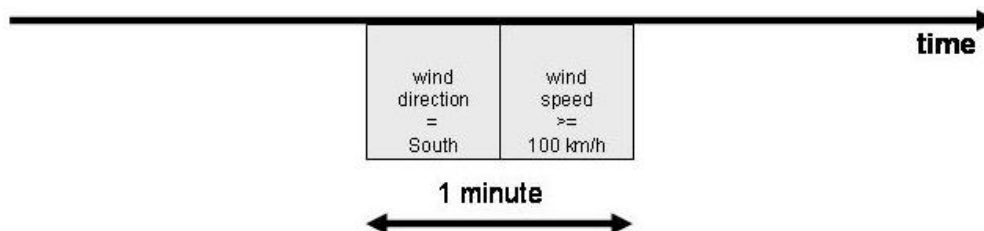


Figure 4-4: Sequence operator.

From an abstract point of view, the concepts of *repetition* and *sequence* work in the same way: they place condition boxes horizontally next to each other. Each separate condition has to be fulfilled one after the other, starting with the left-most box and ending with the right-most box. Therefore, it was decided to subsume both representations into a single representation after the pilot study.

The concepts showed two differences: The first difference was of minor significance—the *sequence* contained no gaps but the *repetition* contained gaps between the horizontally placed boxes. However, this was not done to convey any particular meaning but stemmed from the initial paper-based attempts at specifying the language. When drawing by hand, it was easier to draw a frame around a box that was detached from its neighbouring box. After this pilot study, a decision was made to remove this gap.

The *repetition* operator is represented analogously to the *sequence* operator as shown in Figure 4-5. The orange marker frame is omitted as it is redundant and seemed to imply that there could also be cases where it would be possible to place it around any other condition box than the last. Additionally, the *repetition* places the boxes without a gap. Figure 4-5 shows the same example as shown in Figure 4-3 for the new representation of the *repetition* (as introduced in Section 3.6.2).

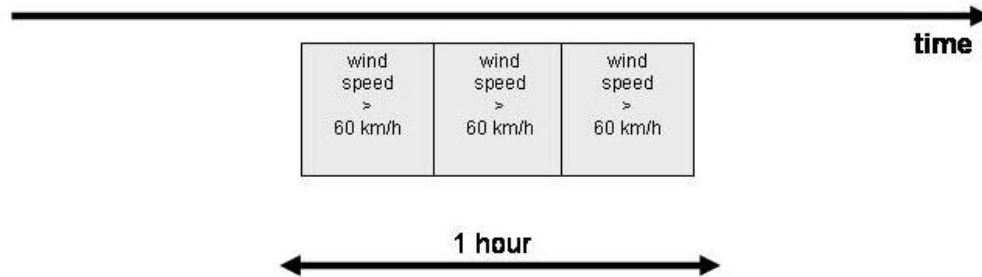


Figure 4-5: New *repetition* operator.

The changes described before were incorporated into the study material before the full study was undertaken.

4.3 Quantitative Findings

The study analysed the time taken for the tasks of each condition and the overall time taken. The other parameter analysed was the success rate of specifying and interpreting profiles.

4.3.1 Task Completion Time

Mean completion times for the two task sets (representing one condition each) as well as of the entire study are shown in Table 4.2. Overall, interpretation tasks were solved considerably faster than specification tasks.

Table 4.2: Task completion times, (N=12).

	Total duration of study	Interpretation duration	Specification duration
Arithmetic mean	01:14:45	00:10:45	00:15:20
Median	01:12:30	00:10:30	00:15:00
Standard deviation	00:19:24	00:02:00	00:05:16

Participants were asked to think aloud; some gave more feedback than others. This had a considerable impact on completion times. So the completion times measured

are not entirely representative for the time subjects required to solve the task. Rather their individual inclination to give feedback has factored into the time taken.

Table 4.3: Task completion times computer scientists versus non-computer scientists.

	Total Duration of Study		Interpretation Duration		Specification Duration	
	CS	Non-CS	CS	Non-CS	CS	Non-CS
Arithmetic mean	01:13:20	01:16:10	00:11:00	00:10:30	00:16:00	00:14:40
Median	01:12:30	01:12:30	00:10:30	00:10:30	00:15:00	00:15:30
Standard Deviation	00:14:43	00:24:37	00:02:06	00:02:04	00:06:19	00:04:27

Table 4.3 differentiates between computer scientists and non-computer-scientists.

On average, for each task set in each condition non-computer scientists were faster than computer scientists. However their completion time for the entire study was slower than that of the computer scientists.

4.3.2 Accuracy

Overall Success Rate. Overall the study has shown a success rate of 94% with 247 out of 264 tasks having been solved correctly by the participants. Non-computer scientists did better than computer scientists.

Task Levels. Table 4.4 gives details about the distribution of the results regarding the different task levels; non-computer scientists showed better results than computer scientists in all levels.

For the three easier task levels, the participants showed high success rates that were in the upper nineties. The medium-level tasks showed slightly less good results than simple and advanced tasks. The success rate for the level of professional tasks is not as good but still 79%.

Table 4.4: Success rates of task completion for specification and interpretation together.

Tasks (N)	All subjects (N=12)	Computer Scientists (N=6)	Non-computer Scientists (N=6)
All (22)	20.58 (94%)	20.0 (91%)	21.17 (96%)
Simple (6)	5.83 (97%)	5.67 (94%)	6.00 (100%)
Medium (8)	7.64 (95%)	7.40 (93%)	7.83 (98%)
Advanced (4)	3.92 (98%)	3.83 (96%)	4.00 (100%)
Professional (4)	3.17 (79%)	3.00 (75%)	3.33 (83%)

Table 4.5: Success rates specification tasks versus interpretation tasks, (N=12).

Tasks (N)	Specification	Interpretation
All (11)	10.00 (91%)	10.58 (96%)
Simple (3)	3.00 (100%)	2.83 (94%)
Medium (4)	3.83 (96%)	3.83 (96%)
Advanced (2)	1.92 (96%)	2.00 (100%)
Professional (2)	1.25 (63%)	1.92 (96%)

Table 4.5 shows a comparison of the success rates for specification versus interpretation tasks. It indicates that a difference between the success rates for specification and interpretation tasks could be observed.

For simple tasks, specification seems to be the easier activity. This tendency levels out with medium tasks and changes slightly with the advanced tasks and more with the professional tasks. Here participants were better in the interpretation than in the specification. Overall, a success rate for the specification tasks of 91% could be observed whereas the success rate for the interpretation tasks was 96%. Thus, the impact of the performance on the professional tasks influences the overall rate.

Individual Tasks. Figure 4-6 presents an overview of the success rates for each separate task. The y-axis shows the number of correct profile specifications or interpretations, respectively. A maximum of twelve could be reached.

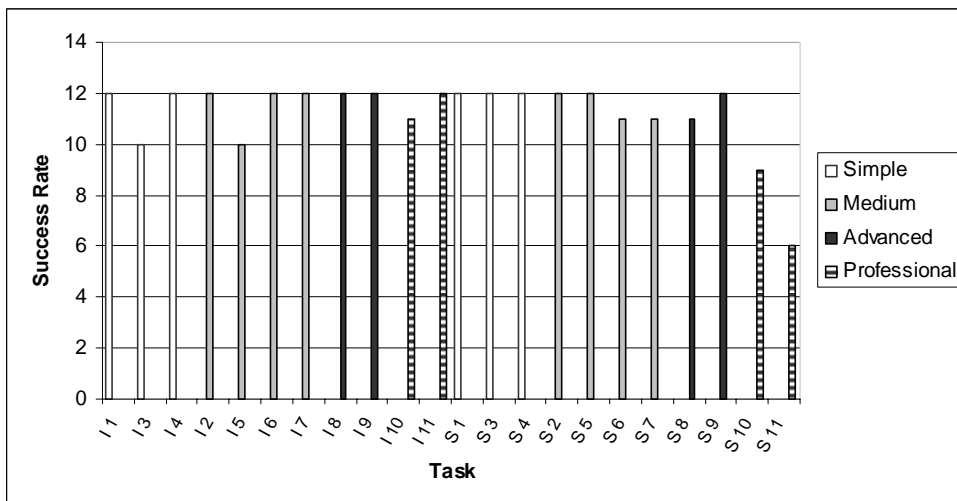


Figure 4-6: Overview of success rates for separate tasks. “In” denotes interpretation tasks; “Sn” denotes specification tasks¹⁷.

For some of the more difficult tasks (mainly professional specification tasks) the performance was worse than for the other tasks. Also I3 and I5 showed distinctive behaviour. Therefore, the semantics of these two tasks was analysed more closely.

For I3 (cf. Appendix C, Evaluation Interpretation 3), participants mixed up logical and time concepts; from the task description they assumed that both condition boxes should be fulfilled *at the same time* but did not know how to express that. However, the task did not ask for a simultaneous occurrence of the given condition boxes.

I5 (cf. Appendix C, Evaluation Interpretation 5) seemed to involve a coincidental accumulation of different minor slips; participants omitted to specify different parts of the profile.

¹⁷ The order of the tasks in the figure is different from the classification of tasks into their complexity levels. Tasks I2 and S2 belong to the medium level of complexity; however, they are in second position as analogous tasks in the tutorial and the training phase were taught as a second task.

4.4 Qualitative Findings

The qualitative parameters studied covered Likert-scale responses concerning the subjective uptake of GPDL, observations about the participants' manipulation of GPDL and the participants' comments regarding the language.

4.4.1 Likert Responses

General Distribution. Subjects rated the language against several criteria:

- Intuitiveness of the language
- Ease of use
- Satisfaction of using the language

A five-point scale was used with higher values reflecting a more positive response to the language.

Intuitiveness. All participants rated the intuitiveness of the language as positive. 25% rated it as very positive. (refer to Table 4.6).

Table 4.6: Results for intuitiveness, (N=12).

Intuitiveness	Count	Percent
4	9	75.00
5	3	25.00

Table 4.7: Results for ease, (N=12).

Ease	Count	Percent
3	1	8.33
4	7	58.33
5	4	33.33

Ease. The results found for ease of use (see Table 4.7) suggest that most participants found it easy to use GPD. This is shown by a proportion of 91.66% in the positive range of the scale and no one in the negative range.

Satisfaction. The participants also were satisfied with the experience of using GPD (see Table 4.8). They all rated it in the positive range of the scale.

Table 4.8: Results for satisfaction, (N=12).

Satisfaction	Count	Percent
4	8	66.67
5	4	33.33

4.4.2 Observations

Boxes. Boxes contain conditions that are represented by attributes and values that are combined using a comparison operator. While computer scientists seemed to have no problems understanding these components, they seemed to make minor mistakes due to lack of concentration. Some of the non-computer scientist occasionally had problems with deciding which comparison operator to use. The \neq symbol was unfamiliar to a couple of subjects. However, with the help of the tutorial they were able to figure out its meaning and application.

Some subjects were confused and distracted by unfamiliar unit names such as hPa for pressure. However, this did not lead to mistakes. Quite a number of subjects experienced problems if they had to use an attribute label that was not given in an example previously; it was challenging for them to deduce potential attribute labels from the given problem descriptions.

Operators. Overall, the participants did well with most of the operators.

Boolean Operators versus Time Operators. There was a learning curve involved with the differentiation of the two dimensions, that is, the use of the vertical dimension for logical operations and the use of the horizontal dimension in order to repre-

sent time operations. This applies mainly for the specification of profiles as it seemed to be intuitive for the interpretation tasks.

No-occurrence. Most subjects did not immediately understand the *no-occurrence* operator. However, once they understood this operator, they were able to handle it correctly. One participant wanted to know when the system starts monitoring for the *no-occurrence*.

All versus Some. Regarding the *all* and *some* operator, the participants had to learn the distinction between the use of a gap or no gap in the representation. However, they learnt quickly. Some remembered instantly and did not have to ask or check in the tutorial. A couple of subjects wondered whether the order of the condition boxes for the logical operators mattered. Nonetheless, with a bit of thought they were mostly able to realise that it does not matter.

Repetition versus Sequence. While the subjects had to analyse whether the given task was a *repetition* or a *sequence*, they had no trouble selecting the appropriate operator to express this.

All versus Sequence. The greatest challenge was the differentiation between *all* and *sequence*.

The participants understood the difference in the concepts immediately. However, when they were given a problem, they had to analyse which of the operations to use, and then to remember how the graphical symbols varied from each other in their orientation (vertical versus horizontal). This was only relevant for the specification tasks. When given a profile for interpretation, most participants had hardly any problem in interpreting it correctly.

One participant had no concept of time direction when using the *sequence*. This particular subject repeatedly mixed up the orientation. For example, when dealing with a rather advanced specification task, the participant drew the subcomponents correctly on different sheets of paper and verbalised correctly how they should be arranged to-

gether in order to make sense for the complete *sequence*. However, when the subject subsequently drew them together on one sheet of paper, they were drawn in the wrong order. This observation has to be read in the context that the subject does not come from a left-to-right writing background.

Moreover, similarly to the *repetition*, a couple of subjects were not sure whether to leave gaps between the condition boxes or not. Mostly, the participants chose not to leave gaps. One participant deduced from the given set of symbols and their meanings that it would not matter if there would or would not be a gap. Therefore, this subject sometimes left gaps.

Time Concepts. Time concepts posed a challenge to subjects. This regards their differentiation as well as the combination of some time concepts with other operators.

All with Duration. At first, a few participants were not able to combine the logical *conjunction* operator with a duration. Once they understood the concept of this combination, one participant had trouble with the simplified nature of the study, that means there were tasks that tested the use of the *conjunction* operator that did not involve the time duration. As there was no time given, the subject incorrectly deduced that this would imply that a simultaneous occurrence of the two events combined by the *conjunction* operation would be required.

Units. Another issue related to the duration that could be observed was caused by certain units used in the condition boxes. There are units that per definition relate to time such as the definition of rainfall. Rainfall is usually measured as mm per hour. Some participants were unsure whether the one hour's time implied by the unit had to be displayed as a duration arrow or could be omitted.

Duration versus Timeframe. Quite a few participants repeatedly required explanation of the difference between the duration and the timeframe. This required some mental processing on their part in the training phase. Nevertheless, once the participants had undergone the training phase, they mostly had understood the different concepts and were able to apply them correctly.

Timeframe Boundaries. It was not obvious to all subjects whether the timeframe boundaries are inclusive or exclusive. So some subjects asked about this while others returned to the tutorial and checked the examples to find out about this issue. One subject found the use of colours for the timeframe unintuitive. While green for the starting time frame could very well symbolise the (inclusive) beginning of time, there existed ambiguities with the ending timeframe. The participant explained that red is the colour of danger and therefore should never be “touched”. Hence, the ending timeframe must be exclusive.

Combination of Several Time Concepts. The combination of several time concepts appeared as a particular challenge. For example, one participant wondered whether it would be possible to have two durations occur in one task. Analysing the required time concepts for tasks at the professional level was very complex for some participants. It distracted them from simpler operations that were involved in the task which they mastered well when they occurred on their own but not in the professional expression.

A couple of participants had a problem understanding the option of nesting timeframes. While they were fine with the two general timeframe concepts, they were unsure of the meaning of their annotations. This was due to the fact that they did not quite know when the annotation denoted a repetitive timeframe and when it denoted an absolute timeframe. This caused a couple of mistakes in the training phase but was resolved when it came to the evaluation phase.

Timeframe Annotations. Some problems with the annotations of the timeframe occurred. The study employed an inconsistent way for the annotations of nesting timeframes. This confused the subjects.

Grouping Boxes. The participants had some problems with the use of grouping boxes. While all of them immediately understood the general idea, they had problems using them. The challenges that occurred can be split into two categories.

Firstly, they had to undergo some cognitive processes in order to determine how the grouping boxes—including condition boxes and operators they combined—had to be drawn. This involved their orientation in the plane, as well as their orientation and dimension to each other.

Secondly, they were confused when to use grouping boxes. This mainly involved the combination of grouping boxes with time concepts. Some participants neglected the use of grouping boxes with the duration arrow. Instead they indicated grouping by inserting arrows below the operators and condition boxes they wanted to group.

The evaluation tasks that involved a change of precedence of *all* and *some* operators were understood well by all participants.

Further Comments. Several participants suggested features they would like to be included into GPD or gave some general thoughts regarding the language.

Two subjects wanted a concept that could express aggregate functions.

Several participants were interested to hear more about the background of alerts while the study only comprised the specification part of the filter part of profiles.

A participant with a training background at a tertiary institution pointed out that GPD would be very useful for scientists and even for primary and secondary school students.

General Approach to Profile Specification. Generally, it was fun for the participants to use GPD. Three participants pointed out that they liked the playful element in it; the majority appeared to have fun and seemed to be quite at ease with the language. The majority of problems that arose involved the semantics of the problem description rather than the use of the elements of the language.

It stood out that the majority of participants—stemming from other areas than computer science, especially administrative areas—took more pride in getting the phrasing of the interpretation tasks exactly right. The majority of the computer scientists

was merely interested in getting across the general idea of the task; however, they tended to neglect details assuming they were trivial, which resulted in errors.

The majority of subjects successfully made use of the tutorial with ease. They repeatedly stated their appreciation for it and pointed out that it was very helpful for them.

One participant appeared to have a lot of trouble with the specification tasks. In order to draw the specifications, the tasks had to be understood correctly. However, the researcher and subject realised that the subject had misunderstood several parts of the description. It can be assumed that further parts were also misunderstood since this participant was the only one that showed a big difference in the correctness between specification and interpretation tasks. This subject gave the English language descriptions of the interpretation tasks in fairly simple language.

Several subjects focussed on giving an elaborate feedback regarding GPD. This caused the times of their evaluation tasks to be elongated.

Most of the participants needed some training to be introduced to the language concepts. Nevertheless, with the help of the tutorial they were quickly able to use GPD.

While the subjects did not remember all the concepts immediately after having been introduced to them once, with the help of the tutorial they were able to access all the information they needed.

4.4.3 Participants' Comments

Affirmative Comments. When asked in the concluding questionnaire to describe their general impression of using GPD, most participants stated their appreciation for the language. No subject expressed a negative reaction towards the language. Among the more positive reactions were:

*“useful in *quickly* defining complex queries (once language understood of course)”*

“easy to understand, informative”

“I can see how this would be useful to define a profile in an alerting service. It feels like it would work well and covers all cases (in a profile) that I can think of. It's a bit difficult to keep track of all those meanings. But I guess that wouldn't matter if I used the language regularly and/or had a graphical interface for it.”

“Seemed easy to pick up. Perhaps I needed another lesson. Very good for simple conditions, the more complicated require more thought. But that's life.”

These comments demonstrate that the subjects appreciated the general idea of the language. While some subjects acknowledged that more complex tasks naturally required a higher amount of mental processing, they appreciated that this is inherent in the nature of complex tasks.

“I liked the language. The basic concept is intuitive. Just got a bit confused with the duration sometimes. Nonetheless, I think that's only a matter of practise.”

“The symbols and language to me seem clear to use and seem in a logical pattern. When components of several different times, that is during 2007, during x months at this time, it was a bit more difficult but it generally was clear.”

While still being appreciative, these comments also indicate that the time concepts should be clearer to guarantee that GPD can be used to specify correct profiles with ease.

Critical Comments. A couple of participants gave more critical comments than most of the other subjects. They commented on issues such as the following:

“Would like to see more complicated real life problems but I see it could be possible to define them with it. Unclear if start and stop can be defined precisely.”

“It seems to be quite simple to understand, although there may be some (Boolean) operations that may not be possible to define (uniqueness). Adding colours may help to identify operators.”

“It is easy to describe an event. But it is bit confusing when it is getting complicated (box).”

Again, one subject commented on the difficulties involved with the time concepts. Another referred to an issue that repeatedly posed a challenge for the participants: the correct use of grouping boxes was not always straightforward.

4.5 Discussion

The discussion summarises the major findings that the study yielded and puts them into perspective to the intended research purpose of this study. The study was geared towards exploring four key issues:

- How accurately can users specify profiles using GPD L?
- How accurately can users interpret profiles expressed in GPD L?
- What is users' subjective experience of GPD L for profile specification?
- What is users' subjective experience of GPD L for profile interpretation?

Overall, the results for both specification and interpretation were very encouraging. GPD L supports a high level of accuracy even for complex tasks. Also the subjective responses were very good; the subjects liked the language. In more detail the major lessons that could be learned from the study are described in the following sections.

4.5.1 Accurate Profile Specification and Interpretation

GPD L is successful for specifying and interpreting profiles for alerting systems; the overall success rate was 94%.

GPD L is well suited for users of alerting systems to correctly specify and interpret profiles of simple, medium and advanced levels. It also is well suited for interpreting profiles at a professional level. For specifying profiles at a professional level further research needs to be addressed to tackle some challenges that were identified through the study. These challenges regarded the following areas:

- The use of grouping boxes

- The differentiation between time concepts
- The differentiation between several operators (mainly *all* versus *sequence*, *all* versus *some*, gap usage for *sequence/repetition* and assumed implicit simultaneity of *all*)
- Spacing and location difficulties when drawing elements of the graphical profile definition language

The fundamental concepts were taken up well. It was observed that the box representation for the smallest unit of information was accepted unchallenged and easily understood. Moreover, the breakdown of operators and dimensions into logical-vertical versus time-horizontal was easily understood and well applied.

For easy and medium tasks, participants were better in specifying profiles in GPD than they were in specifying them using natural language. This is most likely due to the fact that the English language is more complex and thereby leaves more ambiguity in formulation than when using GPD. Hence, it is more error-prone to formulate something in the English language. Also GPD has a clearer look than the English language. Thus mistakes can be seen more easily when revising the specification.

4.5.2 Subjective Uptake of GPD

Both Lickert-scale responses and participants' comments showed a positive reaction towards the handling of GPD. The intuitiveness, ease and satisfaction of using GPD had proportions of 100%, 91.66% and 100% in the positive ranges of the scales (with N=12, max=5, most positive). This suggests that the participants liked GPD.

4.5.3 Target Group

Non-computer scientists performed better than computer scientists in using GPD. When GPD was developed the intention was to develop a language that can be used

by the average user to specify their alerting needs. Thus, the observation that GPD L yielded better results for non-computer scientists is positive.

4.5.4 Teaching GPD L

The tutorial that was developed turned out to be helpful for the participants' understanding of the language. This was pointed out by them repeatedly.

Since specifying profiles is not an every day task for the average user, the participants had to learn the concepts that they had to represent using GPD L even before they could learn the language itself. This challenge was identified as a problem that would have to be addressed in an interface that can be used for specifying profiles in GPD L.

4.5.5 Completion Time

From observing the participants, it appeared that non-computer scientists were slower in solving the given tasks than computer scientists, but this seems to be disproved by the measures taken of the completion time. However, quite a number of the participants did not focus on performing the given tasks as fast as possible. Rather they were interested in giving detailed feedback on GPD L—which was the purpose of the study in the first place.

4.6 Summary

This chapter presented a paper-based user study to evaluate the single-user profile definition language GPD L that was developed in Chapter 3. The foci of this study were to analyse the subjective experience of the participants when specifying profiles using GPD L and to evaluate the accuracy when defining and interpreting profiles in GPD L.

Section 4.2 described the experimental design that was chosen for the user study. Its results, divided into quantitative and qualitative findings, were presented in Section 4.3 and Section 4.4; Section 4.5 presented a discussion of the results.

Overall, it could be observed that the general approach of GPD is very promising and fulfils its design goals. By implementing some refinements to the language, GPD merits further research. This further research is undertaken in the following chapter, which focuses on implementing GPD within a software application and on the interaction design of the software application.

5 The Editor GPDUI for the Specification of Single-user Profiles

Having introduced GPDUI in Chapter 3 and undertaken a paper-based study in Chapter 4, this chapter presents the editor GPDUI that can be used for the specification of profiles using GPDUI. The evaluation of this editor is described in Chapter 6.

The chapter is structured as follows: Section 5.1 describes GPDUI; it starts with a general overview of the interface, then goes into more detail about decisions for layout and interaction mechanisms, and concludes with examples of the profile specification process.

Section 5.2 provides details about how the options for changing profiles in GPDUI are presented to the user. Finally, Section 5.3 explains how GPDUI profiles are presented in an unambiguous way to the user by automatically grouping profile expressions.

5.1 GPDUI: The GPDUI Interface

The following subsections introduce GPDUI that was designed for the specification of single-user profiles using GPDUI as well as the editor's interaction mechanisms.

5.1.1 Overview

GPDUI (Figure 5-1) has a central working area that is split into two sub-areas: The bigger sub-area is the workspace (marked by 1 in Figure 5-1) that holds the currently valid profile (here empty). There is also a component bar (2) that holds components that are not currently in use but can, like building blocks, be added to the current pro-

file, one by one, if desired. The workspace contains an indicator that its horizontal dimension is used to display time (3).

The component bar is scrollable if more components are held than fit into the available space. There can always be only one connected profile in the workspace at any given time. Components can be swapped between the workspace and the component bar by drag and drop.

The name of the current profile is displayed in a tab (6). It is possible to zoom in and out of the profile that is presented in the workspace. It is also possible to navigate within a profile. The component that has focus is displayed in the current component display (4).

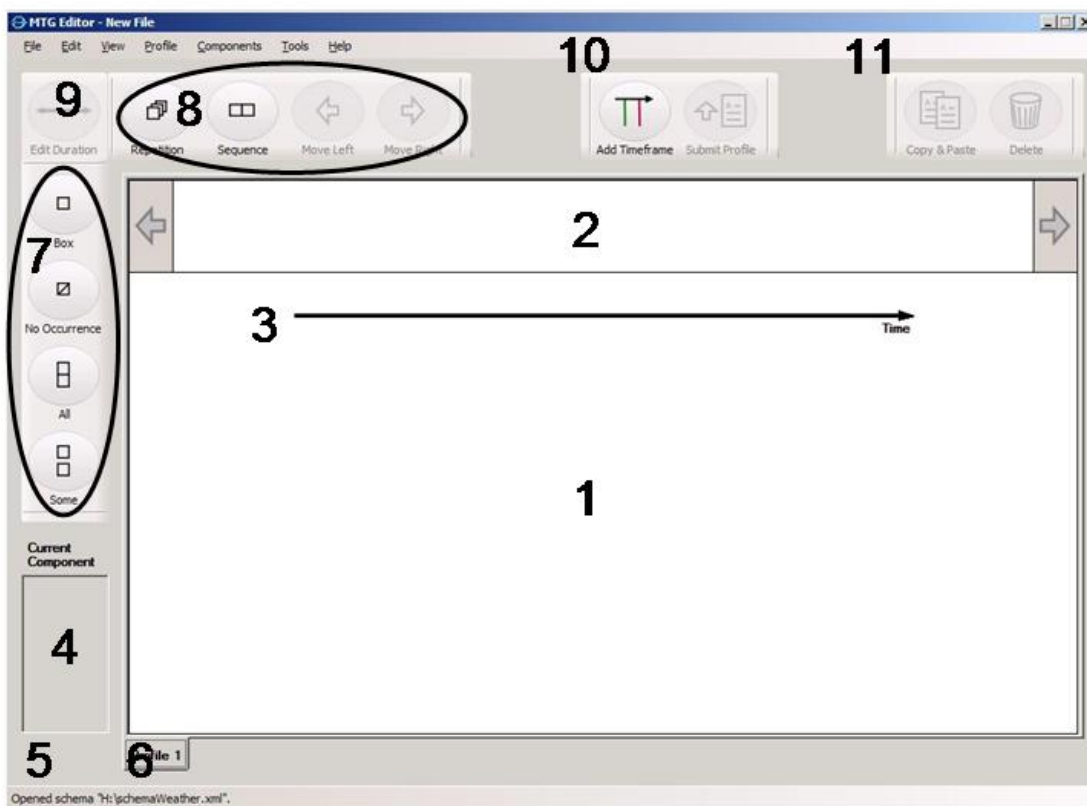


Figure 5-1: Empty interface of GPD-L-UI.

Each profile relates to a given schema, describing available attribute domains, attributes, operators and values. The current schema is displayed in the status line (5). Even

though the schema includes information about event types, this information is not presented to the user; it is only required for transforming GPDL profiles to profiles in the underlying PDL (explained in Chapter 3).

Outside of the working area, there are buttons that offer the basic building blocks, operators, conditions and parameters that can be added to a profile.

Vertically grouped (7) are those operators that use the vertical dimension in GPDL, that is, the *all* and the *some* operator. This group also contains the main building block for conditions, the condition box, and the *no-occurrence* operator. Horizontally grouped (8) are the time-operators *sequence* and *repetition* that use the horizontal dimension in GPDL. Additionally, this group contains left and right manipulators for moving the position of the operands of time-operators. Between them in the corner is the duration parameter that can be specified for both of these operator groups (9).

There are two controls (10) that relate to the entire profile—the timeframe button and the submit profile button. In the right-most horizontal position (11), the standard functionalities copy and paste as well as delete are located.

Depending on the current system state, the buttons are active, inactive or offer related functionalities that are only applicable to certain system states (cf. Section 5.2).

5.1.2 Layout

Button Placement. Generally, it was decided to place the buttons above and left of the working area as this goes along with the natural searching behaviour of westernised users, due to their reading direction.

The buttons are grouped according to their functionality. System operations such as copy and paste as well as delete are separated from profile-specification-specific tasks. Additionally, operations that apply to the entire profile (submit profile and timeframe) are separated from operator-specific operations.

The chosen arrangement of the operator buttons is analogous to the operator approach of GPDUI. Since GPDUI uses the horizontal dimension for time-related operators, the buttons for the *repetition* operator, the *sequence* operator and the manipulation of their operands are placed horizontally above the working area.

Similarly, the vertically-placed buttons left of the working space offer the *all* and *some* operators that cover the vertical dimension in GPDUI. This area also includes the condition box and the *no-occurrence* operator.

The duration button is placed at the corner where horizontally- and vertically-presented operators meet, as this parameter is applicable to both groups of operators.

Split-up of Available Space. An initial design decision that had to be made was how to present profiles and components that can be used to assemble profiles: Each profile can be assembled out of a number of components that in themselves form sub-profiles. The question was how to present this. Should it be in one area or should it be split up into two areas? The decision was made to have a workspace that holds the current profile and a component bar that functions similarly to a clipboard holding all components that are not yet integrated into the profile.

Workspace. The workspace only holds the current profile. The reason for this is that any other solution might be confusing, as a profile does not necessarily consist of one connected geometric representation due to the gap that defines the *some* operator. If all profile components and sub-profiles were stored in the same area, there would be room for misinterpretation of the arrangement of those components and sub-profiles. It would be unclear which components belonged to the current profile and which would be a component that coincidentally happens to be placed nearby to the current profile.

Component Bar. The component bar is the storage space for components that a user wants to define and assemble prior to completing the entire profile they are working on. It is intended to help solve problems via the divide-and-conquer method. For the

profile specification process, this means that the component bar supports a bottom-up approach when creating profiles.

If the component bar needs to display more components than can fit in its display, a horizontal scrolling mechanism is employed.

As the components that are displayed in the component bar are miniature presentations of sub-profiles, it was decided to not fully display the texts describing their conditions in that presentation. For nested components it would be too small to present them like this. The idea is to enable the user to get an idea of the structural overview and then to discover the details. These details are shown with the aid of tooltips once the user hovers over the displayed component.

Exchange Mechanism between Component Bar and Workspace. Separate components are displayed next to each other in the component bar. If a component is moved from the workspace to the component bar (by drag and drop), it is inserted at the current mouse location when it is being dropped.

If a component is moved from the component bar to an empty area on the workspace and the workspace is currently occupied by another component, the component on the workspace is automatically moved to the position of the component that is being moved down from the component bar. That is, the components in the component bar and on the workspace are being exchanged for each other.

If the workspace is empty and a component is moved down from the component bar to the workspace, then all other components that are stored in the component bar are automatically shuffled one position to the left in order to avoid gaps from developing.

If a component is moved from the component bar to an empty box on the workspace, this empty box is filled with the component from the component bar. Again, other components in the component bar are shuffled one position to the left.

5.1.3 Interaction Mechanisms

The major design challenges concerning the interaction with GPDUI were the manipulation of profiles on the workspace as well as the issue of how to navigate through a given profile expression.

Drawing versus Button Approach. The mechanism of how profiles should be manipulated was a key issue when designing GPDUI. Two different approaches were seriously considered. One approach, the drawing approach, embraces a drag and drop mechanism similar to those used in drawing applications. The other approach, the button approach, targets to guide the user in the profile specification process using buttons for manipulating profiles. Following is a summary of the main arguments that led the decision process for choosing the best approach.

Drawing Approach. The drawing-oriented drag and drop approach is more error-prone than the button approach, as users are free as to how they want to move and drag components in the workspace. Moreover, this approach demands for a single working area that holds both the current profile and those sub-profiles that only exist temporarily, in that separate form. This way, it is made harder for users to determine what sub-profile is valid and what sub-profiles only exist temporarily but, if submitted, would not form a valid profile.

It would be near impossible to integrate an automatic correctness check, as it is unclear which component the user determines to be the current profile and which components are randomly spread in the working area as temporary sub-profiles.

Handling the boxes that have to be moved around might be physically challenging for users like certain groups of patients that have less than perfect vision or dexterity. Having to place the mouse over the edge of a box in order to change its border is difficult for them. Also for health care staff that have to work in a hectic environment, this might pose a challenge. It was considered to use boxes supplied with little “handles”, thus making it easier to manipulate them. However, this approach was considered to disrupt the clarity of profiles and thereby diminish their readability.

The drawing approach yields a more chaotic workspace making it difficult to locate profiles. The scrolling mechanisms used for the workspace are horizontal and vertical. Thus, a given component can be located anywhere on the workspace and thereby challenging to find.

The drawing approach was considered in combination with an underlying grid-snap mechanism. If a box is moved, then the bordering boxes automatically adapt their size to only display valid profiles. This, however, leads to a flickering effect in the displayed profile; it might leave the users not understanding their available options and being visually confused.

However, the drawing approach was considered to be cognitively less challenging than the button approach as the user can manipulate the profile basically in any given way. This might lead to invalid profiles, though.

While the physical effort might be more challenging on the dexterity of the user, it would potentially be less demanding on the amount of work required to create a profile.

Button Approach. In contrast, the button approach requires more mental effort to produce profiles, but it avoids invalid profiles. While editing in this approach might require more steps for certain profiles, there is a clear separation between finished—valid profiles—and temporary components. This is because the button approach supports the separation into workspace and component bar.

This separation also leads to less visual confusion than the drawing approach. Additionally, it makes it easier to locate temporary components.

By offering only those profile-manipulation operations that are valid operations at a certain system state, the user is better directed and, thus, avoids mistakes.

Conforming to Fitts' law, the physical tasks that have to be performed in the approach are less demanding than in the drawing approach, as buttons are bigger than edges of boxes and hence easier to click.

Directing the user in the specification of profiles was also considered to be beneficial for the understanding of the structure of a profile expression.

In the end, it was decided to use the button-approach for GPD-L-UI: it better supports the underlying concept of nested expressions that the user has to understand in order to build proper and valid profiles.

Navigating Nested Expressions. The second major challenge that was encountered during the design phase of GPD-L-UI was how to lead the user in navigating through nested profile expressions. An expression semantically consists of three dimensions—the time dimension, the logical dimension and the nesting of components. However, the chosen presentation method only uses two dimensions. Thus, a technique was required that helps the user to recognize this third dimension.

The nesting of components is represented by the sizes of boxes. Additionally, the user is offered help with navigating through expressions: By clicking boxes within the currently active component, users can navigate further into the respective sub-expression. Either by clicking outside of the profile on the workspace or by selecting another, currently inactive component of the profile, a user can navigate out of sub-expressions again.

Whenever the user has a choice of where to navigate in a profile expression, that is, in any other case than having an active single condition box, hovering over a box highlights the entire component that box belongs to. This entire component forms an operand of the operator that currently has focus. This highlighting mechanism is to indicate the operands of the active component (operator in focus) and is used in addition to the use of the sizes of boxes for this purpose in GPD-L.

Once a user clicks into a box to navigate into the respective sub-expression, the component comprising the clicked box is selected as the active component. Boxes outside the active component are greyed out in order to show that they have lost focus. Again,

the highlighting effect regarding operands on the same nesting level is offered. This navigation process can be repeated until a single condition box is reached. Then, clicking on the box allows editing the contents of this box. Alternatively, a user can choose to navigate out of an expression at any given time using the mechanism described before.

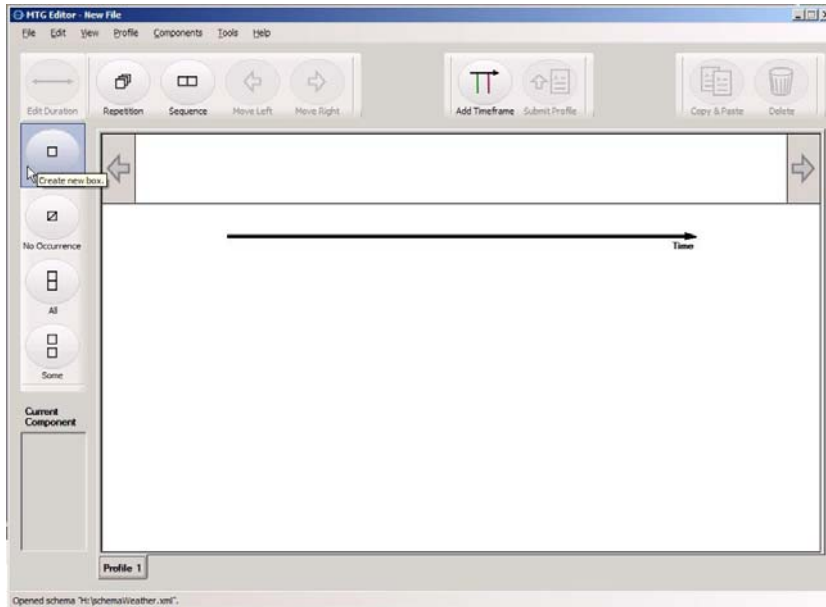
It was decided to offer a current component display in GPDUI to give further information to users of where in a profile expression they are currently located. This display shows an abstract miniature symbol of the profile and the name of the current operator that has been selected.

5.1.4 Profile Specification Process

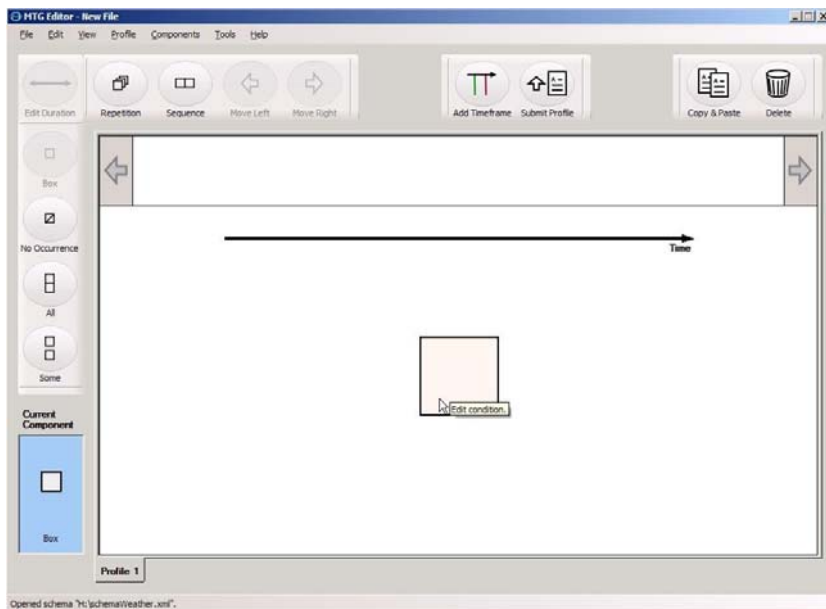
The user can specify profiles in two different ways: bottom-up profile definition and top-down profile definition. Both options are described in the following section.

Bottom-Up Approach. In the bottom-up approach, a user first specifies detailed components and then assembles them into a complete profile. If one would imagine the underlying tree structure of a profile (cf. Chapter 3), the user would be starting the profile specification at the leaf-level of the tree.

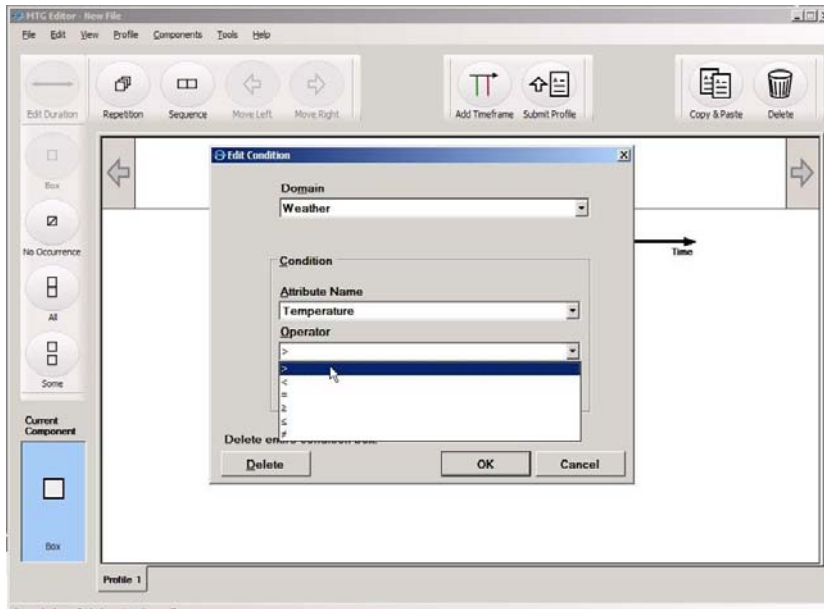
Table 5.1 demonstrates the use of GPDUI when employing a bottom-up approach to specify a *sequence* with two operands and a duration.

Table 5.1: Bottom-up profile specification.

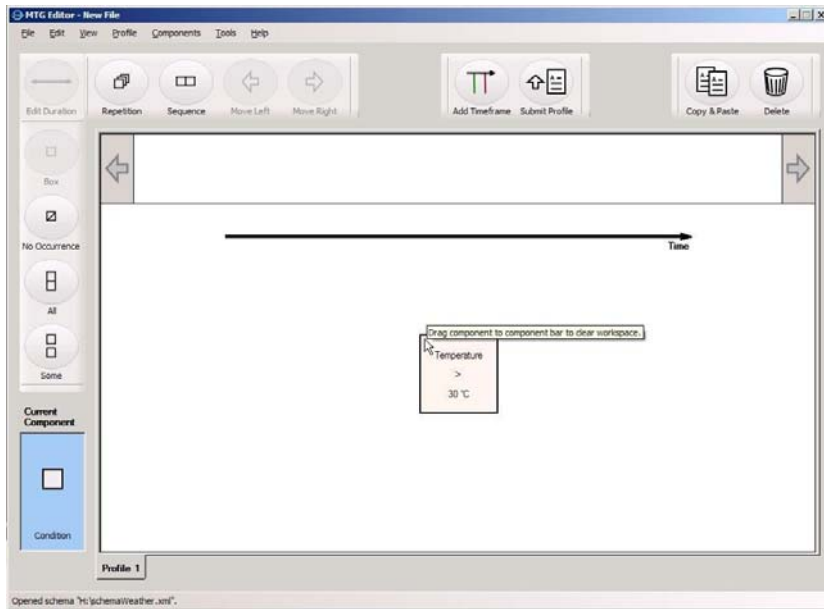
The user chooses to create the first operand, a condition box, by clicking the “Box” button.



The content of the operand is about to be edited by clicking the box.

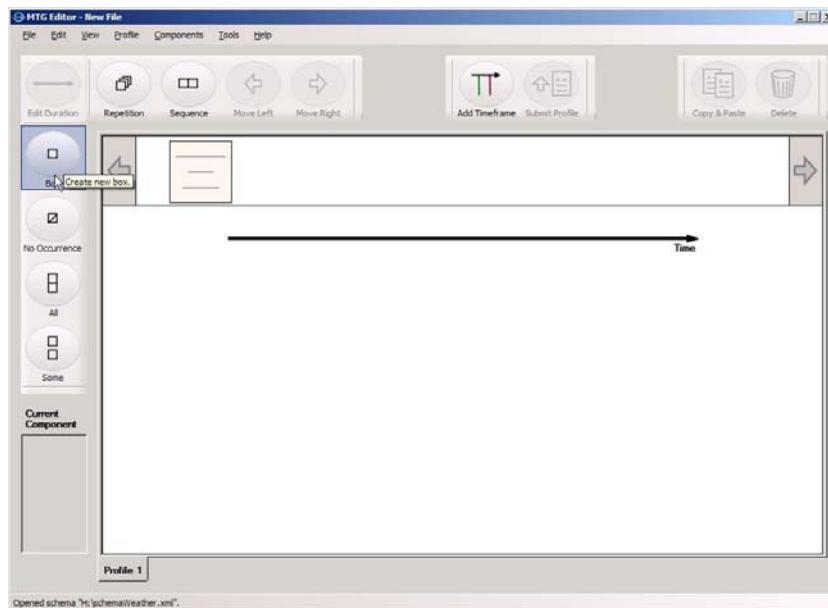


The content of the condition box (attribute, comparison operator and value) is edited.

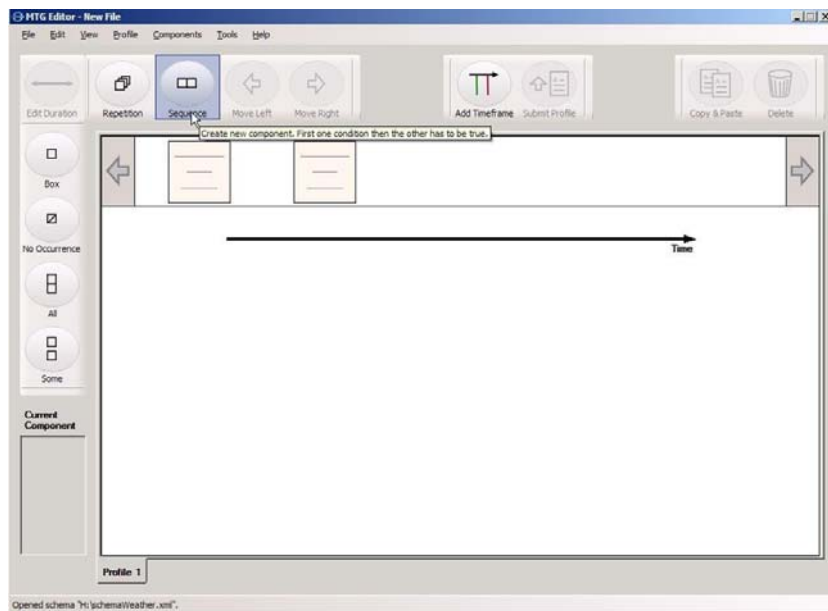


The workspace can be emptied by dragging components up to the component bar.

The first condition box is about to be dragged to the component bar.

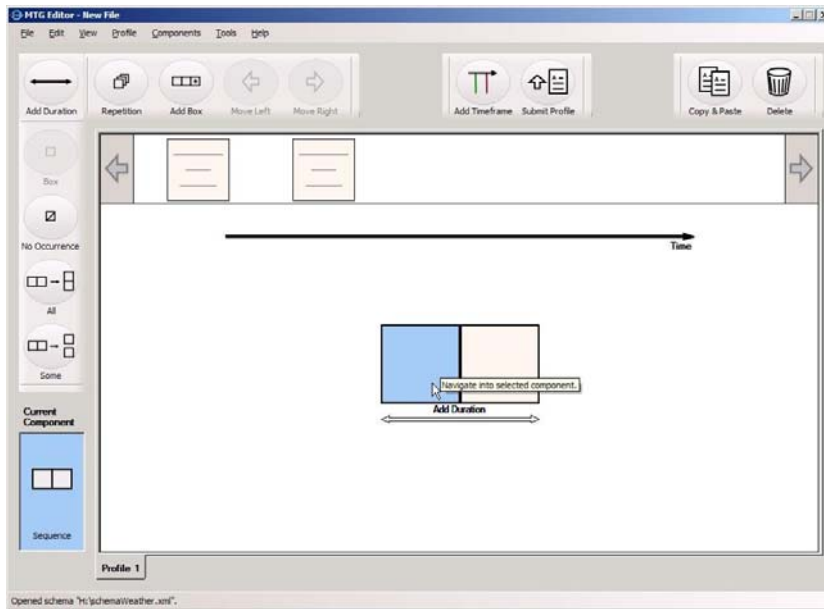


The second operand is going to be created by clicking the “Box” button.

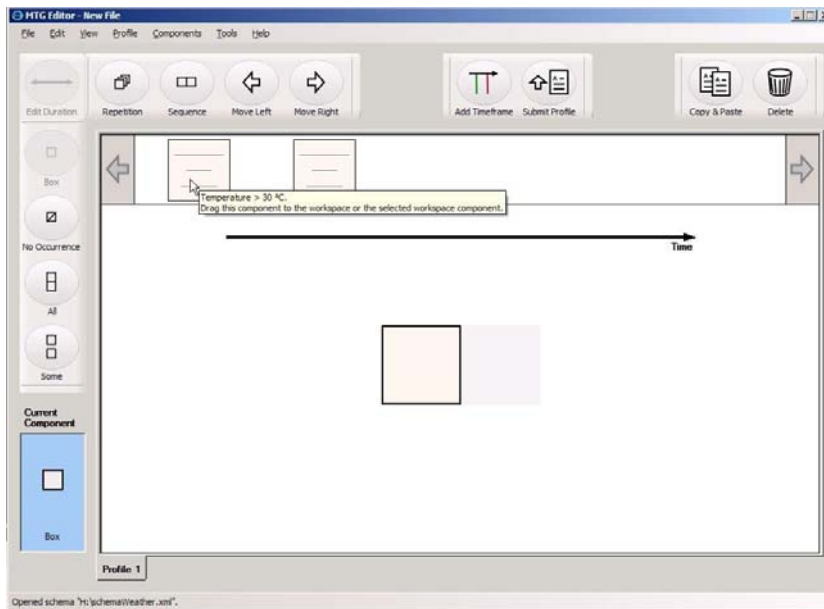


The second operand has been created, the condition box has been edited, and finally been dragged to the component bar.

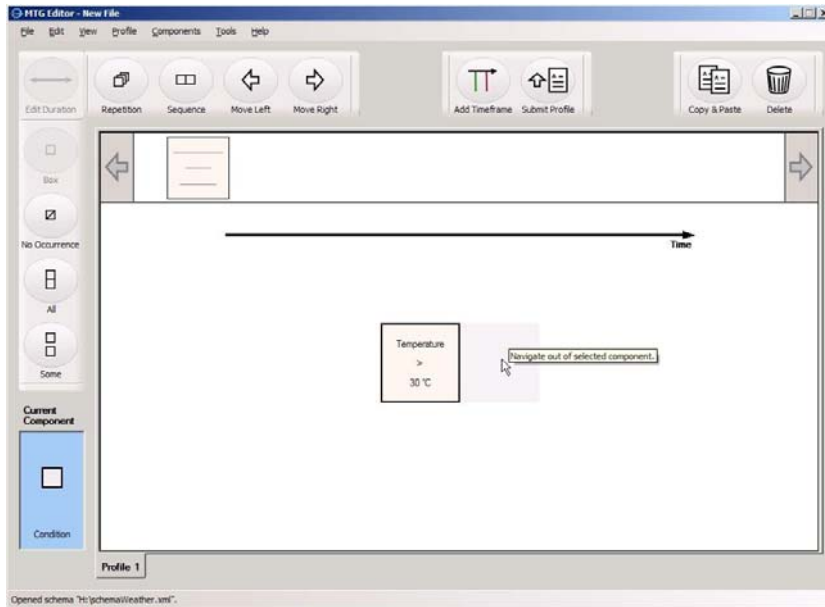
A *sequence* is going to be created by clicking the respective button.



By clicking the left operand of the *sequence*, the user indicates that this operand should be filled by a condition (highlighting is shown in blue).

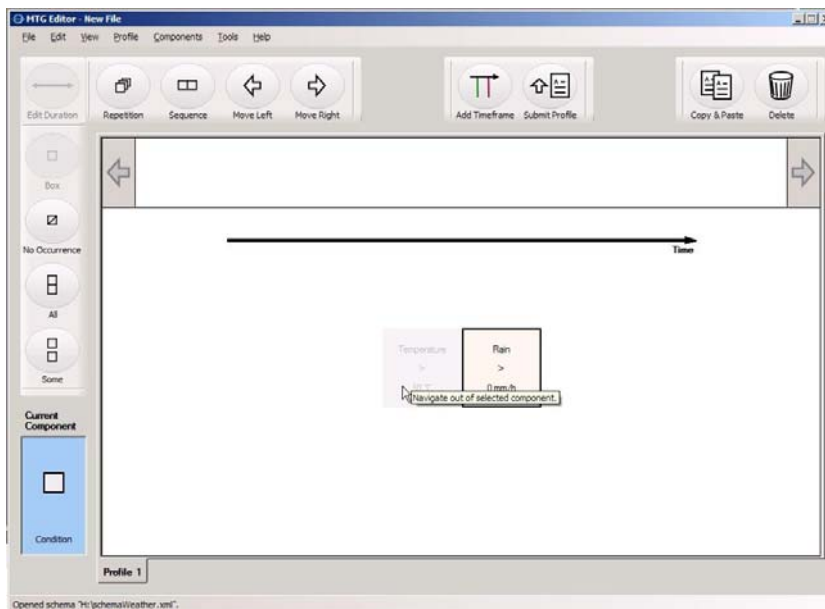


The first operand is going to be dragged from the component bar to its position (left) in the *sequence*.



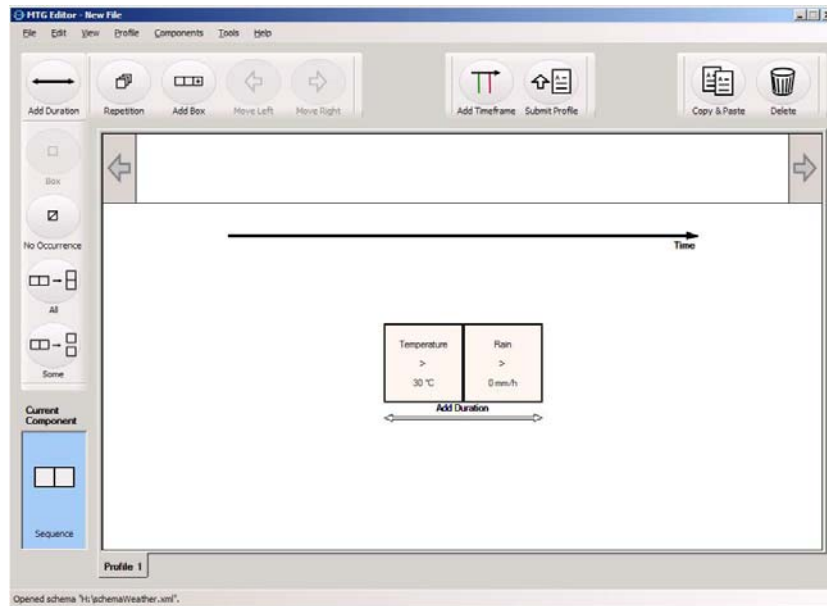
The first operand has been positioned.

The user chooses the position for the second operand (the first click navigates out of the current level of nesting, and the second click selects the right operand of the *sequence*).



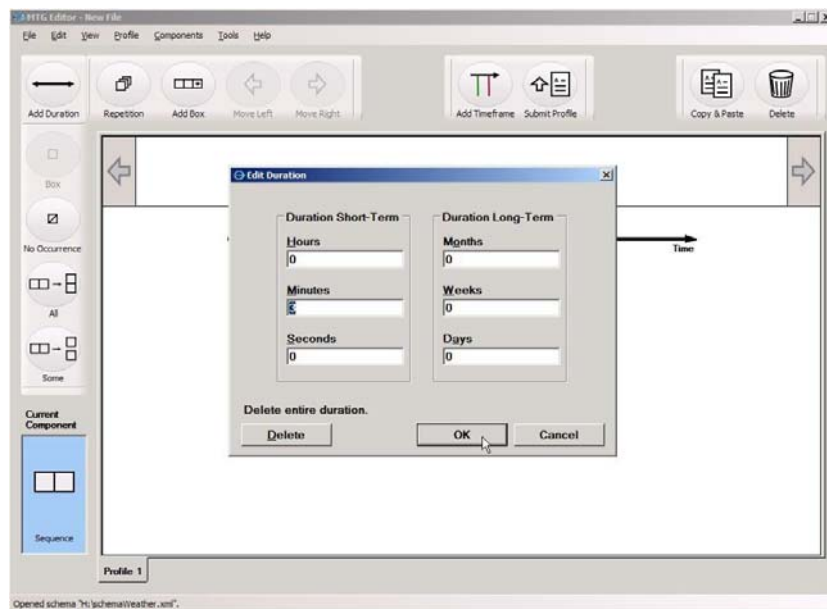
The second operand has been positioned using drag and drop.

The user is navigating to the *sequence* operator (by clicking outside the currently active condition box).

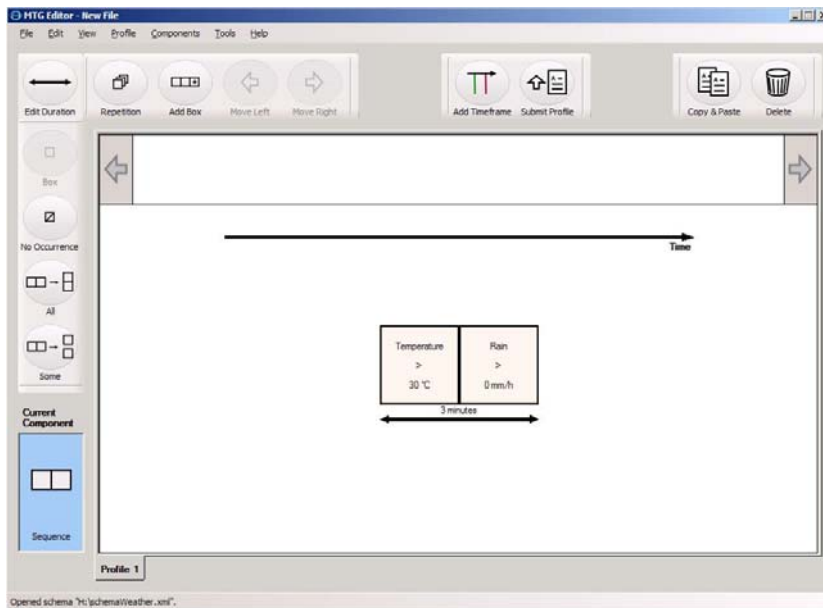


The *sequence* operator is currently active (the current component display shows a *sequence*).

As the *sequence* supports a duration, this parameter is offered to the user (indicator under the profile).



The user edits the duration time (after having clicked on the indicator for the duration).

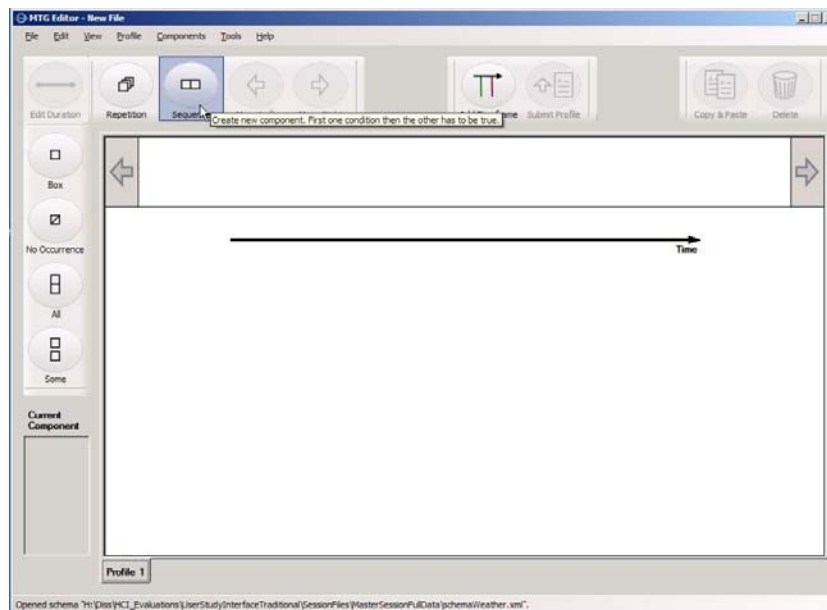


The profile definition is finished.

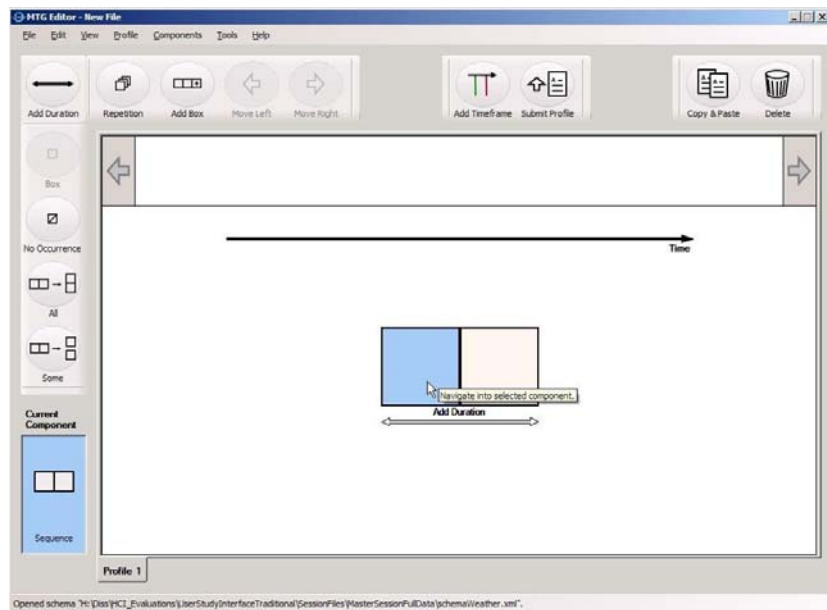
The general bottom-up approach presented in Table 5.1 is applicable to arbitrarily complex profiles involving nested sub-profiles.

Top-Down Approach. For the top-down approach of profile specification, the user first creates the top-most operator of a profile. Imagining the underlying tree structure of the profile expression, they would start the profile definition by specifying the root node and then its operands step by step.

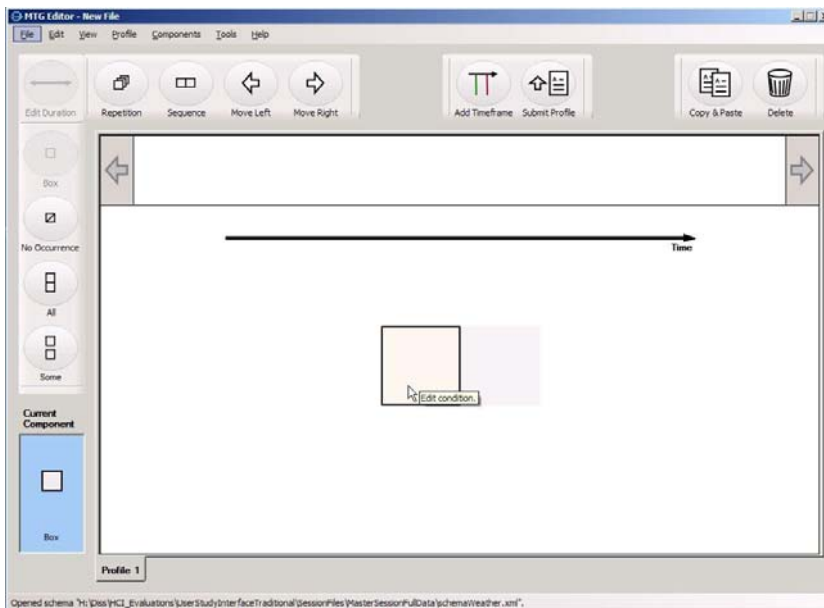
Table 5.2 demonstrates the use of GPDL-UI when the user employs a top-down approach to create a *sequence* with two operands and a duration.

Table 5.2: Top-down profile specification.

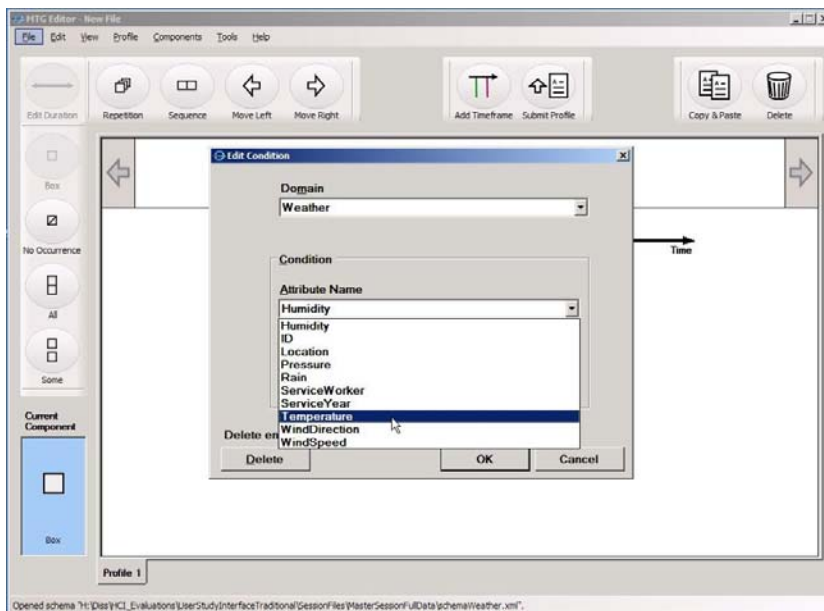
The user creates the *sequence* operator by clicking the respective button.



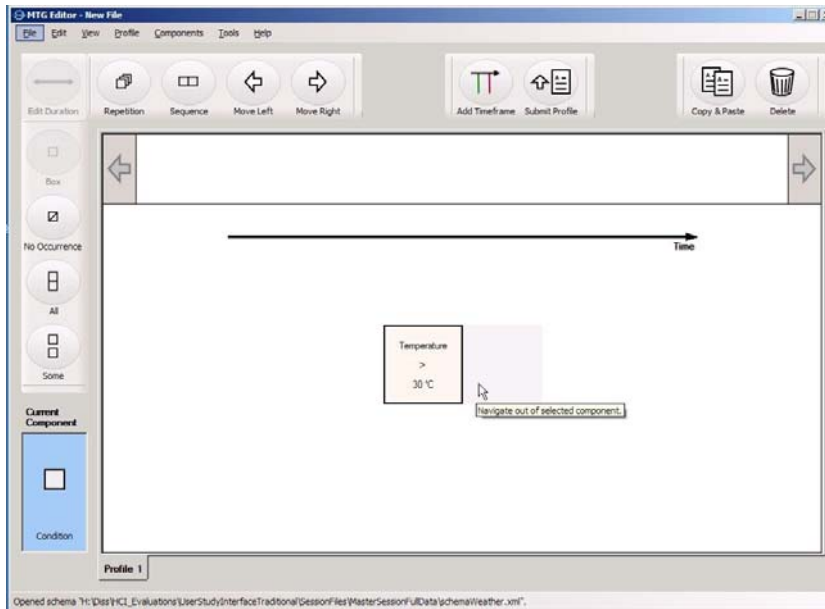
The user is going to navigate into the left-most operand by clicking the left condition box.



The user clicks the condition box to edit the first operand.

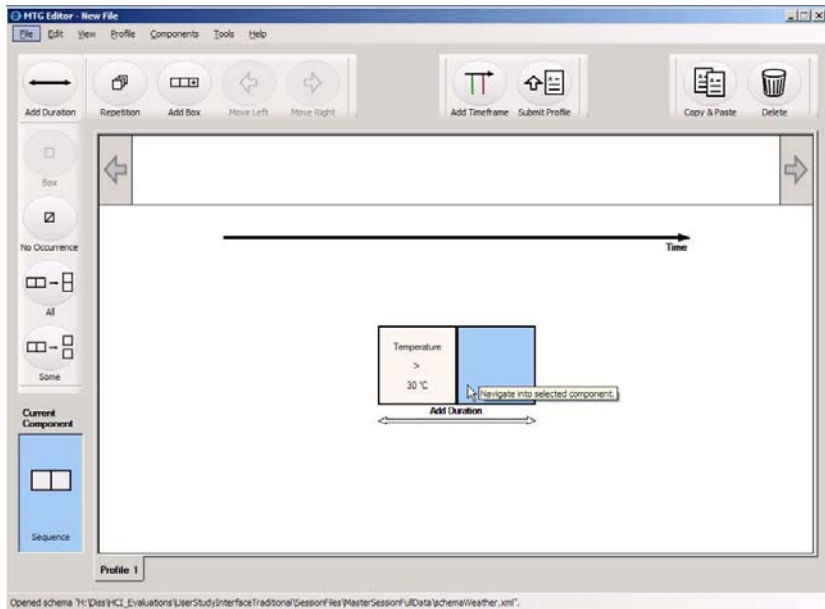


The content of a condition box is being edited.

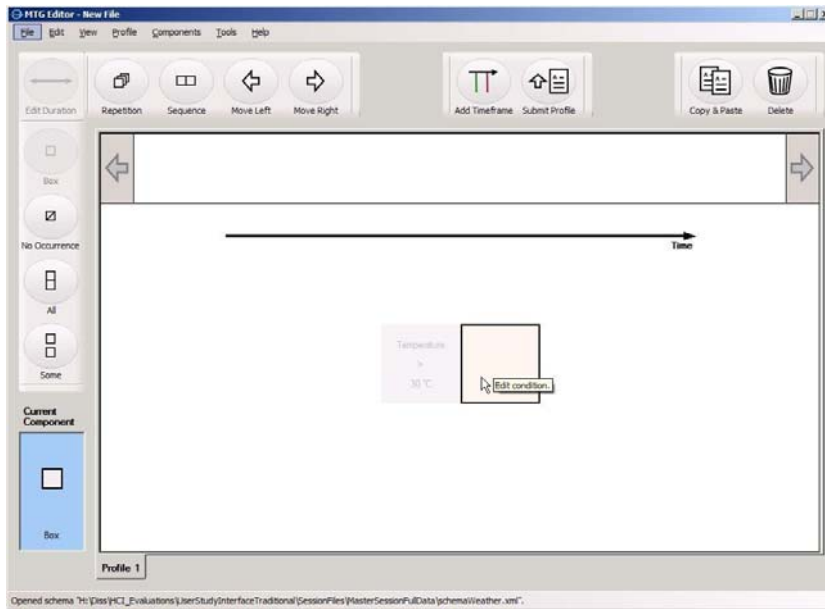


The first operand has been specified.

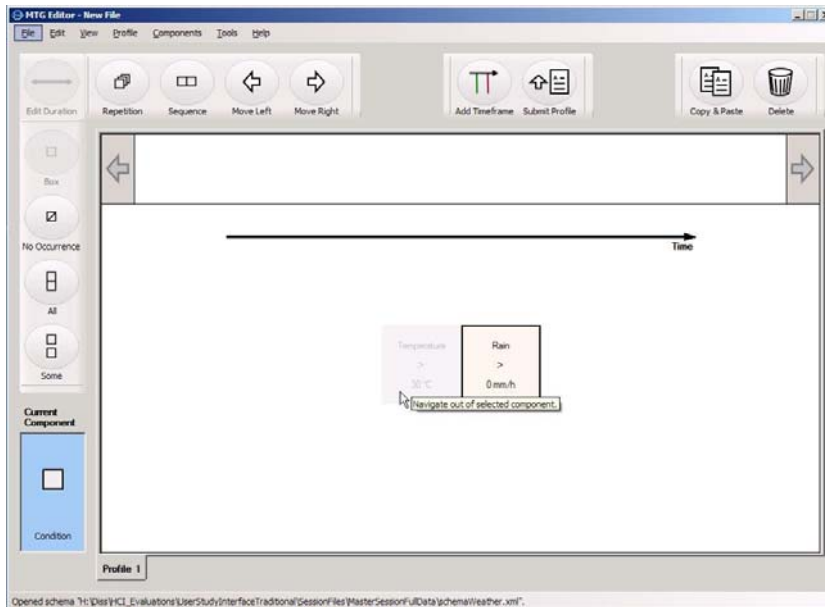
The user is going to navigate out of the left operand by clicking the right operand.



The user is going to navigate into the right operand by clicking it.

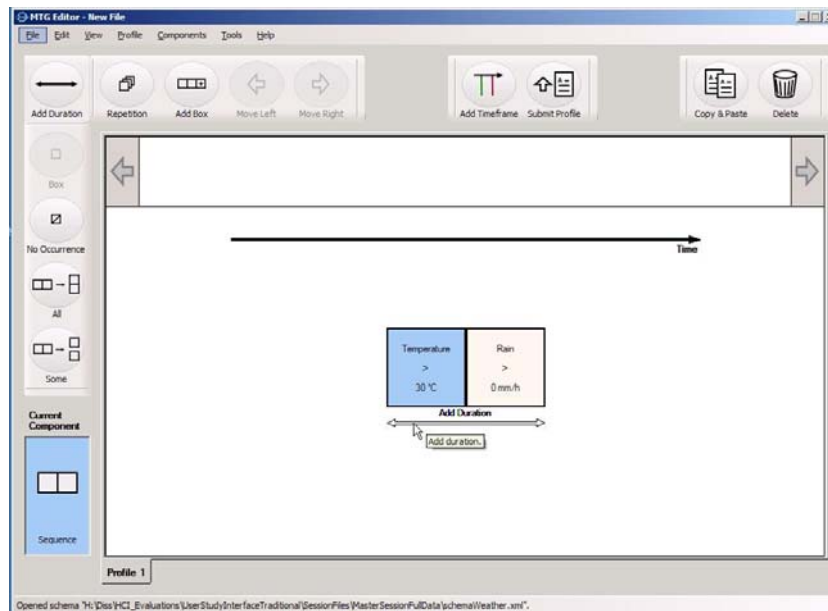


The second operand is going to be edited.



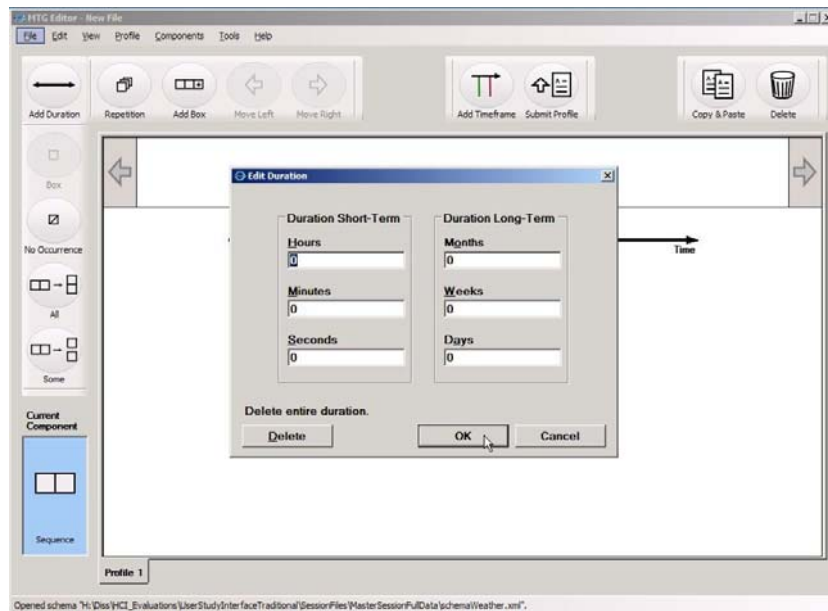
The second operand has been edited.

The user is navigating to the *sequence* level by clicking outside the currently active components (right operand).

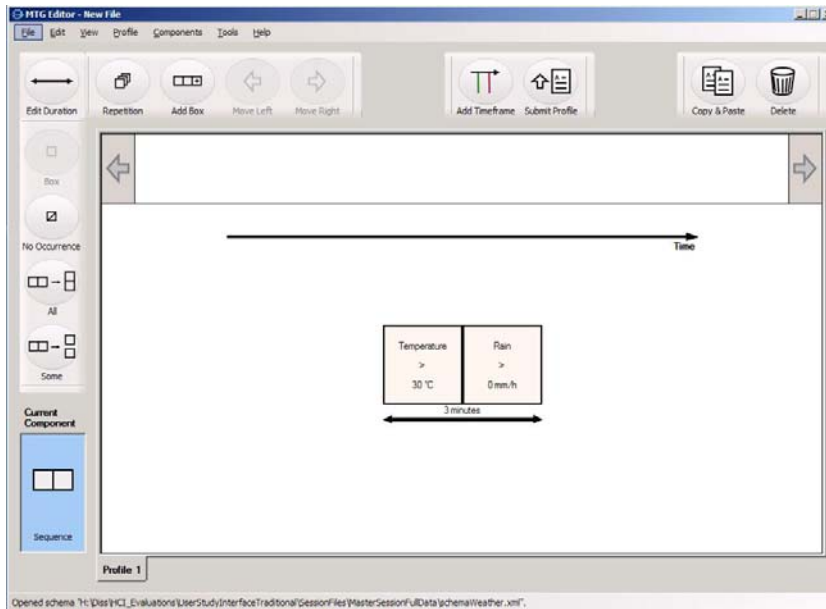


Currently the top-level of the profile is in focus (*sequence*).

The duration parameter is offered for the *sequence*.



The duration time is being edited.



The profile specification is finished.

This top-down approach to specify profiles allows the creation of complex profiles involving various levels of nesting.

Both the top-down and bottom-up approaches can be used together in a mixed approach. This gives users the freedom to define profiles in the way they prefer and to change their approach at any time.

The following section describes in detail how various manipulation operations can be performed on profiles.

5.2 Profile Manipulation

A major GPDUI-specific design challenge regarding the functionality of GPDUI concerned the profile-manipulation operations that should be offered to users. First of all, Section 5.2.1 presents a categorisation of possible edit operations. Then, Section 5.2.2 describes the operations that were chosen to be presented to the user and the

way in which they are presented. The handling of deletions is covered separately in Section 5.2.3.

5.2.1 Taxonomy of Edit Operations

Conceptually, profile manipulation operations can be subdivided into six categories. The following descriptions use the terminology and concepts of both the standard (graphical) GPDL representation and the underlying syntax tree. This procedure allows for the discovery of the interrelation between the presented manipulations on both conceptual levels. Generally, inner nodes in the underlying syntax tree represent operators in GPDL and leaf nodes represent condition boxes. Unary inner nodes are the *repetition* and the *no-occurrence* operator.

Create. The create operation adds new nodes to the syntax tree. Specifically, these can be leaf nodes, that is, condition boxes, and inner nodes, that is, operators with their respective operands. When creating unary operators, one operand is added. For n -ary operators (with $n \geq 2$), it was chosen to automatically create two operands. Two is the minimum number of operands required. Adding further operands is handled by the change-number-of-operands operation (see below).

Node Exchange. The node exchange operation replaces a current n -ary inner node of the syntax tree with $n \geq 2$ by another node, while keeping the children of the original node. Thus, from a graphical viewpoint, the node exchange operation replaces one GPDL operator by another GPDL operator, for example the *some* operator could be changed into the *all* operator.

Embedding. This operation inserts the currently selected node into a newly created operator node (that is, inner node). If the new node represents an n -ary ($n \geq 2$) operator, its children are the currently selected node and an additional empty node. For unary operators, embedding merely adds the current node as the only child to the operator node. As the *repetition* requires an additional parameter (how often to be repeated), this parameter needs to be specified for this operation.

Graphically, this means that a new condition box is placed in some relation to the current component (except for the *no-occurrence* operator). The way both components are located relative to each other expresses what operator was chosen for embedding. For the *no-occurrence* operator, the existing component is crossed out.

Change Number of Operands. The change-number-of-operands operation is designed to add further children to an inner node or to remove existing children.

For GPD-L, it was decided to differentiate between different operators and to offer individual realisations for the change-number-of-operands operation:

- *Repetition*: For the *repetition* operator, which is a unary inner node with a parameter, the operation offers an opportunity to change the number of repetitions. Graphically, this appears as if further operands would be added or some operands would be deleted rather than changing the number of repetitions. Choosing a repetition number of one leads to the parent-deletion operation (see below).
- *No-occurrence*: The *no-occurrence* operator is a unary inner node with exactly one child. Therefore this operation is not applicable to this operator.
- *All, some and sequence*: For the *all*, *some* and *sequence* operators (inner nodes with at least two children), this operation offers adding another operand.

Parent Deletion. The parent deletion operation replaces the parent of the currently selected node with the current node, that is, the child. Hence, the child node is moved up one level in the syntax tree. This operation is directly applicable to the operand of the *no-occurrence* operator. Semantically, it means that the requirement that the operand must not occur is cancelled out. Graphically, the crossing out of a component is removed.

Parent deletion is also possible for the *repetition* operator as a special case of the change-number-of-operands operation (described before). When changing the number of operands to one, the *repetition* is removed semantically; graphically, only one

condition box remains. Parent deletion might also become a special case when deleting operands, as described later on in Section 5.2.3.

Change Operand Position. The change-operand-position operation moves the currently selected operand of an operator to a different position (that is, child nodes are reordered). This operation only has an impact on the semantics of the *sequence* as for all other operators, the order of operands is irrelevant. Thus, it was only implemented for this particular operator.

Considering the combination of each type of edit operation and each type of operator reveals that there is a large number of operations that is possible. The following section analyses different approaches of offering these operations to users and presents the approach that is taken within GPD-L-UI.

5.2.2 Presentation of Edit Operations

An effective manipulation of profiles contains two parts: Firstly, users need to know what manipulation options are possible. Secondly, users need to be able to specify what operation they want to perform.

The following subsection describes two approaches of grouping edit operations in order to allow a structured presentation to users. Based on one of these approaches, the remainder of this section explains how edit operations are presented to the user within GPD-L-UI.

Grouping of Edit Operations. Directly presenting all available edit operations to users leads to an overwhelming amount of possibilities. Structuring these possible operations can reduce this overwhelming effect.

Structuring According to Operation Category. One option of structuring operations is to use the six conceptual categories of operations presented in the previous

subsection. For example Figure 5-2 shows the group of all node-exchange operations that are possible for binary operators.

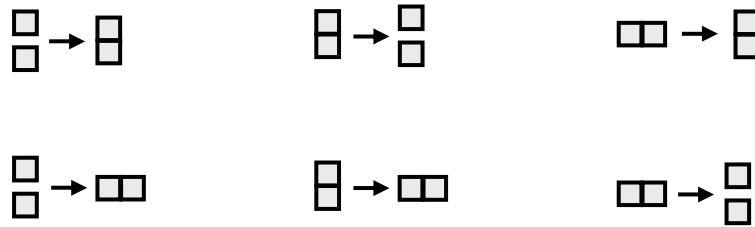


Figure 5-2: Node exchange profile manipulation operations.

This approach asks for a thorough mental analysis by the user in order to determine the operation they want to perform. In particular the required abstraction to the conceptual categories assumes that users understand the internal mechanisms leading to these operations. However, the underlying concept (syntax tree) is hidden from the user in GPD-L-UI.

Nevertheless, once these mechanisms are understood, the common schema of symbols could guide the user in finding the appropriate operation.

Structuring According to Current Operator. The other major structuring option is based on the currently selected operator. For example, one group of operations contains all possible operations for the *all* operator, whereas another group contains all operations for the *sequence* operator. Figure 5-3 shows an example of the group for the *all* operator: the first line contains the create operation, the second line contains both node-exchange operations, the third line the six possible embedding operations and the last line the change-number-of-operands operation.

Taking this approach to structuring does not require an abstraction on the side of the user, as users are aware of what component they have currently selected.

The following presentations are thus based on the option of grouping edit operations according to their operator (current operator) rather than using a grouping approach by operation category.

Selecting an Edit Operation. The edit operations that are shown at any given time in GPDL-UI are restricted to one group, the group corresponding to the operator of the currently active component. The user decides which of these operations to select, for example one of the operations shown in Figure 5-3 if the active component is an *all* operator.

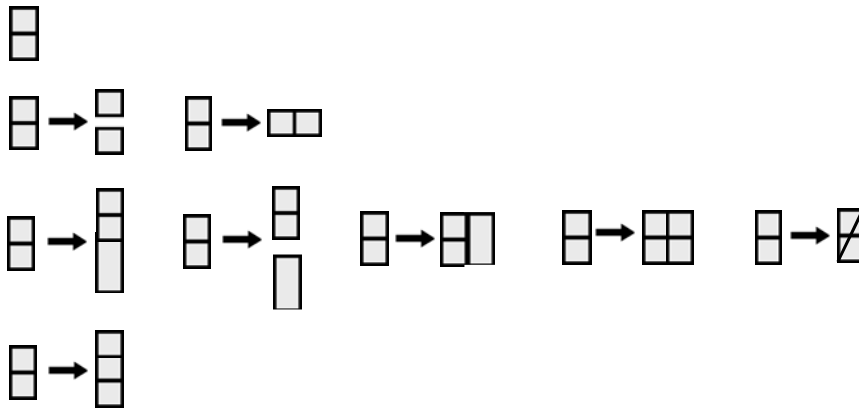


Figure 5-3: Profile manipulation operations for the *all* operator.

Selection According to Manipulation Operator. Taking a task-oriented presentation approach, users will know what particular operator they want to apply within their manipulation. Thus, it is possible to restrict the potential operations by the help of this manipulation operator. For example, of the operations shown in Figure 5-3 (all possible operations for an *all* operator with two operands), two operations involve the *some* operator as manipulation operator (node exchange in line 2 and embedding in line 3) whereas only one operation involves the *no-occurrence* operator.

Restriction to Relevant Operations. It is favourable if only one operation would be possible for each pair of current operator and manipulation operator. For those manipulation operators that allow more than one option, one possible operation can be chosen to be available. This operation should be the one that is required in most cases. This restriction of operations does not mean that other operations can not be performed. The separation of GPDL-UI into a workspace and a component bar still allows these operations, as it is detailed later on.

Taking the restriction approach, at most one possible operation remains for each manipulation operator at any given system state (defining the current operator). This allows for an elegant presentation of edit operations to users, as described in the following subsection.

Presentation to the User. It suffices to offer six buttons in GPD-L-UI (shown in areas 7 and 8 in Figure 5-1) to access all relevant operations apart from the change-operand-position operation (there are two additional buttons for this). The six buttons are overloaded and their images change according to the functionality that is offered at a given time. Buttons without functionality at a certain state are deactivated.

The general idea is that each button is responsible for exactly one operator (or the individual condition box). This avoids users searching for the functionality they want to access. As textual description, buttons always contain their operator name (or box) except in case of offering the change-operand-number operation (stating “Add Box” for the *sequence*, *all* and *some* buttons, and “Change Box No” for the *repetition* button) and the delete-parent operation (stating “Occurrence” for the box button).

For example, anything related to the manipulation operator *some* will always be found at the same button: If the current component on the workspace is an *all* operator, the button for *some* is overloaded with the node exchange operation that replaces the *all* operator by the *some* operator. If the workspace is currently empty, the same button (for *some*) creates a new component involving a *some* operator with two empty operands.

Table 5.3 presents the overloading of all buttons for all possible system states. Buttons are represented by the columns of the table whereas the current state of the workspace is represented by the rows. The table indicates what kind of operation is offered for each button (columns) under all possible circumstances (rows). The table also shows the symbols that are placed on the buttons as an explanation to the user of the editor.

Table 5.3: Overloading of the buttons for condition box and operators. Buttons are shown in the columns, current workspace component in the rows.

	Box button	No-occur. button	All button	Some button	Repetition button	Sequence button
Empty workspace						
	Create	Create	Create	Create	Create	Create
Box						
	Inactive	Embed	Embed	Embed	Embed	Embed
No-occurrence						
	Parent deletion	Inactive	Embed	Embed	Embed	Embed
All						
	Inactive	Embed	Change operand number	Node exchange	Embed	Node exchange
Some						
	Inactive	Embed	Node exchange	Change operand number	Embed	Node exchange
Repetition						
	Inactive	Embed	Embed	Embed	Change operand number	Embed
Sequence						
	Inactive	Embed	Node exchange	Node exchange	Embed	Change operand number

As described before, some operations cannot be directly accessed by taking the *structuring according to current operator* presentation approach in combination with each

button being responsible for exactly one operator. For example, embedding a component with the *all* operator into a *some* operator cannot be directly performed. Instead the user is required to move the *all* operator to the component bar, to create a new *some* operator, and to drag the original component from the component bar into one operand of the workspace component.

5.2.3 Deletion Logic

The currently selected component can be deleted by using the deletion button at any time. Furthermore, condition boxes allow deletion by clicking a little closing symbol (an “x” in the upper right corner) whenever they are selected. Nevertheless, special considerations need to be taken into account for some situations.

A special case occurs when deleting all but the last operand of n -ary operators ($n \geq 2$). It needs to be considered that only retaining one operand makes the presentation of the operator invisible. It was decided to internally remove the operator and to replace it by its remaining operand (child in the internal syntax tree). This corresponds to the parent-deletion operation, described in Section 5.2.1.

As an extension to this, if the duration of an n -ary operator has been defined and the user attempts to delete the second last operand, another special case occurs. There is an information message that makes the user aware of the fact that by deleting this operand, they can not retain the duration that belongs to this n -ary operator. The message lets the user decide whether to remove the operand (and thus the operator and its duration) or to replace the operand by an empty condition (retaining operator and duration).

5.3 Automatic Grouping

GPD-L contains the concept of grouping boxes (Section 3.6). Within GPD-L-UI, grouping of profiles is done automatically in order to lead the user through nested profiles. Hence, certain operations that alter the structure of a profile might introduce grouping boxes. Other operations, however, might lead to the removal of grouping

boxes that are not required anymore. Users thus do not need to consider grouping their profiles properly as GPDL-UI ensures unambiguous profiles.

5.3.1 Reasons for Grouping

Normally, the widths and heights of boxes visualise the structure of a profile. Additionally, the widths of duration arrows normally indicate what operator and operands this duration belongs to: Visually this can be expressed as everything that is above the duration arrow. However, neither technique works for all profiles as under certain circumstances the profile becomes ambiguous.

Figure 5-4 shows an example of a profile that is ambiguously presented without grouping boxes. It would be unclear which of the following two interests are described by the profile:

- Alert if Mr X has at least three high blood pressure readings (diastole above 180mmHg and systole above 120mmHg)
- Alert if there are three readings for Mr X and some patient's diastole is above 180mmHg on three occasions and some patient's systole is above 120mmHg on three occasions

Therefore, grouping boxes are required for clarification in these situations. Assuming the first of the previously described interests is intended by the profile, three vertical grouping boxes create an unambiguous visualisation of the profile.

The general goal of automatic grouping is to ensure unambiguous profiles. The rules presented in the following sub-section fulfil this goal. They represent general rules that do not aim to minimise the created grouping boxes but ensure a common and consistent approach to grouping profiles in GPDL-UI.

<p>patient = Mr. X</p>	<p>patient = Mr. X</p>	<p>patient = Mr. X</p>
<p>systole > 180 mmHg</p>	<p>systole > 180 mmHg</p>	<p>systole > 180 mmHg</p>
<p>diastole > 120 mmHg</p>	<p>diastole > 120 mmHg</p>	<p>diastole > 120 mmHg</p>

Figure 5-4: Example of ambiguous profile.

5.3.2 Grouping Rules

This subsection presents the grouping rules for profiles. To be able to formulate them, the GPDUI grammar needs to be extended slightly.

Extension of GPDUI Grammar. The grammar that is used to define the syntax tree of a GPDUI profile was introduced in Section 3.7 (Figure 3-27). Within GPDUI this grammar is slightly extended as it is required to specify boxes with empty conditions (as can be seen, for example, in Table 5.1 and Table 5.2). Figure 5-5 shows this extension of the production rule for the condition.

```

condition : attribute ',' comparisonOperator ','
              value
              | 'UNSPECIFIED' ;

```

Figure 5-5: EBNF of the extension to the internal GPDUI representation.

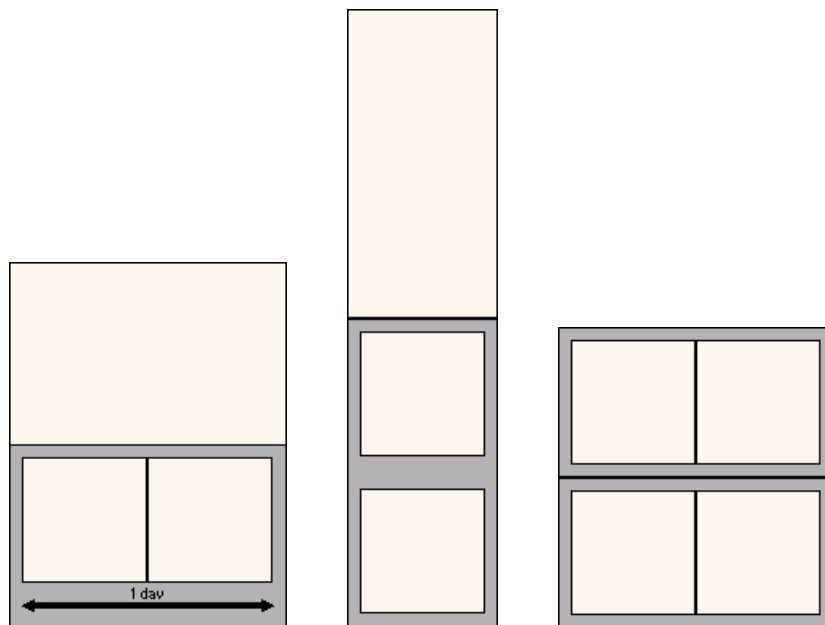
The following subsections specify the automatic grouping rules that are applied within GPDUI. These rules are applied recursively in a bottom-up fashion: As a first step, all operands of the current operator are grouped. As a second step, the rules are applied to the current operator.

The grouping rules differ for the available operators. Within examples, the time-frames of profiles are omitted for brevity. Furthermore, empty conditions are used throughout, as grouping is performed based on the structure of profiles and not based on the contents of conditions.

Grouping Rules for the *All* Operator. For the *all* operator, three grouping rules are applied:

- Rule 1 (*all*): Group each operand that is specified by a duration.
- Rule 2 (*all*): Group each operand that is a *some* operator.
- Rule 3 (*all*): Group each operand that is a *repetition* or *sequence* if one of its adjacent operands (considering the syntax tree, this would be a direct sibling) is a *repetition* or *sequence*.

Example 29 in combination with Figure 5-6 shows the application of these rules.



**Figure 5-6: Grouped profiles using the rules for the *all* operator (Example 29):
Rule 1 (left), Rule 2 (middle) and Rule 3 (right).**

Example 29 In the following profile (Figure 5-6, left), the second operand (sequence) is grouped according to Rule 1 (all):

```
( ALL
  UNSPECIFIED
  ( SEQUENCE {1 d} UNSPECIFIED UNSPECIFIED )
)
```

In the following profile (Figure 5-6, middle), the second operand (some) is grouped according to Rule 2 (all):

```
( ALL
  UNSPECIFIED
  ( SOME UNSPECIFIED UNSPECIFIED )
)
```

Rule 3 (all) is applied to the two operands in the following profile (Figure 5-6, right), that is, both sequences are grouped:

```
( ALL
  ( SEQUENCE UNSPECIFIED UNSPECIFIED )
  ( SEQUENCE UNSPECIFIED UNSPECIFIED )
)
```

Grouping Rules for the *Some* Operator. There are three grouping rules for the *some* operator (similar to the *all* operator):

- Rule 1 (*some*): Group each operand that is specified by a duration.
- Rule 2 (*some*): Group each operand that is an *all* operator.
- Rule 3 (*some*): Group each operand that is a *repetition* or *sequence* if one of its adjacent operands (considering the syntax tree, this would be a direct sibling) is a *repetition* or *sequence*.

These rules are applied in Example 30 and in Figure 5-7.

Example 30 The second operand (sequence) of the following profile (Figure 5-7, left) is grouped according to Rule 1 (some):

```
( SOME
  UNSPECIFIED
  ( SEQUENCE {1 d} UNSPECIFIED UNSPECIFIED )
)
```

The first operand (all) of the following profile (Figure 5-7, middle) is grouped according to Rule 2 (some):

```
( SOME
  ( ALL UNSPECIFIED UNSPECIFIED )
  UNSPECIFIED
)
```

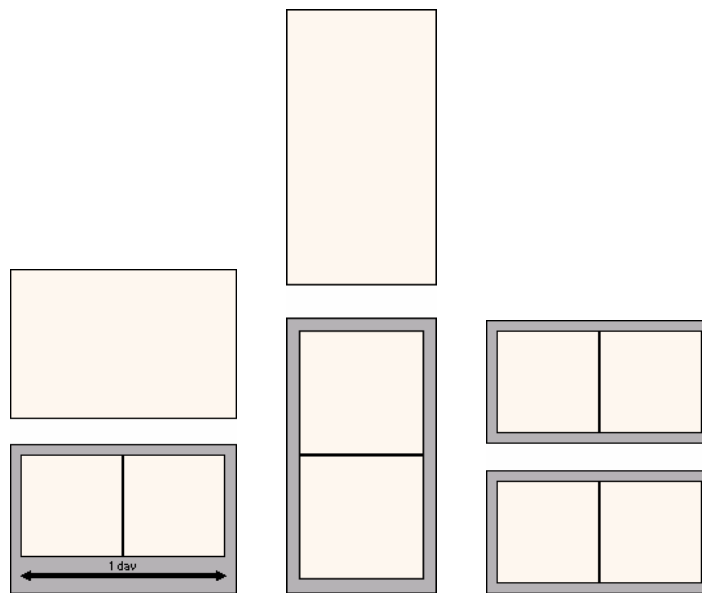


Figure 5-7: Grouped profiles using the rules for the *some* operator (Example 30): Rule 1 (left), Rule 2 (middle) and Rule 3 (right).

Finally, Rule 3 (some) is applied to both operands (sequence) of the following profile (Figure 5-7, right):

```
( DISJUNCTION
  ( SEQUENCE UNSPECIFIED UNSPECIFIED )
  ( SEQUENCE UNSPECIFIED UNSPECIFIED UNSPECIFIED )
)
```

)

Grouping Rules for the *Repetition* Operator. There are two grouping rules for the *repetition* operator:

- Rule 1 (*repetition*): Group each operand that is specified by a duration.
- Rule 2 (*repetition*): Group each operand that is an *all*, *some*, *repetition* or *sequence* operator.

Example 31 in conjunction with Figure 5-8 shows the application of these rules.

Example 31 *The operand (no-occurrence) that is repeated in the following profile (Figure 5-8, left) is grouped applying Rule 1 (repetition):*

```
(REPETITION[3] {2 M}
  (NOOCCURRENCE {1 M } UNSPECIFIED)
)
```

Rule 2 (repetition) leads to the grouping of the repeated sub-profile (all) within the following profile (Figure 5-8, right):

```
(REPETITION [2]
  (ALL UNSPECIFIED UNSPECIFIED)
)
```

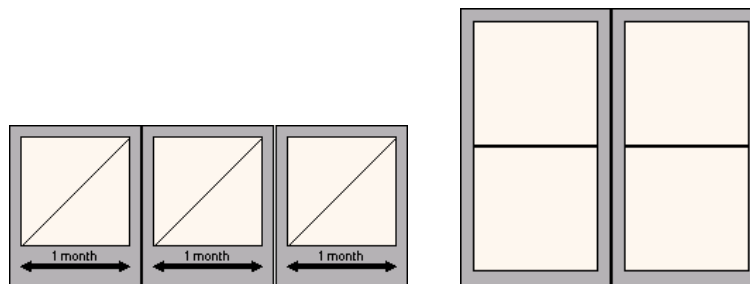


Figure 5-8: Grouped profiles using the rules for the *repetition* operator (Example 31): Rule 1 (left), Rule 2 (right).

Grouping Rules for the *Sequence* Operator. The *sequence* operator requires three grouping rules:

- Rule 1 (*sequence*): Group each operand that is specified by a duration.
- Rule 2 (*sequence*): Group each operand that is a *repetition*.
- Rule 3 (*sequence*): Group each operand that is a *some* or *all* operator if one of its adjacent operands (considering the syntax tree, this would be a direct sibling) is a *some* or *all*.

Example 32 in combination with Figure 5-9 and Figure 5-10 show the application of these three rules.

Example 32 *The first operand in the following profile (Figure 5-9, left) is grouped by applying Rule 1 (sequence), whereas the second operand is not grouped by applying this rule.*

```
(SEQUENCE
  (SEQUENCE {1 w} UNSPECIFIED UNSPECIFIED)
  (SEQUENCE UNSPECIFIED UNSPECIFIED)
)
```

Rule 2 (sequence) groups the second operand (repetition) of the following profile (Figure 5-10):

```
(SEQUENCE
  UNSPECIFIED
  (REPETITION[3] UNSPECIFIED)
)
```

Both operands (all, some) of the following profile (Figure 5-9, right) are grouped by applying Rule 3 (sequence):

```
(SEQUENCE
  (ALL UNSPECIFIED UNSPECIFIED)
  (SOME UNSPECIFIED UNSPECIFIED)
)
```

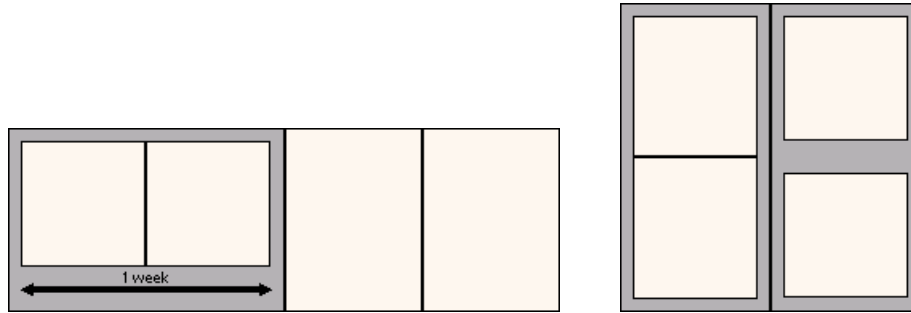



Figure 5-9: Grouped profiles using the rules for the *sequence* operator (Example 32): Rule 1 (left), Rule 3 (right).

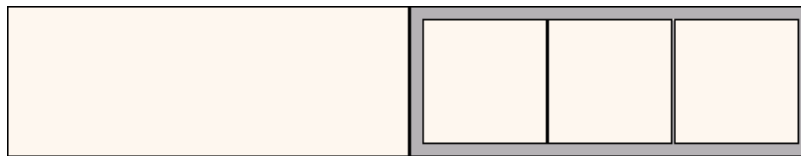


Figure 5-10: Grouped profiles using the Rule 2 for the *sequence* operator (Example 32).

Grouping Rules for the *No-occurrence* Operator. The concept of *no-occurrence* requires two grouping rules to ensure unambiguous profiles.

- Rule 1 (*no-occurrence*): Group the operand if it is specified by a duration.
- Rule 2 (*no-occurrence*): Group the operand if it is a *some* operator or a *no-occurrence* operator.

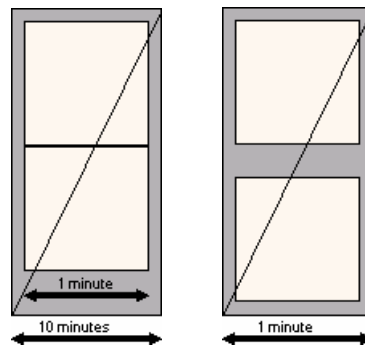


Figure 5-11: Grouped profiles using the rules for the *no-occurrence* operator (Example 33): Rule 1 (left), Rule 2 (right).

These rules are detailed in Example 33 in conjunction with Figure 5-11.

Example 33 *As the operand (all) of this profile (Figure 5-11, left) is specified by a duration, it is grouped by applying Rule 1 (no-occurrence):*

```
(NOOCCURRENCE {10 m}
  (ALL {1 m} UNSPECIFIED UNSPECIFIED)
)
```

Rule 2 (no-occurrence) is applied in case of the following profile (Figure 5-11, right):

```
(NEGATION {1 m}
  (DISJUNCTION UNSPECIFIED UNSPECIFIED)
)
```

5.4 Summary

This chapter introduced the editor GPDL-UI for specifying single-user profiles in GPDL. Section 5.1 described the general interface, and its layout and interaction mechanisms.

Section 5.2 focused on describing the major design challenge for GPDL-UI—the presentation of the available profile manipulation operations to the user. Based on a categorisation of available operations, an approach was developed that offers an overview of possible operations while maintaining a consistent presentation to the user.

GPDL-UI automatically groups profiles to ease the work of users and to ensure an unambiguous profile presentation. The content of Section 5.3 described how this can be achieved.

The following chapter evaluates GPDL-UI in a user study.

6 User Study of the Editor GPDL-UI for the Specification of Single-user Profiles

This chapter presents a user study to evaluate the usability of GPDL-UI, the interface for the single-user profile definition language GPDL. The goal of the study is to explore the usability and effectiveness of the interface and the underlying profile specification approach of GPDL-UI.

Introductory details of the user study are presented in Section 6.1 and Section 6.2; quantitative results are shown in Section 6.3, followed by qualitative results in Section 6.4. A discussion of the study and its results conclude this chapter in Section 6.5.

6.1 Goals of the Study

This study examined users' interaction with the interface GPDL-UI for the specification of profiles for alerting systems using the underlying graphical profile definition language GPDL that was evaluated in a previous user study (Chapter 4). The goal was to find out how effectively and correctly users are able to specify profiles in GPDL using GPDL-UI. The study followed the same goals as the paper-based study that was described in Chapter 4. The goals were to determine:

- How accurately users can specify profiles using GPDL-UI
- How accurately users can interpret profiles shown with GPDL-UI
- Users' subjective experience of GPDL-UI for profile specification
- Users' subjective experience of GPDL-UI for profile interpretation

In order to analyse the answers to the above goals in detail, the users' general experience using the interface was observed. Several variables were analysed: The intuitiveness of the interface as well as the ease and satisfaction that users were having when using GPDUI. Moreover, the mental and perceptual activity users had to invest in order to use the software were analysed as well as the physical activity that was required for the profile specification. Participants were asked about their frustration levels when using the software and had to consider how successful they thought they were in completing the given tasks correctly. The correctness of the profiles that were given by participants of the study was also analysed.

6.2 Experimental Design

6.2.1 Method

The study was an observational computer-based laboratory study that used a within-subjects design. The study was controlled by randomisation. For the entire study the participants were asked to think aloud. No time constraints were posed on the participants because the aim of the study was to discover the general approach used by participants when using the interface and GPDUI, rather than undertaking an efficiency analysis.

6.2.2 Tasks

Condition 1 (Specification Tasks). Each specification task contained a description of an alerting task in English. With the help of the interface the participants had to specify profiles that express the same semantics as were given in the English language description.

Condition 2 (Interpretation Tasks). The subjects were shown profile specifications in the interface. They were asked to describe the semantics of these profiles in English by writing them down on paper.

The condition order was alternated for consecutive participants. For each of these two conditions, the tasks increased their degree of difficulty in order of their presentation—for both the tutorial and the training phase.

Profile Levels. The study used the same profile levels as were used in the paper-based study of GPDL; refer to Section 4.2.2 for details.

6.2.3 Procedure

A copy of the Bill of Rights was given to each participant. Two copies of the Research Consent Form were read and signed by the researcher and the participant. One copy was retained by the researcher, the other given to the participant. At the beginning of each interview the researcher verbally explained these documents, with particular reference to the participant's right to withdraw from the study at any point without explanation.

Each participant was guided through a tutorial on the computer that introduced them to the profile specification approach in question. This was followed by a training phase and an evaluation phase for each of the two conditions.

Tutorial. The tutorial was consistent with the tutorial from the paper-based study presented in Section 4.2.3 using weather domain data.

Training. The training phase entailed eight tasks for each condition. All participants were given the same tasks. The tasks were presented with increasing difficulty and complexity to enable the subjects to learn GPDL and the use of the interface progressively in small increments. During this phase the participants were asked to think aloud.

Evaluation. The evaluation consisted of 11 tasks in each of the two conditions. The evaluation phase was randomised for each condition. All participants were presented with the same tasks. The tasks in each group belonged to levels of varying difficulty and complexity. During the evaluation the participants were asked to think aloud.

Conclusion. The study was concluded with a questionnaire that collected information about the background of the participant followed by a short interview about the experience of the participant with GPDUI and the use of the interface.

6.2.4 Data Captured

A questionnaire was used to collect the participants' background information and to collect their comments on GPDUI. The observer took notes as the participants were completing the tasks and thought aloud. The profile specifications of the subjects were saved as files. The English language descriptions of the given profile were written on paper by the participants and retained by the researcher. For each session an audio track and a video of the participant was recorded. The computer screen was also captured on video.

6.2.5 Participants

The pilot study was undertaken with four participants; three postgraduate students and one faculty member, all computer scientists. One of these participants had already taken part in the paper-based study. Two participants in the full study had already heard about the project and had completed some sample tasks of the paper-based study in a tutorial of one of their previous undergraduate papers.

The full study was undertaken with 18 participants. Among them were two groups of participants: One group of students were undergraduate computer graphic design students. While they were well-versed in using computers, their education has no elements of formal logic training or education on data structures such as trees or graphs. Therefore for the purpose of the study, they can be regarded as non-computer scientists. The other group that participated were undergraduate computer science students who at the time of the study attended a course on usability engineering. Two of the participants of the full study were female, the other sixteen male. Their ages ranged from 21 to 31 years—with a mean of 22.06 years and a standard deviation of ~ 2.86 years. 3 out of the 18 participants did not have English as their native language.

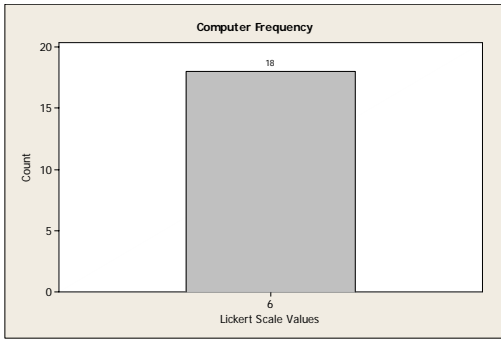


Figure 6-1: Computer frequency.

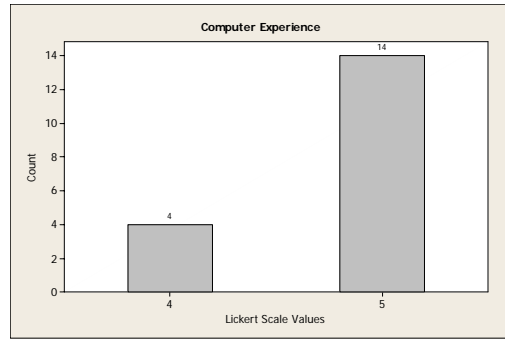


Figure 6-2: Computer experience.

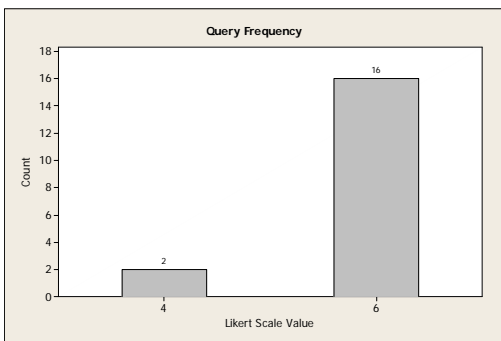


Figure 6-3: Query frequency.

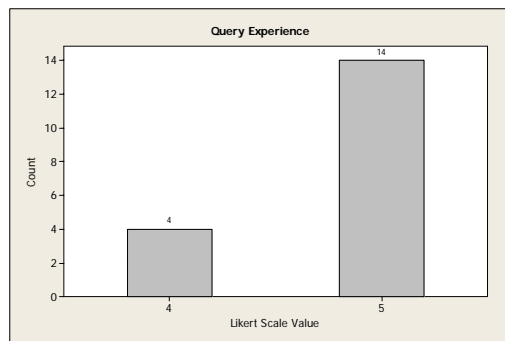


Figure 6-4: Query experience.

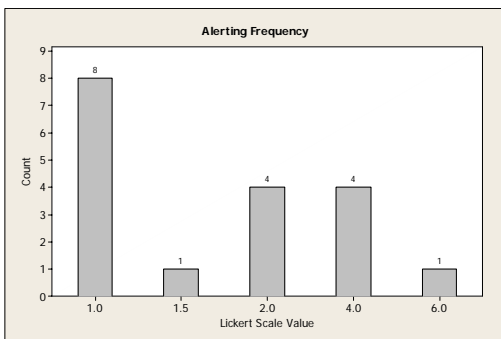


Figure 6-5: Alerting frequency.

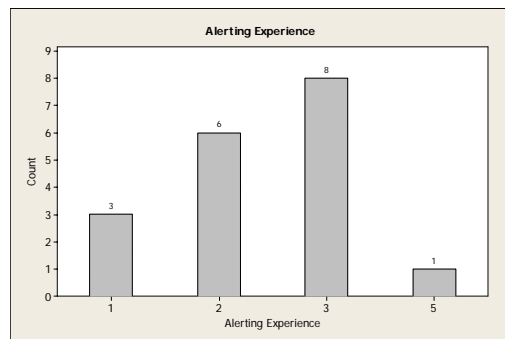


Figure 6-6: Alerting experience.

All participants were very experienced in dealing with computers (see Figure 6-1 and Figure 6-2) and in forming queries (see Figure 6-3 and Figure 6-4). However, most of them had little experience in setting up alerts. The only alerts participants had set up prior to the study were alerts for online auction sites like TradeMe¹⁸ or similar. For details refer to Figure 6-5 and Figure 6-6.

6.2.6 Pilot Study

Observations. The observer observed the participants point out a number of smaller issues. Their comments showed no recurrent patterns, that is, no issue accumulated comments of the same kind. Some observations concerned problems with GPD L, some with the layout of the software’s interface. Other observations covered the interaction processes that were used. Some observations concerned the task descriptions that were used in the study. Some examples of the observations made are as follows.

GPD L. Two groups of issues concerned concepts that are directly related to the language that is used by the interface. They covered the duration and the timeframe.

One of the users preferred to be able to specify a new duration by clicking into the area typically covered by the duration arrow in order to define a new duration.

The major issue regarding the duration that was revealed in the pilot study was an issue that had already been alluded to in the previous paper-based study of the GPD L language. However, it had not been possible to fully identify the cause for the problem in that study. A participant in this study was confused on how to express that “all” of a set of given conditions were supposed to occur within one hour. A discussion with that participant showed that they had misunderstood the exact meaning of the *all* operator. They assumed it meant that all conditions should occur “at the same time”. However, it means that they occur “within” a given time duration.

¹⁸ An online auction site in New Zealand, <http://www.trademe.co.nz/>

A couple of subjects did not understand the distinction between the absolute and repetitive timeframes straight away. One participant mentioned not understanding the graphical symbols that were used in the edit window to represent the two kinds of timeframe. That participant mentioned that it might be helpful if the information about the absolute time would be also shown in the tab for the specification of the repetitive elements of the timeframe.

Interface Layout. Other comments made by the subjects regarded issues that in the broader sense are related to the layout of the interface and its elements, that is, dialogue windows, buttons, labels and symbols, component bar and workspace.

At first glance, none of the participants knew what the function of the component bar was (which they commonly referred to as “that thing”). However, after a short introduction into the possibilities of this concept and after a couple of training tasks they all thought it as helpful for subdividing problems.

Regarding the workspace, two of the subjects criticised issues regarding the size of some profiles on the screen and the lack of scrolling. There was a scroll mechanism but it was not fully working.

The button symbols adapting to the current operating mode (cf. Section 5.2.2) were appreciated.

Interaction Process. Another area covered by some comments of the participants was the interaction processes of the program.

Two participants mentioned that they would like a stronger contrast of the currently selected component (Section 5.1.3). This particularly seemed to pose a problem when working with the *repetition* operator.

One participant suggested that using a drag and drop mechanism instead of a click mechanism for moving components between the component bar and workspace would be good.

The undo functionality was commented upon positively by nearly all participants.

Success Rate and Likert Scale Responses. The overall success rate for completing the given tasks was relatively high for the participants of the pilot study. Also the Likert scale responses regarding intuitiveness, ease and satisfaction were very positive. The replies for mental activity, physical activity, frustration and success were mostly encouraging.

Participants' Comments. The participants mostly seemed to like the interface. Some of them expressed comments that referred to their expectations after they already had taken part in the previous paper-based study regarding GPDUI.

“Works very well and (as far as I can remember) exactly matches the expectations I had after taking part in your earlier study.”

“This is fun!”

They also considered up to what kind of level of complexity the application would scale and regarded it as being sufficient for easy and medium level tasks.

“Useful for formulating specific (non-vague) information needs. Limitations would be to small to medium sized problems, and to clearly specified domains.”

Changes Needed. After the pilot study, a range of minor changes were undertaken. However, there were no drastic changes either in the software or the study set-up.

Tasks and Phrasing. Some task descriptions and some of the questions in the questionnaire were changed in order to make them clearer to understand for the participants.

GPDUI. Some changes that were undertaken regarded the duration and timeframe concepts of the specification language used.

To help the users distinguish between the absolute and the repetitive timeframe, an additional visual aid was introduced. This visual aid had previously only been used in the symbols on the tabs for the absolute timeframe and the repetitive timeframe in the timeframe edit window. For the full study, the same symbols were used for the presentation of the two concepts on the workspace. Figure 6-7 details the use of a hard line for the representation of continuous time flow in the absolute timeframe. In contrast to this Figure 6-8 shows a dashed line to indicate the discontinuous nature of the repetitive timeframe.

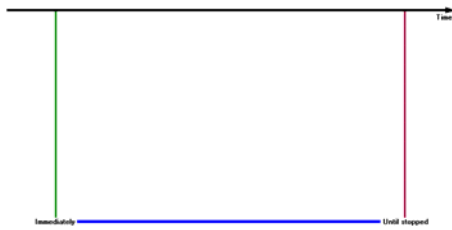


Figure 6-7: Absolute timeframe.

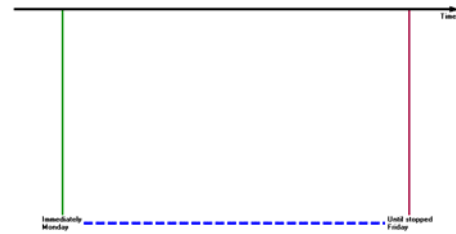


Figure 6-8: Repetitive timeframe.

Interface Layout. Regarding the layout of the interface a couple of minor issues were changed regarding dialogue windows, buttons, labelling, symbols, tooltips, component bar and workspace.

Interaction Process. A navigational help was added in order to reduce the risk of mode confusion while navigating within the profile expression on the workspace. This navigational help displays the “Current Component”, which is that part of the profile expression that has currently been given focus by the user (Section 5.1.3).

Moreover, the one-click approach that could send the workspace component to the component bar was replaced by a drag and drop approach. The aim of this was to help the user to distinguish between navigation methods within the profile expression (clicking), and transfer of components between the two screen elements workspace and component bar (new drag and drop mechanism).

6.3 Quantitative Findings

The quantitative findings of the full study covered the accuracy of participants' profile specifications.

6.3.1 Accuracy

After the completion of the study, saved files from the evaluation phase for each session were analysed for the correctness of the given solution.

Several variables were taken into account: The overall success rate was looked at; the success rates were analysed for each task level and for the individual tasks. Additionally, the influence of several categorical variables on the success rate was examined.

Overall Success Rate. Overall, the participants solved 341 out of 396 tasks correctly; a total of ~86% correctly solved tasks.

Task Levels. The results for the four task levels were analysed in three ways: regarding the results for each level for condition 1 and condition 2, and the total of specification and interpretation tasks for each level (both conditions together). Figure 6-9 and Figure 6-10 detail the percentages of these results, and also give the accumulated sum of correct tasks and the accumulated number of total tasks in the respective category.

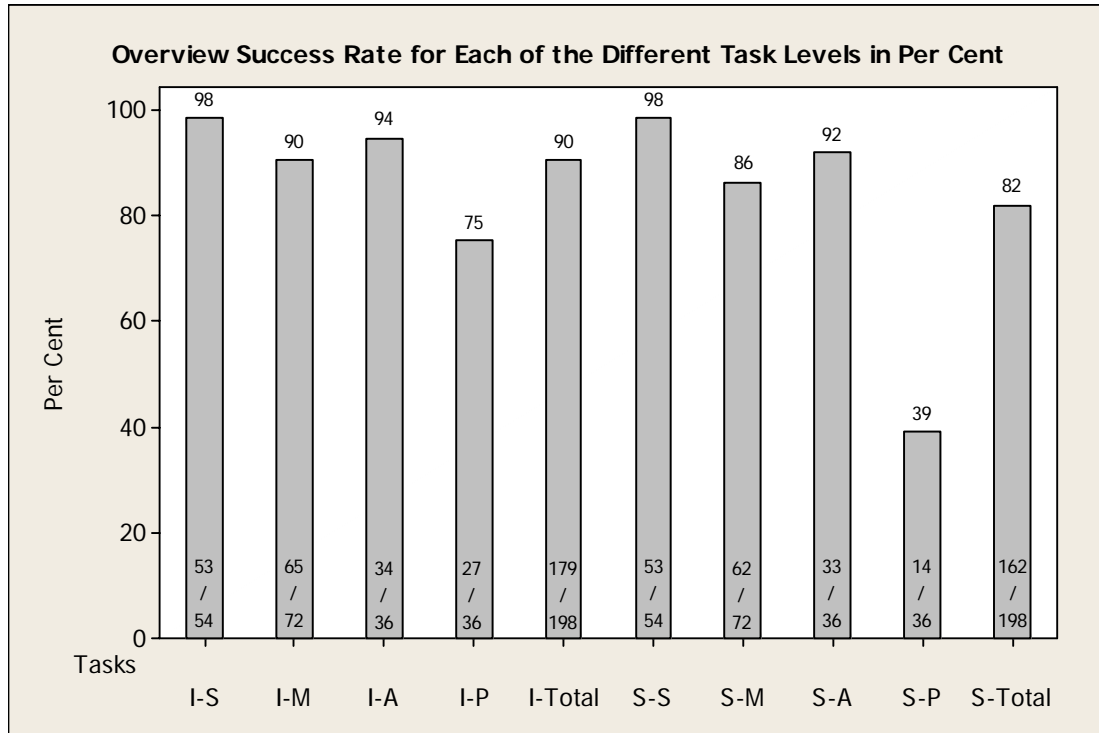


Figure 6-9: Overview of the success rates for each task level in percent and absolute, conditions: I=Interpretation, S=Specification, levels: S=Simple, M=Medium, A=Advanced, P=Professional.

For the simplest level, the success rates are identical between the two conditions as can be seen when comparing I-S and S-S in Figure 6-9. However, with increasing task complexity the success rate changes more and more between interpretation and specification task (cf. I-M, I-A, I-P to S-M, S-A, S-P). Overall, with ~90% (I-Total) the interpretation tasks show better results than the specification tasks with ~82% (S-Total). The biggest gap can be found for the professional tasks. Here, the result for the interpretation is much higher.

The overall tendency (shown in Figure 6-10) is that with increasing task complexity the success rate drops. This is more pronounced in the data detailing the success rates for specification than in the interpretation data (Figure 6-9).

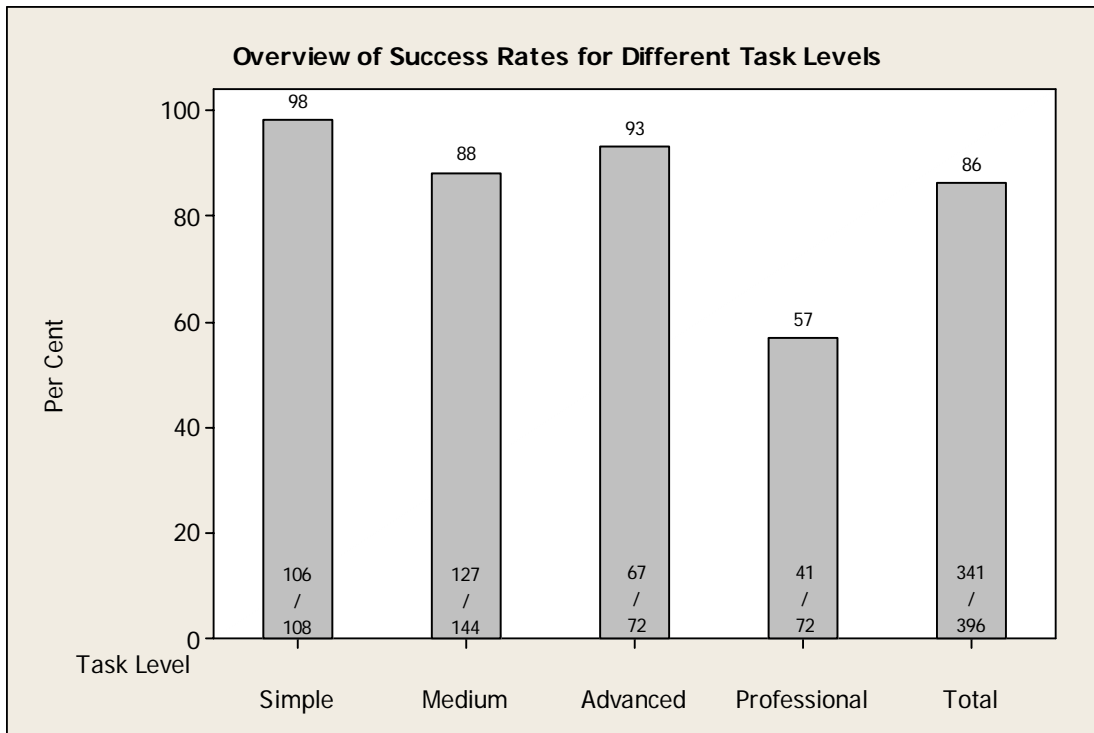


Figure 6-10: Overview of the overall success rates for each task level in percent and absolute.

However, participants were more successful in solving tasks belonging to the level of advanced tasks than those from the level of medium tasks. For more detailed observations on this, please refer to the analysis of the success rates of the individual tasks in the following subsection.

Individual Tasks. Figure 6-11 and Figure 6-12 show the accumulated success rates for each individual task for both interpretation and specification tasks. There were 18 participants; thus for each task the maximum number of correct solutions that could be reached was 18.

The success rates for several tasks in the group of medium and professional tasks stand out from the general tendency in these groups. Task S6 in the medium group shows unusually low success rates. Task S10, S11 and I10 show low success rates in comparison to the overall success rates of other groups, with task S11 being much lower than the other two success rates. If comparing the participants' results relative

to the results within each task level, it stands out that task S6 and S11 are the tasks that deviate the most from the results of other tasks in their individual groups. These are the only specification tasks that involve the *no-occurrence* concept.

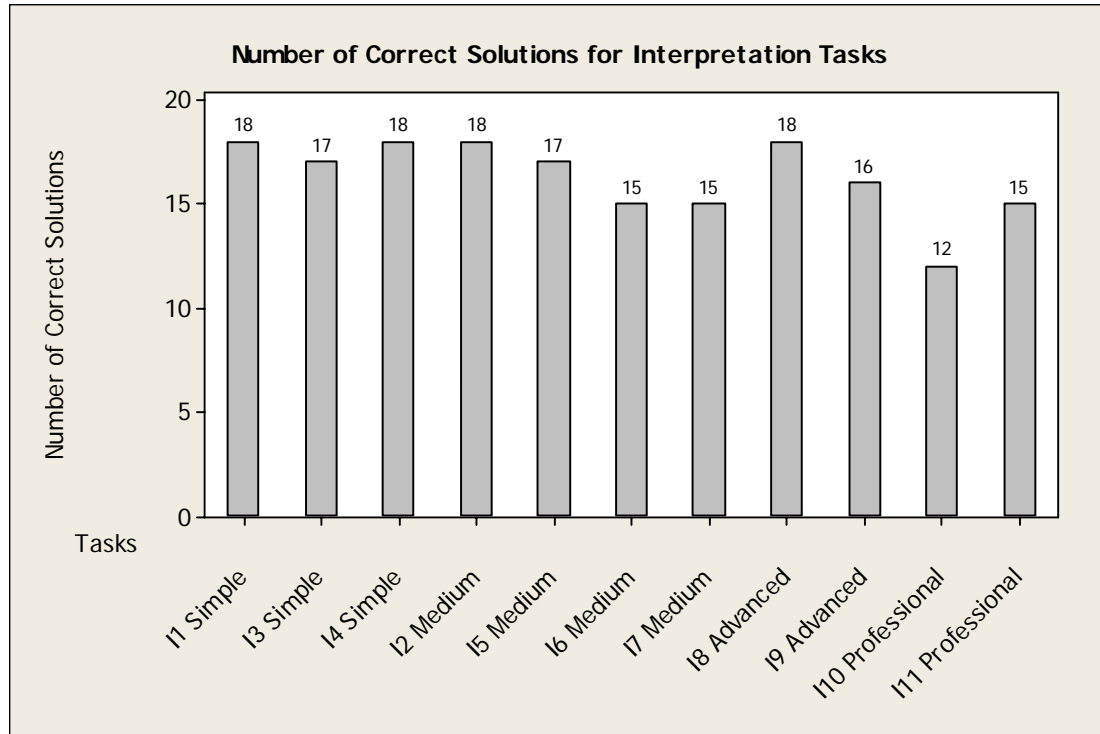


Figure 6-11: Overview of success rates for individual interpretation tasks.

All the other tasks show a success rate that seems relatively consistent with the other success rates within their individual groups.

The very low success rate of S6 causes the entire level of medium tasks to have a lower success rate than the level of advanced tasks. This explains the inconsistent result regarding the task-complexity-success rate behaviour—meaning erroneous specifications of the *no-occurrence* operator negatively influenced the entire specification of profiles at the medium level.

Influence of Categorical Variables on the Success Rate. The influence of several categorical variables on the success rate was analysed. Overall the highest in-

fluence was to be found in the mother tongue of the participants and the presentation order of the tasks.

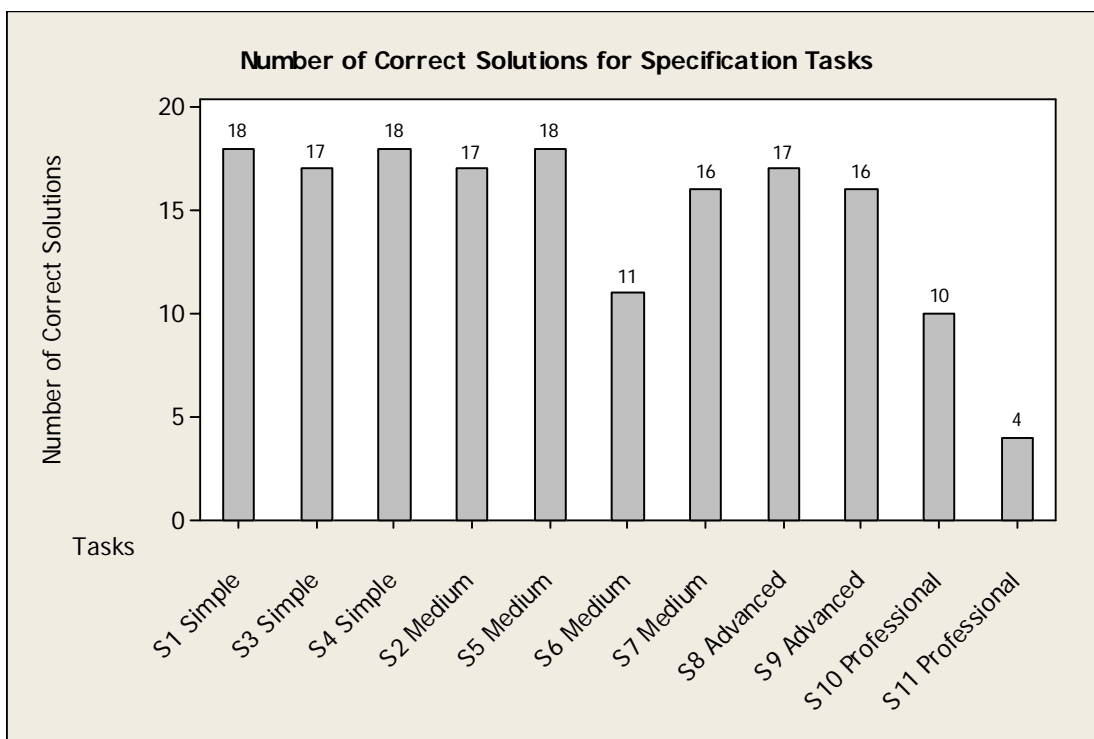


Figure 6-12: Overview of success rates for individual specification tasks.

Task Order. As can be seen in Figure 6-13, overall, with ~90% (IS, Total) the success rates were higher when the interpretation condition was the first condition that was presented to the participants. For the other order the total success rate with ~82% (SI, Total) lay lower. This trend also holds for the four separate profile levels.

Native Speaker. Three out of 18 participants were from a non-English speaking background. With ~89% the overall success rates were higher for native English speakers than they were for non-native speakers who solved ~73% of the tasks correctly. This was similarly relevant for interpretation tasks and specification tasks.

Profession. Overall, the results regarding the success rates were similar for the participants without a computer science background in comparison to those with a computer science background. However, the more complex tasks became, the relatively

better the computer scientists were than the non-computer scientists. The simpler tasks were solved with more success by the non-computer scientists. This holds for both specification tasks and interpretation tasks.

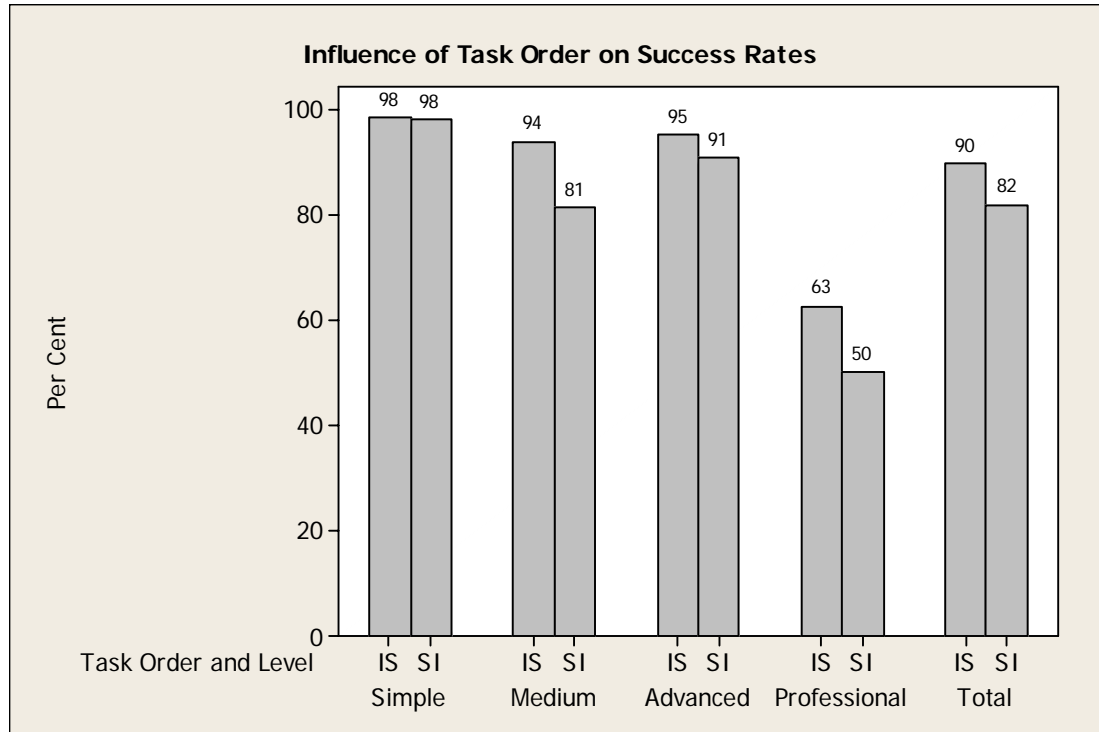


Figure 6-13: Overview of the influence of the task order on success rates, IS=first interpretation then specification, SI=first specification then interpretation.

Duration. Figure 6-14 details the influence of the study duration on the success rates for each task level and in total. For each profile level on the x-axis, it is shown how many per cent of correct solutions were accomplished in correlation to the study duration.

After observing the study, it was obvious to the observer that there was one extreme outlier. One participant took much longer than all the others due to a high interest in the study, a high motivation to complete the tasks successfully and the readiness to give a large amount of feedback. This individual showed a success rate of 100%.

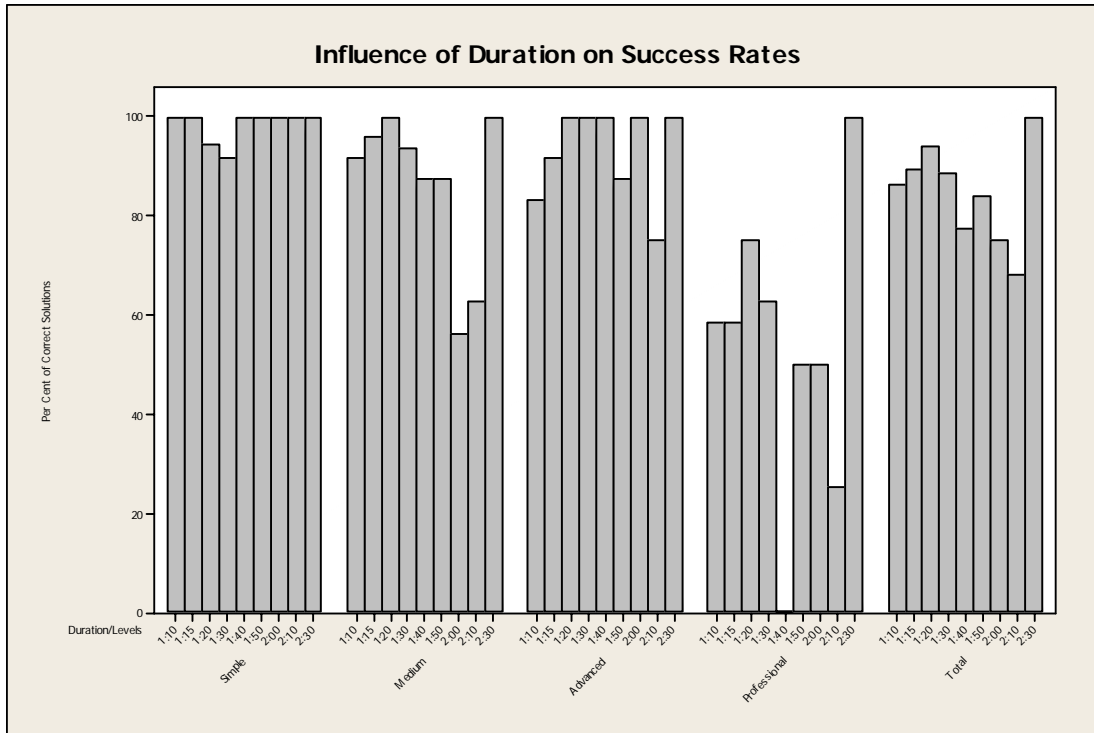


Figure 6-14: Overview of the influence of the study duration on success rate.

Ignoring the long duration this individual took, at first glance it appears that the success rate of the study is influenced by the duration. It appears that there is an initial improvement of the success rate with a longer duration of the study. Nevertheless, after a certain time this changes into results that indicate that the longer the study took the lower the success rate would be. However, a closer look at the relationships between variables shows that there are different variables that cause this effect. The three individuals whose native language was not English took longer for the study. It was also observed that their general success rates were less high than those of the English native speakers. Thus, it is not directly an influence of the study duration that was observed but rather the effect of the native language. Obviously, it takes longer to process tasks you are given in a second language than in your native language. This suggests that the duration and success rates do not correlate.

6.4 Qualitative Findings

The qualitative findings of the full study cover Likert responses, observations and participant comments.

6.4.1 Likert Responses

General Distribution. The general distribution of several parameters was analysed: intuitiveness, ease, satisfaction, mental and physical activity, frustration and subjective success. All of them used a 5-point Likert scale, with five representing the highest value (for example very easy).

Intuitiveness. As shown in Table 6.1, overall, the subjects judged the interface to be relatively intuitive. 61.11% rated it on the positive end of the scale, whereas only 11.11% rated it on the negative end.

Table 6.1: Results for intuitiveness, (N=18).

Intuitiveness	Count	Percent
2	2	11.11%
3	5	27.78%
4	6	33.33%
5	5	27.78%

Ease. Most participants considered the interface as easy to use. 72.22% lay on the positive end of the scale, no one on the negative end. Details are given in Table 6.2.

Table 6.2: Results for ease, (N=18).

Ease	Count	Percent
3	5	27.78%
4	7	38.89%
5	6	33.33%

Satisfaction. Table 6.3 shows the satisfaction of the participants in using the interface. 83.34% of the participants were satisfied or very satisfied with using the interface.

Table 6.3: Results for satisfaction, (N=18).

Satisfaction	Count	Percent
3	3	16.67%
4	12	66.67%
5	3	16.67%

Mental Activity. The participants judged how much mental activity they had to put into using the interface. Their judgement was clustered around the middle of the scale (see Table 6.4). So people had to think about what they were doing though not extremely so.

Table 6.4: Mental activity, (N=18).

Mental Activity	Count	Percent
2	8	44.44%
3	7	38.89%
4	3	16.67%

Physical Activity. Information was gathered about the amount of physical activity the subjects had to put into interacting with the software. The results are shown in Table 6.5. 55.55% judged it to be less, whereas 22.22% judged it to be more work.

Table 6.5: Physical activity, (N=18).

Physical Activity	Count	Percent
1	4	22.22%
2	6	33.33%
3	4	22.22%
4	4	22.22%

Frustration. 83.33% of the participants rated using the interface as positive. For details refer to Table 6.6.

Subjective Success. Table 6.7 shows the participants' subjective ratings of their individual success in solving the tasks correctly. 88.89% were in the positive range of the scale. Half of the participants chose the maximum of the scale that said that they felt very successful in completing the tasks. No one thought they were unsuccessful.

Table 6.6: Frustration, (N=18).

Frustration	Count	Percent
1	7	38.89%
2	8	44.44%
3	1	5.56%
4	2	11.11%

Table 6.7: Subjective success, (N=18).

Success	Count	Percent
3	2	11.11%
4	7	38.89%
5	9	50.00%

Influence of Categorical Variables on Likert Responses. The influence of the task order, mother tongue, profession and duration on the Likert responses was analysed.

Task Order. The task order had a slight influence on several quantitative variables: Participants that were first presented with the interpretation tasks found using the interface slightly more intuitive than those that worked with the other order.

However, they found using the interface slightly easier if presented with the specification tasks first.

They judged the physical activity they had to perform as slightly higher if they were presented with the interpretation tasks first.

Participants that were presented with the specification tasks first felt to be more successful in the correct completion of tasks.

Native Speaker. No major influence on any of the variables could be observed.

Profession. Those participants that came from a non-computer science background found the interface to be more intuitive and more satisfying to use than those from a computer science background. In contrast to this, the computer scientists found it slightly easier to use than the non-computer scientists.

The non-computer scientists found that somewhat more physical activity is involved whereas the computer scientists felt that slightly more mental activity is required in order to use the interface.

The computer scientists were slightly more frustrated than the non-computer scientists. However, they were more likely to think that they were successful in solving the tasks correctly.

6.4.2 Observations

The observer studied the reactions of the subjects, kept note of the problems that occurred while they were using the software and kept track of the utterances they made. The first impression the observer had after each session was reassessed by analysing the recorded session. The areas that were perceived to have attracted the most problems were navigational issues, conceptual timeframe questions, confusion regarding the *repetition* operator, as well as issues involved with drag-and-drop mechanisms.

The observations that were made were categorised; in each category some sample observations are given:

Condition Boxes. A few participants had minor problems with some condition-related interaction mechanisms and language concepts regarding defaults, comparison operators and editing of conditions.

Operators. The greatest problem some participants had was to distinguish between the *repetition* operator and the *sequence* operator. All other issues were experienced by a few participants only or even by just one individual.

Repetition and Sequence. A number of participants had to learn the distinction between the *repetition* and *sequence* concepts. Two subjects found the icon used to denote the *repetition* operator unintuitive.

Sequence versus All. There were a few participants that had to learn the distinction between the *sequence* and *all* operator.

Some versus All. One participant had to explicitly learn the difference between *some* and *all* operators, though they remarked that the ideas were “really cool”.

Simple Negation versus Composite Negation. Two participants recognised that there is a difference between the *simple negation* and the *composite negation* (*no-occurrence*) and made a mental effort in order to fully understand the nature of this difference. Two other participants had to explicitly learn the meaning of the *no-occurrence* concept.

All and Duration. It took two participants longer than the others to understand that the conjunction *all* can be used as a Boolean-time operator and not only as a purely logical concept.

Time-related Concepts. Issues that involved time-related concepts concerned the timeframe of profiles and the duration parameter of operators. With the former being one of the areas that proved to be most problematic, the timeframe of profiles yielded the largest number of relevant observations.

A number of participants found it slightly challenging to learn to distinguish between the *repetitive* and *absolute* timeframes. Several problems occurred. Some participants were confused about whether they were allowed to use both tabs for the definition of the timeframe (see Figure 6-15).

Another problem was whether the blue lines in the workspace (see Figure 6-7 and Figure 6-8) represented different kinds of interpretation modes for the time annotations. None of the subjects seemed to find the hard line—used to represent continuous time—and the dashed line—used to represent a discontinuous time flow—to be intuitive in their meaning. At best the participants ignored those lines and at worst the participants were confused regarding their meaning and inquired what they meant.

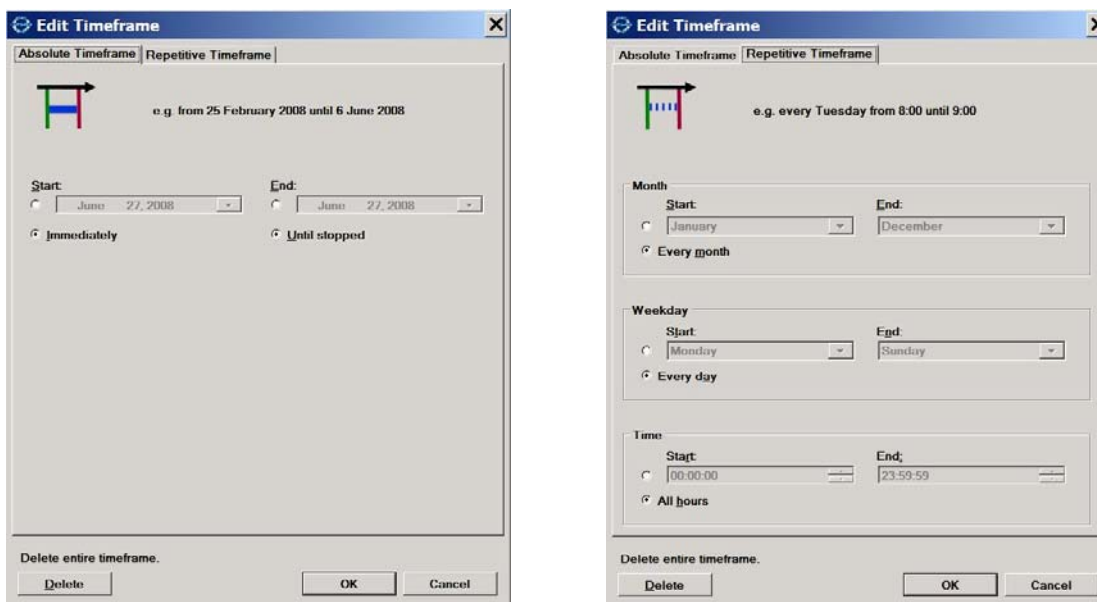


Figure 6-15: Overview of the timeframe edit window.

Left: Absolute timeframe tab. Right: Repetitive timeframe tab.

With two participants, discussions about the concept came up. One ascribed their problems with the concept to the fact that the repetitive timeframe tab did not actually show the absolute time chosen. The other participant attributed the problems to the

presentation of the timeframe concept in the tutorial. It showed the two concepts being separate from each other rather than one being an extension of the other.

Duration versus Timeframe. The observer noticed that a couple of participants had to learn the difference between the duration and the timeframe.

Interaction Mechanisms and Layout. The last group of problems that were observed covers several interaction mechanisms, as well as several interface structure and layout issues. The most pressing of these problems concerned navigation within individual profiles.

Navigation. The greatest problem the participants appeared to have had concerned the navigation mechanisms within the profile. They sometimes were confused at what level of the profile they were. In particular this manifested itself when they were dealing with the two unary operators (*no-occurrence* and *repetition*). They were confused at what times it was possible to add a duration to the operator and when it was not possible.

Copy and Paste. Most people did not try to use copy and paste. However, those who did noticed some problems with it. One did not understand how to use it and just succeeded to use it by coincidence. The other individual suggested that it would be better if one could paste directly into where a component should go rather than to the component bar first.

Drag and Drop. Several participants wanted to be able to drag a subcomponent to the component bar. One participant wanted to be able to use drag and drop with a snap mechanism in order to move components on the workspace. One participant remarked that it would be good to actually see an object while it is being dragged.

Workspace Use. Some participants would have liked to be able to have more than one component at a time within the workspace.

Component Bar. One participant suggested to have the same content in the component bar for each tab, that is, for each profile.

Another participant was interested in being presented with a label describing the contents of each component in the component bar rather than having only a tooltip when hovering over the component.

Grouping. There was one subject who was not aware of the fact that the dark grey grouping box also embodies the functionality of the normal box and thereby can be connected by operators (that is, the relative position of the boxes to each other) to other boxes.

6.4.3 Participants' Comments

At the end of their session each participant gave comments in the questionnaire that summarised the overall impression they had with using the interface. Most participants gave positive comments. No one remarked anything negative. Some added some constructive critique to their general positive judgement.

Affirmative Comments. In summary, the subjects stated that the interface was simple, intuitive and easy to use. They found it clearly labelled and with icons that represent their functionality. They commented that the subject matter dealt within the program is relatively complex and the amount of effort they had to put into learning the interface was justified.

“It was a pretty good interface. Everything was clearly labelled and had tool tips so that I could find the program option I was looking for easily.”

“It’s good and it works”

“Generally its very intuitive—simplistic. Everything I wanted to do was almost achievable without having error...”

“I found the interface very easy to use.”

“Quite easy to use and learn.”

“Once learnt the interface is pretty simple to use. I would be interested to see how well it would scale (that is, how well it would perform on really complex conditions.)”

“Relatively easy to follow”

“Good easy to use interface, all is there just takes a little getting used to.”

“Icons were representative of their function which is very helpful”

“Interface works well and is not hard to follow.”

“Minimalistic which I like. It’s a bit compsci but it deals with an advanced subject so this is fine.”

“Once I became familiar with the functions it was reasonably easy to use.”

“Comprehensive results could be obtained.”

“I was quit tired so concentration on my part was quite average. But it wasn’t overly difficult or anything,”

Critical Comments. Critical comments that occurred repeatedly concerned an issue with the navigation in a profile, and some minor problems with copy and paste as well as problems with the complexity of the given tasks. Some other issues commented upon were only mentioned once.

Navigation. Quite a number of participants remarked that they had problems figuring out in what level of the profile they were in. Other subjects had problems that could be backtracked to this problem.

“Generally easy to use and understand, only had one difficulty during editing which is noted below. Had a problem at times figuring out what level I was editing, though I did mostly ignore the current component box.”

“Quite effective with some small issues to navigation, need a better feel for the programme”

“Some clicks didn’t register... also probably need more feedback of the currently selected box.”

“Some slight issues occurred when trying to set a time value to a single box”

“Seemed very simple. The order of actions needed to get a task done wasn’t very clear. That is making a sequence or timeframe. Turns out the order doesn’t matter but it meant I needed to choose a flow rather than have it mapped out. Once I figured out that I could do things in any order and that timeframe and duration were different it became a lot easier to use.”

Copy & Paste. The mechanism used for copy and paste was commented upon by two participants. One of them had further commented in the questionnaire that they had obtained the correct result using the program’s copy and paste mechanism but said this was merely a coincidence and they did not really fully understand how it worked.

“Just be able to select a different box & copy and paste easier perhaps?”

“Regarding frustration: only the time when I couldn’t find out how to cut/paste part of a repetition.”

Task Complexity. One issue commented upon by a few participants was the complexity of the tasks to be solved. They said that while the interface was easy to use, they had sometimes problems working out what they were supposed to do in the first place.

“Using the interface was easy but deciding if it’s [the solution to the tasks] right or wrong was difficult.”

“I didn’t have trouble with interface, more trying to understand what the questions wanted me to do.”

“At one point in task 11 I wasn’t frustrated but unsure of where the time component should’ve gone. But that was more the task not the interface.”

Other. Other critical thoughts regarded topics such as use of the component bar, button symbols and drag and drop mechanisms.

One participant made note of the fact that they had not had reason to use the component bar.

“Never used the component bar. Although never had chance/reason to use it.”

Another individual commented the fact that they did not like the slightly changing icons on the buttons that were used in order to clarify their slightly changing context. (While the general action of the button stays the same, they change icons depending on the state the current profile is in. Depending on the state of the profile, different functionalities are possible as described in Section 5.2.)

“It is rather easy to use. Found the number of buttons confusing as a lot of them did similar functions, sometimes changing functions (such as All changing to Add Box).”

One subject would have liked more possibilities to use drag and drop in order to place individual profile components.

“For some questions I found it difficult to place the blocks. Drag and drop would have been good here.”

6.5 Discussion

This section presents and discusses major issues that arose out of the study.

The study examined one major question, namely whether the interface is an effective means in order to produce correct profiles for alerting. This question can not be answered quite independently of the question whether the participants of this study consider GPDFL an effective language in order to specify correct profiles, as the interface is using this language.

The study results show a positive outcome regarding the research goals. These goals were to determine:

- How accurately users can specify profiles using GPDFL-UI
- How accurately users can interpret profiles shown with GPDFL-UI
- Users’ subjective experience of GPDFL-UI for profile specification
- Users’ subjective experience of GPDFL-UI for profile interpretation

The overall observations that were made were positive, and the comments the participants gave about the GPDL language and interface of the program were very appreciative. The participants were happy to use GPDL-UI in order to specify profiles for alerting systems and to interpret them.

Also the Likert scale responses that were gathered were supportive. The program was judged as intuitive ($\sim 3.8/5$), easy to use ($\sim 4.1/5$) and satisfying to use ($4/5$). The frustration level was kept within a limit (frustration $\sim 1.9/5$) and the participants overall felt very successful in correctly solving the given tasks (subjective success $\sim 4.4/5$). In order to solve the tasks, the effort the subjects had to make was well balanced between mental activity ($\sim 2.7/5$) and physical activity ($\sim 2.4/5$). Thus, the evaluation of the participants' subjective impression of the interface yields to a positive outcome.

This research outcome is confirmed by the success rates the participants gained when completing the given tasks. The overall success rate with $\sim 86\%$ is high. The interpretation tasks especially showed a good result with $\sim 90\%$. The specification tasks with 82% were still reasonably good. Generally, the more complex the task was the lower the success rate was. This is to be expected and results are not extraordinarily abnormal.

6.5.1 Presentation of Timeframe Concept

From the problems the participants had with understanding the concept of the timeframe and from the discussions about this concept, it could be deduced that participants misunderstood the concept. From the tutorial, they assumed the absolute timeframe and the repetitive timeframe would be two different concepts. However, in the definition window it was treated as one concept that can be extended. This has to be presented in a consistent manner in order to foster a quicker learning curve.

Therefore, the hard blue line that represented the absolute timeframe and the dashed blue line that denoted the repetitive time frame were removed, as the use of these two different symbols increased the misconception about the timeframe concept being split into two concepts.

Moreover, due to the explanations that had to be given during the training phase it could be deduced that users can be better supported by giving more detailed annotations for the workspace presentation of the timeframe.

6.5.2 Navigation

Several observations, comments and results showed that there is room for improvement for the navigation process in the program.

No-occurrence and Repetition. From participants' comments, observations and results regarding the success rates, it could be seen that a number of subjects had problems with the *no-occurrence* and the *repetition* and the use of their duration parameter. This seems to be caused by the fact that due to their unary nature and the way they are presented in the workspace, it is difficult to differentiate on what level the user is navigating. The condition level and the operator level look very similar and thereby navigation is difficult for an individual that is not aware of their own actions. Thus, the presentation of the *no-occurrence* versus the presentation of the condition was changed to a greater contrast of the active and inactive negation cross out line. The newly added deletion-cross also poses a means for distinguishing the levels of the nested expression for the *repetition* operator and for the *no-occurrence* operator: A deletion cross is only visible if a box has been given focus.

No-occurrence. It became obvious that the participants seem to have issues with the *no-occurrence* concept or more in particular with the navigation within this concept. This became clear from the relatively low success rates for S6 and S11 (the only specification tasks containing the *no-occurrence* concept).

6.5.3 Learning GPD and the Interface

While testing the interface, insights were gathered about the learning process of the interface and underlying language.

Task Complexity. The program was testing profile specifications that were divided into four complexity levels. Inherent in the nature of the study was, that the more

complex the profiles became, which the subjects had to specify, the more complex the task descriptions became. From observation and from some of the comments the participants gave it could be seen that they struggled to understand the scenarios given in natural language. This problem can not be out factored in the overall result of the study. It especially posed a problem for one group of participants as can be seen in the following subsection.

Native Speaker. It was observed that it is beneficial to the success rate if the participants were native speakers. This can be expected as the concepts are explained using the English language and the tasks that have to be solved also were in English.

Task Order and Success Rate. It was realised that generally the success rate was higher when the subjects first had to complete interpretation tasks and afterwards specification tasks. This seems to indicate that a prolonged learning period increases the understanding of GPDUI and the interface. Additionally, with this order (interpretation followed by specification), there is a separation in two steps during the initial learning process—which does not exist if starting to learn with completing specification tasks. First the subjects learn the GPDUI language while completing interpretation tasks. It is only when they complete the subsequent specification tasks that they are confronted with the interface. This seems to be beneficial to the learning process.

Task Order and User Experience. For the presentation order interpretation followed by specification, the majority of participants considered the program to be more intuitive but the physical activity they had to put into using the interface seemed to be marginally higher.

For the other order, specification followed by interpretation, they found the program slightly easier to use and felt to be more successful in the correct completion of tasks.

6.5.4 Temporal versus Logical *Conjunction*

In the pilot study it was observed that two participants had problems with the concept of the *all* operator. It can be used with and without duration. This appeared to confuse

the participants. The reason for this confusion seems to lay in the fact that when the concept is used without duration arrow, it has slightly different semantics as when it is used with duration arrow. In the former case it describes the logical combination of all the conditions it combines, that is, a Boolean operator. The latter denotes the *conjunction* as a Boolean-time operator.

This also explains why in the paper-based study the specification task using the *all* operator in conjunction with the duration parameter was prone to errors.

6.5.5 Layout and Interaction Mechanisms

Minor problems arose regarding some layout issues and interaction mechanisms.

Drag and Drop. While the existing drag and drop mechanisms were improved, it was decided against introducing a drag and drop mechanism in combination with a snap mechanism for manoeuvring components on the workspace. This had been an initial design decision in the development of the interface. If such a mechanism would have been introduced, it would have made the support of a semi-automated profile hierarchy building process impossible. If users could freely move components around on the workspace, this would lead to a permanent change in semantics of the profile. Moreover, due to the representation of operators in regards to the location of the components, the program would need to constantly adjust the size and orientation of the components. This would lead to the user experiencing visual disturbances; users would have trouble locating parts of components that have been resized and due to that moved their location on the workspace.

Workspace Use. It was decided to remain with one component on the workspace at a time. The reason for this is based on the operator representation by location. If there were several components on the workspace at the same time, it would be difficult to differentiate which components form one bigger component and which are independent of each other.

Component Bar. It was decided to keep the component bar as there needs to be some space where users can put components they currently have no need for but want to keep. This follows the same reasoning as was given above for the workspace use.

6.5.6 Target Group

Non-computer scientists found the interface more intuitive and more satisfying to use. However, they commented that more physical activity was required. They were less frustrated. In contrast to this, computer scientists found the program easier to use and felt that slightly more mental activity was required. They considered themselves to be more successful.

From this, it could be learned that non-computer scientists and computer scientists seemed to take a different route of how to tackle the problems given for specification. Non-computer scientists appeared to take a more active approach using trial and error, whereas computer scientists appeared to analyse problems further before taking action.

Despite their training in logic, the computer scientists did not score significantly better success rates than the non-computer scientists. While computer scientists were better at the complex tasks, non-computer scientists outperformed the computer scientists at the simpler tasks.

Overall, GPDUI and the GPD language are more suitable for non-computer scientists.

6.6 Summary

This chapter presented a user study of GPDUI: Section 6.1 and Section 6.2 set the focus of the study and described its experimental design. The results were presented in Section 6.3 and Section 6.4; a concluding discussion was led in Section 6.5.

The paper-based GPD study in Chapter 4 and the GPDUI study in this chapter followed the same experimental design and used the same tasks. Comparing both

studies leads to the finding that the GPDL-UI study followed the GPDL study in its positive outcome. Subjective responses were equally appreciative and measurements for success rates were similarly high. This validates the results of the paper-based GPDL study and confirms that GPDL and GPDL-UI merit further consideration. This targets the support of collaboration in the specification process of profiles and is presented in the following chapter.

7 Collaborative Profiles and their Specification

This chapter introduces a collaborative alerting model, allowing for the specification of profiles by several parties. The chapter starts by providing illustrative examples for the requirement and usefulness of collaborative profiles (Section 7.1); followed by Section 7.2 that provides an overview of related research.

Section 7.3 then introduces the concept of collaborative alerting, the mechanisms underlying the profile specification and the integration of collaborative concepts into existing alerting system architectures. CGPDL, the Collaborative Graphical Profile Definition Language, is the content of Section 7.4; it is an extension of GPDFL.

Section 7.5 discusses CoastEd (Collaborative Alerting System Editor), an editor for collaborative profiles using CGPDL. This section gives an overview of the interface of CoastEd, describes how the CGPDL concepts can be accessed in CoastEd and concludes by outlining the interaction processes to collaboratively define profiles.

7.1 Motivating Examples

This section presents some examples to motivate the requirement of collaborative profiles in alerting systems. For further details about this requirement and an extended analysis, refer to Chapter 2. The following subsection contains examples from the health care area. Section 7.1.2 lists other application areas that can benefit from collaborative profiles.

7.1.1 Examples from Health Care

The following examples motivate the requirement of collaborative profiles in health care.

Example 34 *A nurse wants to be alerted if patient Smith's pulse is too high. She does not exactly know, though, what value would be too high for this individual patient. Therefore she leaves this information unspecified within the profile—the exact pulse value is left vague. This part of the profile has to be provided by someone else.*

The nurse starts to specify the profile by incorporating the knowledge and information she has. Then she gives the doctor the job of completing missing information within the profile.

The doctor has the knowledge that for the patient Smith, 90 BPM is the threshold for a pulse that is too high. Accordingly, he refines Mr. Smith's profile.

Figure 7-1 shows the described situation, the collaborative profile specification (nurse and doctor) and the different information all participants supply.

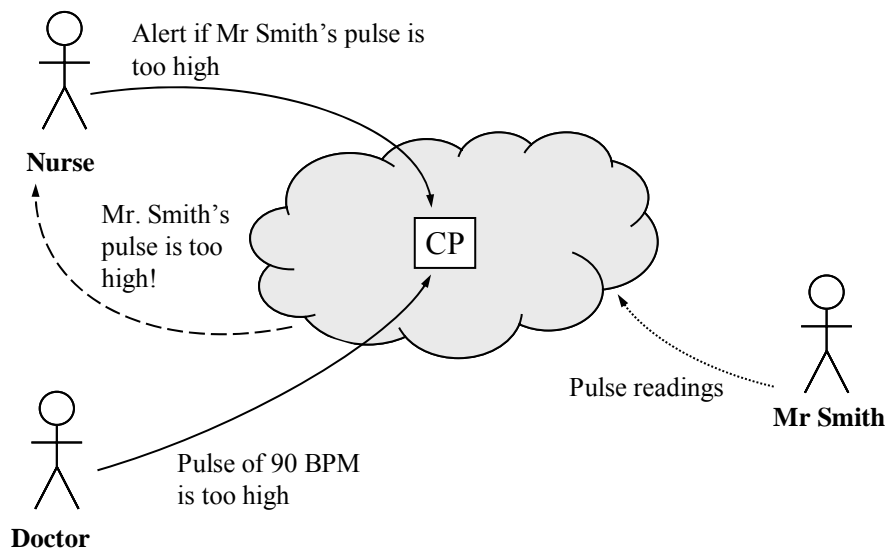


Figure 7-1: Collaborative profile definition described in Example 34.

Example 35 *A medical practitioner wants to be alerted if his patient's pulse is too high within one hour after he has received an unusually high dosage of his medication.*

The medical practitioner does not know what pulse would be too high for that particular patient. Moreover, he does not know what dosage of medication would be unusually high as normally the dosage is set by someone else.

The medical practitioner starts the specification process of the profile, but the information about the exact pulse value and the exact dosage amount is left blank. Two different doctors need to complete these values.

Doctor 1 has to fill in the value for the pulse based on patient's medical history.

Doctor 2 has to provide the dosage amount.

Independently of each other, both doctors refine the profile that is provided by the medical practitioner by filling in the missing information. This way, the collaborative profile, which exploits the knowledge of all participants, is created.

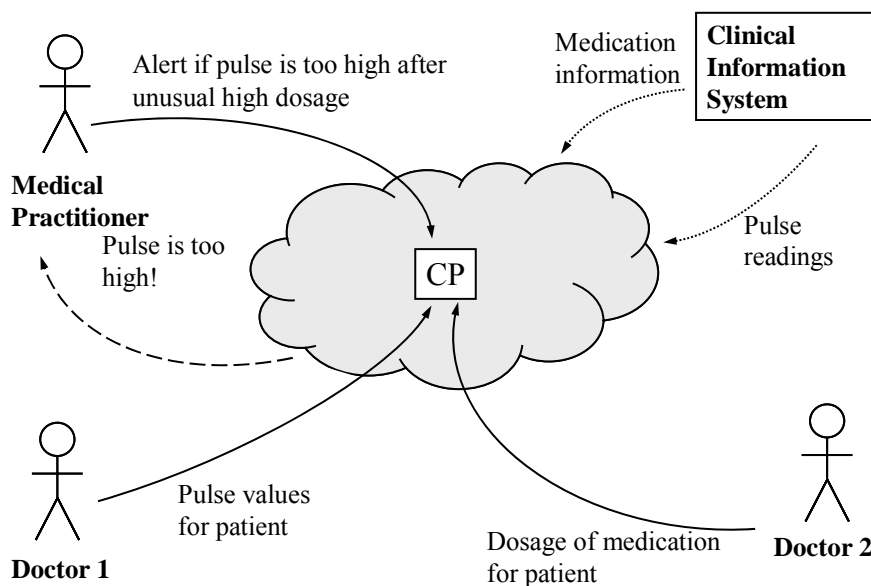


Figure 7-2: Collaborative profile definition described in Example 35.

Figure 7-2 shows an overview of the profile specification process, its participants and external providers of events, that is, information.

Example 36 *A doctor prescribes some medication to a patient who is known to forget to take prescribed medicine. The doctor specifies a profile with all the required information: The patient should be alerted in the morning that it is time to take the medication. However, the doctor is unaware of the exact time that suits this particular patient. Furthermore, the patient might want to change this time, for example if he is on holiday (the medication does not need to be taken at exactly the same time each day).*

The patient needs to complete the profile from the doctor as only the patient knows details about his lifestyle.

Figure 7-3 gives an overview of this situation and the different kinds of information involved.

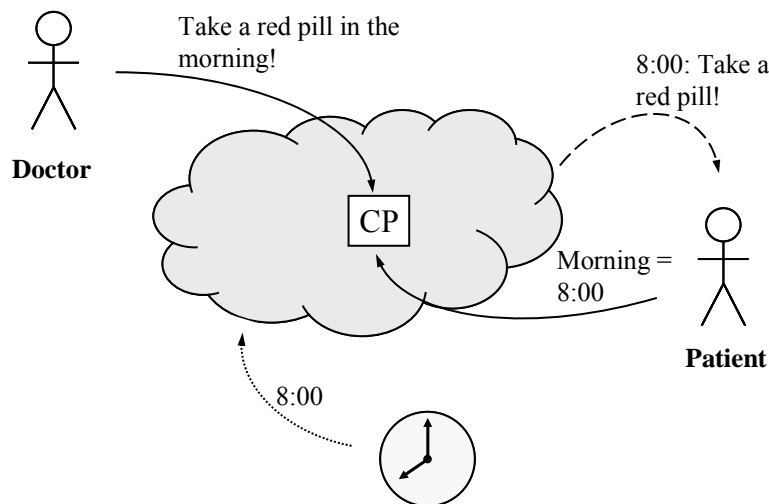


Figure 7-3: Collaborative profile definition described in Example 36.

Example 37 *Let us assume an automatic evaluation of a patient's blood values, which are taken at a laboratory. The evaluation looks at an inflammation marker, ESR, which has to be considered for the treatment of inflammatory bowel disease.*

A nurse wants to be alerted when their patient's ESR is too high so they can adjust a certain treatment. The nurse starts setting up a profile. The doctor has the knowledge about potential values for the inflammation marker. Generally, it is too high if it exceeds a value of 15. However, it may be as high as 20 without being indicative for inflammatory bowel disease if the patient has a cold.

Information in this example, that is, events, is provided by the patient as well as by the laboratory. Figure 7-4 gives an overview about this situation, requiring a collaborative profile.

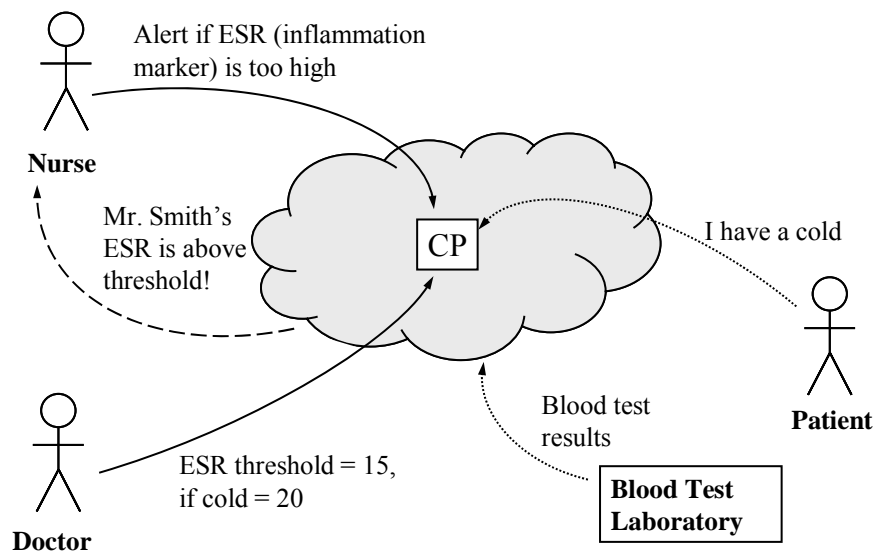


Figure 7-4: Collaborative profile definition described in Example 37.

7.1.2 Other Application Areas

Apart from the health care scenario in the previous examples, in various other application areas it is desirable and helpful to consider the interests of several users or to exploit the expert knowledge of several parties in the definition of profiles. Some of them are presented in this subsection.

Electronic Commerce. In the area of electronic commerce, online auctions and online shops can provide alerting functionalities to improve their customer service, for example described by Bittner [54] and Cilia and Buchmann [91]. Let us assume

the situation described in the following example to get an impression of the usefulness of collaborative profiles.

Example 38 *A person is interested in bargains that are available within an online store, but they do not have knowledge about the market price of various items.*

So, they specify their general interests within a profile, leaving blank all the pricing information they are unsure about. The person has various friends that have expert knowledge on different items. The person asks them to refine the original profile. In doing so, the expert knowledge of various parties as well as the general interests of the person is captured in a collaborative profile.

Facilities Management. In the area of facilities management, alerting systems can be applied to control various devices, such as heating and lighting, within buildings [92]. The following example describes a scenario requiring collaborative profiles.

Example 39 *A caretaker has configured profiles for each flat within the house they are responsible for. These profiles specify general settings for the heating within the house. It reflects the general heating policy, for example considering the temperature outside, the current time of day as well as the solar radiation.*

Obviously, each tenant prefers different temperatures within their flat, depending on their personal preferences and their daily routine. Hence, each tenant refines the profile that has been specified by the caretaker to take into account these personal circumstances.

Tourism and Recommendation. Tourism is a typical application area for alerting mechanisms [93]. Collaborative profiles can be used to improve the mechanisms of single-user profiles.

Example 40 *A group of people, such as friends or a family, wants to travel and receive recommendations on activities that they could do together. Instead of setting up*

individual profiles themselves, they get the idea to collaboratively specify their interests.

One person starts specifying their interests in a profile. This profile is set up in a way that several of such specifications can be given. Only if all specifications describe a mutual interest in an activity, this activity is considered to be relevant, that is, the activity is recommended.

This original profile is then refined by the other parties in the group. Each person specifies their own interests in the profile. Overall, this collaborative profile serves the purpose of satisfying the whole group.

7.1.3 Implication

From the examples provided in the previous subsections it can be seen that the concepts of single-user profiles (Chapter 3) do not suffice to describe profiles in the state before all required information has been collected and brought together to form the collaborative profile. Additionally, the processes required for the collaborative specification of profiles can not be undertaken with the mechanisms provided by single-user profiles and conventional alerting systems.

Within the area of alerting systems, the concept of collaboratively specifying profiles does not exist so far. The following section thus analyses related work from areas other than alerting systems to gain insights into the requirements and concepts of systems that support collaborating users and of systems that bring together different kinds of information.

7.2 Related Approaches

There are a number of concepts and systems in areas other than alerting that target and solve problems related to the requirements of collaborative profiles. Nevertheless, these approaches neither target nor solve the requirements inherent in the novel concept of collaborative profiles.

The area of CSCW targets systems that support a collaboration of users. For the introduction of collaboration to the alerting area, it is thus only logical to consider and to build on the expertise and research issues identified within the CSCW area. Section 7.2.1 serves this purpose.

At first glance, recommender systems appear to target issues that are related to collaborative profiles; they even apply a similar terminology, including concepts such as collaboration and profiles. Nevertheless, the problems targeted by recommender systems are orthogonal to the problems targeted by collaborative profiles, as analysed in Section 7.2.2. This similarly holds for user profiler systems, that are the focus of Section 7.2.3.

7.2.1 CSCW

There are quite a number of aspects typically found in CSCW research that are relevant for the design of collaborative profiles (cf. Dix et al. [94] for an overview). Addressing all of these aspects is clearly beyond the scope of this thesis. Nevertheless, the work from the CSCW area impacted on various design considerations for collaborative profiles. It also reveals various questions that can not be fully answered within this thesis but could be addressed in future work.

Definition of Collaborative Profiles. In the definition process of collaborative profiles, users define sub-profiles that, at a later stage, are resolved to one collaborative profile. Typical concerns that need to be taken into account include the following:

Time Dimension in the Definition Process. There is a typical differentiation between synchronous and asynchronous collaborative processes. Which of these two is required for defining collaborative profiles? An asynchronous model better suits the requirements of the collaborative profile definition process, as typically expert users are not available at the same time.

Location Dimension in the Definition Process. Considering the location when collaborating, collaborative profiles should allow for a remote definition process rather than restricting the participants to work co-located.

Concurrency Control. Is it necessary to integrate concurrency control mechanisms into the definition process of the collaborative profiles? This clearly seems to be necessary if different parties work on the same sub-profile. Furthermore, for this consideration it is significant whether the sub-profile of one party is actually semantically influenced by the definition of the sub-profile of another party. That is, is a sub-profile based on the information provided in other sub-profiles? Then, the definition order of sub-profiles matters and it makes a difference whether certain sub-profiles are defined or modified simultaneously. Moreover, it needs to be considered whether the definition order implies syntactic consequences as well.

Several other questions arise regarding concurrency issues: How should one deal with people forgetting to unlock a profile? Which granularity should be used for the locks of sub-profiles? Moreover, would different levels of sharing (like different levels of locks in databases) be helpful? Generally, it is important to offer a mechanism that effectively avoids deadlocks in the definition and resolution process of collaborative profiles.

Roles. If roles would be used for the realisation of concurrency control, these roles should be flexible in their use to allow for changing situations in the real world.

Sub-Profile Modification. When modifying some part of a collaborative profile, the following question arises: Should other parties be alerted if one of the other sub-profiles has been modified? Obviously, this only concerns parties that are involved in the definition of this particular collaborative profile.

Tagging of Sub-Profiles. Should it be visible what party has defined what sub-profile? If so, should there be an option to hide this information? A decision on this matter is influenced by social as well as by legal matters. Legal matters might depend on the application domain of the system and country of use.

Moreover, it might be reasonable to offer the possibility for users to comment on sub-profiles of other users. This might be necessary if definitions of sub-profiles are or might be erroneous. However, this option also involves social aspects in so far as participants with a lower position in a work hierarchy might be unlikely to comment on profile definitions of people who carry a higher work position.

Resolution of Collaborative Profiles. After the participating users have defined their sub-profiles, these sub-profiles are subsumed into one collaborative profile. Again, this task touches considerations that are inherent in CSCW research.

Social Aspects. When working with a collaborative system, one issue to be considered is social aspects of the individuals involved. This is due to the fact that collaboration involves the social interaction of several individuals. This entails dealing with hierarchical relationships as well as taking into account the individual requirements of people.

Conflict Resolution. If the resolution of several sub-profiles leads to a conflict, the system could try to resolve these conflicts. This could be done by using priorities assigned to these sub-profiles. Another option is to expand the ranges of values defined in individual sub-profiles. The combination of both approaches is also possible.

Resolution Time and Space. Regarding the resolution of collaborative profiles, one also has to consider where and when sub-profiles are going to be resolved into a collaborative profile. This can for example either happen on a centralised server or on a mobile device functioning as a client. Furthermore, the resolution can be done immediately at the time of the definition of the final sub-profile or only at the time of filtering incoming events.

These choices also lead to considerations about centralisation versus replication: Should the resolved profiles be located only at a centralised computer or be replicated to every single involved party? If the choice is to use a client-server architecture, then it is good to make the server poll the clients or work event-driven rather than for the

server to wait for client responses, as clients might have crashed and consequently will not answer.

Alteration of Collaborative Profiles. Similarly to single-user profiles, collaborative profiles, or the sub-profiles they have been built from, might be changed by their users over time. This requires special mechanisms for the collaborative profile that has been subsumed from potentially altered sub-profiles.

Update Races. Additionally, it is worthwhile to consider how update races can be avoided. A possible approach can use timestamps for each sub-profile. Alternatively, locks can be used in order to avoid races. Locks could be realised by requiring a login for users that define sub-profiles.

Comments. The possibility to comment on their own changes of sub-profiles should be given to users. This allows for a better understanding and recollection of the reasons for each particular change.

Notification about Changes. Several questions arise regarding notifications about changes: Should there be a notification to the other parties if someone further refines a collaborative profile by adding a sub-profile or changing an existing one? Should there be a notification once the collaborative profile is resolved? Should there be a notification if a collaborative profile is not resolved in a certain period of time?

History. Should the history of the definition and modification process be stored within the system to enable a backtracking of potential errors that might have occurred? If the system is used in a domain where liability issues are of high concern, such as in the health care domain, then such a history will be required most definitely.

Further Social Aspects. It might be helpful to offer an awareness list of all users currently logged on (like a buddy list in some chat programs). This might support a better collaboration between the parties taking part in the profile definition.

As stated before, the previously described aspects had an influence on the design of collaborative profiles. Some of the questions that arise within this subsection are already answered. Others are picked up later on when describing the conceptual design of collaborative profiles. To ensure clarity, it is again pointed out that not all issues identified before can be dealt with within the scope of this thesis.

7.2.2 Recommender Systems

Recommender systems (or recommendation systems), such as described by Resnick and Varian [95] and by Balabanović and Shoham [96], are systems that give recommendations about items, events, sites, etc. to their users. The retrieval of recommendations may happen implicitly or explicitly and may use this particular user's or other people's feedback on the quality of recommendations and on their general interests.

Primarily, research differentiates between three different approaches for gaining recommendations: the content-based approach, collaborative filtering and the knowledge-based approach.

Content-based Approach. The content-based approach originates in Information Retrieval and is thus building on the methods of this area. It uses the past interests of users in order to find appropriate recommendations for them. These interests are stored in a profile for each user. The profile is used for comparisons between the content of documents that potentially might be of interest to the user and the user's actual interest.

In order to gain the information required for these profiles, a technique called relevance-feedback is employed. For this method users have to evaluate the recommendations they have received. Relative to the degree of interest of a user, they assign weights to keywords in a document. These weights are then incorporated into the selection of further recommendations by an algorithm. This approach has one drawback: Using standard techniques, the user will receive recommendations only on similar items as before; no new documents will be introduced. This is also referred to

as over-specialisation. In order to avoid this phenomenon, randomness is added to the selection algorithms.

Collaborative Filtering. Resolving the disadvantages of the content-based approach, collaborative filtering exploits other people's likes and dislikes rather than an individual's feedback. Thus, it evades the problem of over-specialisation by being based on similarities between users rather than on similarities between certain items. To realise this, collaborative filtering employs nearest neighbour algorithms. Collaborative filtering, however, requires a lot of data to work effectively. A recommender system using this approach, therefore, has to be initialised with a large data set. Furthermore, it is almost impossible for atypical users to receive useful recommendations. As their interests do not show similarity to the interests of other users, there is no data that recommendations for them can be deduced from.

Knowledge-based Approach. The third kind of recommender systems use the knowledge-based approach described by Burke [97]. Some work with decision support tools, others use case-based reasoning. Usually the user initially has to walk through an information catalogue. This gives information to the system about the interests of the user which is then used for finding recommendations for this user.

Thus, recommender systems solve a task completely different from alerting. The use cases for alerting in general, and collaborative alerting in particular, are orthogonal to the use cases of recommender systems. It is merely a coincidence that similar notions like "collaborative filtering" and "profiles" are used: While recommendation systems mainly work using comparisons in order to find suitable recommendations, alerting systems enable users to directly specify their individual information needs.

Collaboration in the context of recommender systems refers to internally considering information from other users to derive recommendations. Using these systems is driven by the wish to receive recommendations, not by the need to actively collaborate with others. Furthermore, recommender systems work in an automated way and

thereby leave little control to the user which in several domains, such as health care, is undesirable.

7.2.3 User Profiler Systems

User profiler systems, for example described by Moukas [98] and Shahabi et al. [99], collect information about users, for example while the user browses the web. This information can be about the contents a user is looking at, about the order of sites visited or about the lengths of visits to certain sites or documents. The knowledge is gained by an agent and is used to build a profile of the user which can be used for things such as providing more suitable information to this user. Information can be deduced, for example, from web browser history files, favourite files or cache references.

Similar to recommender systems, user profiler systems automatically gather information rather than letting users themselves actively participate in the definition of their information needs. Such an approach uses orthogonal techniques to alerting and does not offer leeway for supporting active collaboration between several users. The use of the term profile is again merely a coincidence.

7.3 The Concepts of Collaborative Profiles

Following the introductory examples in Section 7.1, this section outlines the general concept of collaborative profiles as well as their details. Furthermore, this section describes how collaborative profiles can be integrated into conventional alerting systems.

7.3.1 General Idea

A collaborative profile, as envisaged by Jung and Hinze [100], expresses an information need that has been defined by several parties collaboratively. One party starts the definition of the collaborative profile by defining the profile skeleton. Other parties refine this skeleton by providing their expert knowledge. If all required parties have

provided their information, the collaborative profile is defined and can be filtered by an alerting system. Figure 7-5 shows this concept of collaborative profiles in the context of an alerting system. In this figure, three parties collaboratively define the collaborative profile: one party provides the skeleton whereas two parties refine this skeleton.

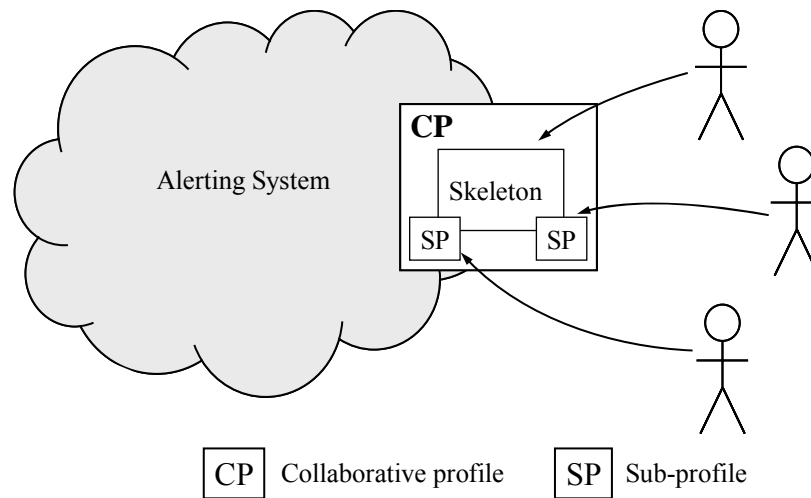


Figure 7-5: Collaborative profile within an alerting system.

Skeleton. The skeleton of a collaborative profile is created by the owner of this profile. In Example 34, the nurse is the owner of the profile; the medical practitioner in Example 35 also acts as an owner.

The profile skeleton describes the basic structure of a collaborative profile. It can contain parts that are built based on single-user profile mechanisms, such as conditions, operators, duration parameters and timeframes. Furthermore, the skeleton contains descriptions of information that should be provided by other parties. These descriptions can be seen as vague specifications within the profile. From the viewpoint of the owner of a collaborative profile, other parties refine the vague specifications that are provided.

Refinement. Parties that provide their expert knowledge for the specification of a collaborative profile work on the basis of a profile skeleton. The skeleton contains vague parts, that is, descriptions of the required information. These vague parts are refined through several parties by providing this information. Parties refining a skeleton are referred to as providers.

In Example 34, the doctor acts as provider of the profile skeleton defined by the nurse. For the skeleton defined by the medical practitioner in Example 35, both doctors act as providers for the profile.

Options for Refinement. Considering the concepts of single-user profiles, which are represented through GPD (cf. Chapter 3), there are different profile parts that can be left vague and allow for the refinement by providers. Collaborative profiles should allow for all of these parts to take part in the collaborative profile definition process. These vague parts are as follows:

- **Vague condition:** A conventional condition contains three parts, an attribute specification, an operator specification and a value specification. All three of these parts can be defined collaboratively and thus be left vague within a vague condition.
- **Vague operator:** Conditions, or more generally sub-profiles, are combined by operators. Choosing what operator is used for this combination can be part of collaboration. Candidate operators are all operators with at least two operands, that is, *some*, *all* and *sequence*. This concept is referred to as vague operator.
- **Vague duration:** Operators involving the time dimension (*sequence*, *repetition*, *all* and *no-occurrence*) can specify the duration parameter. This duration parameter might be specified by a party other than the party specifying the actual operator. This concept is referred to as vague duration.
- **Vague repetition:** The *repetition* operator is defined by specifying how often the repeated sub-profile should occur. Specifying this number of repetitions

should be part of the collaborative process. This is conceptually referred to as vague *repetition*.

- Vague timeframe: Each profile contains a timeframe, consisting of an absolute component and a repetitive component. Both the absolute and the repetitive timeframe might be specified in collaboration. This concept is referred to as vague timeframe.
- Vague subtree: Apart from assuming a skeleton to specify the full structure of the collaborative profile, there should be an option to allow providers to define any sub-profile. This means that the general structure of a collaborative profile can be specified in a collaborative way. This concept is referred to as vague subtree, as on the abstract level a subtree of the syntax tree of a GPD profile can be defined by a provider.

7.3.2 Specification of Underlying Concepts

Chapter 3 introduced an abstract representation (syntax tree) of the graphical language GPD to specify single-user profiles. For collaborative profiles, such an abstract representation of the required concepts exists as well. This subsection specifies these concepts as underlying basis for the Collaborative Graphical Profile Definition Language (CGPDL) that is introduced in Section 7.4.

Constituents of Collaborative Profiles. Collaborative profiles need to allow for the different options of specifying profile parts in a vague fashion, as presented in the previous subsection. Furthermore, several aspects that can be learned from CSCW research (Section 7.2.1) need to be considered for collaborative profiles. The ones presented in the following are among them. They are restricted to those aspects relevant to CoastEd, the interface for collaborative profile specification (Section 7.5):

- Both the owner of a collaborative profile and its providers have the option to tag sub-profiles with comments.

- Several parties are able to collaboratively define a profile, where each provider is responsible for one or several distinct parts of this profile.
- The granularity of collaboratively defined profile parts should be fine. That is, each individual part of a profile is treated separately with respect to collaboration.
- Collaborative profiles need to contain the original skeleton with vague definitions as well as those parts that have already been refined. So, providers might be shown the refinements of other providers if this is intended or required.
- Collaborative profiles should allow their owners to restrict providers in their refinement.

Grammar for Collaborative Profiles. The grammar for collaborative profiles needs to contain the concepts of GPD L as well as the additional concepts specified before. Figure 7-6 shows a part of the grammar that produces the syntax tree of a collaborative profile. The figure contains those nonterminal symbols that do not exist in GPD L (Figure 3-27 and Figure 3-1); nonterminals not given here are equivalent to the GPD L specification.

```
collaborativeProfile : profileExpression timeframe ;
```

```
profileExpression : box
  | '(' 'SEQUENCE' duration?
    profileExpression profileExpression+ ')'
  | '(' 'SOME'
    profileExpression profileExpression+ ')'
  | '(' 'ALL' duration?
    profileExpression profileExpression+ ')'
  | '(' repetition profileExpression ')'
  | '(' 'NOOCCURRENCE' duration?
    profileExpression ')'
  | '(' profileExpression ')'
  | '(' jobExpression ')' ;
```

```
repetition :
  'REPETITION' '[' INTEGER ']' duration? ;
```

```
duration : durationFixed | durationJob ;
```

```
durationFixed : '{'
  (INTEGER 'M')? (INTEGER 'w')? (INTEGER 'd')?
  (INTEGER 'h')? (INTEGER 'm')? (INTEGER 's')? '}' ;
```

```
timeframe :
  ('{'
    timeframeAbsolute | timeframeAbsoluteJob
  '}')?
  ('{'
    timeframeRepetitiveMonth
    | timeframeRepetitiveJobMonth
  '}')?
  ('{'
    timeframeRepetitiveDay
    | timeframeRepetitiveJobDay
  '}')?
  ('{'
    timeframeRepetitiveTime
    | timeframeRepetitiveJobTime
  '}')? ;
```

```
jobExpression : operatorJob
  | repetitionJob | openJob | conditionJob ;
```

```
conditionJob : 'VAGUECONDITION'
  ('restrictionAttributeDomain' '='
    '{' conditionJobDomainRestriction '}')?
  ('restrictionAttribute' '=' '{'
    conditionJobAttributeRestriction '}')?
  ('settings' '=' '{' jobInformation '}')
  ('defined' '=' '{' condition '}')? ;
```

```
conditionJobValueRestriction :
  conditionJobSelfDefinedValueRestriction
  | conditionJobStringValueRestriction
  | conditionJobIntegerValueRestriction ;
```

```
conditionJobSelfDefinedValueRestriction :
  'allowedElements' '='
  '{' value (',' value)* '}' ;
```

```
conditionJobStringValueRestriction :
  'allowedStrings' '='
  '{' value (',' value)* '}' ;
```



```

conditionJobIntegerValueRestriction :
  'allowed' '=' '{' INTEGER ',' INTEGER '}' ;

conditionJobComparisonOperatorRestriction :
  'allowed' '=' '{' comparisonOperator (','
    comparisonOperator)* '}' ;

conditionJobAttributeRestriction :
  'allowed' '=' '{' conditionJobAttribute
    (',' conditionJobAttribute)* '}' ;

conditionJobAttribute : attribute
  ('restrictionOperator' '=' '{'
    conditionJobComparisonOperatorRestriction '}')?
  ('restrictionValue' '=' '{'
    conditionJobValueRestriction '}')? ;

conditionJobDomainRestriction :
  'allowed' '=' '{' conditionJobDomain
    (',' conditionJobDomain)* '}' ;

conditionJobDomain : STRING | INTEGER | TEXT ;

operatorJob : 'VAGUETREEOPERATOR'
  ('restriction' '='
    '{' operatorJobRestriction '}')?
  ('settings' '=' '{' jobInformation '}')
  ('defined' '=' '{' operatorJobOperatorName '}')?
  ('duration' '=' duration)?
  ('children' '=' '{' (profileExpression)* '}') ;

operatorJobRestriction :
  'allowed' '=' '{' operatorJobOperatorName
    (',' operatorJobOperatorName)* '}' ;

operatorJobOperatorName : STRING | INTEGER | TEXT ;

repetitionJob : 'VAGUESELECTION' ('restriction' '='
  '{' repetitionJobRestriction '}')?
  ('settings' '=' '{' jobInformation '}')
  ('defined' '=' INTEGER)?
  ('original' '=' '{' '(' repetition
    profileExpression ')' '}') ;

repetitionJobRestriction :
  'interval' '=' '{' INTEGER INTEGER '}' ;

```

```

openJob : 'VAGUESUBTREE'
  ('settings' '=' '{' jobInformation '}')
  ('defined' '=' '{' profileExpression '}')? ;

timeframeJob : timeframeAbsoluteJob
  | timeframeRepetitiveJobTime
  | timeframeRepetitiveJobMonth
  | timeframeRepetitiveJobDay ;

timeframeAbsoluteJob : 'VAGUEABSOLUTE'
  ('restriction' '=' '{'
    timeframeAbsoluteJobRestriction '}')?
  ('defined' '=' '{' timeframeAbsolute '}')? ;

timeframeRepetitiveJobMonth : 'VAGUEMONTH'
  ('restriction' '=' '{'
    timeframeRepetitiveJobMonthRestriction '}')?
  ('defined' '=' '{'
    timeframeRepetitiveMonth '}')? ;

timeframeRepetitiveJobDay : 'VAGUEDAY'
  ('restriction' '=' '{'
    timeframeRepetitiveJobDayRestriction '}')?
  ('defined' '=' '{' timeframeRepetitiveDay '}')? ;

timeframeRepetitiveJobTime : 'VAGUETIME'
  ('restriction' '=' '{'
    timeframeRepetitiveJobTimeRestriction '}')?
  ('defined' '=' '{'
    timeframeRepetitiveTime '}')? ;

timeframeAbsoluteJobRestriction :
  'min' '=' '{' timeframeAbsolute '}' ','
  'max' '=' '{' timeframeAbsolute '}' ;

timeframeRepetitiveJobMonthRestriction :
  'min' '=' '{' timeframeRepetitiveMonth '}' ','
  'max' '=' '{' timeframeRepetitiveMonth '}' ;

timeframeRepetitiveJobDayRestriction :
  'min' '=' '{' timeframeRepetitiveDay '}' ','
  'max' '=' '{' timeframeRepetitiveDay '}' ;

timeframeRepetitiveJobTimeRestriction :
  'min' '=' '{' timeframeRepetitiveTime '}' ','
  'max' '=' '{' timeframeRepetitiveTime '}' ;

```

```

durationJob :
('restriction' '=' '{' durationJobRestriction '}')?
('settings' '=' '{' jobInformation '}')
('defined' '=' duration)? ;

durationJobRestriction : duration duration ;

jobInformation : 'providerInformation' '='
  '{' providerInformation '}'
  'refinerInformation' '='
  '{' refinerInformation '}' ;

providerInformation :
  'deadline' '=' timeframeDate ','
  'description' '=' TEXT ','
  'provider' '=' personSpecification ;

refinerInformation :
  'comment' '=' TEXT
  (',' 'persons' '='
  '{' personsSpecification '}')? ;

personsSpecification :
  personSpecification (',' personsSpecification)? ;

personSpecification : STRING | INTEGER | TEXT ;

```

Figure 7-6: Partial EBNF of an internal representation of collaborative profiles.

The vague concepts required in collaborative profiles can be directly found within the specification (nonterminals repetitionJob, conditonJob, operatorJob, durationJob, timeframeAbsoluteJob, timeframeRepetitiveJobMonth, timeframeRepetitiveJobDay, timeframeRepetitiveJobTime and openJob).

Furthermore, vague parts can be tagged with comments (nonterminals providerInformation and refinerInformation) and it is possible to individually specify the provider for each vague part (nonterminal providerInformation).

All vague concepts can also contain restrictions to a particular set or range of values (nonterminals conditionJobValueRestriction, imeframeRepetitiveJobDayRestriction,

timeframeRepetitiveJobMonthRestriction, timeframeRepetitiveJobTimeRestriction, timeframeAbsoluteJobRestriction, durationJobRestriction, operatorJobRestriction, repetitionJobRestriction).

7.3.3 Integration into Conventional Alerting Systems

Collaborative profiles augment those components of alerting systems that perform the filtering of profiles and events: Only once the collaborative profile has been fully specified, that is, only once all vague parts have been refined by providers, is the profile incorporated into the matching process of the system. Figure 7-7 shows this approach: All sub-profiles (including the skeleton) are resolved into a collaborative profile that is eventually registered with the alerting system.

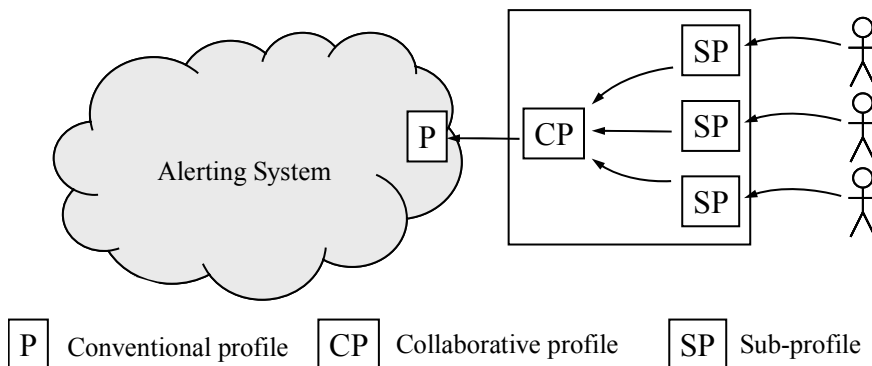


Figure 7-7: Integration of collaborative profiles into alerting systems.

A fully specified collaborative profile can be straightforwardly expressed in GPDL: Collaborative profiles are an extension of GPDL profiles to support collaboration. Once this collaboration has been taken place, a GPDL profile can be extracted.

Using the concepts described Chapter 3, a GPDL profile can be mapped to a profile in the underlying PDL. Furthermore, profiles defined in this underlying PDL can be mapped to profiles for other alerting systems, provided these systems support the required concepts. Hence, once all required providers have worked on a collaborative profile, it can be used within conventional alerting systems.

7.3.4 Application Variant for Collaborative Profiles

The envisaged application of collaborative profiles is to jointly define a profile that is registered with an alerting system after the refinement is undertaken. There is at least one other scenario collaborative profiles can be applied to: personalisation.

The concepts in this scenario remain the same. However, there is a shift in focus and purpose of collaborative profiles. Instead of having the goal of collaboratively specifying and using a profile, in the case of personalisation, the profile owner defines the broad structure of the profile without the intention of actively collaborating with others. The profile owner wants to allow other users to fine tune the skeleton, that is, to personalise the profile. For this scenario, it could be imagined to provide some default refinement of the profile that is applied if other users do not personalise this default setting.

An example application area is facilities management (Section 7.1.2), where building managers could act as the profile owner. Profiles would control the temperature of a heating or the brightness of a room. Tenants could act as providers and, for example, adjust their room temperature directly. Also for short-term changes of requirements, for example for special requirements for exhibitions, the personalisation approach might prove effective.

Having introduced the general concepts of collaborative profiles in this section, the next section analyses how GPDL needs to be extended to be able to express those additional concepts that are required for collaboration.

7.4 The Collaborative Graphical Profile Definition Language CGPDL

The graphical language used for the specification of collaborative profiles, CGPDL, is intrinsically GPDL, the graphical language developed for single-user profiles. The

presentation and concepts of Boolean operators, Boolean-time operators, time operators, condition boxes, nested boxes, grouping boxes, duration parameters and profile timeframes are retained. Extensibility of GPDL to collaborative concepts was one of its design goals.

7.4.1 Colour

In CGPDL, colour is used as an additional dimension to convey information. The colour dimension is applied to indicate the participants that take part in the specification of a collaborative profile. For the profile owner, vague parts of the profile are coloured according to the provider of this information. Each provider, in turn, corresponds to one colour, allowing them to determine what parts of a profile require their expert knowledge.

7.4.2 Placeholders

Section 7.3.1 introduced six possibilities to specify vague parts within profiles. For all of them, it is sufficient to introduce some kind of placeholder to indicate which of these vague concepts is presented (colour indicates a vague concept in general). These placeholders allow for a clear and unambiguous determination of the kind of vague concept by both owner and provider.

- Vague condition: A condition box contains an attribute, an operator and a value. Any combination of these three parts can be left vague by the profile owner. A placeholder is applicable for each of them to indicate that a provider should refine this particular part. Figure 7-8 exemplifies this with all three parts being left vague.

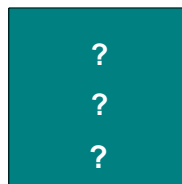


Figure 7-8: Vague condition in CGPDL.

- Vague duration: The duration parameter specifies the allowed time between two (or more) events being combined. The duration value is stated above the duration arrow. Instead of this time, a placeholder can be used, as shown in Figure 7-9.

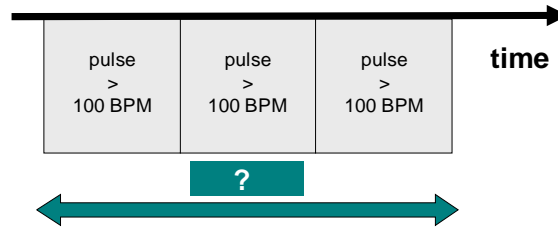


Figure 7-9: Vague duration in CGPDL.

- Vague repetition: The *repetition* describes that a particular sub-profile needs to match a particular number of times. In GPDL, the boxes representing these sub-profiles are placed horizontally adjacent to each other. For the vague repetition, this placement is retained but a placeholder indicates that the exact number of repetitions has not been specified yet. This is shown in Figure 7-10.

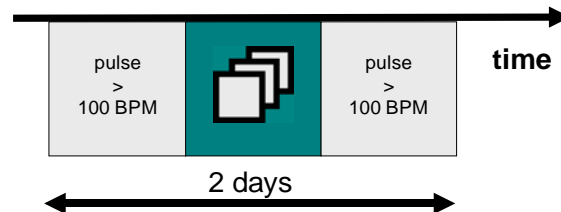


Figure 7-10: Vague repetition in CGPDL.

- Vague timeframe: The timeframe of a profile, specifying when the profile should be matched by the alerting system, is indicated by vertical lines left and right of the profile itself. Instead of stating values for the timeframe annotations, placeholders indicate a vague timeframe (Figure 7-11).
- Vague subtree: The vague subtree allows providers to specify arbitrary sub-profiles. This concept is indicated by a box acting as a placeholder for this sub-profile in CGPDL, as shown in Figure 7-12.



Figure 7-11: Vague timeframe in CGPDL.

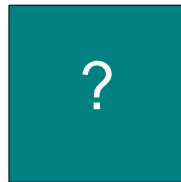


Figure 7-12: Vague subtree in CGPDL.

- Vague operator: The purpose of a vague operator is to let a provider determine by what operator (*all*, *some* or *sequence*) sub-profiles should be combined. In CGPDL, this concept of a vague operator is denoted by a placeholder box, containing those operators that are available to the provider. Sub-profiles themselves, that is, the operands, are expressed using their usual representation. These operands, however, cannot be presented in relation to each other, as their locations would indicate a particular operator, being expressed by the relative locations of its operands in GPD and CGPDL. Nevertheless, CoastEd includes a mechanism to show these operands and the vague operator they belong to, as it will be introduced in Section 7.5. Figure 7-13 shows the placeholder for the vague operator.

Having introduced CGPDL, the following section describes CoastEd, the user interface for specifying profiles in CGPDL.



Figure 7-13: Vague operator in CGPDL.

7.5 CoastEd: The CGPDL Interface

CoastEd is an editor for collaborative profiles using CGPDL. It is an extension of GPDL-UI, presented in Chapter 5. CoastEd is used for all participants in the collaborative profile specification process, that is, profile owners and providers.

7.5.1 Description

CoastEd (see Figure 7-14 for the empty interface) applies the same separation of the general working area into workspace and component bar as GPDL-UI. As an extension, there is a further element in the interface: the job tray. Its purpose is to present collaboration-related information to the user. The set of available buttons is also similar to those known from GPDL-UI. The following subsections go into more detail about these different interface components.

Workspace. As in GPDL-UI, the workspace contains the profile that currently is being edited by a user. For the profile owner, this is the skeleton of the collaborative profile, potentially including vague specifications. For a provider, this is the skeleton and the already specified parts of the collaborative profile.

In CoastEd, the current component display known from GPDL-UI has been excluded as it was found that it is not adopted by users as intended (Chapter 5). Instead study participants asked for a shortcut to navigate to the top level of a profile. The respective button, as well as another button to navigate up one level within a nested profile, was placed in the bottom left corner of the workspace.

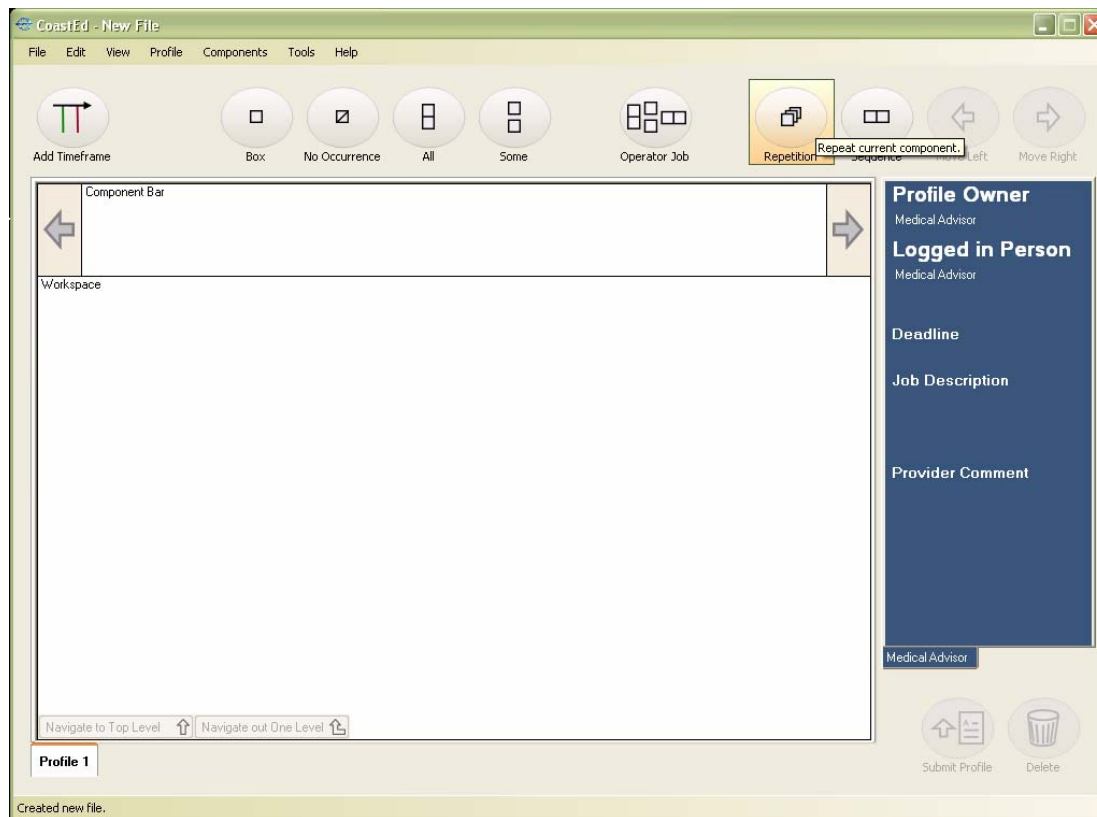


Figure 7-14: Empty interface in CoastEd.

CGPDL Elements. Profiles in CoastEd are displayed in their CGPDL representation, using the visualisations for vague parts that have been described in Section 7.4.2. As an addition, each vague part contains a label, assigning a unique name to this part. In CoastEd, vague parts are referred to as jobs (condition job, duration job, timeframe job, operator job, *repetition* job as well as open job for the vague subtree) and are labelled by “Job 1”, “Job 2”, etc. Each participant in a collaborative profile is assigned a colour. One and the same colour is used for all vague parts that need to be provided by a particular provider.

Component Bar. In contrast to its use in GPDUI, the component bar in CoastEd holds two different kinds of things: On the one hand, it stores all those sub-profiles that are not presently used by the current profile on the workspace, as is the case for GPDUI. On the other hand, it stores those sub-profiles that represent operands of an operator job. These operands are marked as such by a label that assigns

them textually and by colour to the symbolic representation of the operator job on the workspace.

Exchanging components between the component bar and the workspace works analogously to the way it does for GPDL-UI (drag and drop).

Job Tray. The job tray (shown in Figure 7-15) shows collaboration-related information. There is one tab for each provider assigned to a collaborative profile. The job tray shows general information about the current profile and, more importantly, job-related information.



Figure 7-15: Example of the job tray in CoastEd.

General information in the job tray states the owner of the profile and the person that is currently logged onto CoastEd. The tab for each provider (being presented in the colour of this provider) contains a list of all jobs assigned to this provider. For each job, the following information is presented: A description of the job written by the profile owner (after the pilot study, an automatic default description was added), for example giving additional information to the provider of what should be done. Moreover, the job tray shows the deadline by which the assigned job should be finished.

The provider of a profile can give feedback by entering a comment for the profile owner. Depending on who is currently logged on, these information can be editable.

Buttons. CoastEd offers buttons for the building blocks of GPDL—the condition box, the operators *sequence*, *repetition*, *all*, *some* and *no-occurrence*—and for changing the order of the operands of the *sequence*.

The timeframe button is placed separately as it refers to the entire profile. Also the system controls to save a profile and to delete a profile or sub-profile are located separately. The duration button was removed from the interface as it caused confusion in GPDL-UI (cf. previous chapter). Instead the duration can now be added by clicking on the duration indicator that is placed under each operator that can contain the duration parameter.

There is a new button for the operator job that allows for the specification of operator jobs and the refinement of such jobs, respectively. The other kinds of jobs can be created and refined similarly to their single-user counterparts. These functionalities are outlined in the following subsection.

7.5.2 Specification Options

As introduced before, there are six kinds of jobs that can be given to providers of collaborative profiles. The specification of these jobs can be integrated into the specification mechanisms of the corresponding single-user profile concepts. As operators are specified using different buttons in CoastEd, the operator job is offered via an additional button, as described before. The other jobs are offered as options in those windows that allow for the specification of conditions, durations, timeframes and *repetitions*. That is, the provider can choose whether, for example, to specify a single-user *repetition* or a *repetition job*.

When specifying a profile or a sub-profile, users have a choice among four specification options:

- Conventional specification: This is the semantics of conventional profiles that has already been used in GPDL-UI. The specification of a condition, *repetition*, duration, timeframe or operator is done directly by the owner without collaborating with other parties. Exact values are specified.
- Job specification: In this option, the specification of particular parts of a profile is completely left to a provider. For the different kinds of jobs this works as follows:
 - Condition job: The provider can choose an attribute, operator and value. Optionally, attributes, operators or values can be restricted to a set of attributes, a set of operators, a set of values or a range of values, respectively (sets and ranges are assumed to let the provider choose among elements).
 - Operator job: The provider can choose among the *sequence*, *some* and *all* operators. Optionally, the candidate operators can be restricted.
 - Duration job: The provider can choose the duration parameter. The duration can be restricted to a minimum and a maximum value.
 - Timeframe job: A timeframe consists of two kinds of time-periods: an absolute timeframe and a repetitive timeframe. The absolute timeframe specifies two components: a start and an end date; the repetitive timeframe specifies interruptions in using three components: time, week-days and months.

The provider can specify all of these five components. Optionally, components might be restricted to at least two different values.

- *Repetition* job: The provider is allowed to specify the number of repetitions required for a sub-profile. This number might be restricted by minimum and maximum values.

- Open job: The open job allows the provider to specify any sub-profile, that is, to extend the profile skeleton. This semantics does not leave room for any restrictions.
- Mix of conventional and job specification: For some kinds of specifications, the intention is to let the provider extend a partial conventional specification. This is allowed for the following types of jobs:
 - Condition job: For the condition job, a provider might only have to choose a particular value, whereas the attribute and the operator can not be altered by the provider. Other options are also possible, however, the one presented might occur most often in practice.
 - Timeframe job: A timeframe consists of five components. The provider might be able to define only a subset of these components, whereas the other components are already defined by the owner in the conventional way. For example only the start and the end date can be refined by a provider; the interruptions are already given by the owner.
 - Duration job: It is subject to interpretation whether the following situation is regarded as mixed specification, as the number of months cannot be refined: The provider might choose the duration to lie between one month and two days, and one month and fifteen days.
- Refinement specification: Refining parts of profiles that have been assigned as jobs to providers is obviously possible for all six kinds of jobs.

Table 7.1 gives an overview of these different specification options. To get a better impression of how profile specification works in the interface, the following section demonstrates the use of the interface for example profiles.

Table 7.1: Available specification options for selected language concepts.

Concept	Conventional	Job	Mix of conventional and job	Refinement
Condition	yes	yes	yes	yes
Duration	yes	yes	yes / no ¹⁹	yes
<i>Repetition</i>	yes	yes	no	yes
Timeframe	yes	yes	yes	yes
Operator	yes	yes	no	yes
Subtree	yes	yes	no	yes

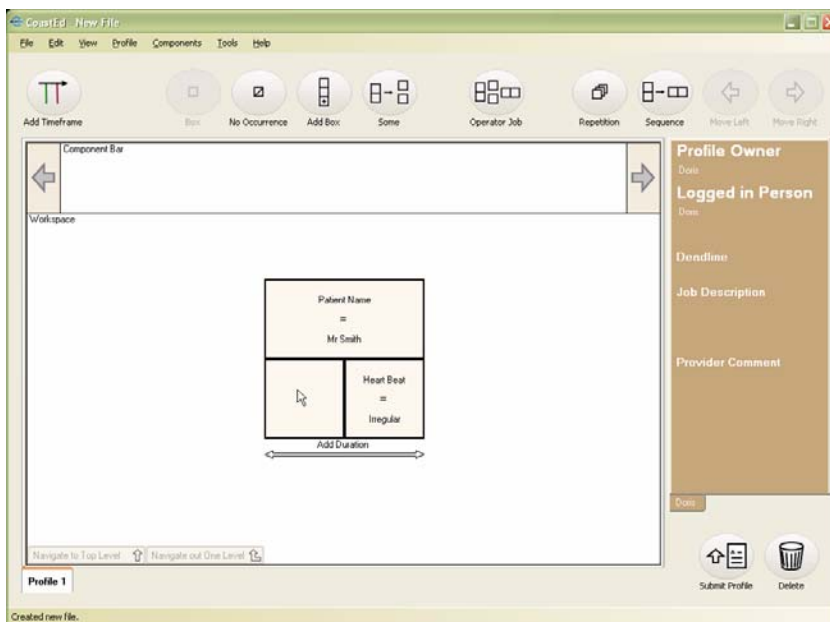
7.5.3 Specification Process

Having presented how collaborative profiles can generally be specified in CoastEd, this section details the specification process. This is done by presenting example interactions with CoastEd. This following subsection demonstrates the definition of two profiles by the profile owner. Afterwards, it is shown how these two profiles are refined by providers. For brevity, only those parts of the specification processes directly related to collaborative profiles are shown (information about the general processes was presented in Chapter 5).

Specification for Owner. Table 7.2 demonstrates the use of CoastEd to specify a collaborative profile, containing a condition job, a duration job and a timeframe job.

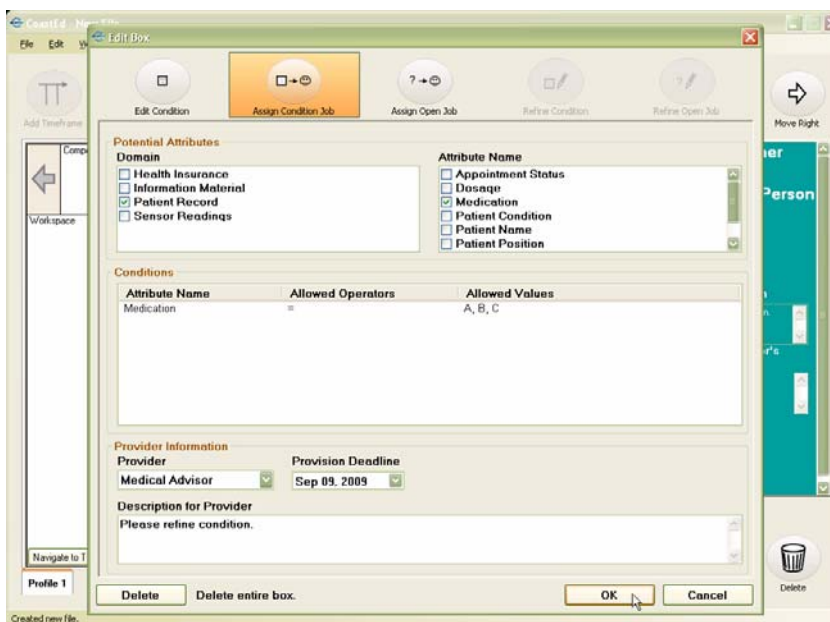
¹⁹ Subject to interpretation.

Table 7.2: Specification of a collaborative profile (condition job, duration job and timeframe job) for the owner.



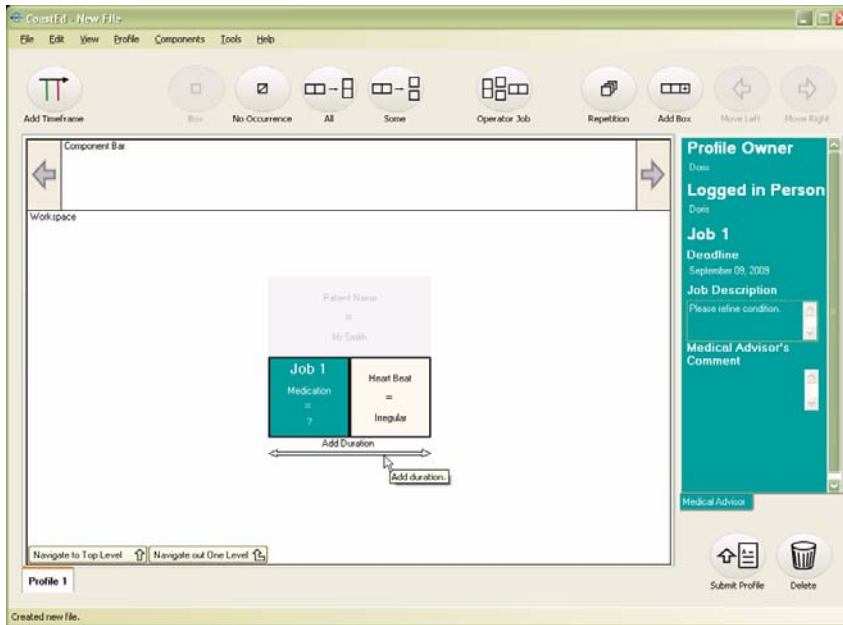
The owner has created some parts of the profile skeleton.

One condition of the profile can not be specified by the owner. It is about to be specified as a condition job to be provided by someone else by navigating into the empty condition box.



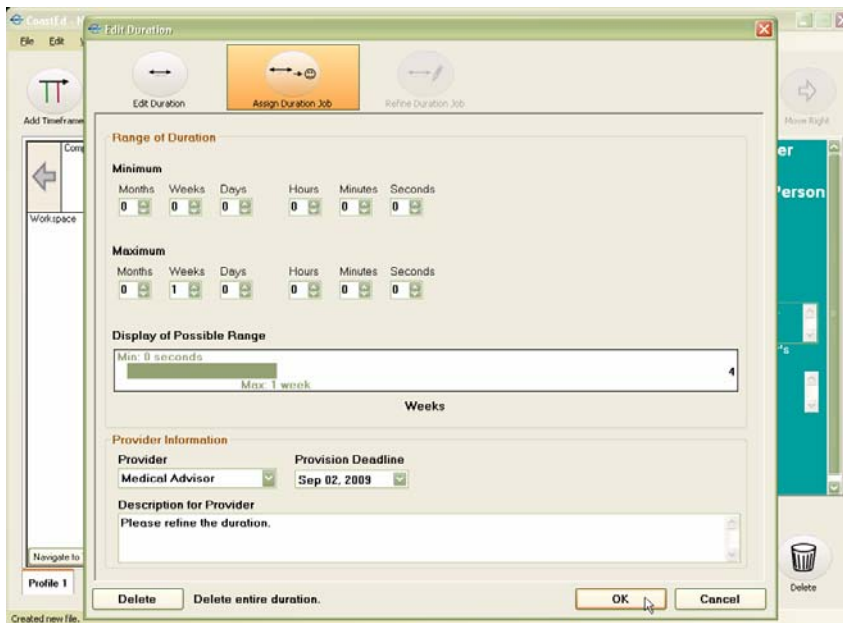
The owner chose to assign a condition job. One attribute and one operator are selected whereas the provider can choose among all possible values. A provider, a provision deadline and a comment to the provider are specified.

Instead of choosing a condition job, the owner could have also selected an open job (option further right).



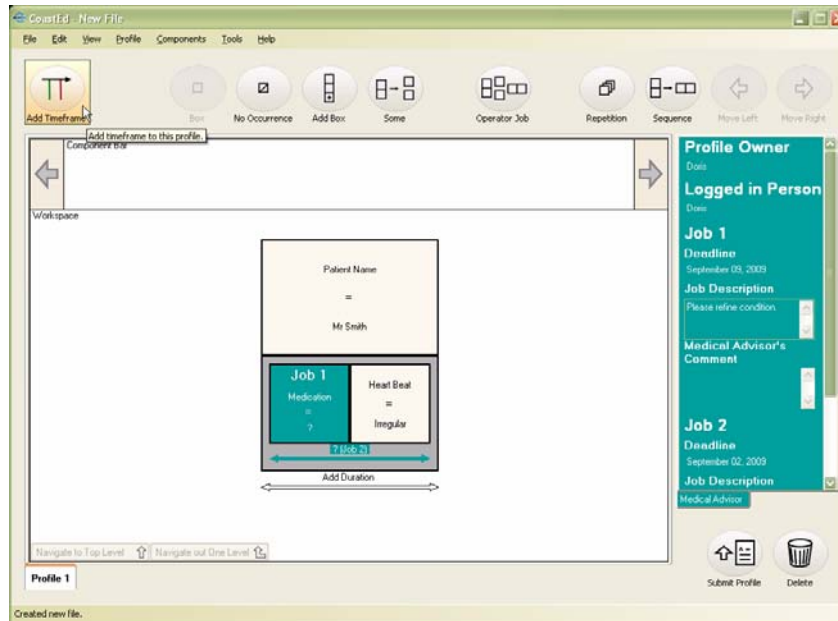
Information about the created condition job is shown in the job tray.

The owner is about to specify the duration of the *sequence*, which should be provided by the same provider as the previous condition.



The owner selected to assign a duration job.

The minimum and the maximum values for the duration are specified. The provider is chosen as well as a provision deadline, and a comment is given to the provider.



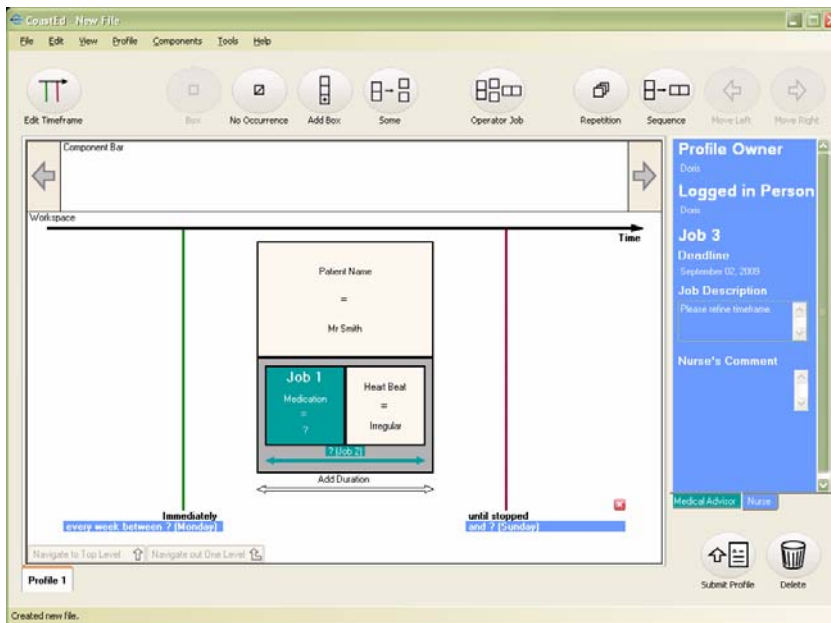
The second job with its information is shown in the job tray. It is shown in the same tab as the previous job, as it was assigned to one and the same provider.

The owner is about to specify the last part of the profile, the timeframe.



A timeframe job was selected. The user has specified the start and the end date of the timeframe. The weekdays to be used for monitoring should be given by the provider.

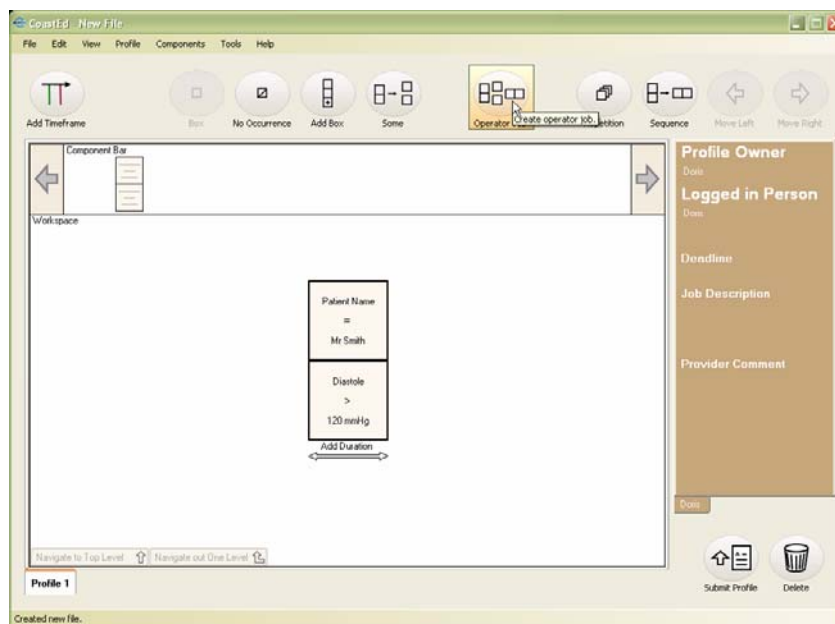
The timeframe is to be specified by a different provider than the provider of the two other jobs.



The profile has been fully specified by its owner.

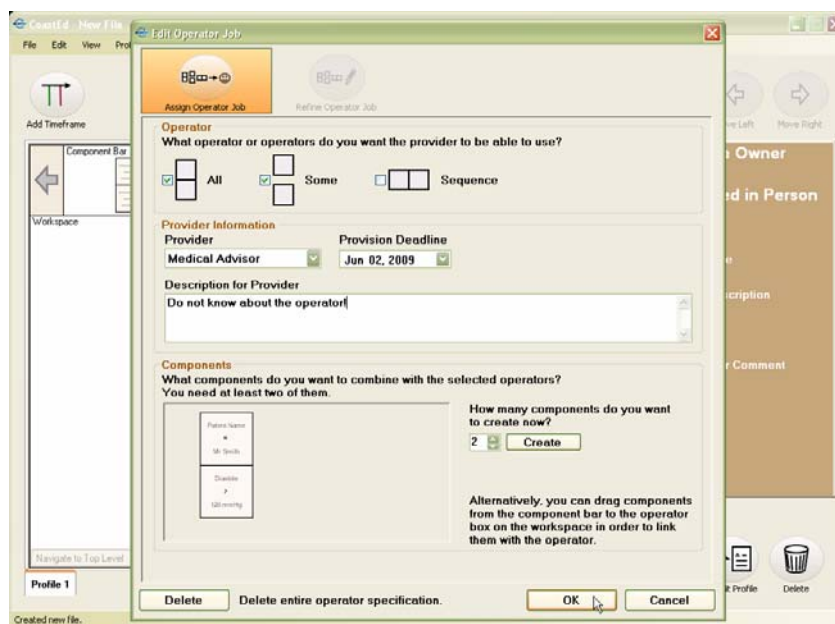
There is a second tab in the job tray, which shows the information that is provided to the provider of the time-frame.

Table 7.3 demonstrates the specification of a profile that contains an operator job. The specification of a *repetition* job is left out for brevity.

Table 7.3: Specification of a collaborative profile (operator job) for the owner.

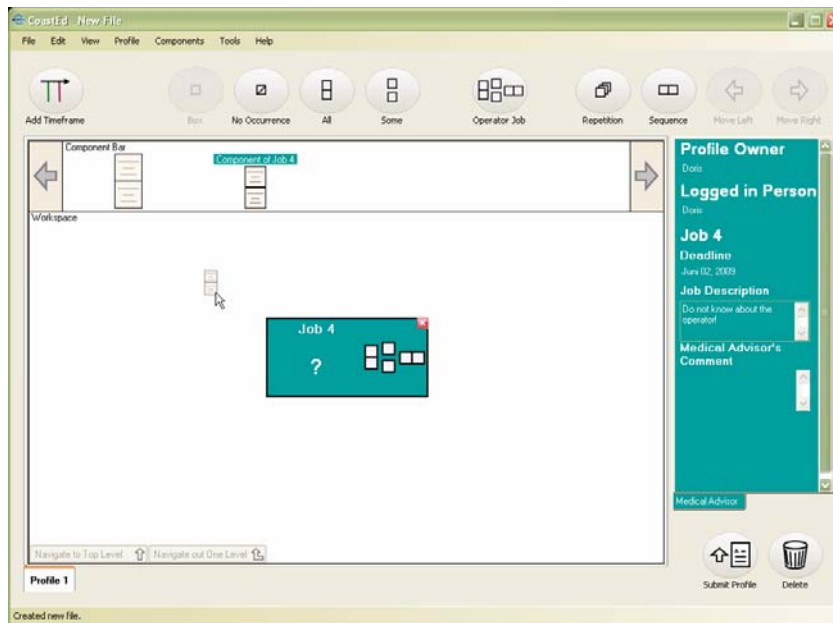
The owner created two operands (one of them in the component bar) but is unsure whether they should be combined by an *all* operator or by a *some* operator.

Therefore, an operator job is about to be created to let someone else decide on the operator.



The owner creates an operator job (by clicking the respective button) and selects that the provider can choose among two operators.

The component that was placed on the workspace before becomes one of the operands of the job.



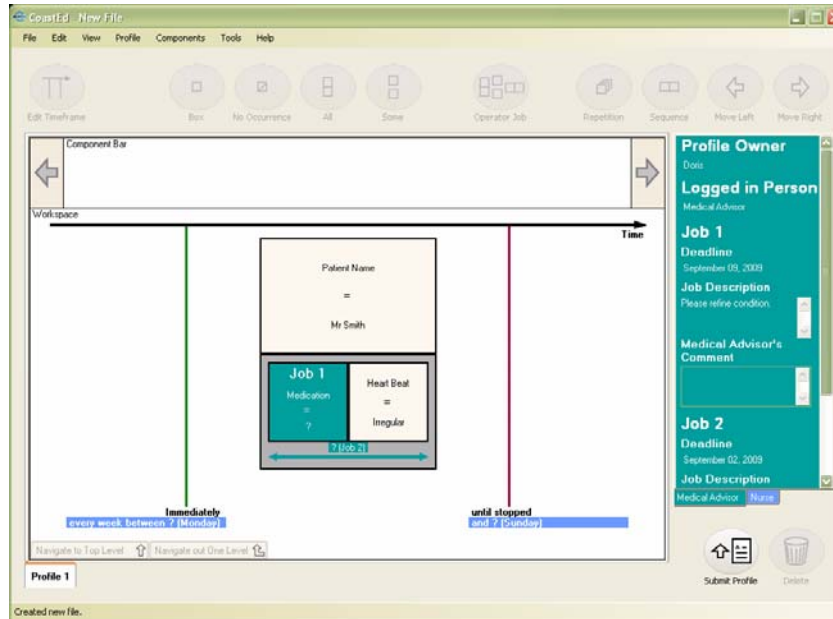
The other component, not being marked as operand of the job yet, is dragged onto the operator job to select it as the second operand. It will then be marked as well (as the first operand on the right in the component bar).

The profile has been specified and can now be refined by the provider.

Specification for Provider. Only those parts of profiles that have been assigned to the provider who is currently logged in can be altered in CoastEd. All other profile components are still shown but set to an inactive status.

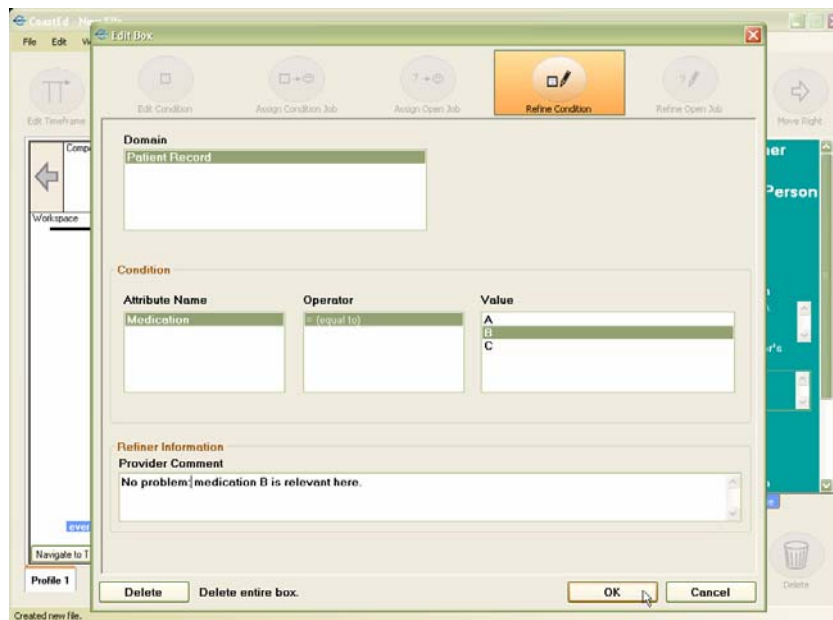
Table 7.4 demonstrates the definition of a collaborative profile in CoastEd by two providers (the profile is the one that is created in Table 7.2).

Table 7.4: Specification of a collaborative profile (condition job, duration job and timeframe job) for the provider.



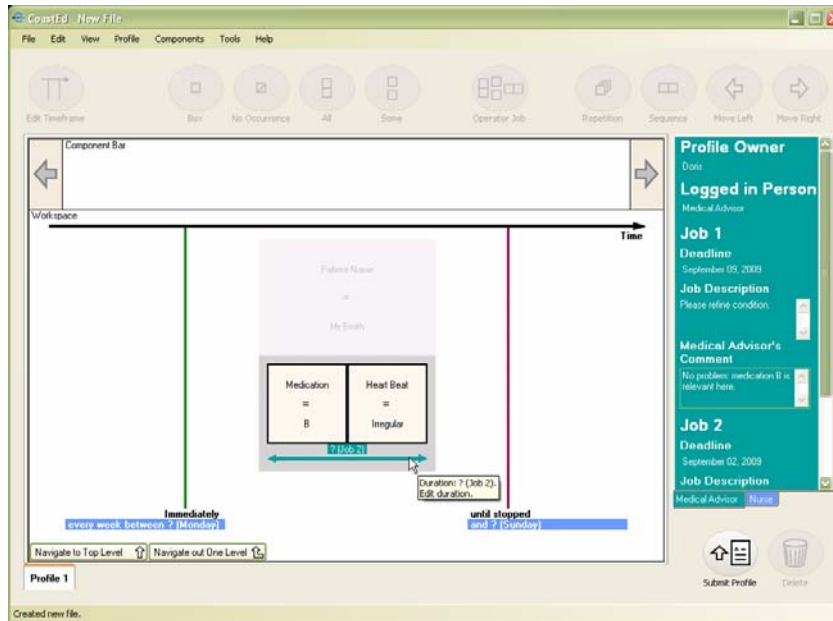
The provider takes a look at the profile. It contains two jobs that require refinement by them; another job needs to be refined by a different provider.

This provider can not edit any parts of the profile other than those assigned to them.

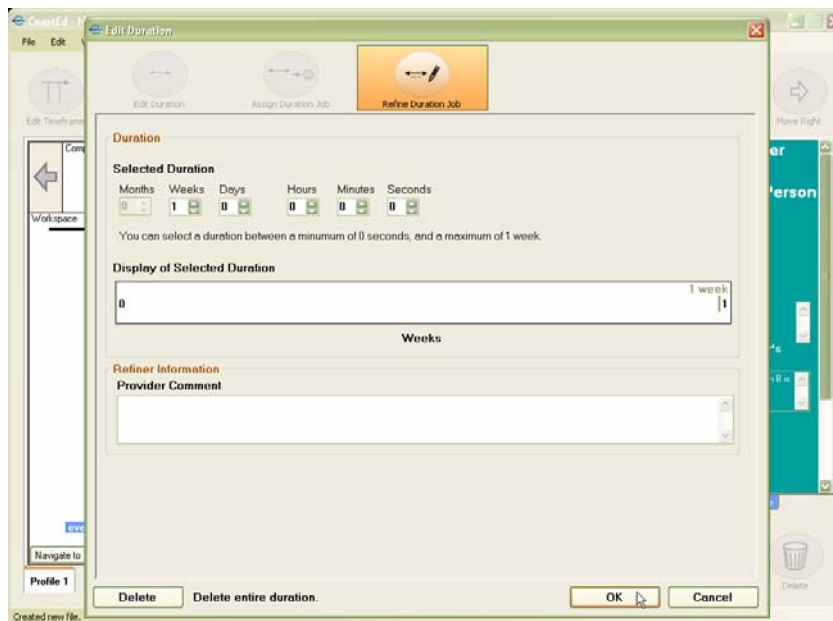


The provider navigated to the condition job and entered the condition.

They have the option to select the value of the condition; attribute and operator have been chosen by the owner. They also enter a comment and finish the job.

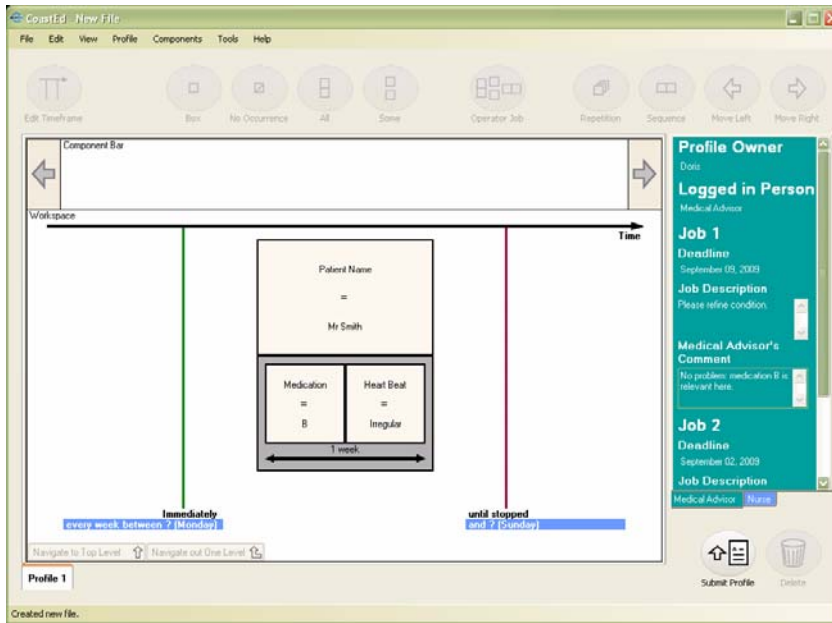


The condition job is specified. The provider navigates to the second job, the duration job, and is about to enter the duration.



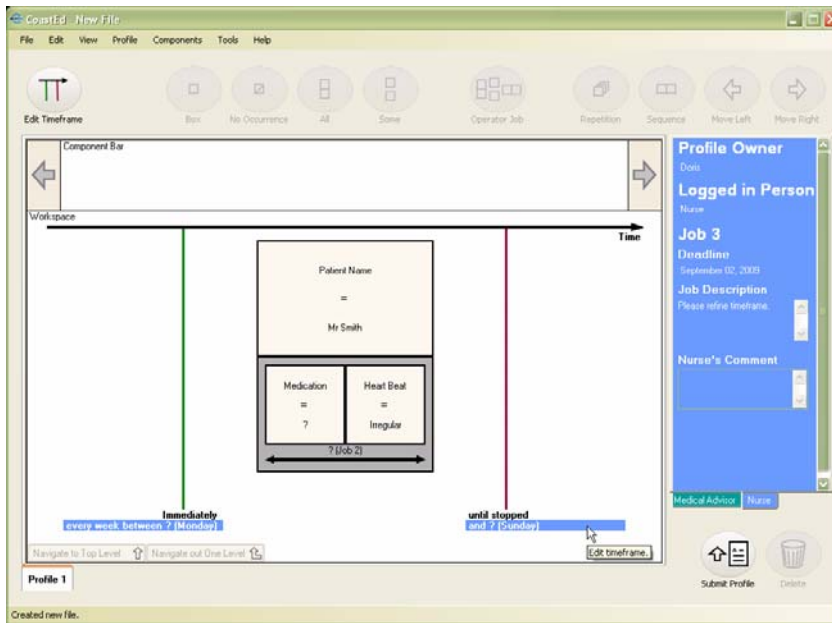
The provider clicked the duration job.

They select a duration within the allowed minimum and maximum values that have been specified by the profile owner.



This provider has refined all jobs assigned to them.

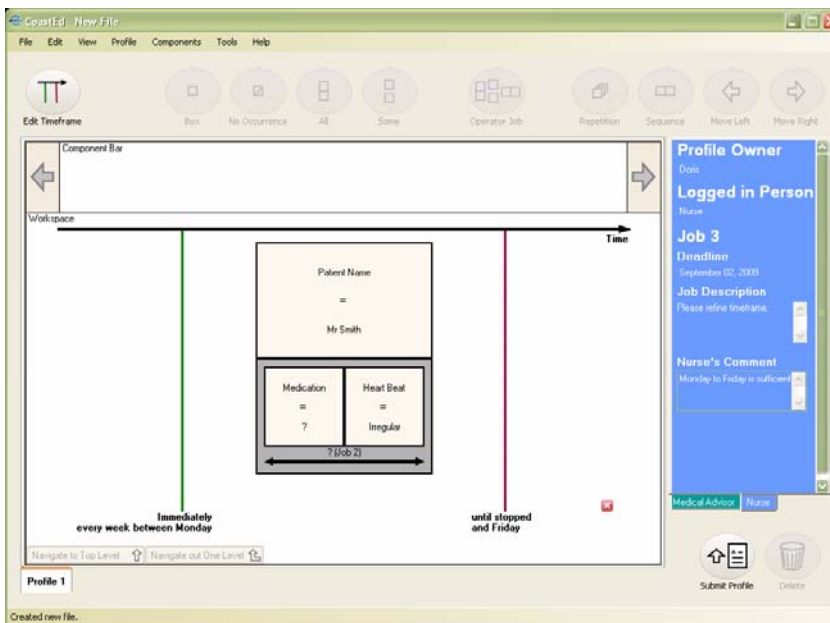
Only the timeframe that needs to be provided by someone else still has to be refined.



The second provider has logged onto the system and is about to refine the timeframe job by clicking on it.

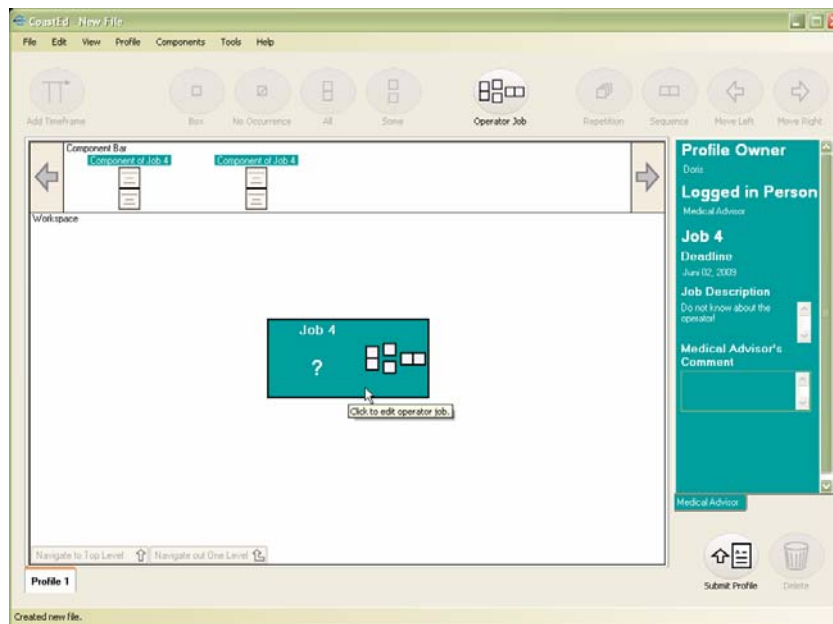


The provider is able to edit the weekdays in the repetitive timeframe; all other parts of the timeframe cannot be changed, as the owner specified them conventionally.



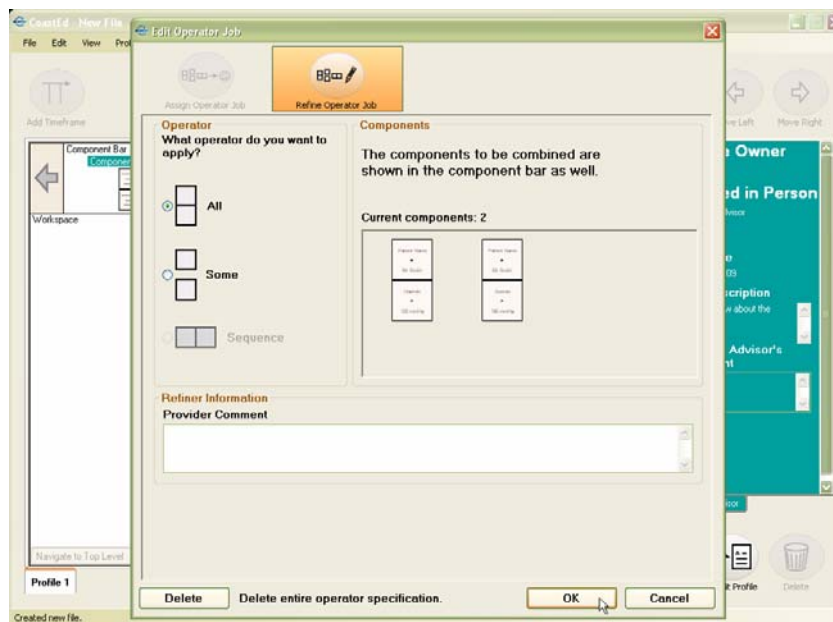
The collaborative profile is fully specified and can now be filtered by the alerting system.

Table 7.5 demonstrates how to refine the profile skeleton that was created in Table 7.3.

Table 7.5: Specification of a collaborative profile (operator job) for the provider.

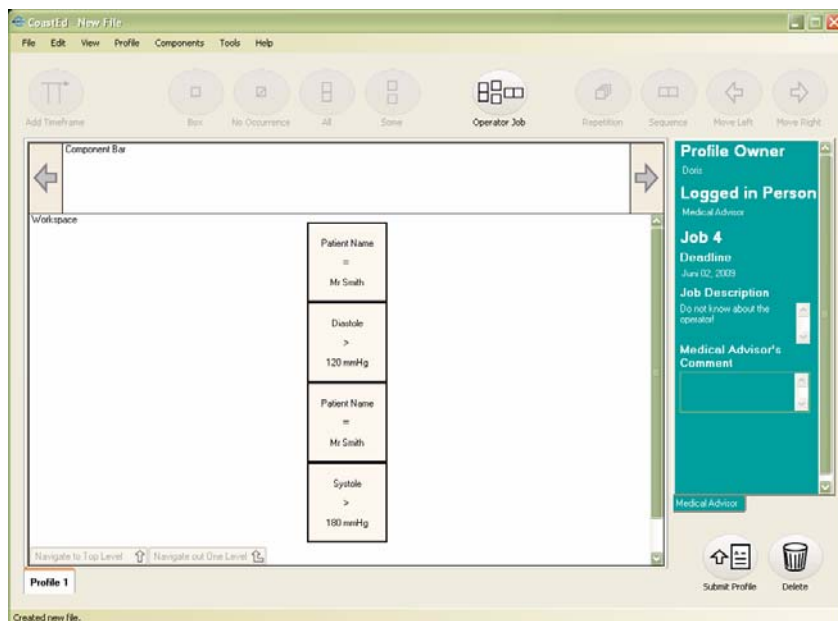
The provider recognises that an operator job has to be refined. They read the instructions in the job tray; they can access the operands if they need to get more information.

The provider is about to click the operator job (they could also have clicked on the operator-job button).



The provider chooses the *all* operator for the combination of the two operands. For information purposes, the operands are shown in the window as well.

If a *sequence* would be chosen, the operands could be reordered.



The profile is fully specified and uses the *all* operator. It can now be registered with the alerting system.

7.6 Summary

This chapter introduced a collaborative model for alerting systems. Section 7.1 started by motivating the usefulness and requirement of collaboration in alerting systems. Related research areas were analysed in Section 7.2.

Based on this work, Section 7.3 proposed the collaborative alerting model, described the underlying concepts and identified different classes of collaboration within profiles. CGPDL, the graphical profile specification language supporting collaboration, was developed in Section 7.4.

Finally, Section 7.5 described CoastEd, an editor for collaborative profiles that uses CGPDL. A user study of CoastEd is presented in the following chapter.

8 User Study of the Editor CoastEd for the Specification of Collaborative Profiles

The study in this chapter constitutes the final study of this PhD research. It analyses CoastdEd, the editor for collaborative profiles that was introduced in the previous chapter. Section 8.1 and Section 8.2 present the study procedure and the study content; they are based on what has been learned in the previous two studies that were undertaken (Chapter 4 and Chapter 6).

Section 8.3 gives an overview about quantitative findings; qualitative findings are detailed in Section 8.4. A concluding discussion of the study and its results is presented in Section 8.5.

8.1 Goals of the Study

There were three major goals to this study. They were to determine

- Whether users are capable of understanding the process of collaborative profile specification
- If the participants are able to effectively use the CoastEd interface in order to follow this collaborative specification process
- Whether CGPDL is a suitable language for the correct collaborative specification of profiles

The first goal is to analyse the participants' general understanding of the specification process of collaborative profiles: Do the users understand that the profile will be set

up by an owner, who will assign jobs to several individuals? Do the users understand that these providers deliver information that refines the profiles given by the owner? Do the users understand that there are different kinds of jobs that can be assigned to providers? Do the users understand in what way these different jobs function?

The second research goal investigates the participants' ability to effectively specify correct collaborative profiles by using the collaborative alerting system editor CoastEd: The intuitiveness of the interface was analysed, as were its ease of use and satisfaction when working with it. An additional aim was to evaluate the physical and mental effort involved with using CoastEd. The subjects were asked how successfully they perceived they had completed the tasks and about their level of frustration using the software. The specified profiles were analysed for correctness.

The third goal was to analyse the collaborative language CGPDL: Is the collaborative extension of GPDL, CGPDL, suitable for the successful specification of collaborative profiles? Does the language pose any hurdles for the successful process of specification? If so, how could they be targeted?

8.2 Experimental Design

8.2.1 Method

The study was a laboratory study that followed an observational within-subjects approach. It was controlled by using randomisation. During the study participants were asked to think out loud.

8.2.2 Tasks

Condition 1. The subjects were given task descriptions in English. These asked them to use the program in order to specify parts of profiles and assign specification jobs of other parts of the profile to other people.

Condition 2. The subjects were shown profile specifications that have been produced with the help of the program. They were asked to complete these profile specifications through completing the jobs that had been assigned to them.

The tasks for each condition varied in complexity. Six different complexity levels of profiles were formed.

Profile Levels. Each of the levels contained a major concept that was tested. The six levels were (cf. Section 7.4):

- Vague condition: In the interface this class is called *condition job*.
- Vague duration: In the interface this class is called *duration job*.
- Vague timeframe: In the interface this class is called *timeframe job*
- Vague operator: In the interface this class is called *operator job*.
- Vague repetition: In the interface this class is called *repetition job*.
- Vague subtree: In the interface this class is called *open job*.

8.2.3 Procedure

A copy of the Bill of Rights was given to each participant. Two copies of the Research Consent Form were read and signed by the researcher and the participant. One copy was retained by the researcher, the other given to the participant. At the beginning of each interview the researcher verbally explained these documents, with particular reference to the participant's right to withdraw from the study at any point without explanation.

Each participant was guided through two phases of a tutorial and subsequent training that introduced them to the profile specification approach. The first phase presented some introductory tasks concerning the single-user profile specification approach and the second phase practised tasks of condition 1 and condition 2.

This was followed by an evaluation phase, during which the participants were asked to perform tasks in both conditions. The participants first performed tasks in one condition, and on the completion of all those tasks, performed tasks in the second condition. The order of the two conditions was alternated.

Tutorial Single-user Profiles. The tutorial began with an example of the domain data that was used in the study. After this introduction, the participants were shown 13 examples of single-user profile specifications and an English language scenario describing this particular profile specification. The researcher talked the subjects through those examples and gave as much information as required. The examples belong to levels of increasing difficulty and complexity and were presented in this order.

Training Single-user Profiles. The training phase entailed similar tasks as presented in the tutorial. There were eight tasks. All participants were given the same tasks. The tasks were presented with increasing difficulty and complexity to enable the subjects to learn GPDL and the use of the basic functionalities of the interface step by step. During the training the participants were asked to think aloud.

Tutorial Collaborative Profiles. The tutorial started off with an example of the domain data that was used in this phase of the study. After this introduction, the participants were shown six examples of collaborative profiles and an English language scenario describing this particular profile. The researcher talked the subjects through those examples and gave as much information as required. The examples belong to different complexity levels.

Training Collaborative Profiles. The training phase entailed similar tasks as presented in the tutorial. The subject was presented with tasks following condition 1 and 2. Of each group there were six tasks. All participants were given the same tasks. The tasks were presented in an analogous order to the tutorial. During the training the participants were asked to think aloud.

Evaluation. The evaluation consisted of two sets of tasks; each set represented one of the conditions and contained six tasks. The order of tasks and the order of sets were controlled. All participants were presented with the same tasks. The tasks in each set belonged to the different profiles levels. During the evaluation the participant was asked to think aloud.

Conclusion. The study was concluded with a questionnaire collecting information about the background of the participant and a short interview about the experience of the participant with the software.

8.2.4 Data Capture

A questionnaire was used to collect background information. While the participants were completing the given tasks and thought out aloud the observer took notes of the issues that involved problems. The researcher also used the task solving notes for the evaluation of the study and saved the results of the profile specifications as files. For each session an audio track and a video of the participants were recorded. The screen was also captured on video.

8.2.5 Participants

The pilot study was conducted with four participants and the full study with fourteen.

Two groups of participants were recruited: those that have a background of computer science and those who do not. The purpose of this was to be able to compare any potential differences in the use of the software between these two groups. For the full-scale study groups of seven participants for each group were recruited. Three of the subjects were from computer graphics design. Due to their lack of formal training in logics, they were counted as non-computer scientists.

The full-scale study was balanced in regards to the gender of the subjects (seven of each gender). Their age ranged from 20 to 41 years (mean 27 years). All of the non-computer scientists were native speakers, however only one of the computer scientists was a native speaker.

All participants use computers on a daily basis (6-point scale, everyone selected 6). The computer scientists were all very experienced in using computers and the non-computer scientists vary between moderately experienced and very experienced (see Figure 8-1, 5-point scale).

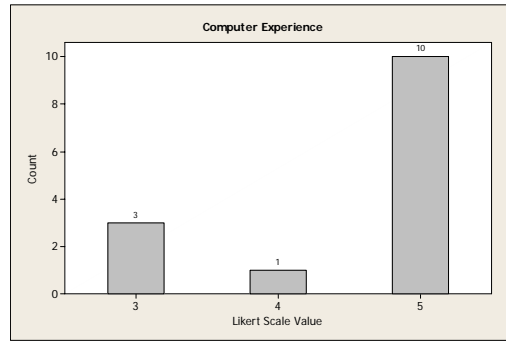


Figure 8-1: Computer experience.

The participants rated their query experience on a 5-point scale and their query frequency on a 6-point scale. So they would often use queries and be relatively experienced in it (see Figure 8-2 and Figure 8-3).

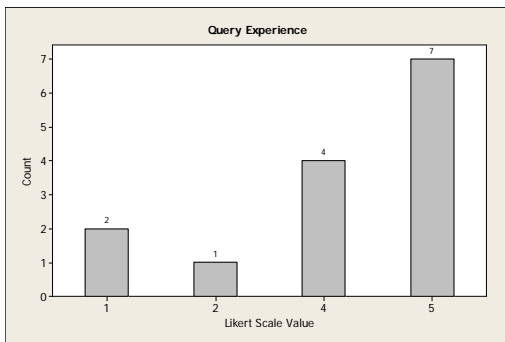


Figure 8-2: Query experience.

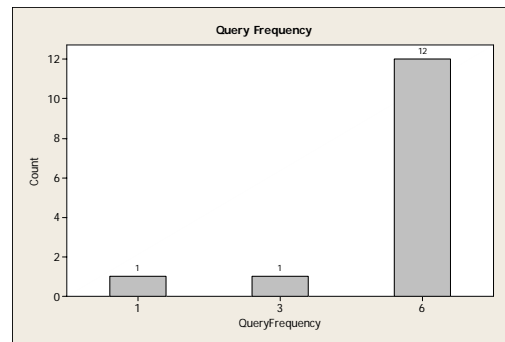


Figure 8-3: Query frequency

Their alerting experience they rated on a 5-point scale and their alerting frequency on a 6-point scale. They were moderately experienced and would use alerts occasionally (see Figure 8-4 and Figure 8-5).

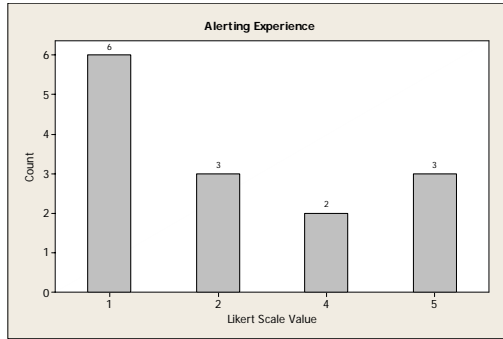


Figure 8-4: Alerting experience.

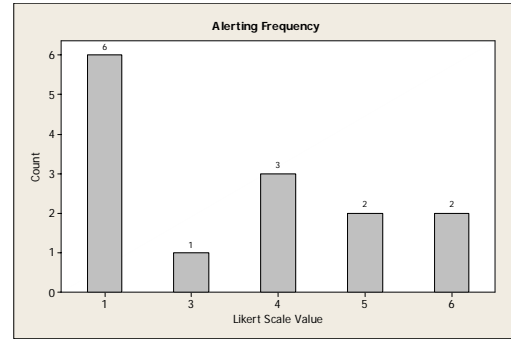


Figure 8-5: Alerting frequency

8.2.6 Pilot Study

Before undertaking the full study a pilot study was conducted that identified errors in the experimental design and tested the experimental procedure.

Observations. The observations made covered some aspects of the study design and a range of issues regarding visualisation and layout of several of the components used in the program. Other observations regarded problems with the interaction with some of the program components.

Presentation and Visualisation. The area with the most problems identified is the representation of information—in some cases visual but mostly textual.

The range indicator in the *edit duration* window (white boxes under the sliders in Figure 8-6) was mentioned as confusing. Its purpose is to display a range for the *assign duration* window. However, in its analogous presentation for the *edit duration* window it only shows a single line rather than a time range. The meaning of this single line remained hidden to the participants.

The graphical *open operator* representation entailed a more severe issue that, misled some of the subjects. The abstract symbol that represents an *open operator* on the workspace contains little boxes for symbolising in an abstract way all operators that can be initially chosen when defining an open operator. However, some of the sub-

jects thought that the mini boxes actually represent editable condition boxes and therefore tried to click into them.

The representation of the operands for the *sequence* did not seem to be suitable. In the provider view for the *open operator*, the participants did not understand which operand was supposed to be the first and which the subsequent ones.

Also, it was challenging for some subjects to tell the *operator job* and the *open job* apart.

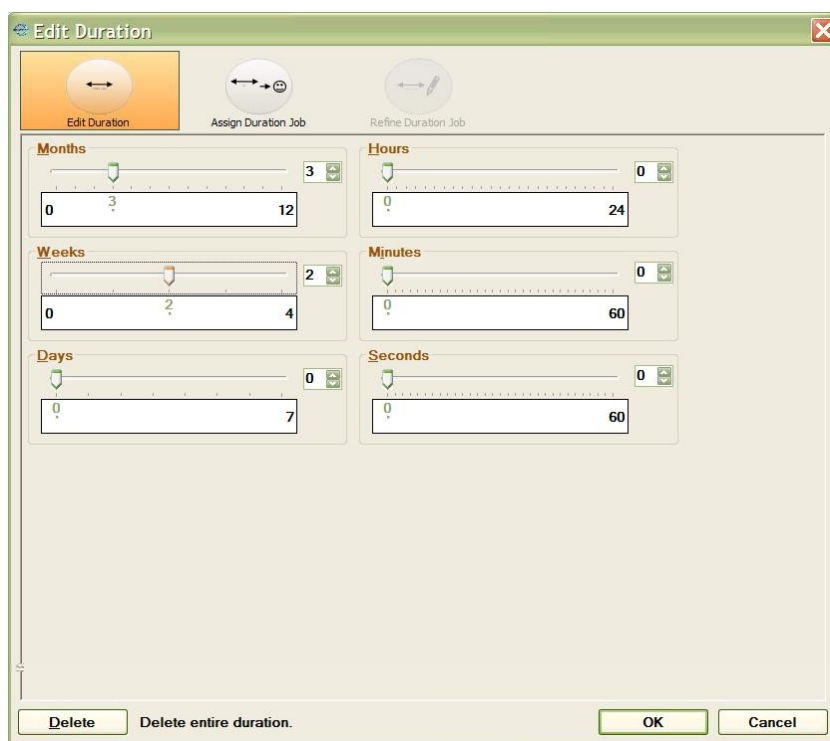


Figure 8-6: Duration pop-up window.

The differentiation of *edit condition*, *assign condition job* and *assign open job* appeared to be difficult for most of the participants. They did not seem to manage to interpret the given symbols as they first ignored most options. When asked to explore the interface further, they had to actually look at the contents of the tabs rather than recognising the intended meaning of the symbols used.

Layout. The major layout issue that was identified concerns the edit window for the duration (Figure 8-6). The layout of the window caused confusion among the participants.

Interaction Mechanisms. Even after changes from the single-user profile study (Chapter 6), some of the users still had navigation issues when dealing with the *no-occurrence* operator. One user suggested an option with the aid of which the entire component could be selected, which in abstract terms means to go back to the root level of the expression.

Timeframe. Two participants had problems with the *assign timeframe* tab. The timeframe consists of several individual time periods that can be assigned for specification to another person. Each of these time periods have the option to be exactly defined, or to be restricted by a range out of which the other person can choose the exact time period. These two options were overlooked by most of the participants when for one or several of the time periods the users did not have to define a range but wanted to leave it open.

Success Rate and Likert Scale Responses. The overall success rate of completing the tasks correctly was relatively high, that is, all tasks on the provider side were entirely correct. However, the owner part showed a lower success rate.

The Likert scale responses regarding intuitiveness, ease, satisfaction, mental activity, physical activity, frustration and success mostly were on the positive side. In comparison to the responses for the pilot study of the single-user profiles (cf. Section 6.2.6), the responses did reflect on the higher complexity of the collaborative approach.

Participants' Comments. The comments given by the participants of the pilot study identified some minor issues and showed that, while most concepts and interaction mechanisms that were used worked well, the *open job* required quite an amount of improvement.

Changes Needed. Based on the observations and questionnaire results, suitable changes to target the problems were identified. The following changes were considered as relevant:

Presentation, Visualisation and Layout. The visualisation of the range indicator for the duration was analysed and it was deduced that the comment made by the participant had not revealed the real underlying problem of the *edit duration* window. It was concluded that the underlying problem partially was caused by the layout of the window: the information was spread too much over the screen and moreover, typical conventions of the presentation of time were disregarded. This was changed. The range indicator itself was retained; however, the indicator was increased in size. To avoid the confusion that some of the subjects were experiencing, it was decided that the range indicators would be summarised into one single time indicator. Depending on what the largest time unit is that would be used, this would be the unit displayed in that time indicator (for example, month in Figure 8-8). The visualisation would be scaled accordingly. Figure 8-7 displays the window that was used in the pilot study. Figure 8-8 presents the new design.

The pilot study identified two problems that had a major impact on the usability of the program. They concerned distinguishing the interaction possibilities of the open operator and the single condition. Their presentation was not representative of their underlying functionality.

To address the differentiation problems between *open operator* and *open job*, and the differentiation problems between *edit condition*, *assign condition job* and *assign open job*, it was decided to change several of their symbols and names. The aim of this was to give each of them a more individual representation that would resemble more clearly the underlying concepts.

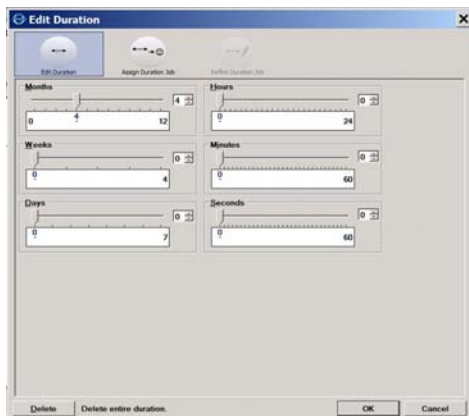


Figure 8-7: Edit duration window, pilot study.

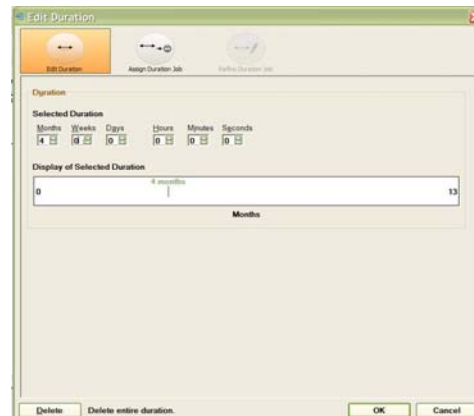


Figure 8-8: Edit duration window, full study.

The *open job* represents the underlying concept of the vague subtree, that is, the provider can extend this part of the profile by any combination of conditions and operators. The *open operator* (*new operator job*) represents the underlying concept of the vague operator, that is, the provider can choose by what operator they wish to combine the given operands. Figure 8-9 shows the old representation of the vague operator which is using a question mark as symbol—to represent that the profile owner does not exactly know which operator to choose—and the name *open operator*. Figure 8-10 shows the new symbol for the vague operator, that is, the conglomeration of the *conjunction*, *disjunction* and *sequence* symbol. It uses a new name *operator job*. This new name is aiming at increasing the difference to the name *open job* that was retained to denote the vague subtree.



Figure 8-9: Vague operator versus vague subtree realisation, pilot study.



Figure 8-10: Vague operator versus vague subtree realisation, improvements based on pilot study findings.

Furthermore, *assign open job* was given a new symbol that looks more different from the symbol used for *assign condition job* (cf. Figure 8-9 and Figure 8-10).

When refining an *operator job*, it was decided to use a new representation for the operands of the *sequence* that enables the free manipulation of their order.

Interaction Mechanisms. To target the navigation problems that were encountered when dealing with unary operators, especially with *no-occurrence*, based on a suggestion by a participant, an option for navigating to the root level of each expression was included. Additionally, an option of navigating one level up was included (cf. Section 7.5.1).

Timeframe. The *assign timeframe* window used in the pilot study (see Figure 8-11) was altered in a way that enforces the user to make a choice between three different ways of specifying the respective timeframe data: They can leave the date unspecified, specify it exactly or they can assign a range to the provider. This enforcement is caused by an additional option that describes the unspecified state of a date, for example “no date specified”. This option is the default and as long as it is selected, the respective data input elements are inactive. They are activated only if one of the other two options is selected. These new options are detailed in Figure 8-12.

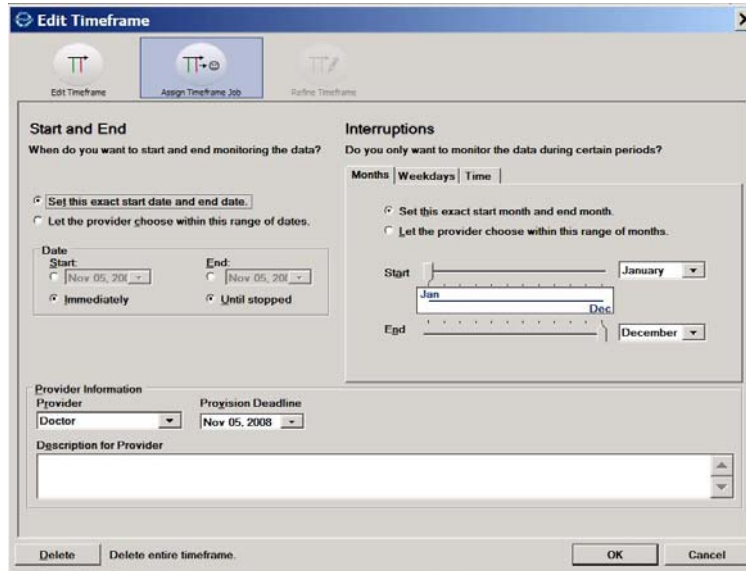


Figure 8-11: Assign timeframe window, pilot study.

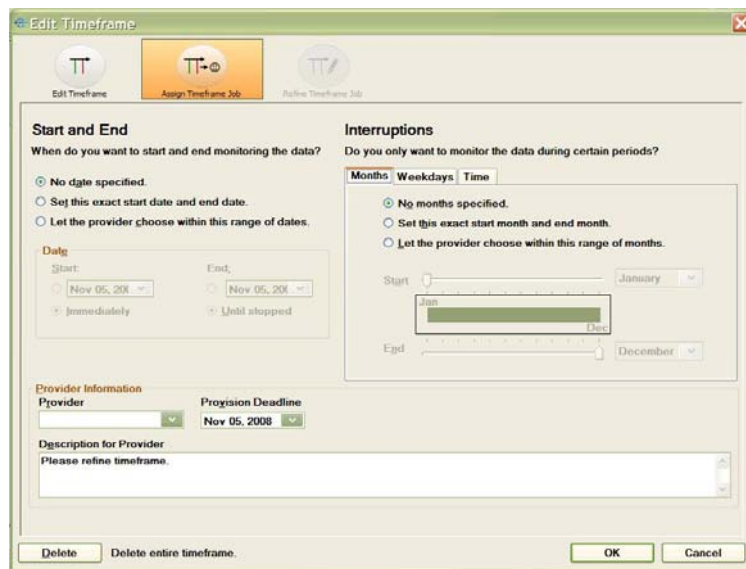


Figure 8-12: Assign timeframe window, full study.

8.3 Quantitative Findings

This section presents the quantitative findings of the full study.

8.3.1 Accuracy

The overall success rate was analysed. Additionally, the individual task levels and the influence of categorical variables on the success rates were examined.

Overall Success Rate. The overall success rates, for profile owners as well as information providers, were calculated.

Total. In total this amounts to 141/168 (84%) correctly solved tasks.

Owner. For 60/84 (71%) owner tasks the profile specifications were correct.

Provider. For the provider part 81/84 (96%) tasks had been correctly solved.

Task Levels and Individual Tasks. The test procedure tested for each task level one task from the owner perspective and one task from the provider perspective. For a detailed overview of the success rates refer to Table 8.1.

Table 8.1: Individual and task level success rates.

Level	User	Concept	Correct	Per Cent
1	Owner	vague condition	11/14	~79%
	Provider		14/14	100%
2	Owner	vague duration	10/14	~71%
	Provider		14/14	100%
3	Owner	vague timeframe	14/14	100%
	Provider		14/14	100%
4	Owner	vague operator	8/14	~57%
	Provider		14/14	100%
5	Owner	vague repetition	11/14	~79%
	Provider		14/14	100%
6	Owner	vague subtree	6/14	~43%
	Provider		11/14	~79%

As shown in the overviews given in Figure 8-13 and Figure 8-14, the success rates for the provider tasks are much higher than for the owner tasks. For all but one level the provider success rates were 100%. Level 6 had lower success rates for both owner and provider tasks. Level 4 shows a drop for the owner task success rate.

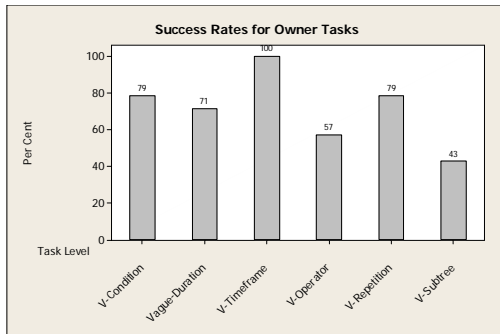


Figure 8-13: Success rates for owner tasks levels (V=vague).

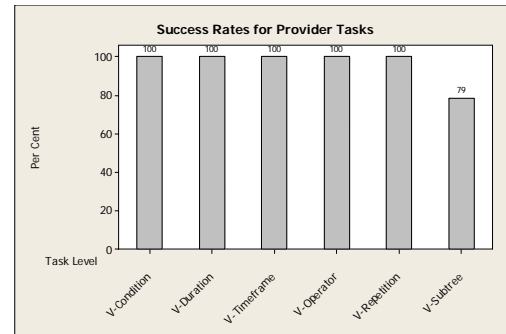


Figure 8-14: Success rates for provider tasks levels (V=vague).

Influence of Categorical Variables on the Success Rate. Only one of the demographical variables gathered, the gender, had an influence on the success rate.

Apart from that, some of the Likert-scale responses correlated with the success rate—the intuitiveness, ease and satisfaction, as well as the physical activity, and the subjective success.

Gender. The gender of the participants appeared to make a difference to the result. As shown in Figure 8-15, females generally showed higher success rates than males. This holds for the total results and for all task levels except level 3 and 4. For level 3 the results were equal for both genders, that is, all participants were able to solve all tasks of that group correctly.

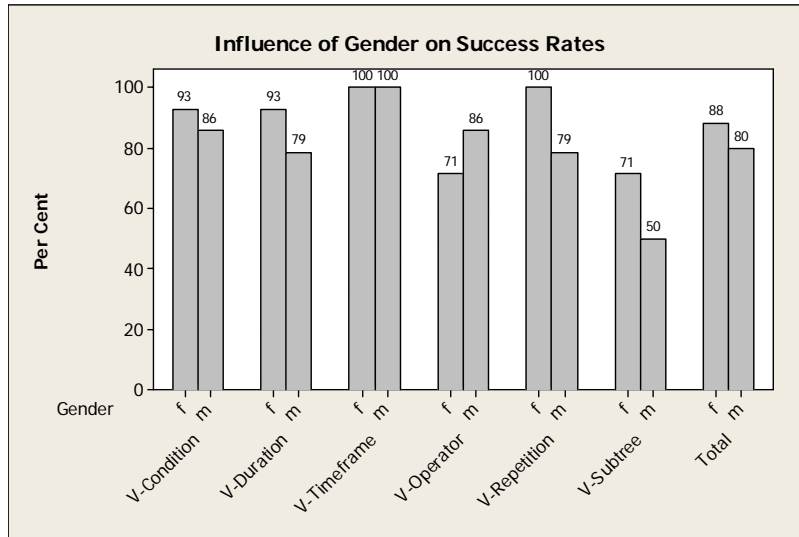


Figure 8-15: Influence of gender on success rate.

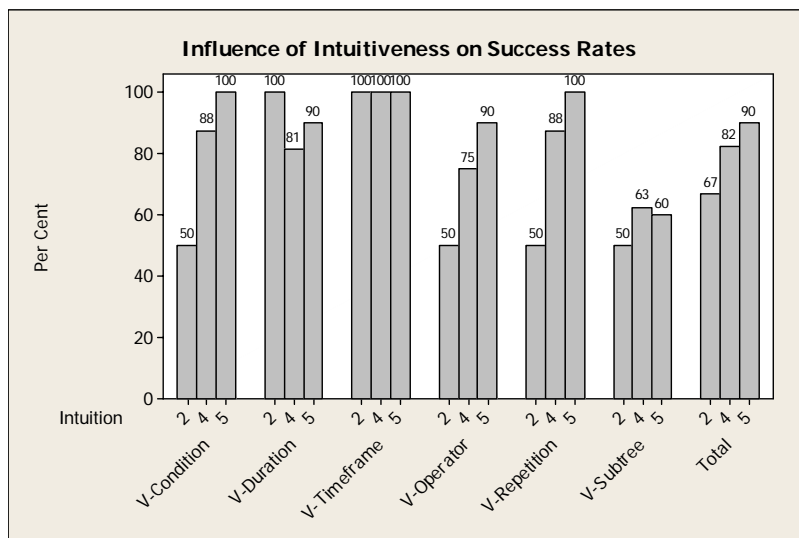


Figure 8-16: Influence of intuitiveness on success rate.

Intuitiveness Ease and Satisfaction. The more intuitive, the easier and the more satisfactory the participants rated the interface, the higher their success rate was found to be and vice versa (cf. Figure 8-16, Figure 8-17 and Figure 8-18).

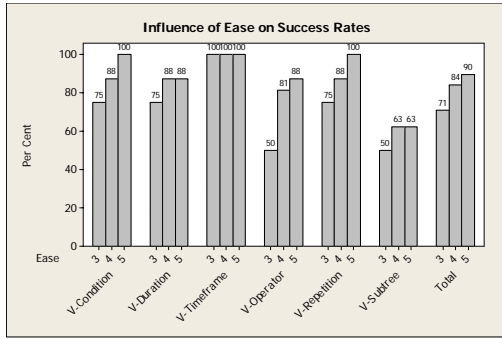


Figure 8-17: Influence of ease on success rate.

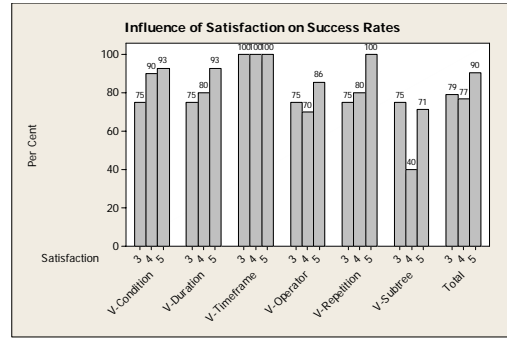


Figure 8-18: Influence of satisfaction on success rate.

Physical Activity. The higher a participant rated the physical activity required to specify a profile, the higher their success rate could be measured. Details are shown in Figure 8-19.

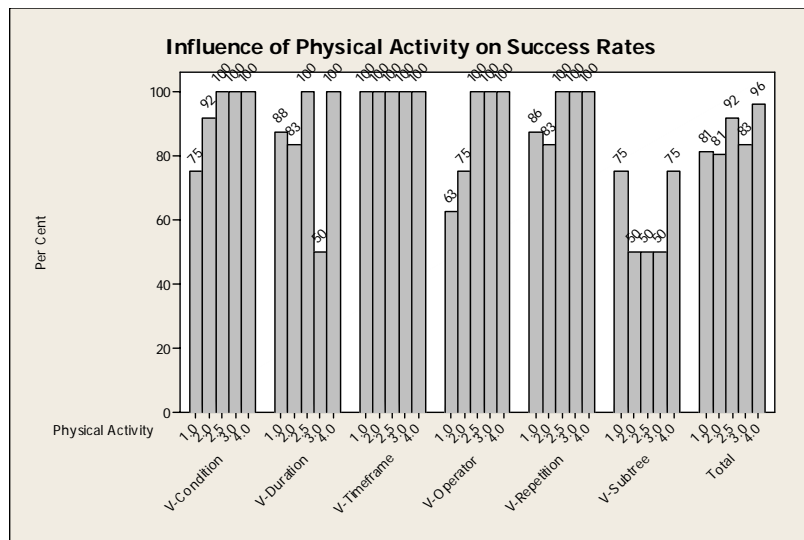


Figure 8-19: Influence of physical activity on success rate.

Subjective Success. Figure 8-20 indicates that the perceived subjective success approximately correlated to their actual success rates.

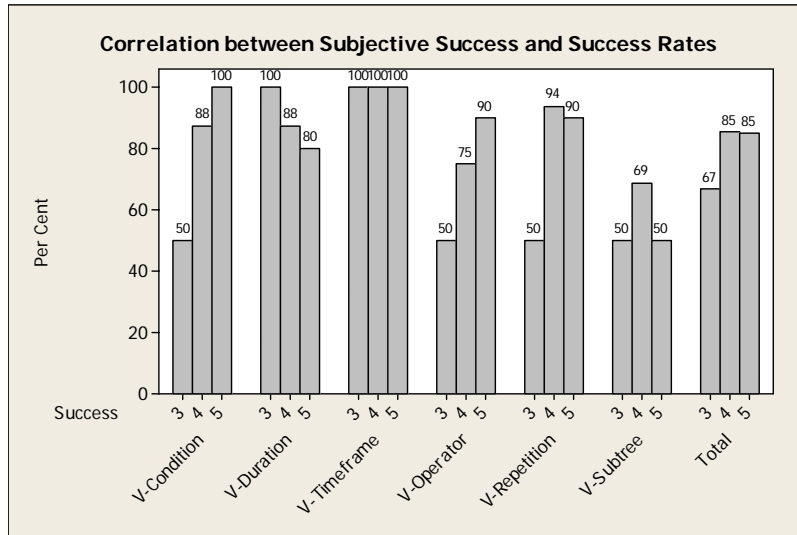


Figure 8-20: Correlation between perceived subjective success and actual success rate.

8.4 Qualitative Findings

This section presents the qualitative findings of the full study.

8.4.1 Likert Responses

All parameters were measured on a 5-point Likert scale.

General Distribution. The questionnaire gathered information from the participants regarding the intuitiveness, ease, satisfaction, mental activity, physical activity, frustration, their subjective success and background data regarding the computer-literacy of the participants.

Intuitiveness. All but one participant rated the interface to be intuitive or very intuitive. This is represented by the mean of ~ 4.2 (cf. Table 8.2).

Table 8.2: Results for intuitiveness, (N=14).

Intuition	Count	Percent
2	1	7.14%
4	8	57.14%
5	5	35.71%

Ease. 85.71% of the participants chose a category in the positive range of the scale, that is, they rated the interface as easy to use (cf. Table 8.3).

Table 8.3: Results for ease, (N=14).

Ease	Count	Percent
3	2	14.29%
4	8	57.14%
5	4	28.57%

Satisfaction. Table 8.4 gives an overview of how satisfying the users rated using the interface. With a proportion of 85.71% the result lies between a rating of satisfying and very satisfying. No participant rated using the interface as not being satisfying.

Table 8.4: Results for satisfaction, (N=14).

Satisfaction	Count	Percent
3	2	14.29%
4	5	35.71%
5	7	50.00%

Mental Activity. Overall, the participants were neither inclined to rate the interface as requiring a lot or little mental activity. Details are shown in Table 8.5.

Physical Activity. The physical activity the users had to invest in order to specify profiles was rated as not very high (cf. Table 8.6).

Table 8.5: Results for mental activity, (N=14).

Mental Activity	Count	Percent
2	5	35.71%
3	5	35.71%
4	3	21.43%
5	1	7.14%

Table 8.6: Results for physical activity, (N=14).

Physical Activity	Count	Percent
1	4	28.57%
2	6	42.86%
2.5	1	7.14%
3	1	7.14%
4	2	14.29%

Frustration. Table 8.7 details the frustration level experienced by the subjects when using the interface. With a proportion of 92.86% it was very low. In addition, the only participant that rated their frustration level as 3, the highest given value, added a comment on the questionnaire that the frustration level was due to tiredness. This could also be observed during the study.

Table 8.7: Result for frustration, (N=14).

Frustration	Count	Percent
1	7	50.00
2	6	42.86
3	1	7.14

Subjective Success. Table 8.8 shows how the subjects rated their perceived success in correctly completing the given profile specification tasks. With a proportion of 92.85% in the positive range of the scale, they rated themselves as being successful. None of the participants thought themselves to be unsuccessful.

Table 8.8: Results for subjective success, (N=14).

Success	Count	Percent
3	1	7.14
4	8	57.14
5	5	35.71

Influence of Categorical Variables on Likert Responses. The study analysed the influence of several categorical variables on the replies given on the Likert scales. The analysed variables that were found to have an influence were: mother tongue, gender, profession, query experience, query frequency, computer experience and having taken part in one of the previous studies.

Native Speaker. The influence of the subjects' mother tongue was analysed. It has to be noted that coincidentally this parameter is locked with the parameter of profession as the non-computer scientists were all native speakers, whereas the computer scientists almost were all non-native speakers.

It appeared that non-native speakers found the interface more intuitive and more satisfying to use than native speakers (see Figure 8-21 and Figure 8-22).

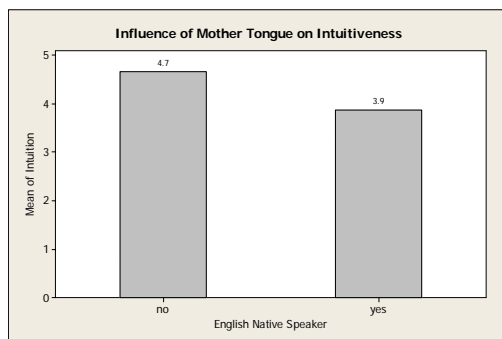


Figure 8-21: Influence of mother tongue on intuitiveness.

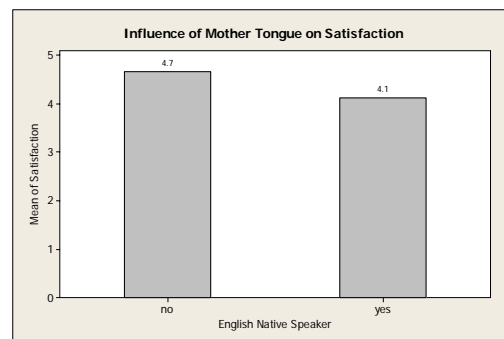


Figure 8-22: Influence of mother tongue on satisfaction.

Moreover, the non-native speakers rated that using the interface required less mental activity and was less frustrating than the native speakers stated (see Figure 8-23 and Figure 8-24).

It should be noted that several participants from the non-native speakers were from an Asian background. The cultural influences of that have to be taken into consideration, that is, they might have held back criticism.

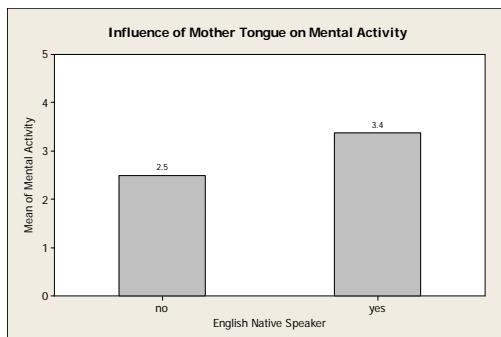


Figure 8-23: Influence of mother tongue on mental activity.

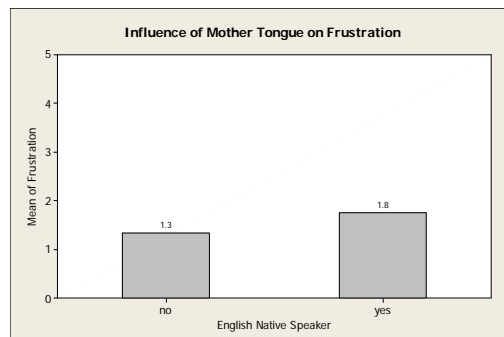


Figure 8-24: Influence of mother tongue on frustration.

Gender. The gender appeared to have an influence on the perceived level of physical activity that was required in order to use the interface and the experienced level of frustration (see Figure 8-25 and Figure 8-26). Both parameters were rated higher by men than by women.

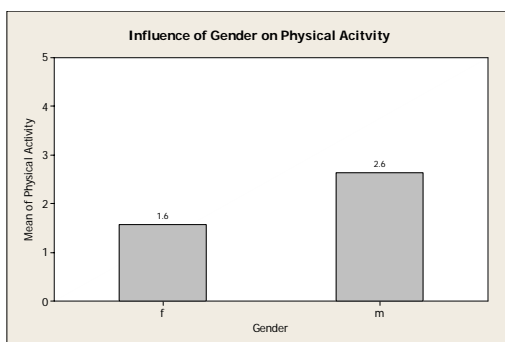


Figure 8-25: Influence of gender on level of physical activity.

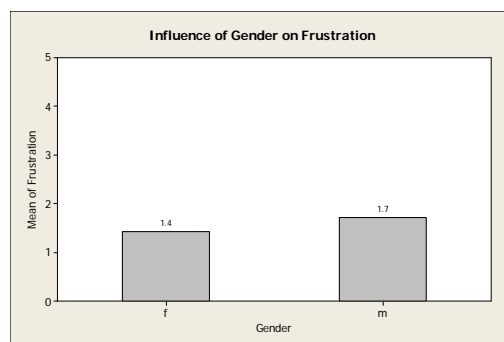


Figure 8-26: Influence of gender on frustration.

Profession. The profession of the participants showed an influence on several variables. Computer-scientists rated using the interface as more intuitive and more satisfying than non-computer scientists (see Figure 8-27 and Figure 8-28).

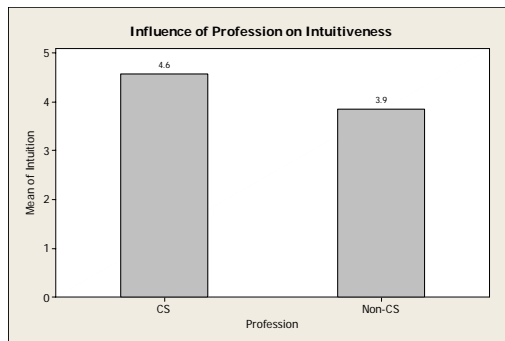


Figure 8-27: Influence of profession on intuitiveness.

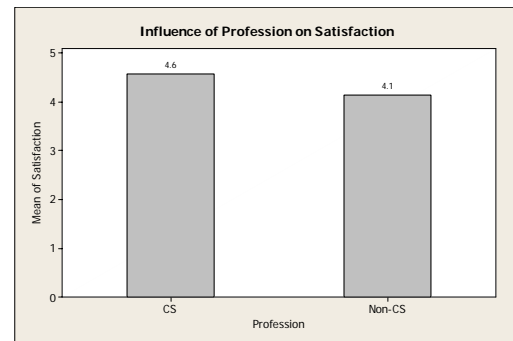


Figure 8-28: Influence of profession on satisfaction.

The mental activity required to specify profiles with the program was rated lower by computer scientists than by non-computer scientists, whereas they rated the physical activity required as higher than was done by the non-computer scientists. Details are shown in Figure 8-29 and Figure 8-30.

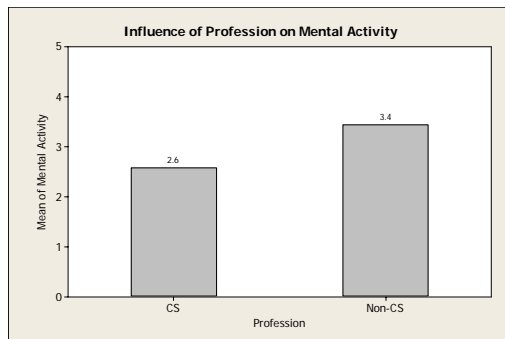


Figure 8-29: Influence of profession on mental activity.

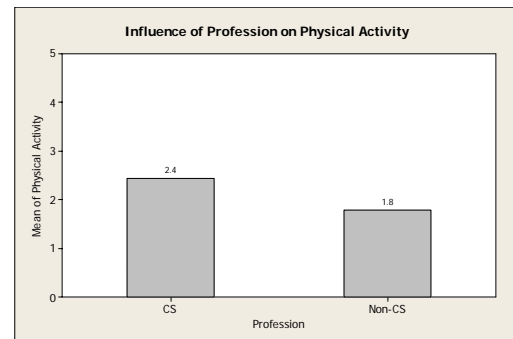


Figure 8-30: Influence of profession on physical activity.

Query Experience. The subject's query experience clearly showed an influence on their rating of the required mental and physical activity (see Figure 8-31 and Figure 8-32). The more experienced the subjects were the smaller they rated the required amount of mental activity and the higher the required amount of physical activity.

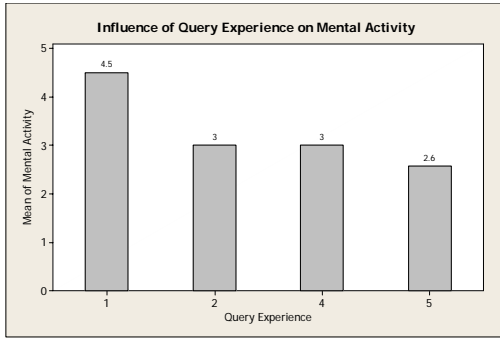


Figure 8-31: Influence of query experience on mental activity.

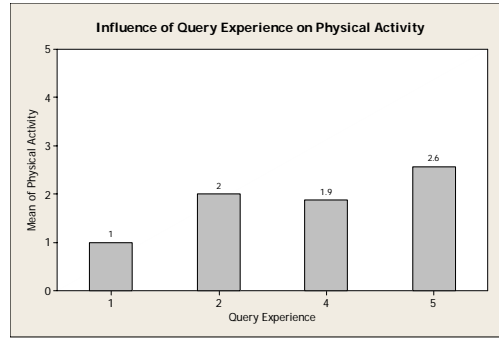


Figure 8-32: Influence of query experience on physical activity.

The participants also rated their subjective success as slightly higher if they had more experience with posing queries. For details refer to Figure 8-33.

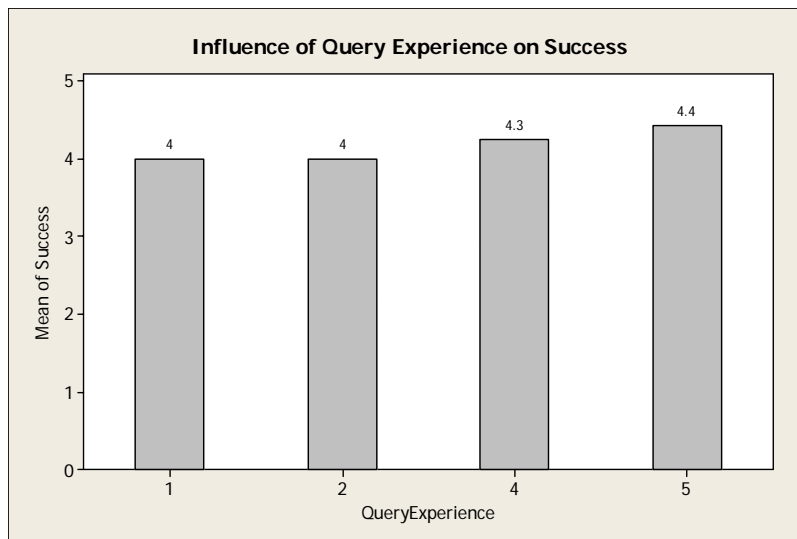


Figure 8-33: Influence of query experience on subjective success.

Query Frequency. Regarding the participants' query frequency it could also be identified that the more often the participants would be using queries, the smaller they rated the required mental activity and the higher the required physical activity (see Figure 8-34).

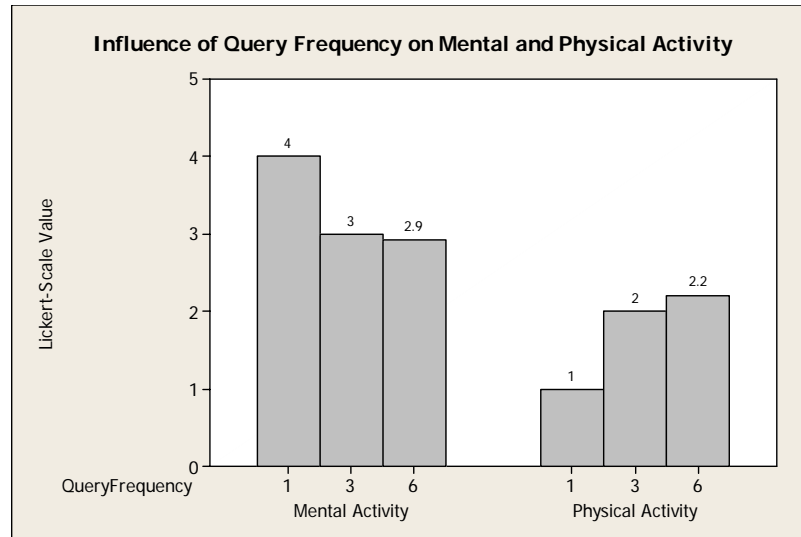


Figure 8-34: Influence of query frequency on mental and physical activity.

Computer Experience. The only variables that were influenced by the computer experience of the participants were the mental and physical activity that were required in order to specify profiles, as shown in Figure 8-35. The more experienced the participants were, the smaller they rated the mental activity and the higher the physical activity.

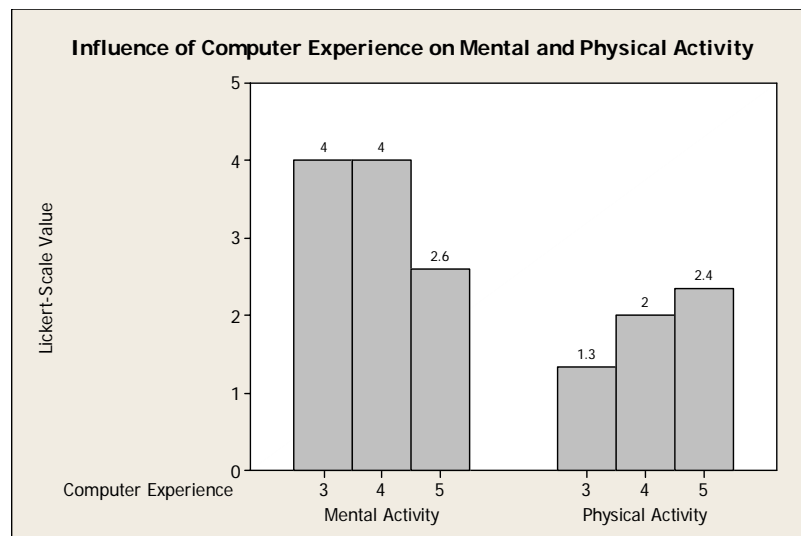


Figure 8-35: Influence of computer experience on mental and physical activity.

Previous Study. It was analysed whether having participated in either of the previous studies had an influence on the given feedback. Participants that had already participated in either the paper-based study or the usability study regarding single-user profiles, rated the intuitiveness, ease and satisfaction in using the program slightly lower than participants that had not participated in one of the previous studies (see Figure 8-36). With a mean of 3 versus a mean of ~1.8 they also rated the required physical activity as higher.

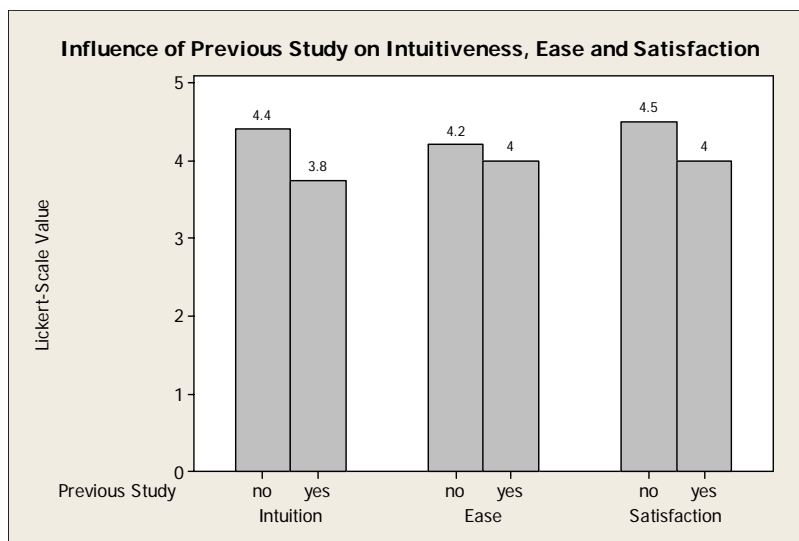


Figure 8-36: Influence of having participated in a previous study on intuitiveness, ease and satisfaction.

Moreover, participants that had already taken part in a previous study rated the frustration as higher and judged their subjective success of specifying profiles as lower (see Figure 8-37).

8.4.2 Observations

The observations made during the study and the deductions made from evaluating the kind of mistakes the subjects made, are grouped by profile level. Additionally, some general observations were made.

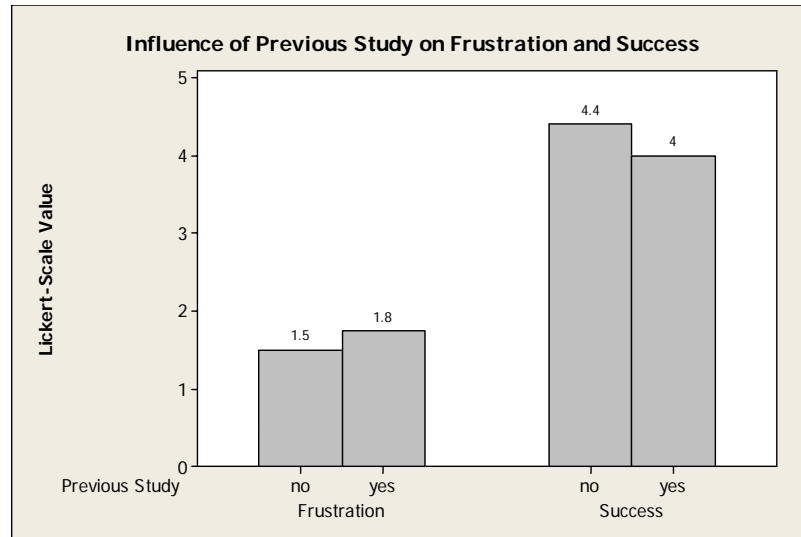


Figure 8-37: Influence of having participated in a previous study on frustration and subjective success.

Vague Condition (1). The differentiation of *edit condition* and *assign condition* was somewhat challenging for the participants. They were unaware that there are several tabs in the window that have to be chosen according to the intended degree of specification of the profile. They were sometimes unaware of the fact that there is a difference between a vague specification and an exact specification.

Vague Duration (2). A couple of subjects had to make themselves aware of the differentiation between timeframe and duration: The task descriptions for the vague duration contained a range that was given by its start and end value. Due to these two given values, these two participants assumed that they were dealing with a timeframe rather than a duration, as the exact duration is defined by a single value whereas the exact timeframe is given by a start and end value.

Vague Timeframe (3). A couple of participants had difficulties with the representation of the specification mode. For each timeframe limit—that is, start and end, months, weekday and time—there is a separate set of three radio buttons, that let the user choose between three modes: Not specifying that particular value at all, specifying the exact values or giving a range. One participant had trouble realising that there

was a different set of buttons for each different timeframe limit. Another one at first entirely overlooked the buttons. However, without choosing any of the buttons no value can be entered into the form.

Vague Operator (4). The biggest problem with the vague operator was the creation of operands.

Operand Creation. While most of the participants were fine with selecting potential operators, they generally had problems grasping the process of operand creation. They did not understand that there are two possible ways of creating new operands: top down—first create the operator and then the operands—and bottom up—first create the operands, then the operator and then link the operands to that operator.

When going bottom-up, one subject was surprised that if one has already created a component and currently placed it on the workspace, it is taken on as operand for the vague-operator job. Another subject was wondering how to create the second operand even though it was already sitting in the component bar. Yet another participant was not sure how to create the operands at all; they wanted to have both operands on the workspace but realised that this was not possible, as their relative position to each other would be defined by the operator; however, the point of the *operator job* is to leave the operator undefined by specifying a set of candidate operators.

One problem was the *Create* and *OK* button in the vague-operator window (*edit operator job*). Several participants selected the number of operands they wanted to create but then clicked the *OK* button rather than the *Create* button. As a result only the candidate operators were selected but not the operands were created.

All of the mistakes that were made in this level were problems with the operands rather than with the operators.

Component Bar. The interaction possibilities with the component bar confused a number of subjects.

One person was confused by the effect of them dragging down a component from the component bar to the workspace. The participant was wondering where the other component had gone that previously was on the workspace. After some playing around with the interface they figured out, though, that the two components had been exchanged.

Vague Repetition (5). There were no major issues observed with this profile level.

Vague Subtree (6). Along with the vague operator, the vague subtree caused the most problems.

Task. In order to be able to solve the given tasks, participants first had to grasp what they were expected to do. For the level of the vague subtree this was a major challenge. In the analysis of the observations and in judging the correctness of the given profile definitions, it was difficult to factor out the subjects' difficulty in grasping the task descriptions.

Location and Presentation of Vague Subtree. One of the main challenges regarding this concept was the localisation of the control for this concept. The tab for the vague subtree is located in the *edit box* window (cf. Figure 8-10), as it is a box that can be extended to represent a vague subtree. However, this seems to be outside of the expectations of the participants. They appear to expect a button that explicitly spells out that operator. A small group of subjects at first chose the vague operator rather than the vague subtree. However, they figured out that this operator could not be used to express the entire flexibility of the concept.

Refining the Vague Subtree. The interaction mechanisms for the completion of the vague subtree were identified as not fully working. Therefore it was a challenge for some of the participants to complete some of the vague-subtree profiles, as not both mechanisms were supported—bottom-up and top-down.

The reason identified for this problem was that the component bar could not be used because of privileges that only allow profile extensions within vague subtrees rather than extending the profile by first creating new components and then integrating them into the profile that has to be extended.

Another reason that could be found was that in order to complete the vague subtree, buttons on the main interface have to be used rather than the specification being done in a pop-up window. Almost all participants would click on the box that represented the vague subtree in order to complete it. Then they would realise that this only opens a comment window and hesitated for a moment pondering what to do. One participant pointed out that the instructions, “Use buttons on main interface for refinement”, given on the box when active, did not give any hint as to their meaning.

Other. The second group of observations that were gathered cover a wide range of issues.

Embedding and Node Exchange. As described in Section 5.2.1, in the design phase the decision was made to offer the most important operations for a given situation only, in order not to confuse participants. While this worked for the situation evaluated in the GPDL-UI study, for the more complex tasks of the collaborative profiles, especially for tasks of the vague-subtree level, this was problematic. There were several situations when the subjects wanted to replace one operator by another, that is, they wanted to use a node exchange operation, but were offered an embedding operation instead. They did not realise that these were limitations of the chosen program approach (that at this instance only supported the top-down approach) rather than their inability to use the interface.

Expression Nesting. To a lesser extent than in the two previous studies, it was observed that some of the subjects got lost in the nesting of the profile expression. However, the conductor of the study realised that to a great extent this was caused by the method used to teach the participants the underlying concept. Once a metaphor of

card board boxes that are boxed into each other was given, the performance of the participants dramatically improved.

Before that some people were confused about or even unaware of the fact that there were nesting levels. Furthermore, they had trouble using operations that were not active when they were on a different level. This for example concerned the duration parameter or adding further operands to the *sequence*.

The two buttons that create a short-cut to the top level of an expression and to the next level up were used by one participant only (cf. Section 7.5.1). This participant started using them at the very end of the study after they enquired about such a short-cut and it was pointed out to them. Either the symbols and text (“Navigate to Top Level” and “Navigate out One Level”) used were not representative of their meaning or their position was inappropriate.

Component Bar. One of the bigger problems of the program was the component bar. The function of the bar was not intuitive to the users. This study was the first study that required the use of the component bar due to the complexity of the tasks. So there were not yet any improvements made to the component bar as result of previous studies.

The following comments were made regarding possible improvements:

- Would like to fully see components in the component bar—not only a tooltip with the content of the condition
- Give the component bar a different name—maybe call it sketch space instead
- Confused by shuffling in the component bar and also by moving up of components

However, there were also participants that after an initial explanation extensively used the component bar.

Terminology. One of the greatest challenges for the participants seemed to be the terminology chosen. The developer had found terms other than those used internally in the program structure and those used in the design phase. However, the new terms still did not seem to speak the user's language. One participant started to discuss the terminology out of their own accord. Examples for this include:

- The participant points out that in the area of computer science the word “job” already has a different meaning (for example batch job or cron job) and therefore it would be misleading. As alternative the participant suggests an expression such as “set of something”
- Use “save” instead of “submit”
- Instead of “assign job” use “ask someone else”
- Use “create task”, “give task” or “incomplete task” were also suggested for “assign job”
- In the “assign condition job” tab, to at least one subject it was unclear what the entry “all values” denotes in the context of restricting allowed values

Provider Part. Overall, it stood out dramatically that the provider part was very straightforward and quick to fill in. The only challenge was tasks of the vague-subtree level.

Learning. Several observations could be made concerning the learning process of the participants. It was obvious that undertaking the study required some learning from the participants. First of all, they had to learn analysing the tasks. Without that ability they were unable to even attempt to use the software.

While the participants in general were quick in picking up the concepts and interaction mechanisms concerning the specification of single-user profiles, it took them longer to understand the interaction mechanisms required for the specification of the collaborative profiles.

8.4.3 Participants' Comments

The comments that the participants gave in the final questionnaire were of two kinds, positive and critically constructive. Overall, they judged the program to be easy to use and—appreciating the complexity of the underlying subject matter—they said that with a small amount of training it would be well suited for its desired purpose. That is, it supports users in successfully specifying collaborative profiles in an easy way.

Affirmative Comments. The positive comments given can be subdivided into several groups. They mainly covered the overall usability of the program, its layout, its flexibility and the usefulness of the collaborative approach.

Overall Usability. The subjects judged the interface as straightforward and found it was easy to use.

“Seemed fairly easy and quick”

“Seemed really straightforward.”

“Easy to use.”

For this judgement, a few of them took into account that it takes a moment to understand the given task and realised that there is a difference between the given task complexity and the interface of the program. They acknowledged that in order to solve problems of the given complexity a small amount of learning would be involved.

“Simplistic & understandable (once you know the symantics [semantics—asked participant—they meant “once you know the syntax of the language”])”

“Once you get the task started and you are on the right track the instructions are clear and easy to understand.”

“It took me a few seconds to understand what I could & couldn't do, but figuring that out was easy and from then on it was straightforward.”

“I think it is very easy to use, especially after the different scenarios have been repeated a few times.”

Layout. The participants also commented that the interface was clearly laid out and well structured.

“Nice interface”

“Good page design. Anything is almost done in a single page, no paging [scrolling] is good.”

“Very clear structure [means clearly laid out interface layout]”

“[...] however it seems nicely laid out and easy to navigate.”

Moreover, they commented that they liked the big buttons and found the icons used representative of their intended purpose.

“I especially like the large buttons, the fact that the interface is not cluttered and the low level of complexity. (It’s not photoshop.)”

“nice big buttons”

“I like the icons, they are illustrative.”

Flexibility. They also commented on the flexibility of the program and pointed this aspect out especially.

“Very flexible.”

“In addition it’s very flexible”

One participant appreciated the adapting symbols of the buttons and pointed out that this was helpful for them in order to know what operations are possible at a given stage.

“Change according to what can be done at the moment (helpful)”

Collaboration Concept. Another group of comments covered the general concept of the program. The general concept was being pointed out as useful and especially the interaction that was required to supply the provider information to complete the collaborative profile was stated to be very easy to do. Additionally, one participant remarked that, whereas some of the tasks were difficult to grasp, the interface aided them in solving the task.

“Filling in the parts for the provider seemed really easy and user friendly, didn’t take too much time.”

“Collaborative information seems useful and collaborator information quick and easy to put in”

“There were some tasks I felt that I was more suited to completing, that is, it suited my way of thinking whereas other problems were more difficult to visualise before I began the task. The program did make it easier to develop a visual chart in my mind.”

Other. Some other affirmative comments were made regarding the good control of permissions, choice of colours and the purpose of the program.

“Well control of the permissions.”

“Good use of colours to guide”

“I can imagine it to be a very practical & extremely helpful tool in many situations where the user needs to be alerted of change.”

Critical Comments. The critical comments mainly regarded training required in order to use the interface, some of the terminology used, and navigation issues.

Training. While most participants pointed out that the interface was easy to use, some of them made a point of saying that it requires some initial training.

“Require training.”

“The interface seems like it would be useful with further training.”

“Generally easy to use with a small amount of training”

One of them pointed out that they were a bit frustrated because they could not remember all the things that they had learned.

“[Regards frustration:] just a touch—but none. Frustrated that ‘I’ couldn’t remember what I’d learnt the task before”

Terminology. One participant pointed out that the terminology could be improved. That participant made a strong point about this issue not only on the questionnaire but

in the conversation that was held while they did the study and especially while they filled in the questionnaire.

“Other words than ‘job’”

“Terminology sometimes difficult to understand [...]”

Navigation in Hierarchy. Despite the changes that were made after the studies regarding single-user profiles and the pilot for the collaborative profile study, there were still comments that concerned problems with the navigation within profile expressions. Some said that the navigation is sometimes confusing:

“[Regarding intuitiveness:] except I got confused between levels”

“Occasionally, it’s not so easy to find the right entry point.”

Another subject remarked that the navigation required quite a high amount of physical activity:

“Re. Q6 [physical activity]. I felt like I had to click a number of times to go from editing something at one level backwards, then back into a level in order to edit that. [Participant said he knew why that was and suggested shortcut using double-click directly above box that should be edited to go straight into that and double-click on white space in order to go to top-level.]”

8.5 Discussion

This section gives an overview about the lessons that could be learned from the study and discusses the results that led to these lessons. Furthermore, it sets the results in correlation to the research goals that were asked initially.

The lessons that could be learned are derived from the results of the full study. In particular there were four sources that were considered:

- The success rates of correctly completing the given tasks
- The replies given in the questionnaire
- The study observations made by the observer

- The comments given by the participants during or after the study

Recurrent patterns that made an impact on several or all of these parameters were determined.

The aim of this study was to evaluate the concept of collaborative profile specification and the usability of the prototypical editor CoastEd for that particular profile definition. In particular the study looked at three aspects:

- Whether users would be capable of understanding the concept of collaborative profile and the specification process involved with it
- Users' ability to effectively specify correct collaborative profiles using the developed prototype
- Whether CGPDL is suitable means for expressing collaborative profiles

The three aspects overall led to positive findings.

On the whole, the concept was understood and accepted by the participants of the study. They quickly picked up the general underlying collaboration idea that involved having an owner and at least one provider completing the profile. They rated the ease (~4.1/5), intuitiveness (intuitiveness ~4.2/5) and satisfaction (~4.4/5) of using CoastEd for specifying collaborative profiles positively. They rated the mental activity (3/5) that was involved as neutral despite the subject matter being a really complex one. The physical activity (~2.1/5) and frustration level (~1.6/5) were judged as low and the subjects found themselves to be successful in solving the given tasks (subjective success ~4.3/5).

Most profiles were created successfully. In total a mean of ~84% tasks was solved correctly. This is composed of a mean of ~71% for the owner and a mean of ~96% for the provider. With some improvements for the owner part of level 4—covering the vague operator—and for the entire level 6—covering the vague subtree—the de-

veloped software is useful for effectively and correctly specifying collaborative profiles.

The study clearly shows that CGPDL is a suitable language for the specification of collaborative profiles. All concepts that were required could be expressed using the language. No language concept had to be added or changed from using the language solely for the specification of single-user profiles to using it for specifying collaborative profiles.

Following are the major lessons learned.

8.5.1 Collaboration Concept

Regarding the central topic of this user study, collaborative profiles, two major findings were identified. One regards the general possibility of specifying profiles correctly in an easy way, and the other one concerns the differentiation between the exact specification of single-user profiles or parts of collaborative profiles and leaving parts of a profile unspecified.

Owner versus Provider. All information sources clearly led to the same conclusion—the completion of vague profiles by the provider was significantly easier than starting up a new profile as owner. For example this was shown by the success rates for the owner ~71% (mean) versus those for the provider ~96% (mean). This is not surprising as the owner has to fully analyse the given task and mentally subdivide into its sub problems, whereas the provider only has to look at a fraction of the original problem.

Challenge of Differentiating between Exactness and Vagueness. One issue that had to be mentally grasped by the participants is the differentiation between the concepts of the exact definition of profiles or parts of them, and the vague definition of them. This is a three-step process—first the users have to understand the differentiation of these two concepts, then they have to analyse the given tasks for which concepts they are asked for, and finally they have to figure out the respective interac-

tion mechanisms and components in the software. The two first steps are mentally challenging and so the users do not have any capacities left to pay much attention to detail in the interface.

This led to errors in the specification when participants tried to use the tabs provided for the exact specification in order to specify a vague concept, for example they would use the *edit condition* tab instead of the *assign condition job* tab. Another example is the participants overlooking the radio buttons for the exact/vague specification of each of the separate timeframe limitations in the pilot study that were changed to a forced input as a consequence for the full study.

8.5.2 Collaborative Profile Levels

The study comprised test tasks for all six collaborative profile levels. Four out of these six turned out to be very successful. The levels covering the vague operator and the vague subtree, however, had some issues. A challenge for the subjects was to tell the underlying concepts of these two levels apart. The problems with both levels, the vague-operator level and the vague-subtree level, were reflected upon in the lower success rates these tasks yielded than the other four task levels.

Vague Operator. Perhaps the greatest challenge of the chosen approach is the impossibility to assign a location to the operands of the vague operator on the workspace. As the relative location of operands to each other is determined by which operator they are combined with, this becomes a catch 22. The owner does not select a specific operator, but leaves it open which one the provider can choose—or gives a set of candidate operators out of which the provider can choose. Thus, there is no way of placing the operands on the workspace. This was challenging for the participants to understand.

Next to the localisation of the operands, a second challenge regarding this concept was identified: the creation of operands. While the process was improved between the pilot and full study, there were still some problems that could not be solved: Most of the participants were unaware that they had two choices—they could either go about

the task top-down or bottom-up. This caused confusion as the same window has to be used for both approaches. Also, within each approach there remained some problems:

- The bottom-up approach entailed the problem that it links only the current component to the chosen operator set (as operand) rather than linking all components (that is, all from the component bar and the current one from the workspace). Once, they have realised that no other operand has been linked; the participants wonder how to link the second or any subsequent operands to the vague operator.
- The top-down approach has a different problem—in order to create new operands users have to select a number and then press the *Create* button (cf. Table 7.3). If they simply skip the *Create* button and go straight for the *OK* button then this will only create the operator set but omit the operand creation. This process is required when the window is used for the bottom-up approach. However, it startled the users.

Vague Subtree. For the owner part, it was difficult for the participants to tell this concept (*open job*) apart from the two condition concepts—*edit condition* and *assign condition job* and localise it in the interface. The expectations seem to be that there would be another operator button in the interface.

It is difficult to meet the expectation of the user for this concept as in fact it is not yet another operator but possible combinations of operators and conditions. A decision as to where to place the control had to be made. The chosen approach puts an emphasis on the fact that the subtree that will be defined by the providers will be embedded into an empty box—with all of its operators and condition boxes. Thus, the control for the owner is found in the *edit box* interaction mechanism rather than appear as yet another operator button.

For the provider part a major problem was identified that has to be solved for any future version of CoastEd: The participants could not use the component bar due to the lack of privileges if they were not the owner of the profile. This, under certain circumstances, makes completing the profile impossible. Only if the users follow a cer-

tain order of steps were they able to complete the profiles. Rather than just letting the users work within the box into which the subtree has to be embedded, they have to be able to temporarily use the full functionality. In doing this, it has to be taken care by the software to only have the users assemble the separate components in the way the owner had intended.

Also the presentation of the *open job* to the provider entailed an unnecessary interaction element. Each single participant made the mistake of clicking onto the box that represented the vague subtree and only later realised that this was not necessary and only opened a comment pop-up window. Instead they could have started to use the normal standard buttons from the main interface. However, the explanatory label in the vague-subtree box did not help them sufficiently with that.

8.5.3 Interaction Mechanisms

The interaction issues that were identified partially already were identified in the previous study. Attempts had been undertaken to improve the situation and in fact the problems were less prevalent than before.

Navigation in Nested Boxes. Again, some of the subjects did not fully grasp that while they were manipulating profiles, they were actually navigating in and out of expressions. This did not pose a problem for n -ary operators ($n \geq 2$) as the natural interaction behaviour of users is to click onto something they want to manipulate. For binary operators this implies that normally after manipulating one operand users will want to work with (one of) the other operand(s). Thus, they automatically click onto the respective operand and thereby navigate out of the previous operand and with the next click into the desired object. However, for unary operators the interaction mechanism was not as intuitive. The participants had to mentally grasp what they were doing in order to navigate out of the expression, rather than just to click and things would happen without their deeper understanding as to why this was.

During the study a metaphor was discovered that helped non-computer scientists to understand the underlying concept. Navigating in and out of expressions was com-

pared to boxing boxes into each other and unpacking them again in order to manipulate them—for example painting the boxes. This helped them a lot in understanding what was going on and avoiding mistakes.

During the pilot study one participant had the idea that the program should offer a mechanism that selects the entire component, that is, the program goes back to the root level of the expression. This was realised in the form of a button—however, it was widely ignored by the participants of the full study. The reason for this, potentially, can be found in the position of that button. If this option had been realised as a context menu or some other interaction mechanism that is located at the position of the component that is being manipulated, it might have been more successful.

8.5.4 Language Concepts

There were some lessons that were learned that did not concern the concept of collaboration as such but entailed things that were already latent in the previous study. However, due to the more complex tasks given in the collaborative study, they stood out more and had more impact than in the single-user study.

Timeframe versus Duration. Some participants had some minor problems with differentiating between the concepts of duration and timeframe. This issue only occurred during the collaborative phase of the study. During the introductory training tasks regarding single-user profiles this did not seem to pose a problem.

During that phase, the participants were capable of understanding that the duration is a parameter that goes with most of the operators and is determined by a single value. Whereas the timeframe is the time span during which the incoming information is monitored. Thus it is determined by a start and an end date.

During the collaborative phase, however, the participants got confused as to the nature of the given tasks. They were asked to give a possible range for the duration when assigning the job to the provider. The two given values—in contrast to the

usual single value normally connected to the duration—appeared to confuse them and made them think they were dealing with a timeframe rather than a duration.

Expressiveness of CGPDL. Another lesson learned was that the underlying language, GPDL is more expressive than originally intended by the developer. A number of subjects managed to specify expressions in a different way than intended. These expressions had the same semantics as the expression suggested by the researcher.

Node Exchange and Embedding. One of the major findings of the collaborative study concerned the underlying operation concepts of node exchange and embedding. It has to be noted that these findings seem to contradict what was identified during the single-user profile study. While in the single-user profile study it was confirmed that the choice to limit the operations to the most straight-forward ones under a given situation, was the most workable, the collaborative study suggested that it is preferable to always offer all kinds of operations.

Due to the more complex nature of the given tasks, the participants had to manipulate the components they had created in more ways than was required for the tasks of the single-user profile study. They got confused if a kind of operation they had just used in one situation could not be found in a different situation. Therefore, it would be more useful to always offer all operations. Even though this requires more buttons and would probably confuse the users initially, in the long run it would be better to show all operations. If the button, for some reason, does not make sense it seems it would be better for the participants to be shown an inactive button rather than no button with that functionality. This merits further research.

8.5.5 Presentation

One lesson learned concerned two issues regarding the general presentation of information and concepts.

Component Bar. The function of the component bar was not immediately obvious to the study participants. They required an explanation of what its purpose was. After the initial clarification most subjects started to use the component bar for subdividing the profiles they were specifying. Several of them were confused by some of the interaction mechanisms and the involved presentation of components. The shuffling in the component bar that had been implemented to avoid big gaps otherwise evolving over time is one example. Another is the lack of content presentation that was chosen due to the miniature presentation of the components in the component bar. The participants did not find the existing tooltips sufficient.

Duration. A minor but important lesson was identified for the *edit duration* window. In the pilot study the information input was spread too far in the window and entailed too many separate components. When this was simplified to a single information visualisation and input components that were located close to each other, the participants' performance was greatly increased.

Terminology. One of the main sources for confusion could be some of the terminology used. It would pay to review a number of labels and explanations given in the software. It was interesting to hear what names the participants used in order to refer to components when they were asking questions or talking about them. The terminology used by them could be a starting point for the revision of terms.

8.5.6 Test Situation

The study also yielded some findings about the study situation. They concerned the test situation regarding the given tasks, the learning curve that is involved with the program and the influence a previous participation of the same research project had on the results.

Understanding Tasks. A factor that highly influenced the study was the complexity of the subject matter. It was impossible to only test the mental effort that was involved in using the interface. Since the subjects could only start using the software

once they had fully grasped the underlying semantics of the given task, they were already on a level that was beyond a neutral basis regarding the mental effort that would be involved solving the tasks with the interface. Even though participants were asked to only consider the effort involved in using the interface when rating it in the questionnaire, this is psychologically challenging and probably could not be adhered to by every participant.

Moreover other parameters like ease of use and how intuitive and satisfying it was to use the software, were highly influenced by the effect of having to understand the given tasks first.

Learning Curve. It was observed that the subjects followed a steep learning curve. The subject matter is quite complex and thus the subjects initially had to learn many concepts. The method of presentation was improved between the single-user profile study and the collaborative profile study. Following some informal feedback of one of the previous subjects it was attempted to interleave the tutorial phases with phases that asked for active interaction of the participant with the software. This mostly increased the participants' concentration and despite this study covering more complex subject matters, the subjects seemed to tire less than in the single-user profile study.

Influence of Previous Participation. It was found that those subjects that had participated in one or both previous studies gave more open feedback than those that were entirely new to the material. They also rated almost all parameters more critically than the new subjects. This seems to lie in the fact that those subjects that participated in several studies became more familiar with the observer and therefore less shy and were more prone to give their honest opinion rather than responding out of politeness.

Likert Scale Responses. From the Likert scale responses, the impression was increased that the subject matter entails a certain degree of complexity. The success rate dropped for those sessions that took very little time. This seems to imply that

those subjects did not take the time required to fully understand the tasks and learn the involved concepts. It seems obvious that a high rating for intuitiveness, ease and satisfaction seems to be reflected in higher success rates.

While for the single-user profile study, non-computer scientists seemed to adopt the program more than computer scientists, in the collaborative study the opposite was found. Non-computer scientists found that more mental effort was involved. A possible explanation for the higher rating in mental activity that was found in the collaborative profiles study could be the high complexity of the tasks that had to be solved. Computer scientists would be more used to these kinds of problems than non-computer scientists. The higher mental effort might also have led to their rating for intuitiveness and satisfaction in using the program. As stated earlier it is impossible to factor out the effect of the complexity of tasks that the participants had to deal with.

8.6 Summary

This chapter provided the final study of this research. It evaluated CoastEd, the editor for collaborative profiles.

Section 8.1 and Section 8.2 detailed the goals of the study and the experimental design approach that was chosen. The quantitative results and the qualitative results of the user study were discussed in Section 8.3 and Section 8.4, respectively. A concluding discussion of the study was the content of Section 8.5.

Overall, the user study showed that the editor CoastEd constitutes an understandable and suitable editor to effectively specify collaborative profiles for alerting systems.

9 Conclusion

This thesis has presented a new end-user-centred approach to specifying single-user and collaborative profiles for alerting systems. An initial analysis of the requirements of end-users of alerting systems was undertaken for the application domain of health care. This analysis gave rise to generic solutions for the specification of single-user and collaborative profiles.

Single-user and collaborative profile definition languages (GPD and CGPD) were proposed. Software tools (GPD-UI and CoastEd) for the specification of profiles in these languages were presented. In doing so, this thesis has shown how single-user profiles can be extended into a collaborative context through an appropriate language and software. In addition, this thesis has made the case for a new collaborative alerting model.

The thesis proposes two major research questions:

***Research Question 1:** Does the graphical profile definition language GPD and the software application developed for its use enable average users to correctly and effectively specify their alerting needs for subscription with alerting systems?*

***Research Question 2:** Are the concept of collaborative profiles and the collaborative alerting model suitable means to satisfy the requirement of collaboration in alerting? Does the representation of collaborative profiles found in this thesis allow domain experts to correctly and effectively express their alerting needs?*

The contributions (see Section 9.2 for a full overview) show that the suggested approach to specifying single-user profiles is suitable for average end-users to correctly and effectively specify profiles (targeting Research Question 1). The contributions

also show that the suggested approach to specifying profiles collaboratively worked in most of the cases: For the initiator of the collaborative profile specification process, two types of profiles call for further research whereas the other four types were successfully specified by users. For the user providing the collaborative information, the specification process was effective and produced correct profiles for all profile types (targeting Research Question 2).

Section 9.1 gives a summary of the thesis and describes the research that was undertaken to answer the research questions to find suitable solutions to support both research questions. Section 9.2 gives an outline of the contributions of this work. Section 9.3 relates the findings of the usability studies to each other and discusses conclusions that can be drawn from this comparison. Moreover, it gives a discussion of other findings that were identified in the course of this research. The chapter is concluded by Section 9.4 which discusses potential future work.

9.1 Summary

The first step taken in this thesis was to undertake a requirements analysis of alerting systems described in Chapter 2, containing a related work analysis, an online survey targeted at patients and interviews with health care providers. The research questions of this thesis can be directly extracted out of the requirements that were found.

Chapter 3 of the thesis proposes an effective representation for a profile definition language to enable users to specify their alerting needs within profiles. The chapter starts by analysing and categorising existing languages and alerting approaches, but it also considers research from areas other than alerting. On the basis of this analysis GPDL, the Graphical Profile Definition Language, was developed. The effectiveness of GPDL was evaluated by a user study in Chapter 4.

The next step (Chapter 5) was to develop appropriate interaction mechanisms and an interface that can be used to effectively and correctly specify single-user profiles using GPDL. To reach these goals the software tool GPDL-UI was developed. This de-

velopment took the findings of the GPDFL user study into consideration. GPDFL-UI was analysed in a user study in Chapter 6.

The remainder of the thesis focused on effective collaboration aspects: Chapter 7 starts with developing the new collaborative alerting model. It continues with extending the single-user language GPDFL to a collaborative profile definition language, CGPDFL. Based on CGPDFL, Chapter 7 proposed CoastEd as an effective interface that can be used to correctly specify profiles collaboratively. Concluding the thesis, Chapter 8 evaluates CoastEd in combination with CGPDFL and shows that both fulfil their design goals of allowing an effective and correct profile specification.

9.2 Contributions

The contributions of this thesis cover two research areas in computer science—Human-Computer Interaction and Information Systems. Within these areas the contributions stem from two goals, improving the usability of interfaces for the specification of users' alerting needs and supporting users in collaborating in the process of the specification of alerting needs. A number of contributions have arisen from addressing these goals.

Requirements Analysis. A requirements analysis was undertaken regarding the user needs for applying alerting systems to support patients with chronic conditions. The requirements that were found by analysing the area—alerting systems in health care—and existing approaches in this area were verified and extended by conducting an online survey with patients and interviews with health care providers. The requirements found are representative of any domain that involves a highly collaborative work environment.

The two main findings of this requirements analysis are that alerting systems lack usable interfaces for the specification of profiles and that collaboration needs to be taken into account when specifying profiles.

Graphical Profile Definition Language (GPDL). Targeting the first research question, this thesis contributes a graphical profile definition language, GPDL. GPDL allows the specification of single-user profiles in alerting systems. This language targets average users of alerting systems and provides such users with a means to effectively specify their alerting needs.

Editor for the Specification of Single-user Profiles (GPDL-UI). Also regarding the first research question, an editor GPDL-UI for the specification of the alerting needs of single users was developed. This editor applies the graphical language GPDL as a representation for profiles.

Collaborative Alerting Model. Targeting the second research question, this thesis contributes the collaborative alerting model as an extension to existing research in the alerting area. This novel alerting model introduces the concept of collaborative profiles that offer the possibility of supporting collaboration between users in the process of specifying profiles.

The interviews undertaken within the requirements analysis on alerting systems for health care targeted this new concept of collaborative profiles in an exploratory way. These interviews gave a first impression that this new concept is useful for health care staff in better supporting them in their daily routine of treating patients.

Collaborative Graphical Profile Definition Language (CGPDL). This thesis contributes the graphical profile definition language CGPDL for the effective specification of collaborative profiles. CGPDL makes the collaborative alerting model available to users of alerting systems; the language was developed in the context of the second research question.

Editor for the Specification of Collaborative Profiles (CoastEd). Further targeting the second research question, this work contributes the editor CoastEd. CoastEd supports users in the specification process of collaborative profiles. The graphical language CGPDL is used by CoastEd to represent collaborative profiles.

Results of User Studies. The single-user approach for specifying profiles was evaluated in two user studies. Both the language GPDL and the editor GPDL-UI were shown to be effective means to correctly specify single-user profiles in a usable way. This contribution directly supports Research Question 1.

The collaborative approach for specifying collaborative profiles—realised through the language CGPDL and the editor CoastEd—turned out to be an effective means to correctly refine existing collaborative profiles in a usable way (provider side of profiles). CGPDL and CoastEd were also shown to be promising for setting up new collaborative profiles from scratch (owner side of profiles). This contribution directly supports the second research question of this work.

Repeatability. The results of the studies are repeatable by using the software developed in this PhD research. It is downloadable from the following page: <http://www.cs.waikato.ac.nz/~dj20/PhDProject/Software.html>

9.3 Cross-Study Observations

This research led to a number of observations that were made in the scope of all three user studies. Some observations could be studied as a consequence of changes made after one of the first two single-user profile specification studies; other observations concerned issues that hold for all three studies; again others became apparent due to a comparison of the results of each study. They are presented in the following.

Overall Success. Both GPDL and CGPDL are languages that are suitable for users with little or no training in formal logics. A tendency was observed for computer scientists to internally translate the operator symbols into symbols they commonly use (for example \wedge and \vee). They also used the technical terms *conjunction* and *disjunction* rather than the descriptive labels chosen for the operators of the interfaces (*all* and *some*). This put a higher cognitive load on them than was required. Non-computer scientists, however, easily learned the concepts of the two languages and applied them mostly correctly. Since the target group for the system was set to aver-

age users rather than trained computer scientists or mathematicians, the proposed languages fulfilled their aim.

Visual Language Semantics. Overall the approaches that were chosen to represent concepts in GPD L were well understood; users understood the representation of a single condition as a box immediately. They also picked up the differentiation of the time- versus logical-concept and its representation by the horizontal and vertical dimension, respectively, without any major hindrances. Users were also able to intuitively apply these representations.

Nested Expression Structure. During the studies, it was observed that in order to explain the nested character of expressions and to clarify the navigation in these expressions, it was useful to use a metaphor of stacked boxes that can only be manipulated (wrapped with wrapping paper or used for storing pens or the like) once the surrounding boxes have been opened. The idea was created on the spur of the moment as the researcher was storing her study material in boxes and had them standing close at hand. When in one of the pilot studies it became clear that words alone were not enough to make the subject understand, a quick stacking of boxes into each other was used to clarify the matter. This metaphor was then used in the entire study for explaining the nested nature of expressions.

Specification Approach. During the initial studies, it could be observed that the participants were using both the top-down and the bottom-up approach for the specification of profiles. Therefore both approaches were integrated into the potential interaction mechanisms of the software. This led to a trade-off: While each user could use their preferred approach, it confused those users that wanted to be entirely led by the system; it took them a moment to realise that it was up to them to decide how to proceed.

Timeframe versus Duration. Distinguishing between time-concepts was one of the biggest challenges for the single-user profile specification. A timeframe is determined by a start- and an end-point; a duration is determined by a single value that de-

scribes a time span. However, with the changes that were undertaken as result of the studies, in each subsequent study these problems were fewer than in the previous study. Removing the button with the duration arrow from the interface and instead only offering the duration arrow directly under the operator it belonged to—whenever there was a point in using this parameter with that operator—helped users to distinguish the duration and the timeframe. That is, the problem was resolved.

Explicit Operator Cues. The symbols on the buttons of the interface for the individual specification of profiles helped users to know which operator to use. This can be concluded from the fact that users used the online tutorial far less in the computer-based studies than the paper-based study. In the paper-based study they forgot what the operator looked like and looked it up. In the computer-based studies the symbols on the buttons helped them recognise which operator to choose. Whereas in the paper-based study participants made mistakes based on confusing operators, in the other two studies that showed the symbols of the operators on the buttons in the interface, instances of these mistakes were reduced.

Mapping between GPDL and PDL. One weakness of GPDL that was identified is an inconsistency between the *sequence* and the *conjunction* operator: The *sequence* without duration means that its duration is regarded as unlimited. In contrast to this, the *conjunction* without duration denotes the logical *conjunction* that has to be regarded more like a simultaneity if it is related to time. It does not assume an unlimited duration if the duration parameter is omitted. This is due to the simplification GPDL does over the underlying profile definition language: It uses one presentation for several underlying concepts, that is, logical operators and time-related operators are represented by the same concept in GPDL. This represents a problem in GPDL that in any future research should be targeted.

Over-abstraction. Another trade-off had to be accepted: One of the main foci in the development of the two graphical profile definition languages was simplicity; they put a level of abstraction over the underlying concepts. In this way they spare their users from having to deal with some amount of the existing complexity. In the

cases of simple, medium and advanced profiles, this worked well and users were able to easily and effectively specify their alerting needs. Nevertheless, for the profile specification at the professional level, this might be a cause for a drop in performance by the participants. Profiles at the professional level involve nesting and using combinations of multiple operators in one expression. A deeper understanding of the expression than aimed at by the graphical profile definition languages might be required at this level of complexity to warrant the effective and easy specification of profiles.

Example for Matching Profile Data. For the time-based concepts it became obvious that a picture with example data helped immensely to foster the participants' understanding of the concepts. Participants were shown a quick scribble of potential data entering the system. Then it was explained which of the data would match a given example profile. This leads back to the researcher's initial idea to visualise the data matched by the profile. However, as opposed to database queries which are posed over a fixed set of data, in alerting, the data that is manipulated is constantly changing. Thus, the data cannot be used for a visualisation of the result set that would be produced by the profile since the result set is a function of the time. However, if the approach researched in this thesis was to be revisited and extended, it might be worthwhile to integrate some visualisations with example data—like these spontaneously drawn during the study—that can show the matches of the profile with that particular sample data.

GUI. It was observed that the general layout and positioning of components in the working area of the program was not optimally evaluated by the studies. The single-user profile specification study used problems that—as it turned out afterwards—were not sufficiently complex to test the use of the component bar in combination with the workspace. Users either ignored the existence of the component bar or simply said it appeared like a good idea. However, only the collaborative profile specification study incorporated a couple of tasks that relied on the use of the component bar. Since these tasks entailed another problem it is difficult to say if the component bar and its interaction mechanisms require improvement.

Task Analysis and Domain Expertise. A major limitation which clearly stood out but which was nevertheless unavoidable lay in the fact that users first had to be able to understand the tasks they were given before they could attempt to specify the corresponding profile. Considering the mistakes that were made, it was almost impossible to test which of them were caused by the inability to use the given language or interface, and which were due to a misunderstanding of the task description. These descriptions posed a challenge to the verbal communicational skills of a number of participants. Other problems might have been caused by a lack of understanding of the domain. However, in the real-world users know what they want to express instead of having to interpret a task description. Moreover, in the real-world profiles would be specified by professionals of their own domain who, therefore, have no problems understanding the data they are dealing with.

The limitation of the study that it was not possible to distinguish between the participants' ability to understand the given tasks and their ability to use GPD and CGPD, also indicates another challenge: In order to specify their alerting need, participants first have to be able to understand the concepts that are offered by an alerting system to specify alerting needs. Then they have to decide which of those concepts they want to use. Only then they can use the given language—in the case of this thesis, GPD or CGPD—in order to specify the profile they have in mind. So the results of the user studies did not merely give insights about how suitable the given representations were for recognising already understood concepts, but also hinted at the intuitiveness of the representation to teach existing concepts.

Collaborative Profiles. It turned out that the specification for single-user profiles was easier to learn than the collaborative approach. However, considering that the collaborative concept is inherently much more complicated, users grasped the concept very fast and were quickly proficient in using most of the sub-concepts of collaborative specification.

Comparing the reactions and statements that were gathered in the interviews that were conducted with health care professionals (in the initial requirements analysis) to

the participants' handling of collaborative profiles in the user study, it can be stated that actually working with profiles, that is, specifying and interpreting profiles, made participants understand the approach much better. That means that the approach is much better understood when practically dealing with it rather than only hearing about it in a theoretical way. This suggests a positive prognosis regarding the concept of collaborative profiles and its adoption.

Not all parts of the process of collaboratively specifying profiles have been addressed. Typical problems that already have been targeted extensively by CSCW research have been deliberately excluded as they were not part of the focus of this thesis.

Extensibility. As considered when constructing GPDL, the language proved easily extendable to CGPDL.

Domain Data. Two of the three studies used data from the weather observation domain, that is, data other than from the healthcare domain. This change of domain data was easily achievable by loading a domain data model file for each of the domains. The studies showed no difference in the usability of the software depending on the domain data file that was used. The only difference that might have been observed was a different domain familiarity among the participants. However, this was independent of the usability of the software tools. This supports the generic applicability of GPDL-UI and CoastEd.

9.4 Future Work

The research presented in this thesis is complete in that it answers the research questions asked in Section 1.3 and confirms the claims made regarding those questions. During the research many other paths were found that could have been followed. This opens up the possibility of further research related to the work of this thesis. Also new challenges were identified on the way.

Different Routes in the Design Approach. One possible piece of research concerns the design approach that was chosen for the interaction mechanism with GPD L that is used in combination with the editors designed in the scope of this thesis. The initial design decision that was undertaken chose an approach that can be referred to as button-oriented rather than being similar to the interaction mechanisms of drag and drop commonly used in drawing applications. It was envisaged that a drawing-oriented approach would yield more errors in building profile expressions on the side of the user.

In order to discover if these deductions are true, research would need to be conducted which entails implementing the drawing-oriented approach, and undergoing the same usability study as was undertaken with the button-oriented approach proposed in this thesis, and to compare the results.

Further Development. Quite a lot of future work concerns aspects that are only relevant for a commercialisation of such a system. However, some areas are research-worthy.

This thesis only considers the specification of the alerting needs of users, however, it disregards what modes of delivery would be considered suitable by the users and how they could specify these modes (for example e-mail or text message). This part was disregarded as it is more straight-forward than the specification of the filter part of a profile that expresses a user's alerting need.

The system internally transforms the profile that has been defined with GPD L into the internally used profile definition language. The next step is to incorporate CoastEd into an existing alerting system. So the internally used language would have to be transformed into the language used by the respective alerting system. Approaches such as those suggested by Jung and Hinze in [68] could be employed.

The integration of CoastEd into an alerting system could include mapping the applied proprietary data model to an existing health care system standard. Along the same lines as that, another step would be to revise the data model that has been used, in col-

laboration with health care providers and adapt it to their needs. Since the data model is imported into the system it is easily interchangeable.

More along the lines of a commercial realisation would be issues of how to integrate CoastEd into an entire architecture of clinical information systems, mobile alerting systems, PCs and backup media.

Further Evaluations. Once CoastEd is connected to an alerting system, a longitudinal study should be conducted to research the uptake of both GPDL and CGPDL and the specification mechanisms offered by the interfaces. This would be following up where the interviews with health care providers failed to yield any further knowledge. While health care providers stated that the concept of collaborative profile specification would be a supportive means for their work and the treatment regime of patients (with chronic conditions), would they actually be able to successfully use such a system? Would it help them in the way that they predicted? Would it improve treatment results? Would it make their work easier?

Such a longitudinal study would also reveal if users would choose a certain approach because it suits them or simply because this approach is what they were shown or figured out first. For example, they might use the top-down approach rather than bottom-up, or instead of using the *repetition* operator, they might create a condition and then copy and paste it multiple times into the *sequence* operator—a valid but more complicated approach. If they use the program over a longer period of time, they would discover which approaches might be most convenient for them.

It might be necessary to repeat the study with different participants, that is, rather than testing with university students and university employees (mostly—some professionals came from other areas) testing the system with health care staff.

In order to support the claim of GPDL-UI's and CoastEd's generic applicability to domains that are characterised by a highly collaborative environment, it could be of interest to evaluate the two software tools with other domain data. This is easily

achievable as other data models and domain-specific data can easily be imported into the systems.

Another aspect suitable for future research would be to test if users are able to specify alerts that they decide are relevant for them rather than to use pre-set tasks. However, this has the limitation of being very subjective. It would not be possible to gather parameters that are comparable to each other as each participant would undertake different tasks. However, this would eliminate the problem that first the tasks have to be understood in order for the participants to specify the profile aimed for by the researcher.

Another avenue that could be chosen is to undertake a cognitive dimensions framework evaluation [101] of the languages, GPDL and CGPDL, as well as their corresponding software applications, GPDL-UI and CoastEd. This would make the results more comparable to other research in the area of visual languages. The framework contains 13 cognitive dimensions. These dimensions are concepts that are relevant to the structure of a language and users' interaction with it. It comprises notions such as abstraction gradient, consistency, error-proneness, hidden dependencies and viscosity. It is difficult to optimise all of these dimensions. For example in this thesis the viscosity of the approach was increased when the button approach was chosen over the drawing approach (see Section 5.1.3). This was done in order to decrease the error-proneness of profile specification. In the drawing approach users could manipulate profiles in any way they would want to whereas in the button approach users are led through the underlying hierarchy of the profile. This requires more steps but avoids the creation of incorrect profile expressions.

9.5 Epilogue

To date, the alerting systems community has focused on excelling in areas such as the scalability of alerting systems, their performance and other technical issues, but has given no consideration to end-users.

By developing effective graphical profile definition languages and by providing usable software tools for their application, this thesis takes a step towards acknowledging the necessity of integrating end-users and their requirements into the design and development of alerting systems.

References

- [1] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra, "Matching Events in a Content-Based Subscription System," in *18th ACM Symposium on Principles of Distributed Computing (PODC'99)*. Atlanta, USA, 1999, pp. 53-61.
- [2] J. Pereira, F. Fabret, F. Llirbat, and D. Shasha, "Efficient Matching for Web-Based Publish/Subscribe Systems," in *7th International Conference on Cooperative Information Systems (CoopIS 2000)*. Eilat, Israel, 2000, pp. 162-173.
- [3] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Design and Evaluation of a Wide-Area Event Notification Service," *ACM Transactions on Computer Systems*, vol. 19, pp. 332-383, 2001.
- [4] G. Li, V. Muthusamy, and H.-A. Jacobsen, "Adaptive content-based routing in general overlay topologies," presented at 9th ACM/IFIP/USENIX International Conference on Middleware, Leuven, Belgium, 2008.
- [5] Y. Huang and H. Garcia-Molina, "Publish/Subscribe in a Mobile Environment," presented at 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE '01), Santa Barbara, USA, 2001.
- [6] L. Mottola, G. Cugola, and G. P. Picco, "A Self-Repairing Tree Topology Enabling Content-Based Routing in Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 946-960, 2008.
- [7] P. T. Eugster, "Type-Based Publish/Subscribe," in *Communications Systems Department*. Lausanne, Switzerland: Swiss Federal Institute of Technology in Lausanne (EPFL), 2001.
- [8] A. Wun and H.-A. Jacobsen, "A policy management framework for content-based publish/subscribe middleware," presented at 8th ACM/IFIP/USENIX International Conference on Middleware, Newport Beach, USA, 2007.
- [9] D. Jung and A. Hinze, "A Mobile Alerting System for the Support of Patients with Chronic Conditions," in *European Conference on Mobile Government (EURO mGOV)*. Brighton, UK, 2005, pp. 264-274.
- [10] D. Jung and A. Hinze, "Patient-Based Mobile Alerting Systems - Requirements and Expectations," in *Health Informatics in New Zealand Conference (HINZ 2005)*. Auckland, New Zealand, 2005.

- [11] D. Jung, "Collaborative Profiles for Event Notification in Healthcare: Lessons Learned from Healthcare Staff Interviews," in *5th New Zealand Computer Science Research Student Conference (NZCSRSC'07)*. Hamilton, New Zealand, 2007.
- [12] I. Purves and P. Robinson, "Knowledge Management for Health: What 'Tools' Can Improve the Performance of Workgroups, Clinicians and Patients?" in *11th World Congress on Medical Informatics (Medinfo 2004)*. San Francisco, USA, 2004, pp. 678–682.
- [13] M. M. Wagner, S. A. Eisenstadt, W. R. Hogan, and M. C. Pankaskie, "Preferences of Interns and Residents for E-mail, Paging, or Traditional Methods for the Delivery of Different Types of Clinical Information," in *American Medical Informatics Association (AMIA) 1998 Annual Symposium*. Lake Buena Vista, USA, 1998, pp. 140-144.
- [14] S. D. Iordache, D. Orso, and J. Zelingher, "A Comprehensive Computerized Critical Laboratory Results Alerting System for Ambulatory and Hospitalized Patients," in *10th World Congress on Medical Informatics (Medinfo 2001)*. London, UK, 2001, pp. 469-473.
- [15] G. Hripcsak, S. Bakken, P. D. Stetson, and V. L. Patel, "Mining complex clinical data for patient safety research: a framework for event discovery," *Journal of Biomedical Informatics*, vol. 36, pp. 120-130, 2003.
- [16] M. M. Shabot, M. LoBue, and J. Chen, "Wireless Clinical Alerts for Physiologic, Laboratory and Medication Data," in *American Medical Informatics Association (AMIA) 2000 Annual Symposium*. Los Angeles, USA, 2000, pp. 789–793.
- [17] C. J. Staes, R. S. Evans, B. H. S. C. Rocha, J. B. Sorensen, S. M. Huff, J. Arata, and S. P. Narus, "Computerized Alerts Improve Outpatient Laboratory Monitoring of Transplant Patients," *Journal of the American Medical Informatics Association (JAMIA)*, vol. 15, pp. 324-332, 2008.
- [18] S. Kheterpal, R. Gupta, J. Blum, K. Tremper, M. O'Reilly, and P. Kazanjian, "Electronic Reminders Improve Procedure Documentation Compliance and Professional Fee Reimbursement," *Anesthesia and Analgesia*, vol. 104, pp. 592-597, 2007.
- [19] G. J. Kuperman, J. M. Teich, M. J. Tanasijevic, N. Ma'Luf, E. Rittenberg, A. Jha, J. Fiskio, J. Winkelman, and D. W. Bates, "Improving Response to Critical Laboratory Results with Automation: Results of a Randomized Controlled Trial," *Journal of the American Medical Informatics Association (JAMIA)*, vol. 6, pp. 512-22, 1999.
- [20] H.-T. Chen, W.-C. Ma, and D. M. Liou, "Design and implementation of a real-time clinical alerting system for intensive care unit," in *American Medical Informatics Association (AMIA) 2002 Annual Symposium*. San Antonio, USA, 2002, pp. 131-135.

- [21] E. G. Poon, G. J. Kuperman, J. Fiskio, and D. W. Bates, "Real-time Notification of Laboratory Data Requested by Users through Alphanumeric Pagers," *Journal of the American Medical Informatics Association (JAMIA)*, vol. 9, pp. 217-222, 2002.
- [22] M. M. Wagner, M. Pankaskie, W. Hogan, F. C. Tsui, S. A. Eisenstadt, E. Rodriguez, and J. K. Vries, "Clinical Event Monitoring at the University of Pittsburgh," in *American Medical Informatics Association (AMIA) 1997 Annual Symposium*. Nashville, USA, 1997, pp. 188–192.
- [23] I. Hoch, A. D. Heymann, I. Kurman, L. J. Valinsky, G. Chodick, and V. Shalev, "Countrywide computer alerts to community physicians improve potassium testing in patients receiving diuretics," *Journal of the American Medical Informatics Association*, vol. 10, pp. 541-546, 2003.
- [24] C. H. Fung, J. S. Tsai, A. Lulejian, P. Glassman, E. Patterson, B. N. Doebeling, and S. M. Asch, "An Evaluation of the Veterans Health Administration's Clinical Reminders System: A National Survey of Generalists," *Journal of General Internal Medicine*, vol. 23, pp. 1497-1525, 2008.
- [25] J. J. Saleem, E. S. Patterson, L. Militello, M. L. Render, G. Orshansky, and S. M. Asch, "Exploring Barriers and Facilitators to the Use of Computerized Clinical Reminders," *Journal of the American Medical Informatics Association*, vol. 12, pp. 438-447, 2005.
- [26] B. Wu and K. Dube, "PLAN: a Framework and Specification Language with an Event-Condition-Action (ECA) Mechanism for Clinical Test Request Protocols," in *34th Hawaii International Conference on System Science (HICSS-34)*. Maui, USA, 2001.
- [27] K. Zheng, R. Padman, M. Johnson, J. Engberg, and H. Diamond, "An adoption study of a clinical reminder system in ambulatory care using a developmental trajectory approach," in *11th World Congress on Medical Informatics (Medinfo 2004)*. San Francisco, USA, 2004, pp. 1115-1119.
- [28] W. L. Galanter, R. J. Didomenico, and A. Polikaitis, "A Trial of Automated Decision Support Alerts for Contraindicated Medications Using Computerized Physician Order Entry," *Journal of the American Medical Informatics Association (JAMIA)*, vol. 12, pp. 269-274, 2005.
- [29] A. W. Steele, S. Eisert, J. Witter, P. Lyons, M. A. Jones, P. Gabow, and E. Ortiz, "The Effect of Automated Alerts on Provider Ordering Behavior in an Outpatient Setting," *PLoS Medicine*, vol. 2, pp. 864-870, 2005.
- [30] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 2nd ed: Sage Publications, 2003.
- [31] D. S. Moore and G. P. McCabe, *Introduction to the Practice of Statistics*, 4th ed: W. H. Freeman and Company, 2003.
- [32] J. W. Creswell, *Qualitative Inquiry and Research Design: Choosing Among Five Traditions*, 2nd ed: Sage Publications, 1997.

- [33] J. Nielsen and R. Molich, "Improving a Human-computer Dialogue," *Communications of the ACM*, vol. 33, pp. 338 - 348, 1990.
- [34] "CORBA Event Service Specification, version 1.0," Object Management Group OMG Document 00-06-20, 2000.
- [35] "Jini(TM) Technology Core Platform Specification," Sun Microsystems, Inc. Platform Specification, 2000.
- [36] B. Oki, M. Pfluegl, A. Siegel, and D. Skeen, "The Information Bus – An Architecture for Extensible Distributed Systems," in *14th Symposium on Operating Systems Principles (SIGOPS)*. Asheville, USA, 1993, pp. 58-68.
- [37] G. Cugola, E. D. Nitto, and A. Fuggetta, "The JEDI Event-Based Infrastructure and Its Application to the Development of the OPSS WFMS," *IEEE Transactions on Software Engineering (TSE)*, vol. 27, pp. 827-850, 2001.
- [38] G. Fitzpatrick, T. Mansfield, S. Kaplan, D. Arnold, T. Phelps, and B. Segall, "Augmenting the workaday world with Elvin," presented at 6th European Conference on Computer Supported Cooperative Work (ECSCW'99), Copenhagen, Denmark, 1999.
- [39] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman, "An Efficient Multicast Protocol for Content-based Publish-Subscribe Systems," in *19th IEEE International Conference on Distributed Computing Systems (ICDCS'99)*. Austin, USA, 1999, pp. 262-272.
- [40] P. R. Pietzuch and J. Bacon, "Hermes: A Distributed Event-Based Middleware Architecture," in *22nd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'02)*. Vienna, Austria, 2002, pp. 611-618.
- [41] S. Brandt and A. Kristensen, "Web push as an Internet notification service," in *W3C Workshop on Push Technology*. Boston, USA, 1997.
- [42] G. Li, S. Hou, and H.-A. Jacobsen, "A Unified Approach to Routing, Covering and Merging in Publish/Subscribe Systems based on Modified Binary Decision Diagrams," in *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. Columbus, USA, 2005, pp. 447-457.
- [43] G. Mühl, L. Fiege, and A. Buchmann, "Filter Similarities in Content-Based Publish/Subscribe Systems," in *International Conference on Architecture of Computing Systems (ARCS'02)*. Karlsruhe, Germany, 2002, pp. 224-238.
- [44] L. Golab and M. T. Özsu, "Issues in data stream management," *ACM SIGMOD Record*, vol. 32, pp. 5-14, 2003.
- [45] A. J. G. Gray and W. Nutt, "A Data Stream Publish/Subscribe Architecture with Self-adapting Queries," in *13th International Conference on Cooperative Information Systems (CoopIS 2005)*. Agia Napa, Cyprus, 2005, pp. 420-438.

- [46] L. Liu, C. Pu, and W. Tang, "Continual queries for internet scale event-driven information delivery," *IEEE Transactions on Knowledge and Data Engineering, Special issue on Web Technologies*, vol. 11, pp. 610-628, 1999.
- [47] L. Liu, C. Pu, W. Tang, and W. Han, "CONQUER: A Continual Query System for Update Monitoring in the WWW," *International Journal of Computer Systems, Science, and Engineering, Special edition on Web Semantics*, vol. 14, pp. 99-112, 1999.
- [48] J. Chen, D. J. DeWitt, F. Tian, and Y. Wang, "NiagaraCQ: a scalable continuous query system for Internet databases," in *2000 ACM SIGMOD International Conference on Management of Data (SIGMOD 2000)*. Dallas, USA, 2000, pp. 379-390.
- [49] J. Pereira, F. Fabret, H.-A. Jacobsen, F. Llirbat, and D. Shasha, "WebFilter: A High-throughput XML-based Publish and Subscribe System," in *27th International Conference on Very Large Data Bases (VLDB 2001)*. Rome, Italy, 2001, pp. 723-724.
- [50] M. Altinel and M. J. Franklin, "Efficient Filtering of XML Documents for Selective Dissemination of Information," in *26th International Conference on Very Large Data Bases (VLDB 2000)*. Cairo, Egypt, 2000, pp. 53-64.
- [51] R. Chand and P. A. Felber, "Efficient filtering of XML documents with XPath expressions," *The VLDB Journal*, vol. 11, pp. 354-379, 2004.
- [52] J. Yang, M. P. Papazoglou, and B. J. Krämer, "A Publish/Subscribe Scheme for Peer-to-Peer Database Networks," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Catania, Italy, 2003, pp. 244-262.
- [53] H. Liu and H.-A. Jacobsen, "Modeling uncertainties in publish/subscribe systems," in *International Conference on Data Engineering (ICDE'04)*. Boston, USA, 2004, pp. 510-522.
- [54] S. Bittner, "General Boolean Expressions in Publish-Subscribe Systems," in *Department of Computer Science*. Hamilton, New Zealand: The University of Waikato, 2008.
- [55] A. Campailla, S. Chaki, E. Clarke, S. Jha, and H. Veith, "Efficient Filtering in Publish-Subscribe Systems using Binary Decision Diagrams," in *23rd International Conference on Software Engineering (ICSE 2001)*. Toronto, Canada, 2001, pp. 443-452.
- [56] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. L. Wiener, "The Lorel Query Language for Semistructured Data," *International Journal on Digital Libraries*, vol. 1, pp. 68-88, 1997.
- [57] H. Hosoya and B. C. Pierce, "XDuce: A statically typed XML processing language," *ACM Transactions on Internet Technology (TOIT)*, vol. 3, pp. 117-148, 2000.

- [58] W. May, "XPathLog: A declarative, native XML data manipulation language.," in *The International Database Engineering and Applications Symposium (IDEAS'01)*. Grenoble, France, 2001, pp. 123-128.
- [59] J. Robie, J. Lapp, and D. Schach, "XML Query Language (XQL)," in *The Query Languages 1998 Conference*. Boston, USA, 1998.
- [60] G. Moerkotte and T. Fiebig, "YAXQL: A powerful and web-aware query language supporting query reuse," Department for Mathematics and Computer Science, University of Mannheim, Mannheim, Germany, Technical Report TR-2000-016, 7 August 2000.
- [61] S. Ceri, S. Comai, E. Damiani, P. Fraternali, S. Paraboschi, and L. Tanca, "XML-GL: a Graphical Language for Querying and Restructuring XML Documents," *WWW8/Computer Networks*, vol. 31, pp. 1171-1187, 1999.
- [62] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu, "XML-QL: A Query Language for XML," <http://www.w3.org/TR/NOTE-xml-ql/>, 1998.
- [63] J. Clark and S. DeRose, "XML Path Language (XPath) Version 1.0," <http://www.w3.org/TR/xpath>, 1999.
- [64] S. Boag, D. Chamberlin, M. F. Fernández, D. Florescu, J. Robie, and J. Siméon, "XQuery 1.0: An XML Query Language," <http://www.w3.org/TR/xquery/>, 2007.
- [65] A. Hinze, Y. Michel, and T. Schlieder, "Approximative filtering of XML documents in a publish/subscribe system," in *29th Australasian Computer Science Conference*. Hobart, Australia, 2006, pp. 177-185.
- [66] D. Zimmer and R. Unland, "The Formal Foundation of the Semantics of Complex Events in Active Database Management Systems," Cooperative Computing & Communication Laboratory, Paderborn, Germany, Technical Report 22/1997, 1997.
- [67] A. Hinze and A. Voisard, "Composite events in notification services with application to logistics support," in *9th International Symposium on Temporal Representation and Reasoning (TIME-2002)*. Manchester, UK, 2002.
- [68] D. Jung and A. Hinze, "A Meta-service for Event Notification," in *International Conference on Cooperative Information Systems (CoopIS 2004), in conjunction with OnTheMove Federated Conferences (OTM'04)*. Agia Napa, Cyprus, 2004, pp. 283-300.
- [69] D. Jung and A. Hinze, "Event Notification Services: Analysis and Transformation of Profile Definition Languages," University of Waikato, Computer Science Department, Hamilton, New Zealand, Technical Report 12/2004, October 2004.
- [70] T. J. Parr and R. W. Quong, "ANTLR: a predicated-LL(k) parser generator," *Software-Practice & Experience*, vol. 25, pp. 789-810, 1995.

- [71] A. Michard, "Graphical presentation of boolean expressions in a database query language: design notes and an ergonomic evaluation," *Behaviour & Information Technology*, vol. 1, pp. 279 - 288, 1982.
- [72] D. Young and B. Shneiderman, "A graphical filter/flow representation of Boolean queries: a prototype implementation and evaluation," *Journal of the American Society for Information Science*, vol. 44, pp. 327-339, 1993.
- [73] T. Davies and S. Willie, "The efficacy of a Venn-based query interface: an evaluation," in *Annual CHISIG (Queensland) Symposium (QCHI '95)*. Gold Coast, Australia, 1995, pp. 41-50.
- [74] T. A. Halpin, "Venn diagrams and SQL queries," *Australian Computer Journal*, vol. 21, pp. 27-32, 1989.
- [75] M. Hertzum and E. Frøkjær, "Browsing and querying in online documentation: a study of user interfaces and the interaction process," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 3, pp. 136-161, 1996.
- [76] S. Jones, S. McInnes, and M. S. Staveley, "A graphical user interface for boolean query specification," *International Journal on Digital Libraries Special Issue on User Interfaces for Digital Libraries*, vol. 2, pp. 207-223, 1999.
- [77] C. Katzeff, "The effect of different conceptual models upon reasoning in a database query writing task," *International Journal of Man-Machine Studies*, vol. 29, pp. 37-62, 1988.
- [78] S. Willie and P. Bruza, "Users' models of the information space: the case for two search models," in *18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*. Seattle, USA, 1995, pp. 205-210.
- [79] A. Spoerri, "InfoCrystal: a visual tool for information retrieval & management," in *Second International Conference on Information and Knowledge Management (CIKM 93)*. Washington, USA, 1993, pp. 11-20.
- [80] D. E. Egan, J. R. Remde, L. M. Gomez, T. K. Landauer, J. Eberhardt, and C. C. Lochbaum, "Formative design evaluation of superbook," *ACM Transactions on Information Systems (TOIS)*, vol. 7, pp. 30-57, 1989.
- [81] R. Chimera and B. Shneiderman, "An exploratory evaluation of three interfaces for browsing large hierarchical tables of contents," *ACM Transactions on Information Systems (TOIS)*, vol. 12, pp. 383-406, 1994.
- [82] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Transactions on Graphics (TOG)*, vol. 11, pp. 92-99, 1992.
- [83] J. Stasko, R. Catrambone, M. Guzdial, and K. McDonald, "An evaluation of space-filling information visualizations for depicting hierarchical structures," *International Journal on Human-Computer Studies (IJHCS)*, vol. 53, pp. 663-694, 2000.

- [84] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone Trees: animated 3D visualizations of hierarchical information," in *SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology*. New Orleans, USA, 1991, pp. 189-194.
- [85] H. P. Kumar, C. Plaisant, and B. Shneiderman, "Browsing Hierarchical Data with Multi-Level Dynamic Queries and Pruning," *International Journal of Human-Computer Studies*, vol. 46, pp. 103-124, 1997.
- [86] J. Lamping, R. Rao, and P. Pirolli, "A focus+context technique based on hyperbolic geometry for visualizing large hierarchies," in *SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. Denver, USA, 1995, pp. 401-408.
- [87] E. R. Tufte, *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press, 1983.
- [88] J. D. Mackinlay, G. G. Robertson, and S. K. Card, "The perspective wall: detail and context smoothly integrated," in *SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology*. New Orleans, United States, 1991, pp. 173-179.
- [89] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman, "LifeLines: visualizing personal histories," in *SIGCHI Conference on Human Factors in Computing Systems: Common Ground*. Vancouver, Canada, 1996, pp. 221-227.
- [90] "OMG Unified Modeling Language Specification," Object Management Group, Inc. formal/2009-02-02, 2009.
- [91] M. Cilia and A. P. Buchmann, "An Active Functionality Service For E-Business Applications," *ACM SIGMOD Record, Special Issue on Data Management Issues in Electronic Commerce*, vol. 31, pp. 24-30, 2002.
- [92] A. Hinze, "A-MEDIAS: Concept and Design of an Adaptive Integrating Event Notification Service," in *Department of Computer Science*. Berlin: Freie Universität Berlin, 2003.
- [93] A. Hinze and A. Voisard, "Location- and Time-Based Information Delivery in Tourism," *8th International Symposium on Advances in Spatial and Temporal Databases*, pp. 489-507, 2003.
- [94] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed: Pearson/Prentice-Hall, 2004.
- [95] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, pp. 56-58, 1997.
- [96] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, pp. 66-72, 1997.

- [97] R. Burke, "Integrating Knowledge-based and Collaborative-filtering Recommender Systems," in *Workshop on Artificial Intelligence for Electronic Commerce*. Orlando, USA, 1999.
- [98] A. G. Moukas, "User Modeling in a MultiAgent Evolving System," in *Workshop Machine Learning for User Modeling*. Chia Laguna, Italy, 1997.
- [99] C. Shahabi, A. M. Zarkesh, J. Adibi, and V. Shah, "Knowledge discovery from users Web-page navigation," in *Seventh International Workshop on Research Issues in Data Engineering*. Birmingham, UK, 1997, pp. 20-29.
- [100] D. Jung and A. Hinze, "Capturing Context in Collaborative Profiles," in *OTM 2005 Workshop on Context-Aware Mobile Systems (CAMS 2005), in conjunction with OnTheMove Federated Conferences (OTM 2005)*. Agia Napa, Cyprus, 2005, pp. 152-155.
- [101] T. R. G. Green and M. Petre, "Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework," *Journal of Visual Languages and Computing*, vol. 7, pp. 131-174, 1996.

Appendix A. Patient Online-Survey

The online survey was targeted towards patients, doctors, nurses and computer scientists working in the health care domain. In the evaluation of the online survey only the patient questionnaires were used. However, for completeness, all questionnaires are included in this appendix.



Requirements Survey for the Project "A Mobile Alerting Device for the Support of Patients with Chronic Conditions"

[Deutsche Version](#) | [English Version](#)

The Project

This survey is connected with the project "A Mobile Alerting Device for the Support of Patients with Chronic Conditions", which is carried out at the University of Waikato. The aim of this survey is to find out requirements of the system we are researching on.

In our project we are developing a system, which helps patients and medical staff in the management of chronic conditions such as glaucoma, diabetes, chronic heart-disease or cardio-vascular problems. This system will be integrated into a little helper patients can take along wherever desired. Thus, help can be at hand whenever needed. So what does this little helper do? To give you a brief impression think of the following tasks:

First of all it can store important and interesting information for patients and medical staff. Also it can remind patients of important condition-related issues such as taking the correct medicine at the correct time and in the correct amount. Furthermore it can help medical staff to perform a permanent remote analysis of new condition-relevant data fed into the helper by the patient. Due to this the condition of a patient can be controlled more thoroughly without having to go to the doctor's office more often. Additionally, patients can also hold a copy of their personal condition-related values such as values of blood sugar. So patients can lead their lives more independently.

Generally, with this little helper we want to improve lives of patients, improve overall treatment results, help doctors in analyzing their patients' condition and generally relieve clinical staff of organizational bureaucratic work.

I have already specified some requirements, which I suggest you to choose from in the actual questionnaire. But I would like to complement these requirements and replace them where appropriate by new ones inspired by your opinion. If you take part in this survey, this would enable me to take your ideas into account developing this helper. So it is your opportunity to get involved now!

I am doing this project for my PhD research. This topic I have chosen since I think that in today's healthcare, information systems are mostly used to support clinical staff, whereas patients with chronic conditions are neglected in the support of the management of their conditions.

The Survey

The survey consists of two parts. The first and major part consists of three or four sections depending on whether you are a patient or work at a clinic. The second part collects some personal data, so that I can put your replies into a greater context. Instructions are written *like this* and questions are written like this . The survey will take about 15 minutes to complete. There will be a final box for remarks. Please write down in there those things you would usually scribble down at the sides of a paper questionnaire or use it whenever you come up with some idea. Your ideas are very valuable to me, even if you might think they are unimportant!

If you have any questions concerning the survey or the project, do not hesitate to contact me at: d (dot) jung (at) cs (dot) waikato (dot) ac (dot) nz. Note that you will not be identified in any publication or dissemination of my research findings. The result from your questionnaire will be sent anonymously to me.

If you proceed to the actual questionnaire you agree with the following research consent form, which you can also read [here](#).

Research Consent Form

 This consent form, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

Research Project Title

 Requirements survey for the project "A Mobile Alerting Device for the Support of Patients with Chronic Conditions".

Researcher

 Ms Doris Jung.

Survey Purpose

 The purpose of this survey is to find out in what way patients with chronic diseases would like to be supported by a technical device, which can help them with the management of their disease.

Participant Recruitment and Selection

You can inform yourself about the Research Participant's Bill of Rights in the following or [here](#).

Research Participant's Bill of Rights

 The following is a list of your rights if you participate in a research project organised within the School of Computing and Mathematical Sciences at the University of Waikato.

As a research participant, you have the right:

- To be treated with respect and dignity in every phase of the research.
- To be fully and clearly informed of all aspects of the research prior to becoming involved in it.
- To enter into clear, informed, and written agreement with the researcher prior to becoming involved in the activity. You should sense NO pressure, explicit or otherwise, to sign this contract.
- To choose explicitly whether or not you will become involved in the research under the clearly stated provision that refusal to participate or the choice to withdraw during the activity can be made at any time without penalty to you.
- To be treated with honesty, integrity, openness, and straightforwardness in all phases of the research, including a guarantee that you will not unknowingly be deceived during the course

The Questionnaire

The survey asks different questions depending on your status.

What status best describes you?

(If several of the statuses offered apply to you, please choose one of the questionnaires applying to you. If possible fill in afterwards a questionnaire again for your other status.

If none of the statuses applies to you chose the one that is similar to your actual status.)

I confirm that I have read the Research Consent Form and Research Participant's Bill of Rights and agree to participate in the survey.

Questionnaire for Patients

Questionnaire for Doctors

Questionnaire for Nurses

Questionnaire for Computer Scientists

Doris Jung is a PhD student at the University of Waikato, New Zealand. She is writing her PhD thesis with the Department of Computer Science. If you have any questions about this project, you can contact her via e-mail at d (dot) jung (at) cs (dot) waikato (dot) ac (dot) nz.

Research Consent Form

This consent form, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

Research Project Title

Requirements survey for the project "A Mobile Alerting Device for the Support of Patients with Chronic Conditions".

Researcher

Ms Doris Jung.

Survey Purpose

The purpose of this survey is to find out in what way patients with chronic diseases would like to be supported by a technical device, which can help them with the management of their disease.

Participant Recruitment and Selection

Members of several condition-related mailing lists are addressed to complete this survey. Moreover some doctors, nurses, patients and computer scientists are directly asked to fill in the questionnaire.

Procedure

This survey will require about 15 minutes of your time. You will be asked to fill in a short questionnaire consisting of two parts. The first part will deal with the management of your or your patients' condition in reference to the supportive technical device proposed by the researcher. This will be followed by an enquiry about some statistical data. So that the results from the first part can be put into perspective.

Data Collection

An online-questionnaire will be used to collect the answers. They will be anonymously sent to the researcher.

Confidentiality

Confidentiality and participant anonymity will be strictly maintained. All information gathered will be used for statistical analysis only and no names or other identifying characteristics will be stated in the final or any other reports.

Likelihood of Discomfort

There is no likelihood of discomfort or risk associated with participation.

Researcher

Doris Jung is working on her doctorate in the Computer Science Department at the University of Waikato. This study will contribute to her research "Adaptable Event Notification Services in Healthcare". Her supervisor is Dr. Annika Hinze. Doris can be contacted in room G2.06 of the School of Computer and Mathematical Sciences building at the University of Waikato. Her phone number is 856 2889 extension 6011 and her email address is d (dot) jung (at) cs (dot) Waikato (dot) ac (dot) nz.

Finding out about Results

The Participants can find out the results of the study by contacting the researcher after April 1, 2005.

Agreement

Ticking the check box before proceeding to the actual questionnaire indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to not answer specific items or questions in interviews or on questionnaires. You are free to withdraw from the study at any time without penalty. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact the researcher.

Research Participant's Bill of Rights

The following is a list of your rights if you participate in a research project organised within the School of Computing and Mathematical Sciences at the University of Waikato.

As a research participant, you have the right:

- To be treated with respect and dignity in every phase of the research.
- To be fully and clearly informed of all aspects of the research prior to becoming involved in it.
- To enter into clear, informed, and written agreement with the researcher prior to becoming involved in the activity. You should sense NO pressure, explicit or otherwise, to sign this contract.
- To choose explicitly whether or not you will become involved in the research under the clearly stated provision that refusal to participate or the choice to withdraw during the activity can be made at any time without penalty to you.
- To be treated with honesty, integrity, openness, and straightforwardness in all phases of the research, including a guarantee that you will not unknowingly be deceived during the course of the research.
- To receive something in return for your time and energy.
- To demand proof that an independent and competent ethical review of human rights and protections associated with the research has been successfully completed.
- To demand complete personal confidentiality and privacy in any reports of the research unless you have explicitly negotiated otherwise.
- To expect that your personal welfare is protected and promoted in all phases of the research, including knowing that no harm will come to you.
- To be informed of the results of the research study in a language you understand.
- To be offered a range of research studies or experiences from which to select, if the research is part of fulfilling your educational or employment goals.

The contents of this bill were prepared by the University of Calgary who examined all of the relevant Ethical Standards from the Canadian Psychological Association's Code of Ethics for Psychologists, 1991 and rewrote these to be of relevance to research participants.

Descriptions of the CPA Ethical Code and the CPA Ethical Standards relevant to each of these rights are available at <http://www.cpa.ca/ethics2000.html> and <http://www.psych.ucalgary.ca/Research/ethics/bill/billcode.html> if you would like to examine them.

The complete CPA Ethical Code can be found in Canadian Psychological Association "Companion manual for the Canadian Code of Ethics for Psychologists" (1992).



Requirements Survey for the Project "A Mobile Alerting Device for the Support of Patients with Chronic Conditions"

Introduction

Imagine for the management of your chronic condition you had a little supportive helper. This helper could remind you of condition-related issues or store vital information and supply you with them whenever needed.

It could also be used for communication purposes with your healthcare provider.

In addition, your healthcare provider would have the chance to perform a permanent analysis of values which are fed by you into the helper. These values could be things such as blood sugar values or blood pressure values. Thus, whenever treatment changes are required this is noticed at once.

A. What do you need?

Imagine you were the designer of the helper. What should the helper do to help you? How would you like being supported? I would like you to think about that for a moment and then tell me about it in the next questions - especially in the "Other" boxes.

What kind of things would you like to be reminded of?

(tick several boxes if applicable)

- Taking the correct amount of medicine
- Taking medicine at the correct time
- Taking the correct type of medicine
- Getting a new prescription
- Doctor appointments
- Critical values of your readings, e.g. too high blood sugar values
- Critical values of your parameters involving a combination of several parameters such as usually only a doctor could judge (e.g. visual fields, cup/disc ratio and eye pressure in glaucoma)
- New educational material concerning your condition
- Other: *(specify all possibilities in this box)*

Nothing

In respect to your chronic condition - which kind of information is interesting for you to be stored? Which information should be available for you personally?

Matters of confidentiality of your data will be addressed in another question, so do not worry about it at this point.

(tick each scale once)

Personal data (e.g., name, date of birth, address)
uninteresting interesting

Medications you are currently taking, including dosage and time to take it
uninteresting interesting

Possible adverse effects of the medications you are taking
uninteresting interesting

Possible interactions of your medications to other medications
uninteresting interesting

Values measured by you or clinical staff (e.g. blood sugar, blood pressure)
uninteresting interesting

Other: *(specify all possibilities in this box)*

B. How could we help?

In part A, we have asked you to slip into the role of the designer of the helper. We now would like you to think about how you imagine this device. What does it look like? How does it work? How should interesting information about your condition be sent to you?

How should your attention be drawn to any new condition-related information? We can send you some kind of signal. This signal may differ depending on the medical priority of the information and on the time of the day.

Here a high medical priority means any change in your condition which requires immediate action by your doctor. A medium medical priority would be an issue which is important within in a time range of a month and a low priority is relevant over the course of a year.

(tick several boxes per row)

Signal	Never	Low medical priority	Medium medical priority	High medical priority	On business	At home	At night
Audio	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Visual	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vibration	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

In what way would you like to receive condition-related information, depending on the medical priority of the cause of the notification and the time of the day?

(tick several boxes if applicable)

Medium	Never	Low medical priority	Medium medical priority	High medical priority	On business	At home	At night
Text message	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Voice message	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E-Mail	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Printout of the notification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Automatic entry into your electronic health record	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other medium: <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The helper will have to be some kind of technical device. There are various options - it could be a mobile phone, a home computer or a device especially designed to carry around (similar to a watch) - just to name a few.

We would like to know what kind of device would suit you best. Again, consider several possibilities and tell us for all of them under which circumstances they might or might not suit you.

Which devices would you like to use to get informed about condition-related issues, depending on the medical priority of the cause of the notification and the time of the day?

(tick several boxes if applicable)

Device	Never	Low medical priority	Medium medical priority	High medical priority	On business	At home	At night
Home computer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Notebook	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PDA such as palmtop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Home or work phone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mobile phone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mobile device such as a device integrated into a watch	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Doctor's visit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other device: <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

We could give you a personal device as little helper to store and show information on your condition to you and help you to remember things. If you had such a personal helper how would it work?

How would your device present the information it holds?

(tick several boxes if applicable)

Visual display

Voice feedback

Other output:

How would you tell it what to do?

(tick several boxes if applicable)

Voice input

Manual input (e.g. keyboard, mouse, touchpad, buttons to press)

Other input:

The device we are planning could offer different modes of functionality. They would determine the ease with which you could use this technical device but they as well determine how sophisticated the functions offered by the device would be.

A device could do a lot, or just a little. How much would you like it to do? How much control would you like to have?

(tick several boxes if applicable)

- Basic functions with easy-to-use interaction
- Several different settings where you can choose from that one which best suits your personal circumstances
- Advanced functions which you can adjust yourself according to your personal circumstances

If the device had only one level of functionality, which would you like?

(tick scale once)

basic functionality with easy-to-use interaction ○ ○ ○ ○ ○ sophisticated functionality with an interface one has to learn

C. Confidentiality and further requirements

Clearly, a combination of personal and medical data is sensitive and confidential. No one would leave their medical details lying around at some piece of paper at a public place, neither would they send these data on a postcard to anyone. At the same time you expect your doctor to keep record of your condition and not to lose any sort of documentation.

These are matters which have to be addressed for electronic healthcare documentation and the electronic transfer of medical data. What do you think about these issues of confidentiality?

If you were using some kind of technical support for the management of your condition, how concerned would you be about issues of confidentiality?

(tick each scale once)

Holding information about you in confidence, so that only authorized persons have access (e.g. you and your doctor)
not concerned ○ ○ ○ ○ ○ highly concerned

Ensuring that data about you is not changed, damaged or lost
not concerned ○ ○ ○ ○ ○ highly concerned

The origin of your data, i.e. who has created them, is known
not concerned ○ ○ ○ ○ ○ highly concerned

Creator of data cannot deny that he created the data (This might be important if mistakes occur)
not concerned ○ ○ ○ ○ ○ highly concerned

Moreover, if you really try to imagine the supportive device we have been talking about, you might actually come up with ideas we have not mentioned here. Please tell anything which comes up to your mind!!

If you were using some kind of technical support for the management of your condition, are there any issues that would be of importance to you such as the size or design of the supporting device or other issues?

(specify)

P. Personal data

I now would like to collect some personal data, so that I can put your replies into a greater context.

Can you additionally be described by any other status than patient? If so, which?

(tick several boxes if applicable)

- Doctor
- Nurse
- Computer Scientist
- Other status relevant to area of healthcare:
- None

How many hours a week do you use a computer?

(specify number of hours)

hours

What is the length of your experience in using computers?

(specify number of years)

years

How experienced are you in using computers and other technical items?

(tick scale once)

inexperienced very experienced

Which gender are you?

(tick once)

Female

Male

Give your age on 1st January 2005

(specify number of years)

years

What is your nationality?

What condition do you have?

(tick several boxes if applicable)

Glaucoma

Diabetes

Cardio-vascular problems

Chronic heart-disease

Other:

Now please submit your answers.

Thank you very much for helping me!

If you have any comments about this questionnaire, please let me know:



Doris Jung is a PhD student at the University of Waikato, New Zealand. She is writing her PhD thesis with the Department of Computer Science. If you have any questions about this project, you can contact her via e-mail at d (dot) jung (at) cs (dot) waikato (dot) ac (dot) nz.



Requirements Survey for the Project "A Mobile Alerting Device for the Support of Patients with Chronic Conditions"

Introduction

Imagine that your patients had a little supportive helper for the management of their chronic conditions. This helper could remind you and them of condition-related issues or store vital information and supply you and your patients with this information whenever needed.

It could also be used for communication purposes between you and your patients.

In addition, you would have the chance to perform a permanent analysis of values which are fed into the helper by your patients. These values could be parameters such as blood sugar values or blood pressure values. You would not have to sit down for each analysis but only once and explain the helper what would be relevant for you (or choose from predefined analyses). Thus, whenever treatment changes are required you would notice this at once.

A. What do you need?

Imagine you were the designer of the helper. What should the helper do to help you and your patients? How would you like being supported? How should your patients be supported? I would like you to think about that for a moment and then tell me about it in the next questions - especially in the "Other" boxes.

What kind of things should your patients be reminded of?

(tick several boxes if applicable)

- Taking the correct amount of medicine
- Taking medicine at the correct time
- Taking the correct type of medicine
- Getting a new prescription
- Doctor appointments
- Critical values of their readings, e.g. too high blood sugar values
- Critical values of their parameters involving a combination of several parameters such as usually only you could judge (e.g. visual fields, cup/disc ratio and eye pressure in glaucoma)
- New educational material concerning their conditions
- Other: *(specify all possibilities in this box)*

Nothing

What kind of things would you like to be alerted of?

(tick several boxes if applicable)

Critical values of your patients' readings, e.g. too high blood sugar values

Critical values of their parameters involving a combination of several parameters such as usually only you could judge (e.g. visual fields, cup/disc ratio and eye pressure in glaucoma)

New educational material concerning your patients' condition

Other: *(specify all possibilities in this box)*

Nothing

In respect to your patients' chronic conditions - which kind of information should be stored? Which information should be available for your patients personally?

Matters of confidentiality of their data will be addressed in another question, so do not worry about it at this point.

(tick each scale once)

Personal data (e.g., name, date of birth, address)

uninteresting interesting

Medications they are currently taking, including dosage and time to take them

uninteresting interesting

Possible adverse effects of the medications they are taking

uninteresting interesting

Possible interactions of their medications to other medications

uninteresting interesting

Values measured by them or clinical staff (e.g. blood sugar, blood pressure)

uninteresting interesting

Other: *(specify all possibilities in this box)*

Again, in respect to your patients' chronic conditions - which kind of information is interesting for you to be stored? Which information should be available for you?

Matters of confidentiality of their data will be addressed in another question, so do not worry about it at this point.

(tick each scale once)

Personal data (e.g., name, date of birth, address)

uninteresting interesting

Medications they are currently taking, including dosage and time to take them

uninteresting interesting

Possible adverse effects of the medications they are taking

uninteresting interesting

Possible interactions of their medications to other medications

uninteresting interesting

Values measured by them or clinical staff (e.g. blood sugar, blood pressure)

uninteresting interesting

Other: *(specify all possibilities in this box)*

Automatic entry into the patient's electronic health record	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other medium: <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The patient's helper will send out information to you. You will have to receive this information with some kind of technical device. There are various options for this device - it could be a mobile phone, a pager, a home computer or a device especially designed to carry around (similar to a watch) - just to name a few.

We would like to know what kind of device would suit you best. Again, consider several possibilities and tell us for all of them under which circumstances they might or might not suit you.

Which devices would you like to use to get informed about issues relating to your patients' conditions, depending on the medical priority of the cause of the notification and the time of the day?

(tick several boxes if applicable)

Device	Never	Low medical priority	Medium medical priority	High medical priority	On business	At home	At night
Home computer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Notebook	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PDA such as palmtop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Home or work phone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mobile phone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mobile device such as a device integrated into a watch	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Patient's visit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other device: <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

If we gave you a device which could be used to communicate with your patient's helper. How should it work?

How would your device present the information it holds?

(tick several boxes if applicable)

Visual display

Voice feedback

Other output:

How would you tell it what to do?*(tick several boxes if applicable)*

- Voice input
- Manual input (e.g. keyboard, mouse, touchpad, buttons to press)
- Other input:

The patients' helper and your counterpart it communicates with could offer different modes of functionality. These modes would determine the ease with which you could use them but the modes as well determine how sophisticated the functions offered by the devices would be.

A device could do a lot, or just a little. How much would you like it to do? How much control would you like to have?

(tick several boxes if applicable)

- Basic functions with easy-to-use interaction
- Several different settings where you can choose from that one which best suits your personal circumstances
- Advanced functions which you can adjust yourself according to your personal circumstances

If the devices had only one level of functionality, which would you like?

(tick scale once)

basic functionality with easy-to-use interaction



sophisticated functionality with an interface one has to learn

C. Confidentiality and further requirements

Clearly, a combination of personal and medical data is sensitive and confidential. No one would leave their medical details lying around at some piece of paper at a public place, neither would they send these data on a postcard to anyone. At the same time you are expected to keep record of your patients' conditions and not to lose any sort of documentation.

These are matters which have to be addressed for electronic healthcare documentation and the electronic transfer of medical data. What do you think about these issues of confidentiality?

If you were using some kind of technical support for the management of your patients' condition, how concerned would you be about issues of confidentiality?

(tick each scale once)

Holding information about patients in confidence, so that only authorized persons have access (e.g. patient and you)
not concerned highly concerned

Ensuring that data about your patients is not changed, damaged or lost
not concerned highly concerned

The origin of your patients' data, i.e. who has created them, is known
not concerned highly concerned

Creator of data cannot deny that he created the data (This might be important if mistakes occur)
not concerned highly concerned

Moreover, if you really try to imagine the supportive device we have been talking about, you might actually come up with ideas we have not mentioned here. Please tell anything which comes up to your mind!!

If you were using some kind of technical support for the management of your patients' conditions, are there any issues that would be of importance to you such as the size or design of the supporting device or other issues?

(specify)

D. Your clinic/practice

We are interested how the patient's helper and your counterpart integrate into the existing infrastructure of your clinic or practice. Therefore we have some questions addressing this matter.

Do you know what Information System is used in your clinic/practice?

(tick several if applicable)

SAP

Soprano (Orion International)

Soarian (Siemens)

Other system:

Not known

Are Electronic Health Records used in your clinic/practice?*(tick once)*

- Yes
- No

Which clinic/practice are you working at?**Do you happen to know what standards for the exchange, management and integration of data are used in your clinic?***(tick several if applicable)*

- HL7
- Dicom
- Other Standard: *(specify all possibilities in this box)*

- Not known

P. Personal data

I now would like to collect some personal data, so that I can put your replies into a greater context.

Can you additionally be described by any other status than doctor? If so, which?

(tick several boxes if applicable)

- Patient

- Nurse
- Computer Scientist
- Other status relevant to area of healthcare:
- None

How many hours a week do you use a computer?

(specify number of hours)

hours

What is the length of your experience in using computers?

(specify number of years)

years

How experienced are you in using computers and other technical items?

(tick scale once)

inexperienced very experienced

Which gender are you?

(tick once)

Female

Male

Give your age on 1st January 2005

(specify number of years)

years

What is your nationality?

What conditions do your patients have?

(tick several boxes if applicable)

- Glaucoma
- Diabetes
- Cardio-vascular problems
- Chronic heart-disease
- Other Conditions: *(specify all possibilities in this box)*

Now please submit your answers.

Submit answers

Thank you very much for helping me!

If you have any comments about this questionnaire, please let me know:



Doris Jung is a PhD student at the University of Waikato, New Zealand. She is writing her PhD thesis with the Department of Computer Science. If you have any questions about this project, you can contact her via e-mail at d (dot) jung (at) cs (dot) waikato (dot) ac (dot) nz.



Requirements Survey for the Project "A Mobile Alerting Device for the Support of Patients with Chronic Conditions"

Introduction

Imagine that your patients had a little supportive helper for the management of their chronic conditions. This helper could remind you and them of condition-related issues or store vital information and supply you and your patients with this information whenever needed.

It could also be used for communication purposes between you and your patients.

In addition, you would have the chance to perform a permanent analysis of values which are fed into the helper by your patients. These values could be parameters such as blood sugar values or blood pressure values. You would not have to sit down for each analysis but only once and explain the helper what would be relevant for you (or choose from predefined analyses). Thus, whenever treatment changes are required you would notice this at once.

A. What do you need?

Imagine you were the designer of the helper. What should the helper do to help you and your patients? How would you like being supported? How should your patients be supported? I would like you to think about that for a moment and then tell me about it in the next questions - especially in the "Other" boxes.

What kind of things should your patients be reminded of?

(tick several boxes if applicable)

- Taking the correct amount of medicine
- Taking medicine at the correct time
- Taking the correct type of medicine
- Getting a new prescription
- Doctor appointments
- Critical values of their readings, e.g. too high blood sugar values
- Critical values of their parameters involving a combination of several parameters such as usually only medical staff could judge (e.g. visual fields, cup/disc ratio and eye pressure in glaucoma)
- New educational material concerning their conditions
- Other: *(specify all possibilities in this box)*

Nothing

What kind of things would you like to be reminded of?

(tick several boxes if applicable)

Giving the correct amount of medicine

Giving medicine at the correct time

Giving the correct type of medicine

Doctor consultations

Critical values of your patients' readings, e.g. too high blood sugar values

Critical values of their parameters involving a combination of several parameters such as usually only you could judge (e.g. visual fields, cup/disc ratio and eye pressure in glaucoma)

New educational material concerning your patients' conditions

Other: *(specify all possibilities in this box)*

Nothing

In respect to your patients' chronic conditions - which kind of information should be stored? Which information should be available for your patients personally?

Matters of confidentiality of their data will be addressed in another question, so do not worry about it at this point.

(tick each scale once)

Personal data (e.g., name, date of birth, address)

uninteresting interesting

Medications they are currently taking, including dosage and time to take them

uninteresting interesting

Possible adverse effects of the medications they are taking

uninteresting interesting

Possible interactions of their medications to other medications

uninteresting interesting

Values measured by them or clinical staff (e.g. blood sugar, blood pressure)

uninteresting interesting

Other: *(specify all possibilities in this box)*

Again, in respect to your patients' chronic conditions - which kind of information is interesting for you to be stored? Which information should be available for you?

Matters of confidentiality of their data will be addressed in another question, so do not worry about it at this point.

(tick each scale once)

Personal data (e.g., name, date of birth, address)

uninteresting interesting

Medications they are currently taking, including dosage and time to take them

uninteresting interesting

Possible adverse effects of the medications they are taking

uninteresting interesting

Possible interactions of their medications to other medications

uninteresting interesting

Values measured by them or clinical staff (e.g. blood sugar, blood pressure)

uninteresting interesting

Other: *(specify all possibilities in this box)*

Automatic entry into the patient's electronic health record	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other medium: <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The patient's helper will send out information to you. You will have to receive this information with some kind of technical device. There are various options for this device - it could be a mobile phone, a pager, a home computer or a device especially designed to carry around (similar to a watch) - just to name a few.

We would like to know what kind of device would suit you best. Again, consider several possibilities and tell us for all of them under which circumstances they might or might not suit you.

Which devices would you like to use to get informed about issues relating to your patients' conditions, depending on the medical priority of the cause of the notification and the time of the day?

(tick several boxes if applicable)

Device	Never	Low medical priority	Medium medical priority	High medical priority	On business	At home	At night
Home computer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Notebook	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PDA such as palmtop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Home or work phone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mobile phone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mobile device such as a device integrated into a watch	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other device: <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

If we gave you a device which could be used to communicate with your patient's helper. How should it work?

How would your device present the information it holds?

(tick several boxes if applicable)

Visual display

Voice feedback

Other output:

How would you tell it what to do?*(tick several boxes if applicable)*

- Voice input
- Manual input (e.g. keyboard, mouse, touchpad, buttons to press)
- Other input:

The patients' helper and your counterpart it communicates with could offer different modes of functionality. These modes would determine the ease with which you could use them but the modes as well determine how sophisticated the functions offered by the devices would be.

A device could do a lot, or just a little. How much would you like it to do? How much control would you like to have?*(tick several boxes if applicable)*

- Basic functions with easy-to-use interaction
- Several different settings where you can choose from that one which best suits your personal circumstances
- Advanced functions which you can adjust yourself according to your personal circumstances

If the devices had only one level of functionality, which would you like?*(tick scale once)*

basic functionality with easy-to-use interaction ○ ○ ○ ○ ○ sophisticated functionality with an interface one has to learn

C. Confidentiality and further requirements

Clearly, a combination of personal and medical data is sensitive and confidential. No one would leave their medical details lying around at some piece of paper at a public place, neither would they send these data on a postcard to anyone. At the same time you are expected to keep record of your patients' conditions and not to lose any sort of documentation.

These are matters which have to be addressed for electronic healthcare documentation and the electronic transfer of medical data. What do you think about these issues of confidentiality?

If you were using some kind of technical support for the management of your patients' condition, how concerned would you be about issues of confidentiality?

(tick each scale once)

Holding information about patients in confidence, so that only authorized persons have access (e.g. patient and you)
not concerned ○ ○ ○ ○ ○ highly concerned

Ensuring that data about your patients is not changed, damaged or lost
not concerned highly concerned

The origin of your patients' data, i.e. who has created them, is known
not concerned highly concerned

Creator of data cannot deny that he created the data (This might be important if mistakes occur)
not concerned highly concerned

Moreover, if you really try to imagine the supportive device we have been talking about, you might actually come up with ideas we have not mentioned here. Please tell anything which comes up to your mind!!

If you were using some kind of technical support for the management of your patients' conditions, are there any issues that would be of importance to you such as the size or design of the supporting device or other issues?

(specify)

D. Your clinic/practice

We are interested how the patient's helper and your counterpart integrate into the existing infrastructure of your clinic or practice. Therefore we have some questions addressing this matter.

Do you know what Information System is used in your clinic/practice?

(tick several if applicable)

SAP

Soprano (Orion International)

Soarian (Siemens)

Other system:

Not known

Are Electronic Health Records used in your clinic/practice?

(tick once)

- Yes
- No

Which clinic/practice are you working at?

Do you happen to know what standards for the exchange, management and integration of data are used in your clinic?

(tick several if applicable)

- HL7
- Dicom
- Other Standard: *(specify all possibilities in this box)*

- Not known

P. Personal data

I now would like to collect some personal data, so that I can put your replies into a greater context.

Can you additionally be described by any other status than nurse? If so, which?

(tick several boxes if applicable)

- Doctor
- Patient

Computer Scientist

Other status relevant to area of healthcare:

None

How many hours a week do you use a computer?

(specify number of hours)

hours

What is the length of your experience in using computers?

(specify number of years)

years

How experienced are you in using computers and other technical items?

(tick scale once)

inexperienced very experienced

Which gender are you?

(tick once)

Female

Male

Give your age on 1st January 2005

(specify number of years)

years

What is your nationality?

What conditions do your patients have?

(tick several boxes if applicable)

- Glaucoma
- Diabetes
- Cardio-vascular problems
- Chronic heart-disease
- Other Conditions: *(specify all possibilities in this box)*

Now please submit your answers.

Submit answers

Thank you very much for helping me!

If you have any comments about this questionnaire, please let me know:

Doris Jung is a PhD student at the University of Waikato, New Zealand. She is writing her PhD thesis with the Department of Computer Science. If you have any questions about this project, you can contact her via e-mail at d (dot) jung (at) cs (dot) waikato (dot) ac (dot) nz.



Requirements Survey for the Project "A Mobile Alerting Device for the Support of Patients with Chronic Conditions"

Introduction

Imagine that patients had a little supportive helper for the management of their chronic conditions. This helper could remind them and medical staff of condition-related issues or store vital information and supply them with this information whenever needed.

It could also be used for communication purposes between patients and medical staff.

In addition, doctors would have the chance to perform a permanent analysis of values which are fed into the helper by patients. These values could be parameters such as blood sugar values or blood pressure values. Doctors would not have to sit down for each analysis but only once and explain the helper what would be relevant for them (or choose from predefined analyses). Thus, whenever treatment changes are required doctors would notice this at once.

A. Data security

Clearly, a combination of personal and medical data is sensitive and confidential. For permanent remote analyses of new condition-relevant data of a patient and notifications about the results of these analyses we obviously have to handle such a combination of personal and medical data.

Also doctors are expected to keep record of their patients' conditions and not to lose any sort of documentation.

These are matters which have to be addressed for electronic healthcare documentation and the electronic transfer of medical data. What do you think about these issues of confidentiality?

If you were working at the IT department of a clinic how concerned would you be about issues of data security?

(tick each scale once)

Holding information about patients in confidence, so that only authorized persons have access
not concerned highly concerned

Ensuring that data about your patients is not changed, damaged or lost
not concerned highly concerned

The origin of patients' data, i.e. who has created them, is known
not concerned highly concerned

Creator of data cannot deny that he created the data (This might be important if mistakes occur)
not concerned highly concerned

B. Your clinic/practice

(Answer section B. only if you work at a clinic or practice (or know the infrastructure of one), otherwise proceed with section C.)

We are interested how the patient's helper and its counterpart at the clinic/practice integrate into the existing infrastructure of your clinic or practice. Therefore we have some questions addressing this matter.

What Information System is used in your clinic/practice?

(tick several if applicable)

- SAP
- Soprano (Orion International)
- Soarian (Siemens)
- Other system:
- Not known

Are Electronic Health Records used in your clinic/practice?

(tick once)

- Yes
- No

Which clinic/practice are you working at?

What standards for the exchange, management and integration of data are used in your clinic?

(tick several if applicable)

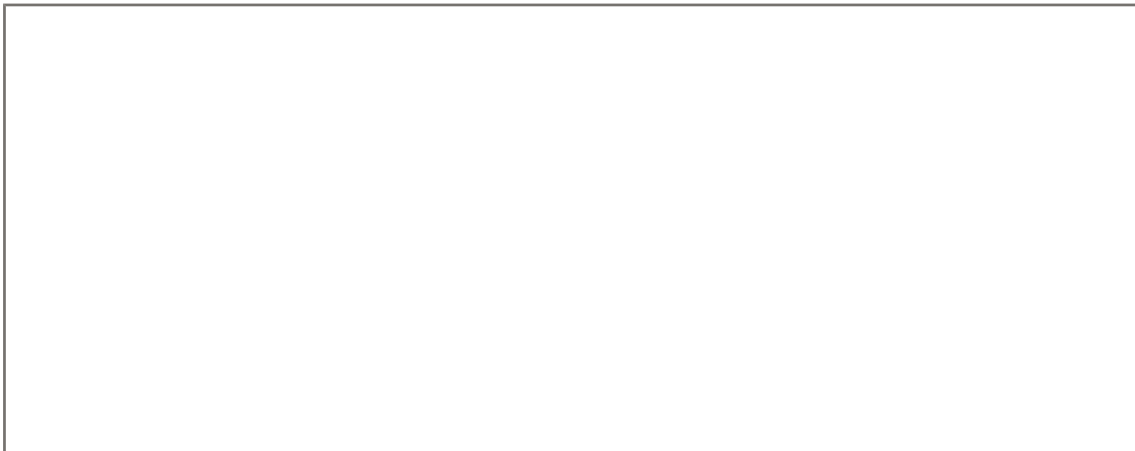
- HL7
- Dicom
- Other Standard: *(specify all possibilities in this box)*



Not known

What further technical details come to your mind as being important?

(specify)



C. Further Requirements

Moreover, if you really try to imagine the supportive device we have been talking about, you might actually come up with thoughts we have not mentioned here. Please tell anything which comes up to your mind!!

What further requirements can you think of in respect to a mobile alerting device for the support of patients with chronic conditions?

(specify)

P. Personal data

I now would like to collect some personal data, so that I can put your replies into a greater context.

Can you additionally be described by any other status than computer scientist? If so, which?

(tick several boxes if applicable)

- Patient
- Nurse
- Doctor
- Other status relevant to area of healthcare:
- None

How many hours a week do you use a computer?

(specify number of hours)

hours

What is the length of your experience in using computers?

(specify number of years)

years

How experienced are you in using computers and other technical items?

(tick scale once)

inexperienced very experienced

Which gender are you?

(tick once)

Female

Male

Give your age on 1st January 2005

(specify number of years)

years

What is your nationality?

Now please submit your answers.

Thank you very much for helping me!

If you have any comments about this questionnaire, please let me know:



Doris Jung is a PhD student at the University of Waikato, New Zealand. She is writing her PhD thesis with the Department of Computer Science. If you have any questions about this project, you can contact her via e-mail at d (dot) jung (at) cs (dot) waikato (dot) ac (dot) nz.

Appendix B. Health Care Provider Interviews

The University of Waikato · School of Computing and Mathematical Sciences

Research Participant's Bill of Rights

The following is a list of your rights if you participate in a research project organised within the School of Computing and Mathematical Sciences at the University of Waikato.

As a research participant, you have the right:

- To be treated with respect and dignity in every phase of the research.
- To be fully and clearly informed of all aspects of the research prior to becoming involved in it.
- To enter into clear, informed, and written agreement with the researcher prior to becoming involved in the activity. You should sense NO pressure, explicit or otherwise, to sign this contract.
- To choose explicitly whether or not you will become involved in the research under the clearly stated provision that refusal to participate or the choice to withdraw during the activity can be made at any time without penalty to you.
- To be treated with honesty, integrity, openness, and straightforwardness in all phases of the research, including a guarantee that you will not unknowingly be deceived during the course of the research.
- To receive something in return for your time and energy.
- To demand proof that an independent and competent ethical review of human rights and protections associated with the research has been successfully completed.
- To demand complete personal confidentiality and privacy in any reports of the research unless you have explicitly negotiated otherwise.
- To expect that your personal welfare is protected and promoted in all phases of the research, including knowing that no harm will come to you.
- To be informed of the results of the research study in a language you understand.
- To be offered an range of research studies or experiences from which to select, if the research is part of fulfilling your educational or employment goals.

The contents of this bill were prepared by the University of Calgary who examined all of the relevant Ethical Standards from the Canadian Psychological Association's Code of Ethics for Psychologists, 1991 and rewrote these to be of relevance to research participants.

Descriptions of the CPA Ethical Code and the CPA Ethical Standards relevant to each of these rights are available at <http://www.cpa.ca/ethics2000.html> and <http://www.psych.ucalgary.ca/Research/ethics/bill/billcode.html> if you would like to examine them.

The complete CPA Ethical Code can be found in Canadian Psychological Association "*Companion manual for the Canadian Code of Ethics for Psychologists*" (1992).

The University of Waikato · School of Computing and Mathematical Sciences

Research Consent Form

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

Research Project Title

Adaptive Alerting Systems in Healthcare (Subproject: Collaborative Profiles for Alerting Systems in Healthcare).

Researcher

Ms Doris Jung.

Experiment Purpose

The purpose of this interview is to find out about the usefulness of collaborative profiles for alerting systems in healthcare (see description of the project) and to collect first hand application scenarios from healthcare staff.

Participant Recruitment and Selection

Doctors and nurses from New Zealand, Germany and potentially from the UK are being recruited for this experiment.

Procedure

This session will require about 30 minutes of your time. You will be explained the idea to be discussed and you will be presented some figures of examples for the idea. Then you will be asked several questions regarding the usefulness of my idea and your expectations regarding it. Subsequently, I will ask you about further examples from your daily work, where my idea could be used.

None of the tasks is a test – my objective is to find out your personal opinion and expectations towards my approach and to receive an overview of how my idea could support you in your daily work.

Data Collection

The interviewer will take notes. If you agree to it, an audio recording of the interview might take place.

Confidentiality

Confidentiality and participant anonymity will be strictly maintained. All information gathered will be used for statistical analysis only and no names or other identifying characteristics will be stated in the final or any other reports.

Likelihood of Discomfort

There is no likelihood of discomfort or risk associated with participation.

Researcher

Doris Jung is working on her doctorate in the Computer Science Department at the University of Waikato. This study will contribute to her research “Adaptable Alerting Systems in Healthcare”. Her supervisor is Dr Annika Hinze.

Doris can be contacted in room G2.06 of the School of Computer and Mathematical Sciences building at the University of Waikato. Her phone number is 07 838 4466 ext 6011 and her email address is d.jung@cs.waikato.ac.nz.

Finding out about Results

The Participants can find out the results of the study by contacting the researcher after March 1, 2006.

Agreement

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive you legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to not answer specific items or questions in interviews or on questionnaires. You are free to withdraw from the study at any time without penalty. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact the researcher.

Participant

Date

Investigator/Witness

Date

A copy of this consent form has been given to you to keep for your records and reference.

Interviews for the Project

“Collaborative Profiles for Alerting Systems in Healthcare”

Description of the Project

These interviews will be undertaken to help me finding out several issues about a system I am researching for my PhD. This system is going to help patients with chronic conditions and medical staff in the management of these chronic diseases.

I propose a small mobile device that can be used by patients with chronic conditions to enhance the management of their disease by reminding the patient of important issues and storing treatment-relevant data in detail. So the condition of the patient can be monitored and doctors can be notified if emergency action is required. Moreover, the patient himself may hold a copy of condition-relevant parameters himself, which enables him to lead his life more independently. Patients may be reminded of condition-related issues such as taking the appropriate medicine at the correct time, doctor appointments or adverse effects of medications to the medication they are already taking. Important parameters gained in the self-reliant handling of the patient's disease such as measurements of blood sugar or blood pressure may be stored into the device and automatically transferred to the doctor's electronic health record. Data can be filtered according to the doctor's guidelines, i.e. data will automatically analysed for unusual values. Thereby critical conditions requiring emergency treatment or general changes required in the treatment plan will be recognized earlier and in doing so can be treated earlier. These guidelines could be default values or be defined in the beginning of treatment and for each change in treatment. The aim of this device is to improve patient compliance in the treatment of their condition, to relieve doctors of organisational bureaucratic work and thereby to improve the overall treatment results.

For the process of filtering patient parameters, someone initially has to define the guidelines (profiles) according to which the parameters are filtered. For my PhD I suggest a new concept of collaborative profiles. Several people, e.g. doctors, patients, nurses and pharmacists, may be collaboratively included in the definition process of profiles in order to fully exploit everyone's expert knowledge.

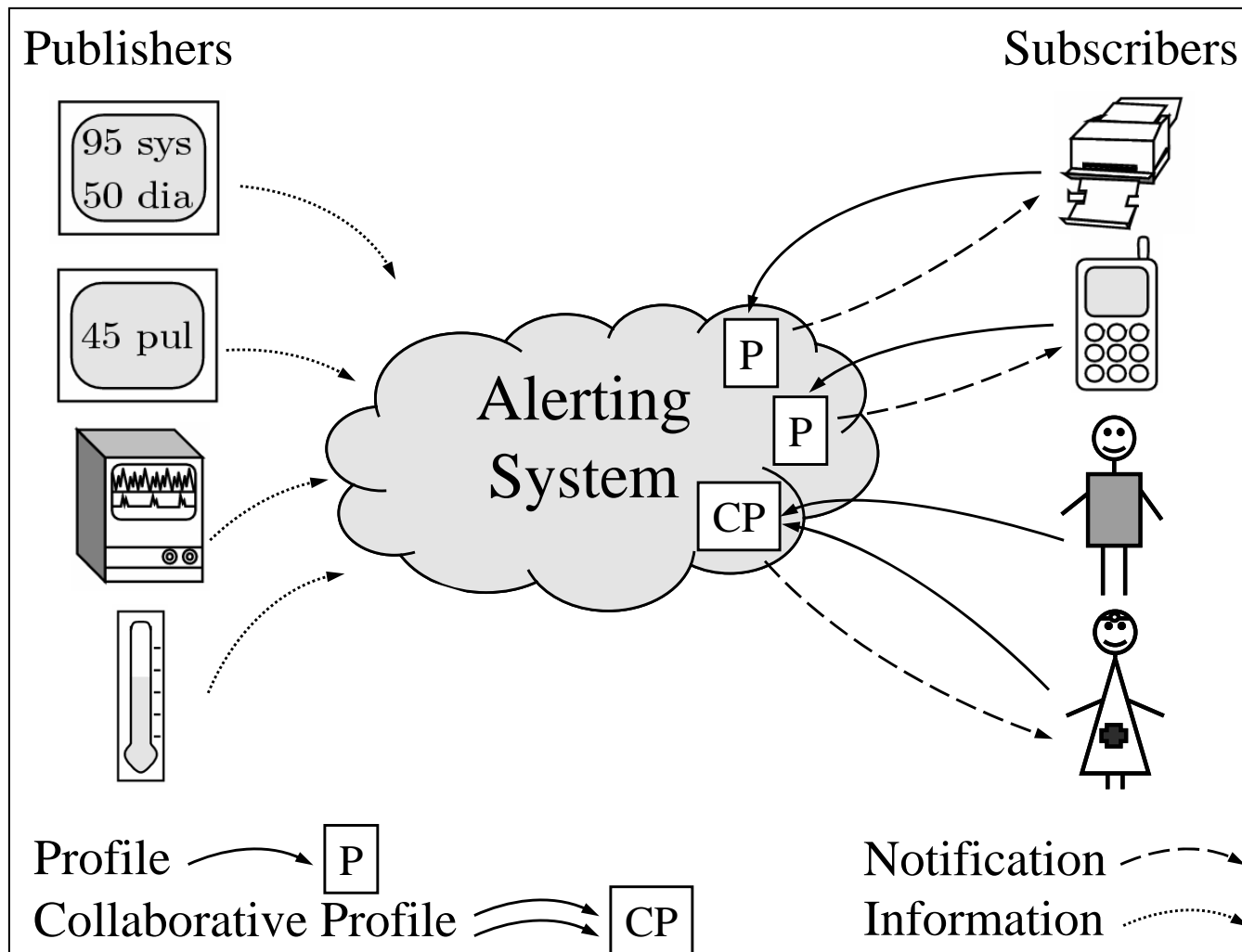
With these interviews I want to discuss the usefulness of a mobile alerting system and the concept of collaborative profiles with doctors. Furthermore, I want to find out about the expectations of doctors regarding collaborative profiles, and collect possible scenarios for their application, deduced from the daily work of doctors.

Description of the Interview

Before starting the interview, I will inform the participant about the “Research Participant's Bill of Rights” and the “Research Consent Form”.

Participants will be introduced to the ideas of a mobile alerting system and collaborative profiles using a suitable language and a figure of such a system to illustrating its use. Next, they will be shown figures of some examples for collaborative profiles. Afterwards, they will be asked for their opinion and their expectations regarding the concept of collaborative profiles and the usefulness of mobile alerting systems. Subsequently, they will be asked to think about examples of their daily work in which they and their patients should be supported with a mobile alerting system and especially collaborative profiles.

Alerting System



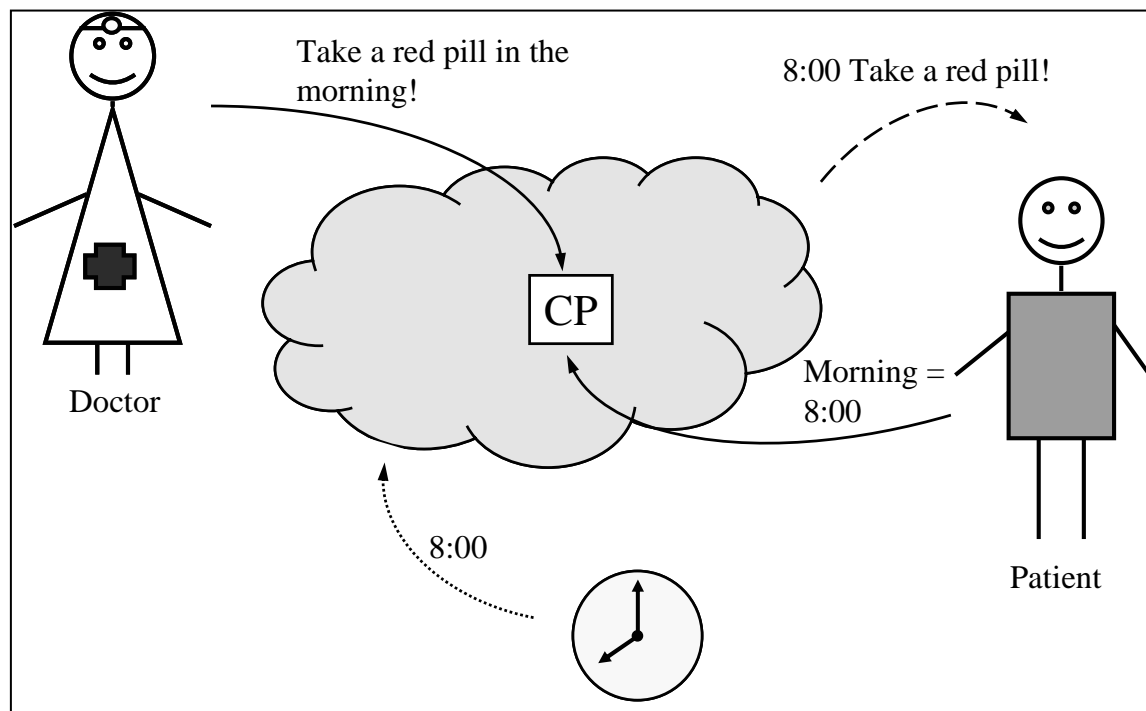


Example 1: Collaborative Profile and Medication

Example for PATIENT ALERT

Profile Doctor: Take 1 red pill in the morning

Profile Patient: Specifies exact time (e.g. baker gets up much earlier than student + might change during holidays)



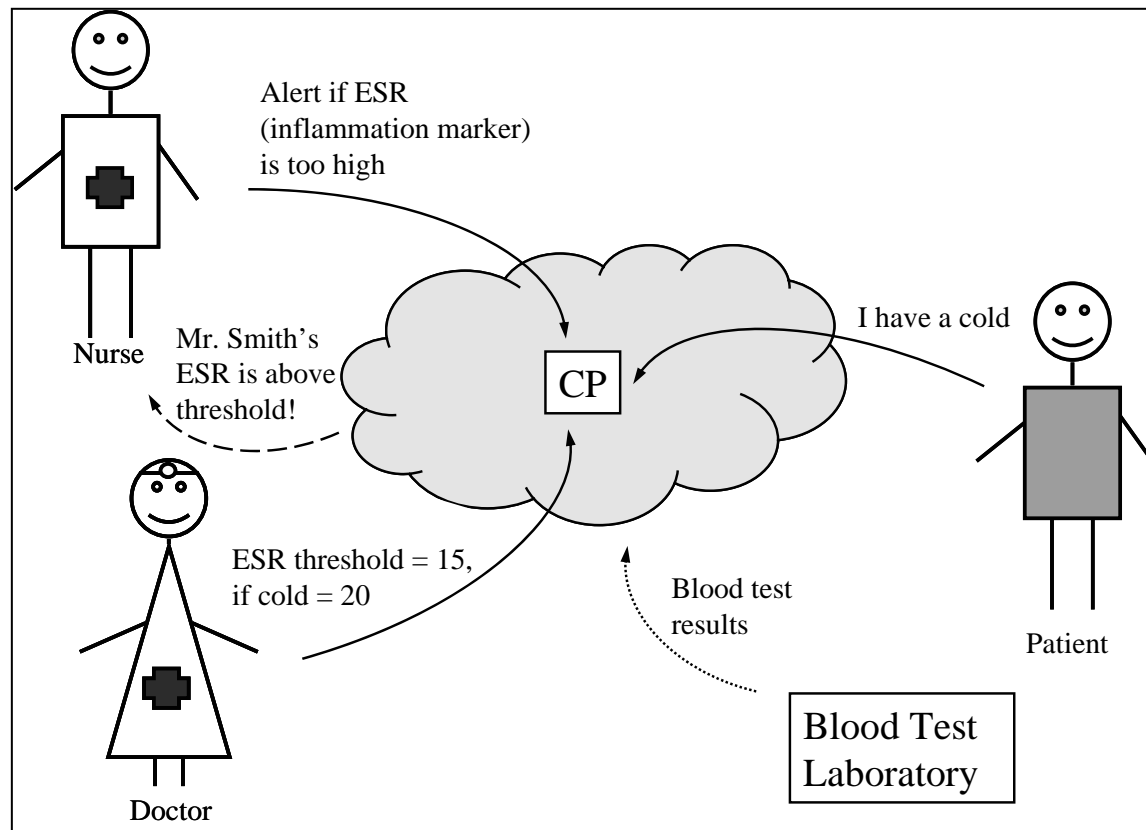
Example 2: Collaborative Profile and IBD

Example for NURSE ALERT

Profile Nurse: Alert when ESR (inflammation marker used for IBD) is too high

Profile Doctor: ESR threshold is 15, if patient has a cold it is 20

Profile Patient: I have a cold





Example 3: Collaborative Profile and Operation

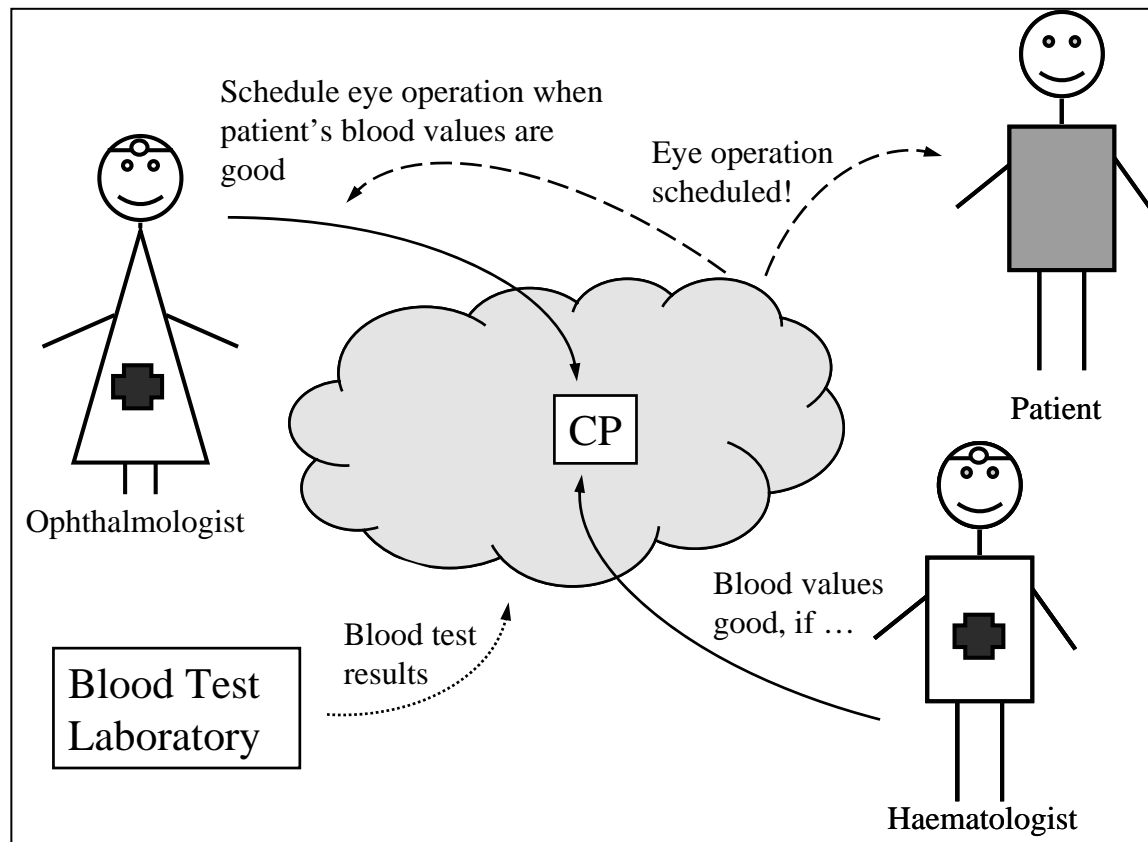
Example for DOCTOR/PATIENT ALERT

Profile Ophthalmologist:

Schedule eye operation when patient's blood values are good

Profile Haematologist:

Specifies what good blood values are





Interviews on Collaborative Profiles for Mobile Alerting Systems in Healthcare

Questions

1. To what extent do you consider mobile alerting systems to be useful for chronic patients and their healthcare providers?
2. To what degree is the concept of collaborative profiles useful for chronic patients and their healthcare providers?
3. What do you expect from collaborative profiles?
4. Can you give me examples from your daily work in which a mobile alerting system with collaborative profiles could support you?



Interviews on Collaborative Profiles for Mobile Alerting Systems in Healthcare

Personal Data

1. What conditions do your patients have?
2. How many hours a week do you use a computer?
3. For how many years have you been using computers?
4. How experienced are you in using computers and other technical items?
Inexperienced OOOOO very experienced
5. Which gender are you?
O Female O Male
6. What age were you on 1st January 2005?
7. What is your nationality?

Appendix C. Graphical Profile Definition Language Study: Participant Workbook



User Study of a Graphical Profile Definition Language

Researcher Doris Jung

Participant Workbook

Session Number:

Date and Time:

Evaluation Task Order:

The University of Waikato School of Computing and Mathematical Sciences

Research Participant's Bill of Rights

The following is a list of your rights if you participate in a research project organised within the School of Computing and Mathematical Sciences at the University of Waikato.

As a research participant, you have the right:

- To be treated with respect and dignity in every phase of the research.
- To be fully and clearly informed of all aspects of the research prior to becoming involved in it.
- To enter into clear, informed, and written agreement with the researcher prior to becoming involved in the activity. You should sense NO pressure, explicit or otherwise, to sign this contract.
- To choose explicitly whether or not you will become involved in the research under the clearly stated provision that refusal to participate or the choice to withdraw during the activity can be made at any time without penalty to you.
- To be treated with honesty, integrity, openness, and straightforwardness in all phases of the research, including a guarantee that you will not unknowingly be deceived during the course of the research.
- To receive something in return for your time and energy.
- To demand proof that an independent and competent ethical review of human rights and protections associated with the research has been successfully completed.
- To demand complete personal confidentiality and privacy in any reports of the research unless you have explicitly negotiated otherwise.
- To expect that your personal welfare is protected and promoted in all phases of the research, including knowing that no harm will come to you.
- To be informed of the results of the research study in a language you understand.
- To be offered an range of research studies or experiences from which to select, if the research is part of fulfilling your educational or employment goals.

The contents of this bill were prepared by the University of Calgary who examined all of the relevant Ethical Standards from the Canadian Psychological Association's Code of Ethics for Psychologists, 1991 and rewrote these to be of relevance to research participants.

Descriptions of the CPA Ethical Code and the CPA Ethical Standards relevant to each of these rights are available at <http://www.cpa.ca/ethics2000.html> and <http://www.psych.ualgary.ca/Research/ethics/bill/billcode.html> if you would like to examine them.

The complete CPA Ethical Code can be found in Canadian Psychological Association "*Companion manual for the Canadian Code of Ethics for Psychologists*" (1992).

The University of Waikato School of Computing and Mathematical Sciences**Research Consent Form**

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

Research Project Title

User Study of a Graphical Profile Definition Language.

(PhD Project Title: Realisation and Representation of Collaborative Profiles.)

Researcher

Doris Jung.

Experiment Purpose

This study will evaluate the usability of a graphical approach to interactive profile specification.

Participant Recruitment and Selection

Undergraduate and graduate students from the University of Waikato are being recruited for this study. Academic and non-academic staff members of any department at the University of Waikato will also be recruited for the study.

Procedure

This session will require about 1:15 hours of your time. You will be given a tutorial and a set of training tasks. Next, you will be asked to solve two sets of tasks writing/drawing the solution on paper. Please think aloud while solving these and the training tasks – describing how you try to solve the tasks. Afterwards we will have a brief discussion about how you experienced solving the tasks and you will be asked to fill in a short questionnaire about your background.

Data Collection

A questionnaire will be used to collect background information. As you are solving the task sets and think aloud the interviewer will take notes. The researcher also will use your task solving notes for the evaluation of the study.

Data Archiving/Destruction

Data will be kept securely stored by the researcher.

Confidentiality

Confidentiality and participant anonymity will be strictly maintained. All information gathered will be used for statistical analysis only and no names or other identifying characteristics will be stated in the final or any other reports.

Likelihood of Discomfort

There is no likelihood of discomfort or risk associated with participation.

Researcher

Doris Jung is working on her doctorate in the Computer Science Department at the University of Waikato. This study will contribute to her research on the realisation and representation of collaborative profiles. Her supervisor is Associate Professor Steve Jones.

Doris can be contacted in room G2.27 of the School of Computer and Mathematical Sciences building at the University of Waikato. Her phone number is 838 4547 and her email address is d.jung@cs.waikato.ac.nz.

Finding out about Results

The Participants can find out the results of the study by contacting the researcher after October 1, 2007.

Agreement

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to not answer specific items or questions in interviews or on questionnaires. You are free to withdraw from the study at any time without penalty. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact the researcher.

 Participant

 Date

 Investigator/Witness

 Date

A copy of this consent form has been given to you to keep for your records and reference.

User Study of a Graphical Profile Definition Language

Researcher: Doris Jung
Supervisor: Steve Jones

Tutorial

User Study of a Graphical
Profile Definition Language

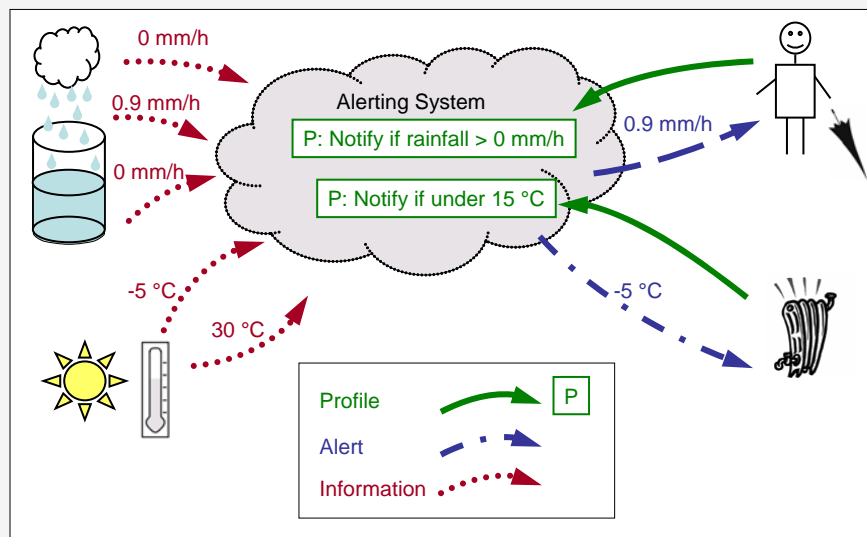
Tutorial 1

Example of Weather Observation

Temperature = 13 °C
 Wind Speed = 32 km/h
 Wind Direction = W
 Rainfall (last hr) = 0.2 mm
 Humidity = 74 %
 Pressure = 992 hPa

Tutorial 2

Introduction to Alerting Services



Tutorial 3

Notify if wind stronger than 50 km/h.

wind
>
50 km/h

Each **condition** is presented by a (grey) **box**.

Tutorial 4

Notify if temperature is different from 0 °C.

temperature
≠
0 °C

Tutorial 5

Notify if humidity is higher than 60 % and the temperature is more than 30 °C.

humidity
>
60 %

temperature
>
30 °C

If **all of several conditions** have to be fulfilled, place boxes **adjacent beneath** each other.

Tutorial 6

Notify if humidity is more than 65 % or the temperature is above 25 °C or the pressure is higher than 992 hPa.

humidity
>
65 %

temperature
>
25 °C

pressure
>
992 hPa

If **one of several conditions** has to be fulfilled, place boxes **with a gap beneath** each other.

Tutorial 7

Notify if humidity is higher than 60 % and the temperature is more than 30 °C, both happening within 2 minutes.

humidity > 60 %
temperature > 30 °C

2:00 min
↔

An **annotated arrow** gives a **duration**. The desired conditions have to be fulfilled within the given time.

Tutorial 8

Notify if the wind repeatedly is stronger than 60 km/h, i.e. the wind is stronger than 60 km/h for at least three times in an hour.

time →

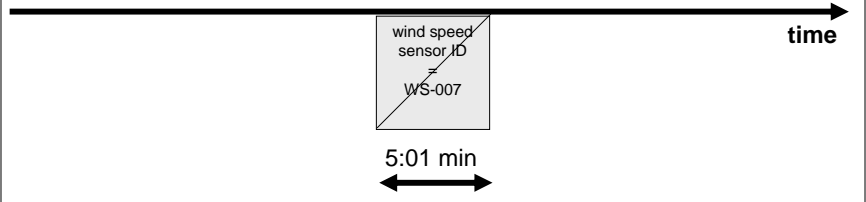
wind speed > 60 km/h	wind speed > 60 km/h	wind speed > 60 km/h
----------------------------	----------------------------	----------------------------

1 hour
↔

If you want to select **one piece of information out of a series of repetition** of that information, place boxes **next to** each other and **put an (orange) frame** around the desired box.

Tutorial 9

Notify if the wind speed sensor with the identification WS-007 potentially has been blown off the roof.



wind speed
sensor ID
WS-007

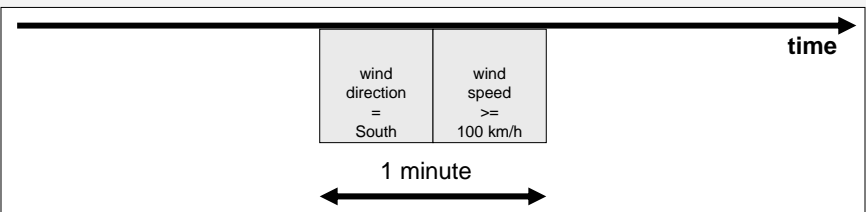
5:01 min

Assumption: Sensor WS-007 sends a value every 5 minutes.

If you are interested in **information that does not** occur, **cross out the box** with that information. You need to know how often information would be send normally in order to determine the **duration** you require.

Tutorial 10

Notify if the wind comes from the South and this is followed by a wind speed of at least 100 km/h within maximally 1 minute.



wind direction = South

wind speed >= 100 km/h

1 minute

If you are only interested in an alert about a condition if some other condition has been fulfilled previous to that, use a **sequence** of boxes by placing them **adjacent next to each other**.

Tutorial 11

Notify if any rainfall has been registered. Check this only from June to August.
 (During these month it is likely that there will be black ice on the desert road and someone has to be sent out to check the road.)

You can indicate the **validity of a profile**.

The **start** is indicated by a **green line** annotated by a time and the **end** by a **red line** annotated by a time.

Tutorial 12

Since 1999, Helen Clark, during the week, has to walk to the Beehive.
 She does not like to be rained on. Therefore notify when there is any rain on those days between 8:30 and 8:35. Then Helen knows to take along her umbrella!

You can combine several validity times. Simply annotate with several times.

Tutorial 13

Notify if the humidity exceeds 70 % or if the temperature is higher than 30 °C.
Also notify if the temperature is higher than 25 °C and this is followed by rainfall. Test this only during the day (6:00 - 23:00).

time

humidity
>
70 %

temperature
>
30 °C

temperature > 25 °C	rainfall > 0 mm
---------------------------	-----------------------

06:00 23:00

Tutorial 14

Notify if the wind direction measured by the sensor with the ID “Fred Flinstone’s Sensor” is registered as being a Southerly for the third time in a row within an hour. Check this from 1001 before Christ until 1000 before Christ.

time

ID = “Fred Flinstone’s Sensor”	wind direction = South
--	---------------------------------

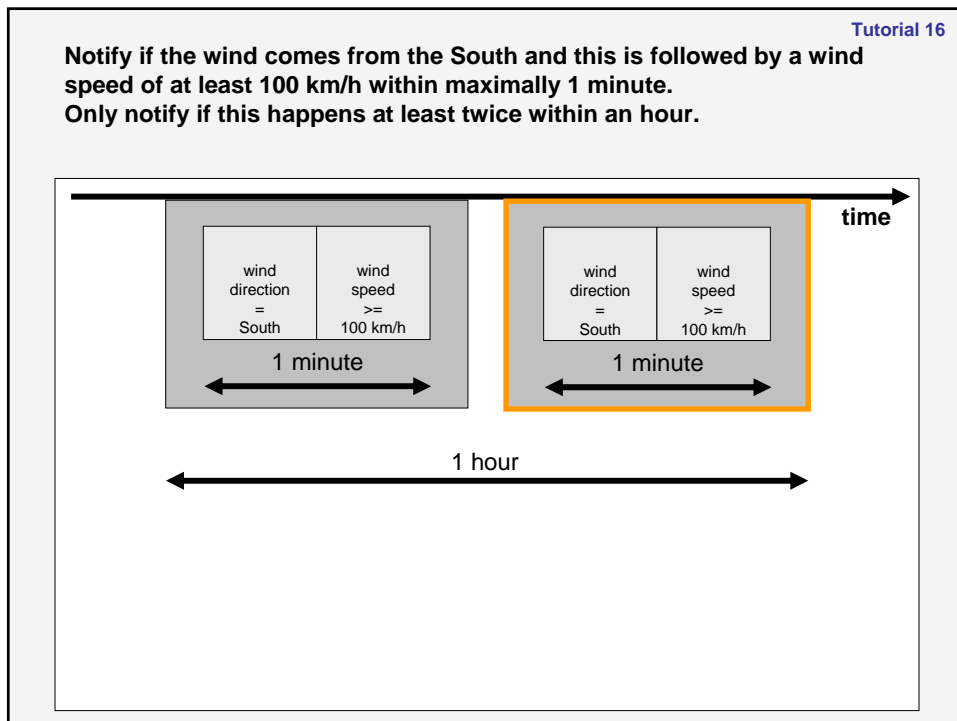
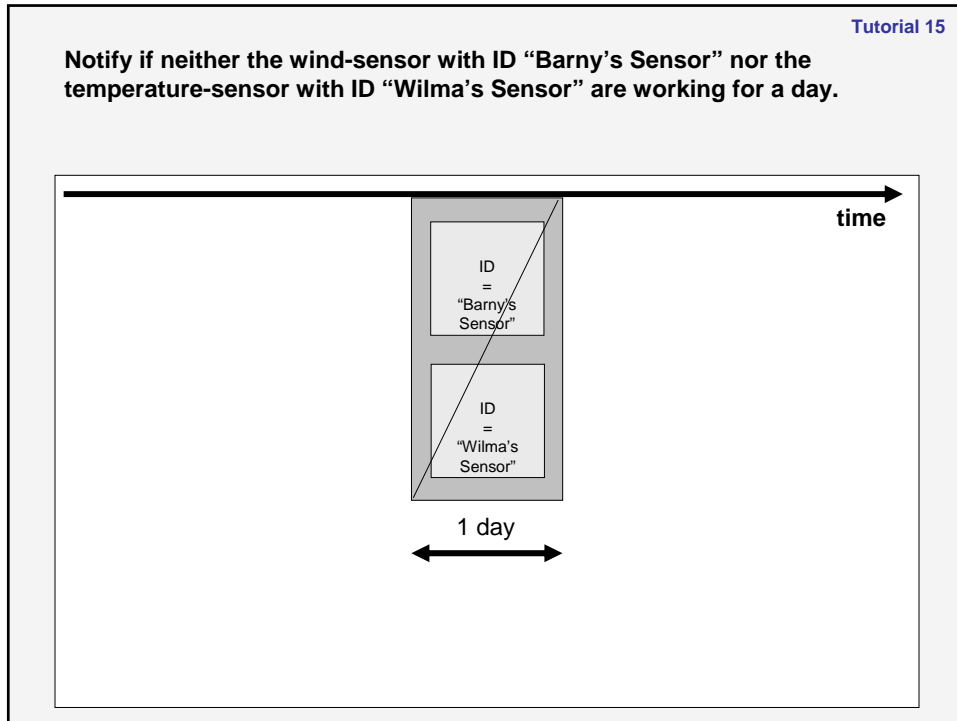
ID = “Fred Flinstone’s Sensor”	wind direction = South
--	---------------------------------

ID = “Fred Flinstone’s Sensor”	wind direction = South
--	---------------------------------

1 hour

-1001 -1000

Conditions are combined by placing them in another box.



Training

User Study of a Graphical
Profile Definition Language

Training: Interpretation Tasks

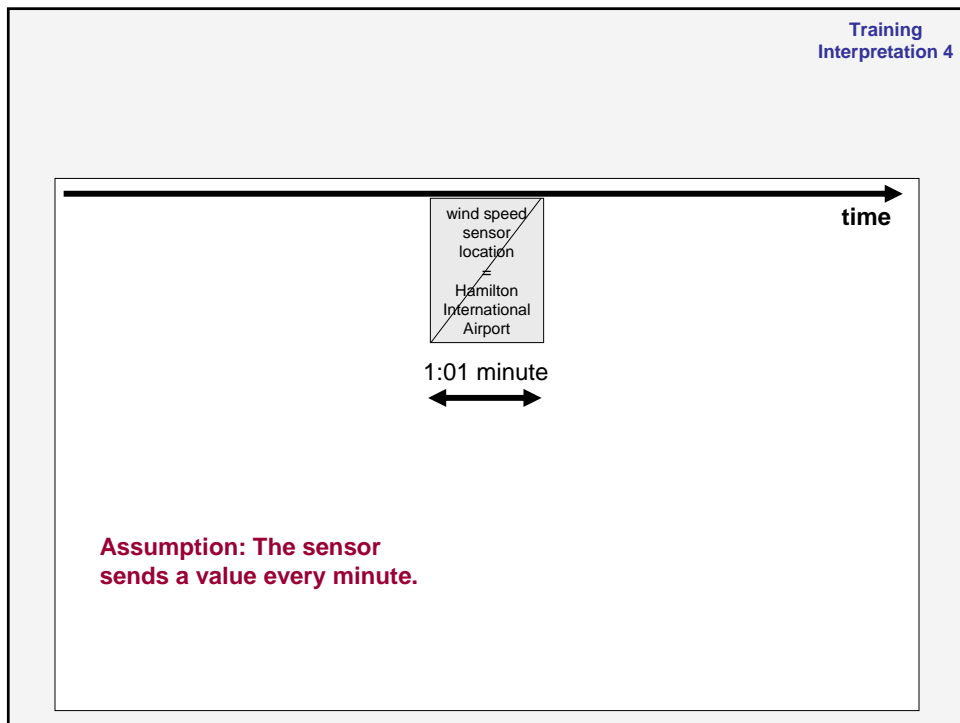
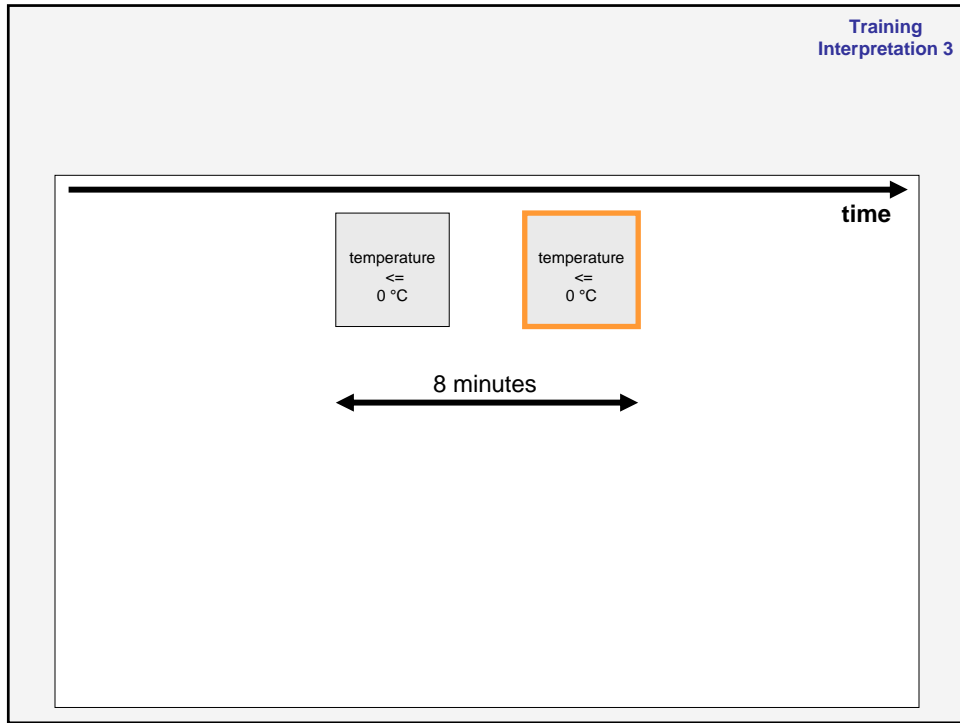
User Study of a Graphical
Profile Definition Language

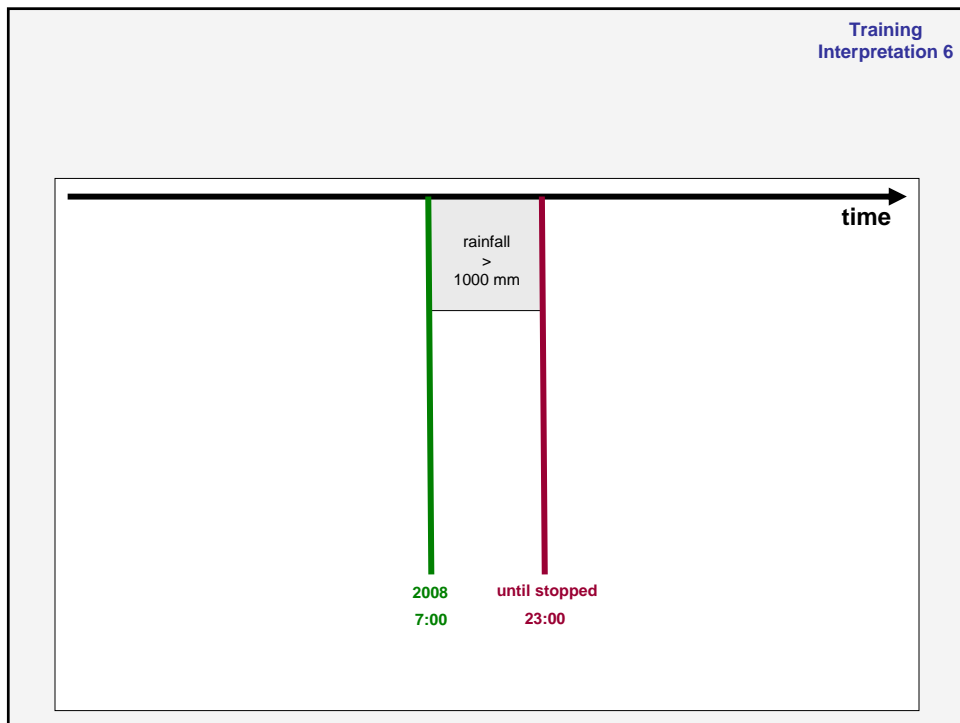
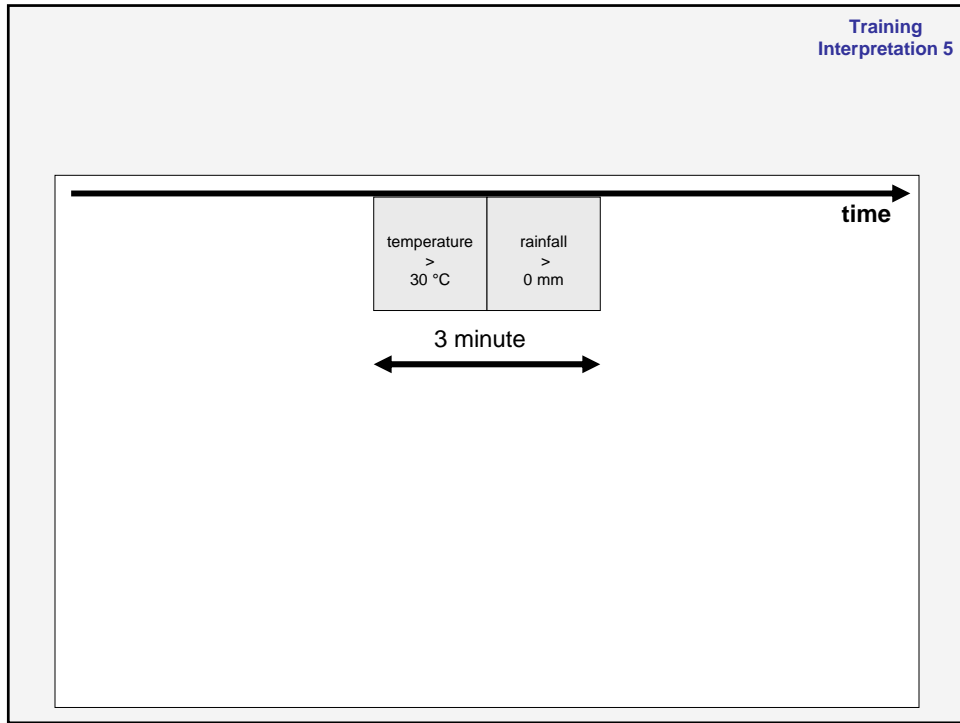
**Training
Interpretation 1**

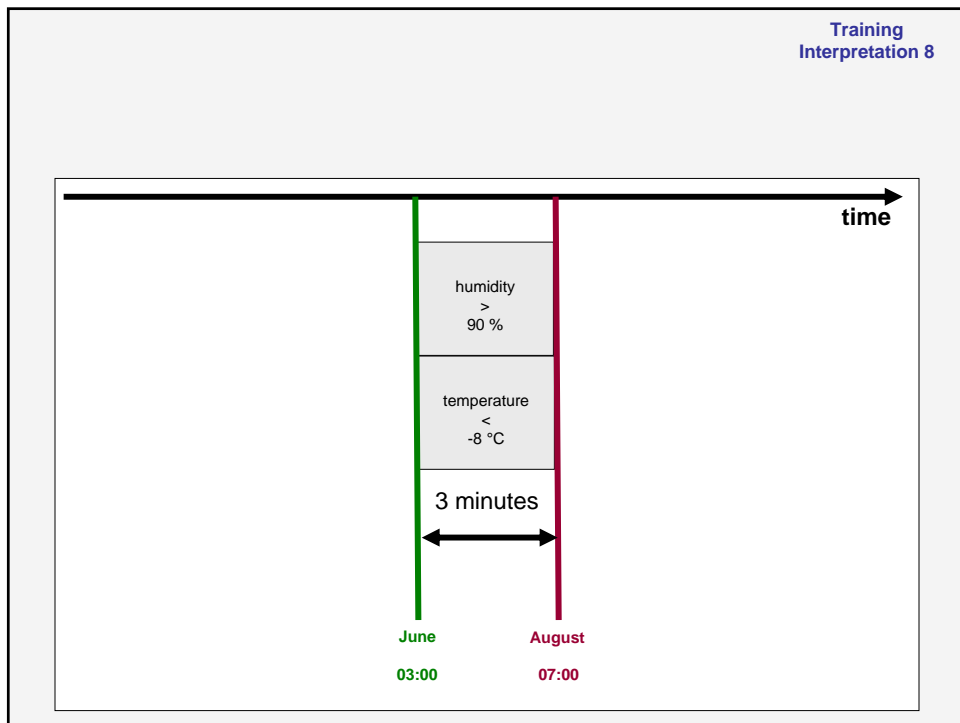
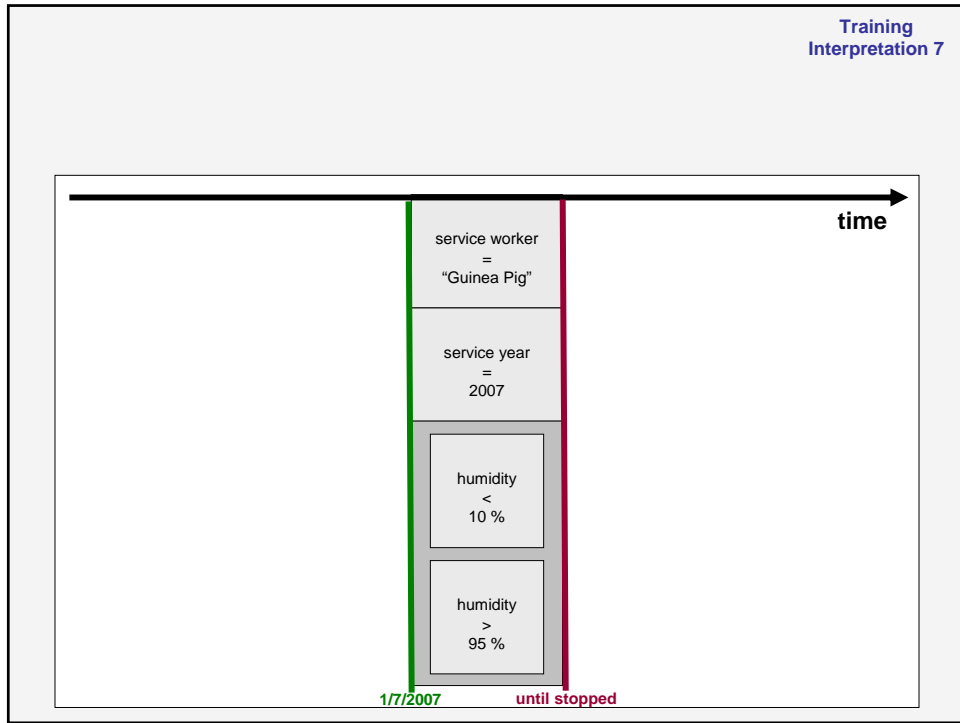
temperature > 10 °C
temperature < 30 °C

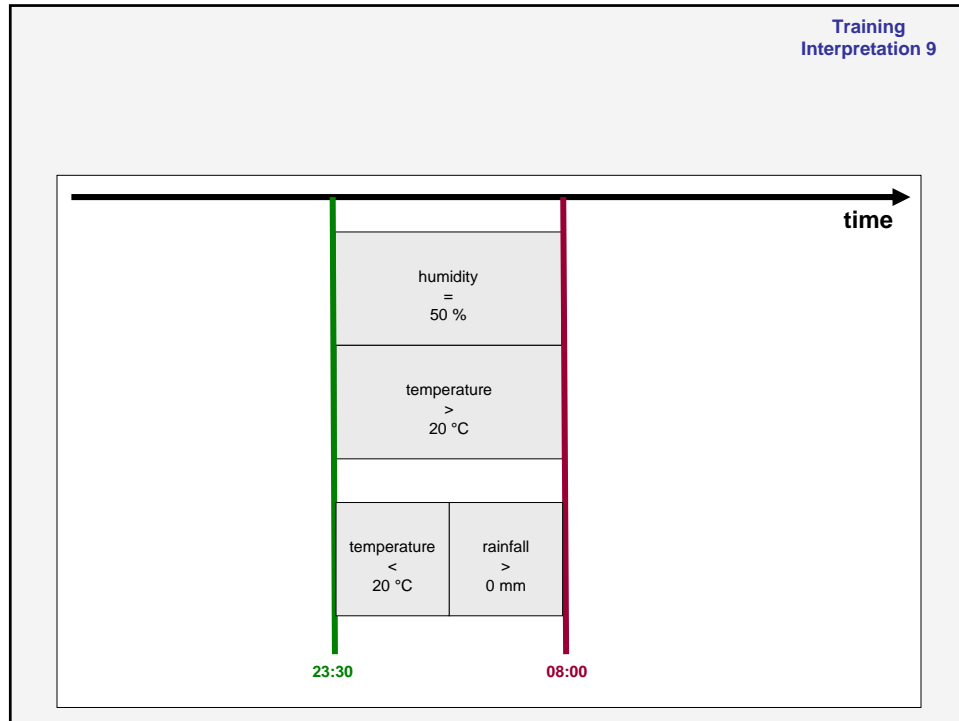
**Training
Interpretation 2**

temperature < 0 °C
temperature > 40 °C
pressure < 800 hPa







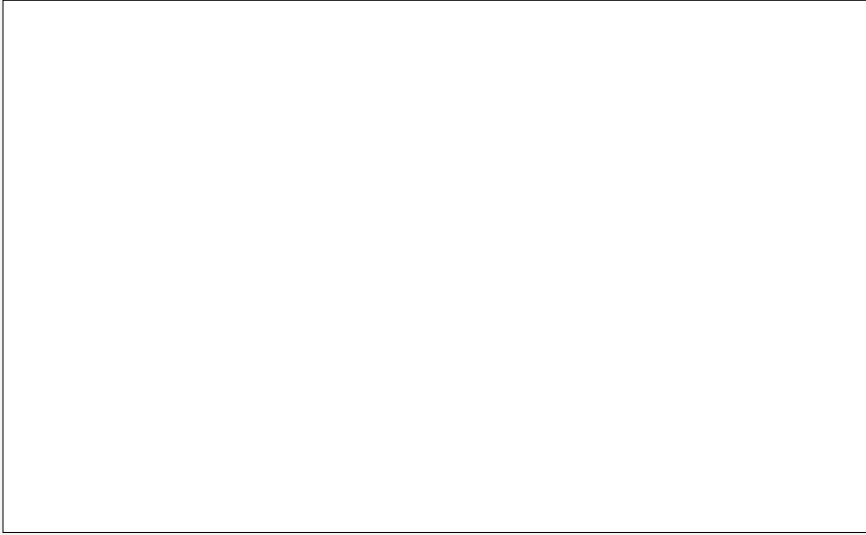


Training: Specification Tasks

User Study of a Graphical
Profile Definition Language

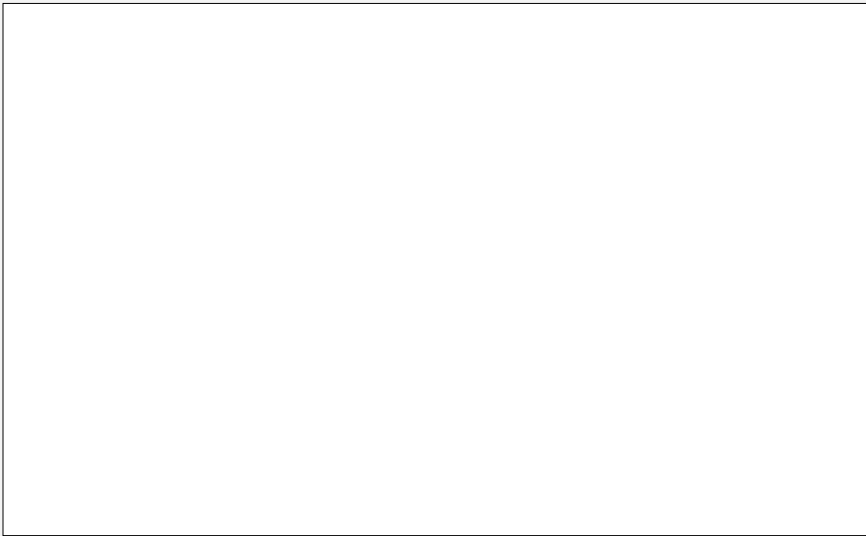
**Training
Specification 1**

Notify if it is raining and the temperature is colder than 20 °C.



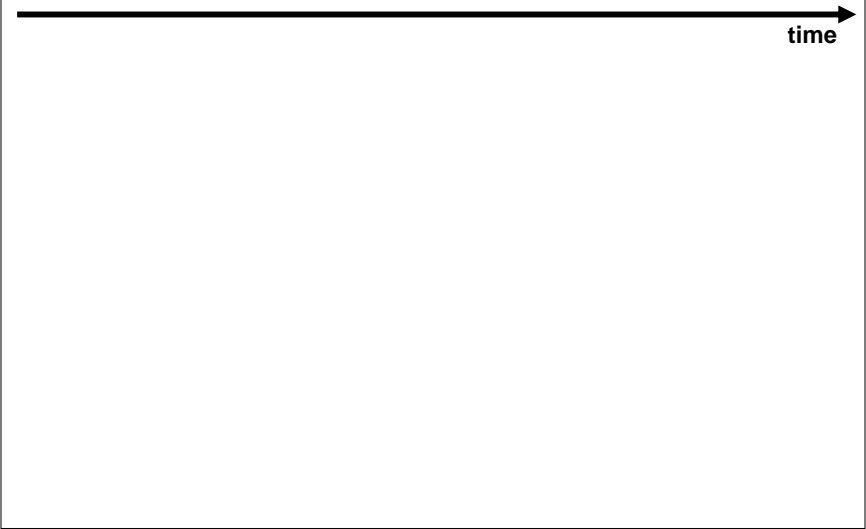
**Training
Specification 2**

Notify if there is rainfall of above 5 mm/h or if the humidity is above 70 %.



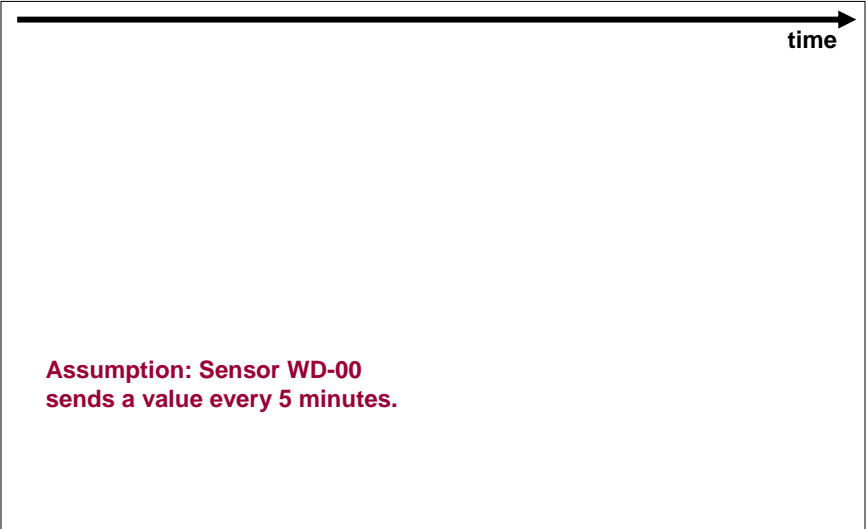
Training
Specification 3

Notify if there is a heat wave, i.e. the temperature is above 40 °C at least three times in 2 weeks.



Training
Specification 4

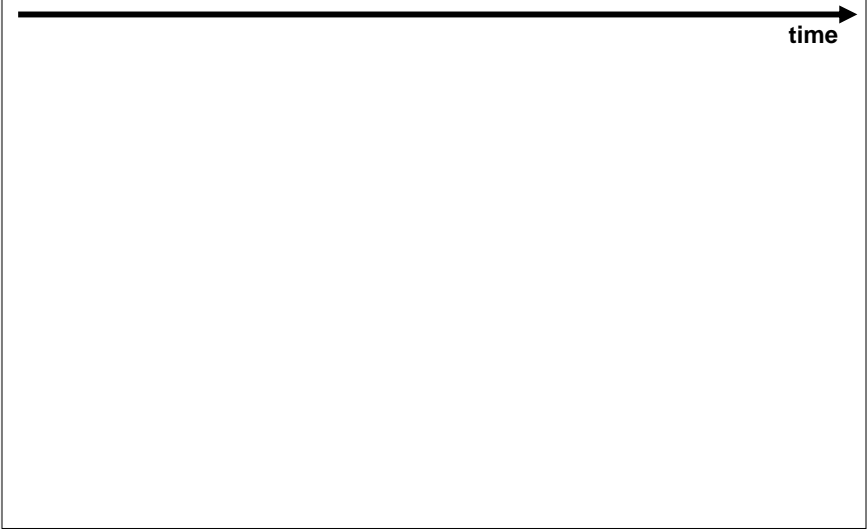
Notify if the wind direction sensor with the identification „WD-00“ potentially is broken.



Assumption: Sensor WD-00 sends a value every 5 minutes.

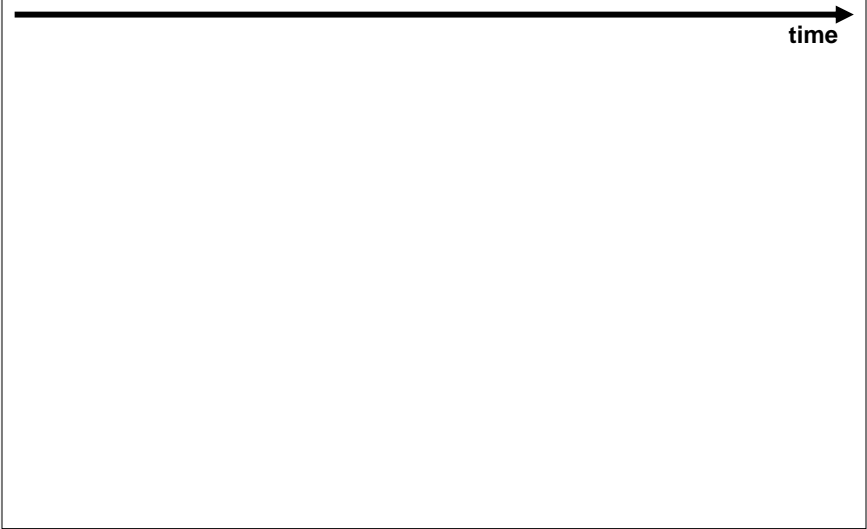
Training
Specification 5

Notify if the wind direction is North and this is followed by a wind speed of at least 150km/h within maximally 30 seconds.



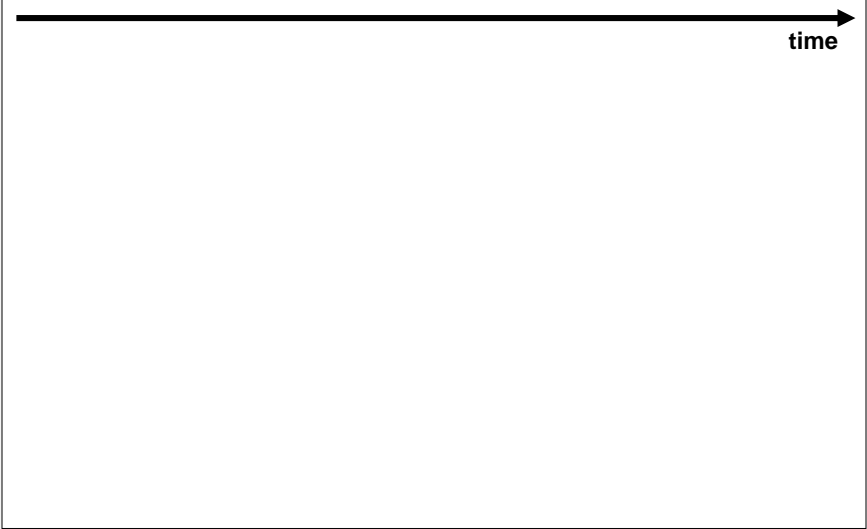
Training
Specification 6

Check whether the temperature in 2008 and 2009 ever falls below -10 °C. To save resources, test this only during winter (June - August).




**Notify if the temperature is smaller than 0 °C.
Also notify if the temperature is smaller than 3 °C and this is followed by
rainfall. Test this only during winter (June-August).**

Training
Specification 7



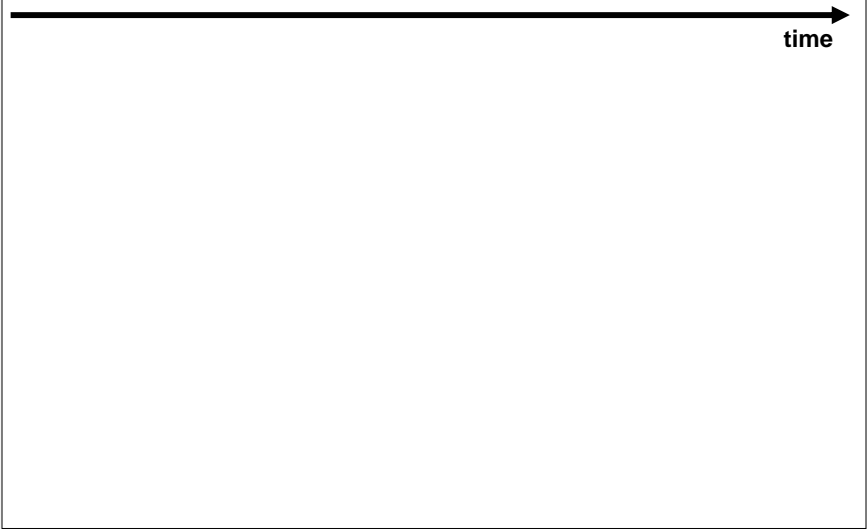
**Notify if, in one month, we have more than 50 mm/h rainfall for
the third time at location Christchurch. We are interested in this during
2007 only.**

Training
Specification 8



Notify if neither the temperature sensor located in Hamilton nor the temperature sensor located in Wellington are working for an hour.

Training Specification 9



The diagram consists of a horizontal axis labeled "time" with an arrow pointing to the right. The axis is enclosed in a rectangular frame. The text "time" is positioned at the end of the arrow on the right side.

Evaluation

User Study of a Graphical
Profile Definition Language

Evaluation: Interpretation Tasks

User Study of a Graphical
Profile Definition Language

Evaluation
Interpretation 1

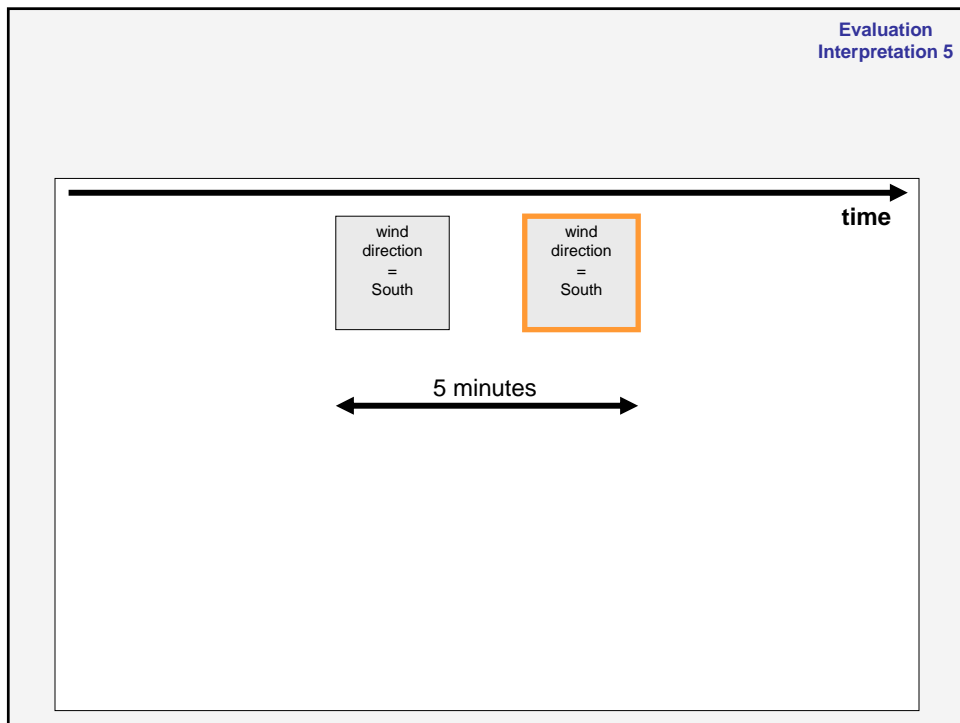
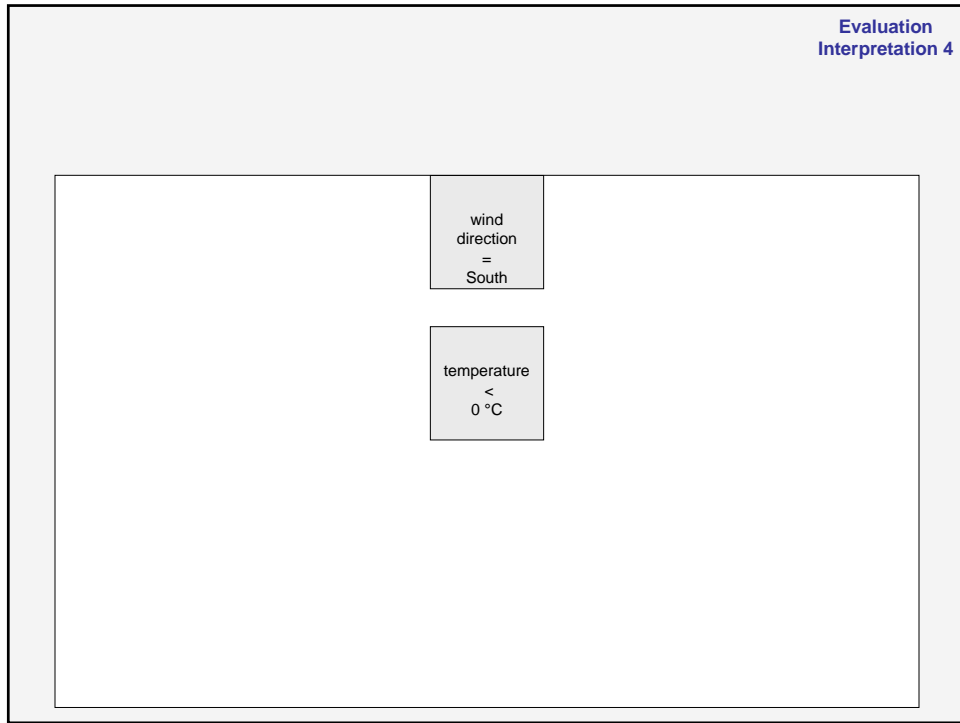
wind
direction
=
North

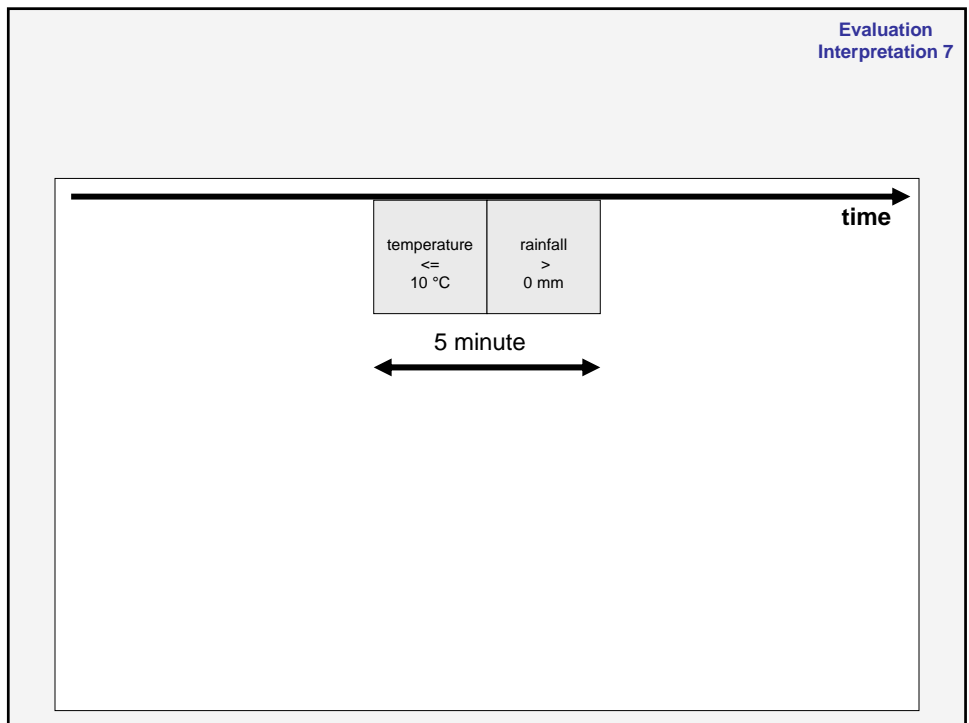
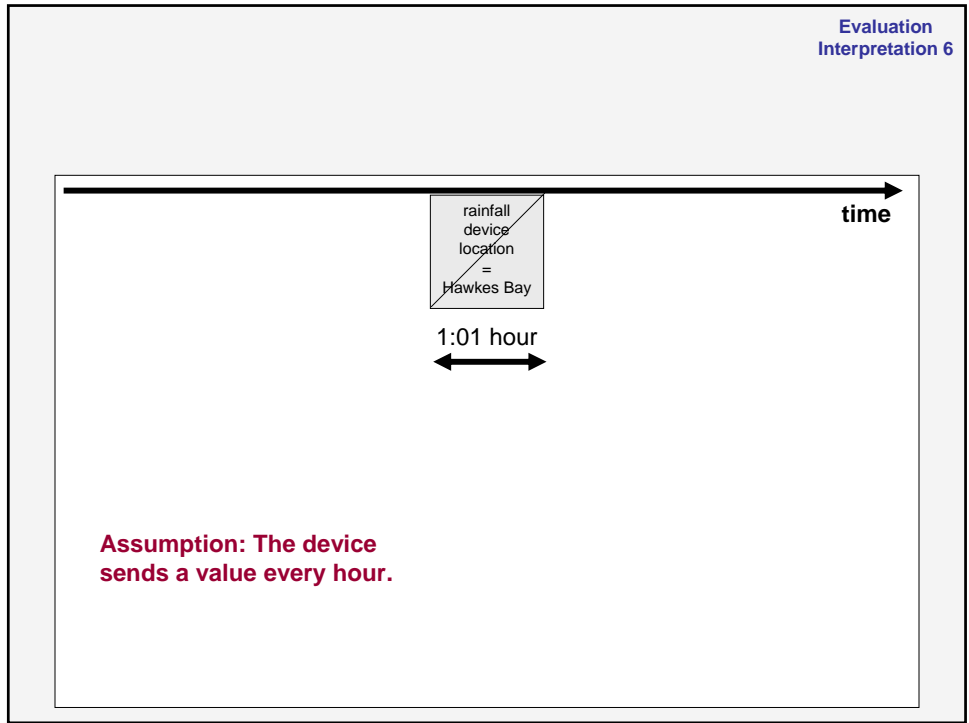
**Evaluation
Interpretation 2**

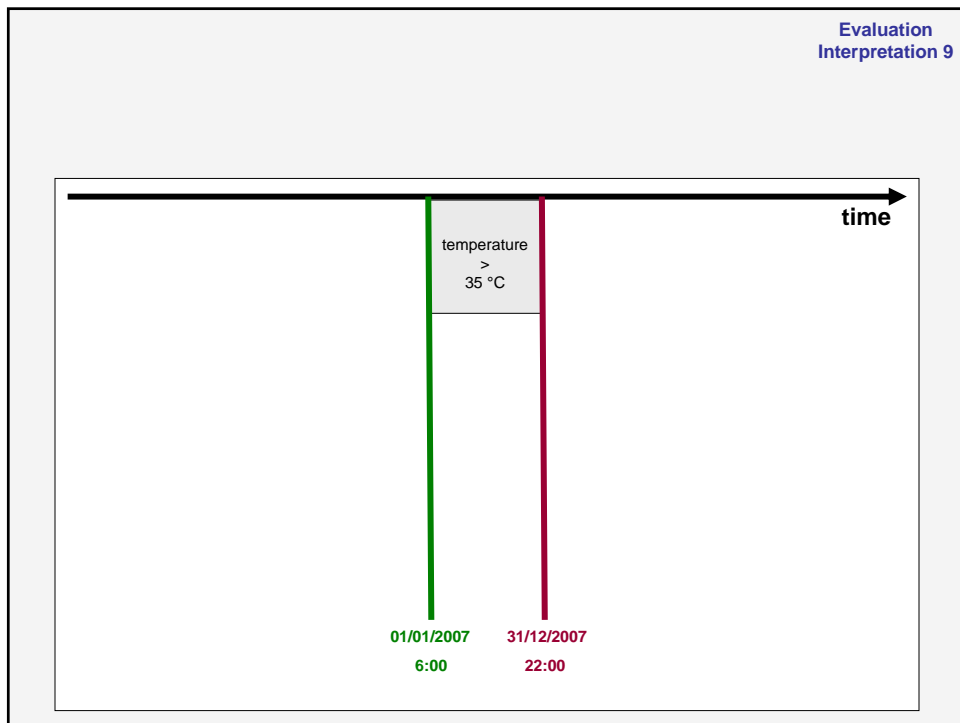
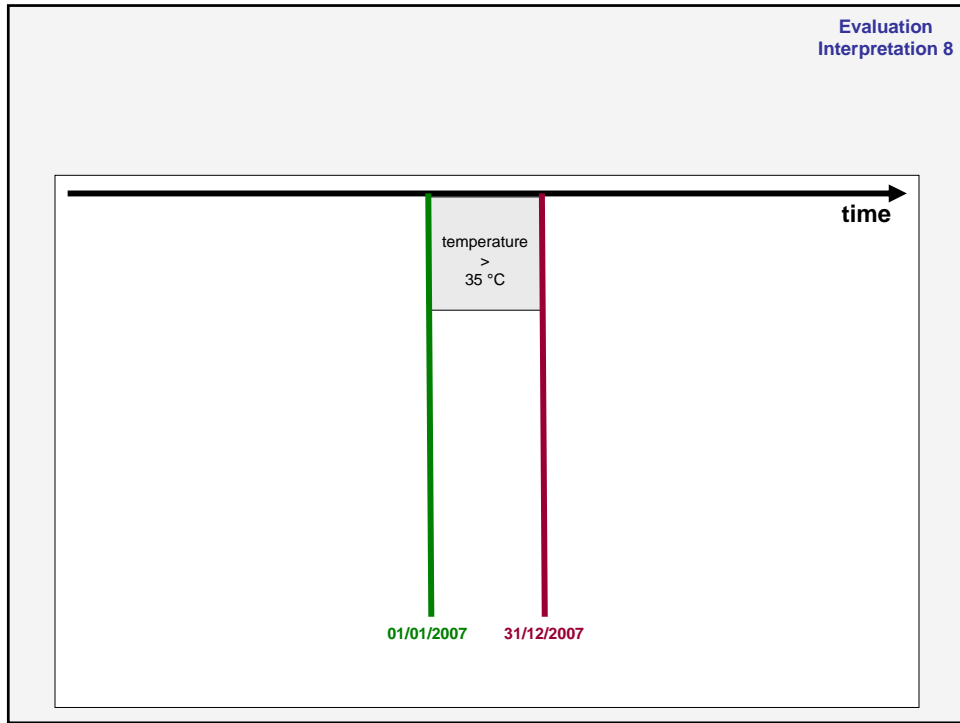
wind = 0 km/h

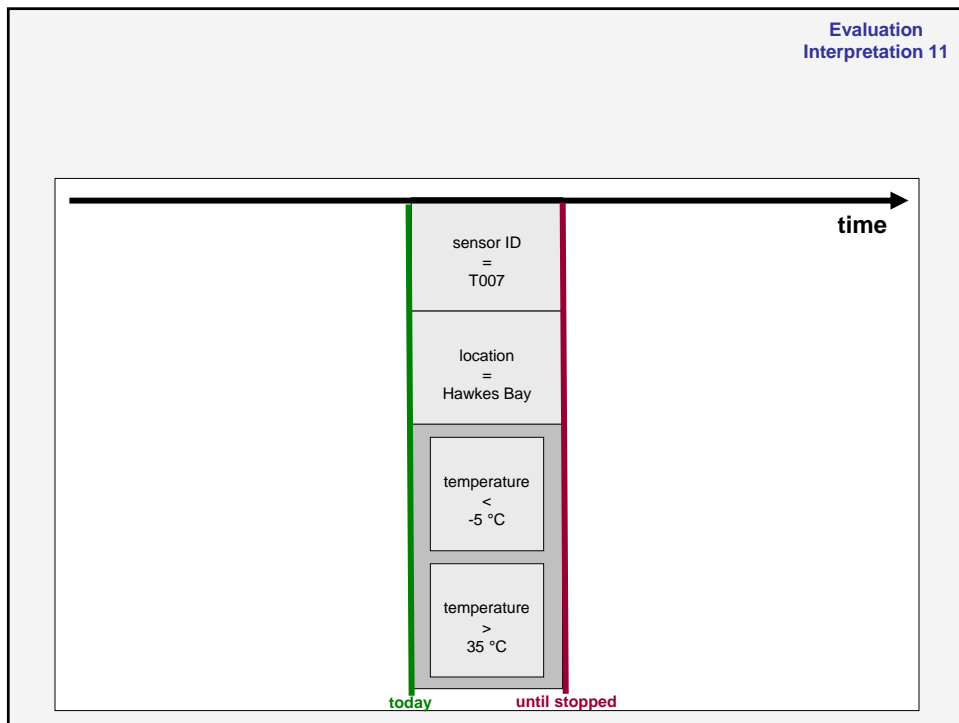
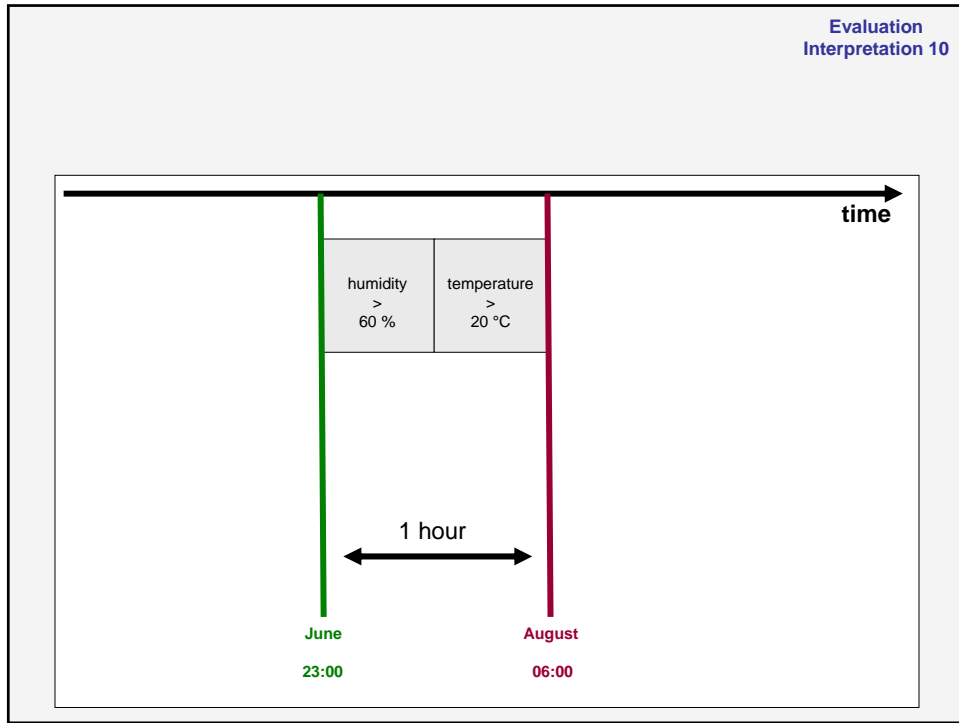
**Evaluation
Interpretation 3**

wind > 50 km/h
temperature = 30 °C







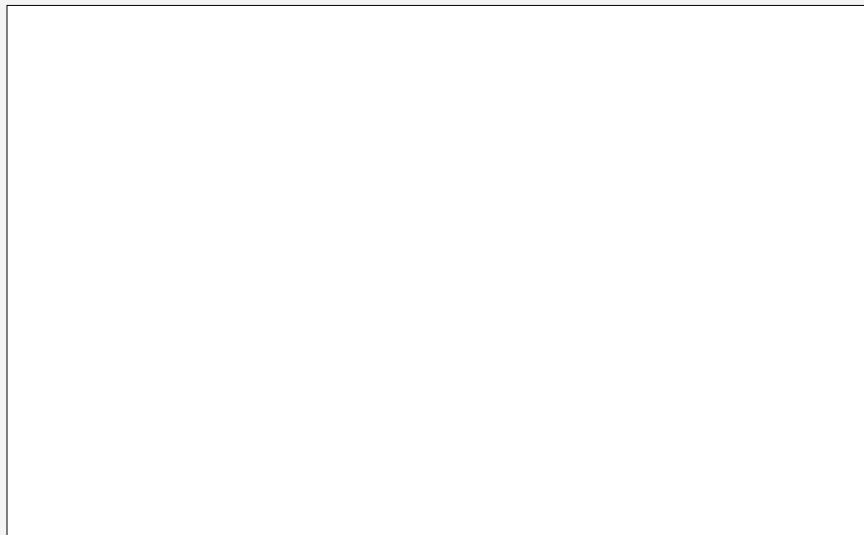


Evaluation: Specification Tasks

User Study of a Graphical
Profile Definition Language

Evaluation
Specification 1

Notify if the pressure is above 1000 hPa.



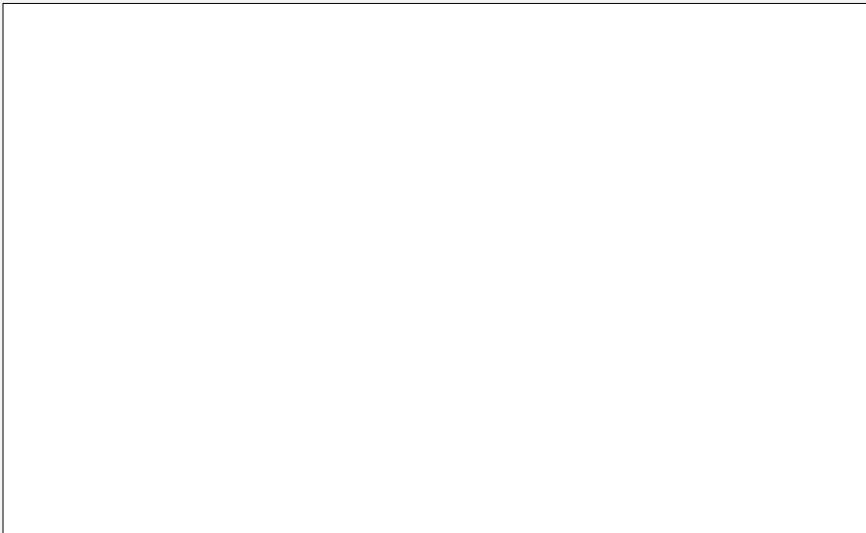
**Evaluation
Specification 2**

Notify if any wind has been registered.



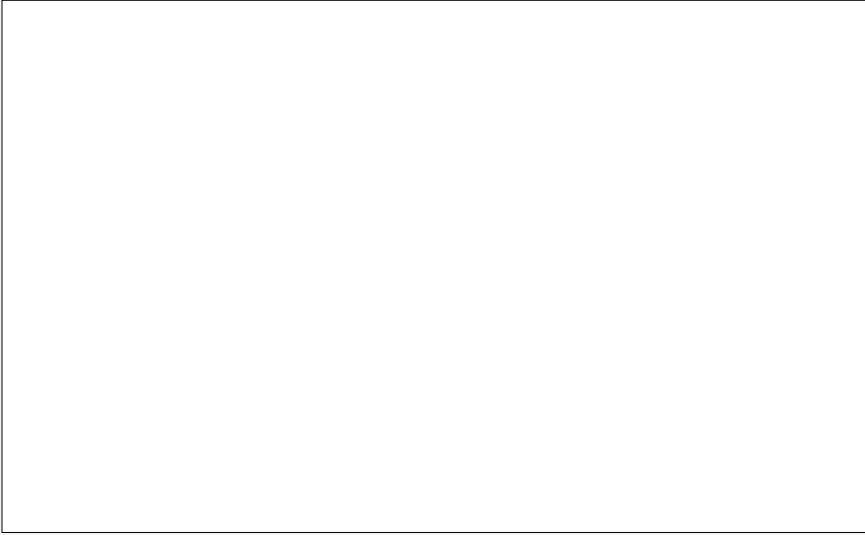
**Evaluation
Specification 3**

Notify if there is a breeze coming from the East.




Evaluation
Specification 4

Notify if the wind direction is either from the South or from the North.



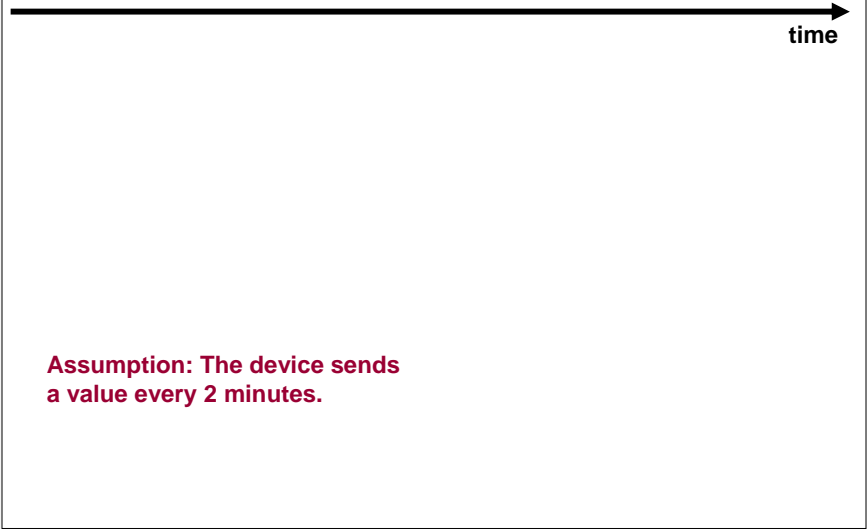
Evaluation
Specification 5

Notify if it is raining. However, make sure that this was not only one short shower, but has been registered more than once within 10 minutes.



Evaluation
Specification 6


**Notify if the wind speed measurement device located at
Mystery Creek is malfunctioning.**



**Assumption: The device sends
a value every 2 minutes.**

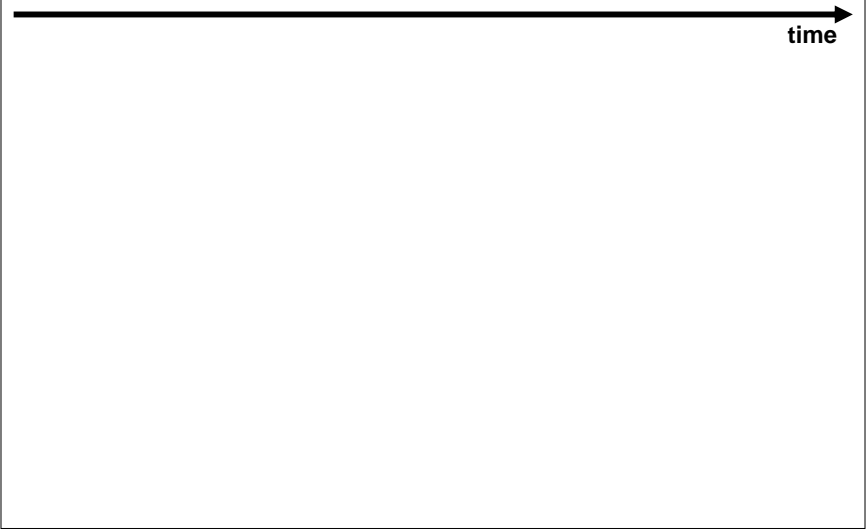
Evaluation
Specification 7

**Notify if the wind speed is at least 100 km/h and this
is followed by rainfall within 1 minute.**




Evaluation
Specification 8

Check whether in August and September it ever rains more than 50 mm/h.




Evaluation
Specification 9

Check whether in summer (December - February) it ever rains more than 50 mm/h during the day (6:00 – 22:00).



Notify if the wind “goes wild”. I.e. it comes from every direction at least once in an hour. Check this only in summer (December - February) and only during the day (7:00 to 23:00).

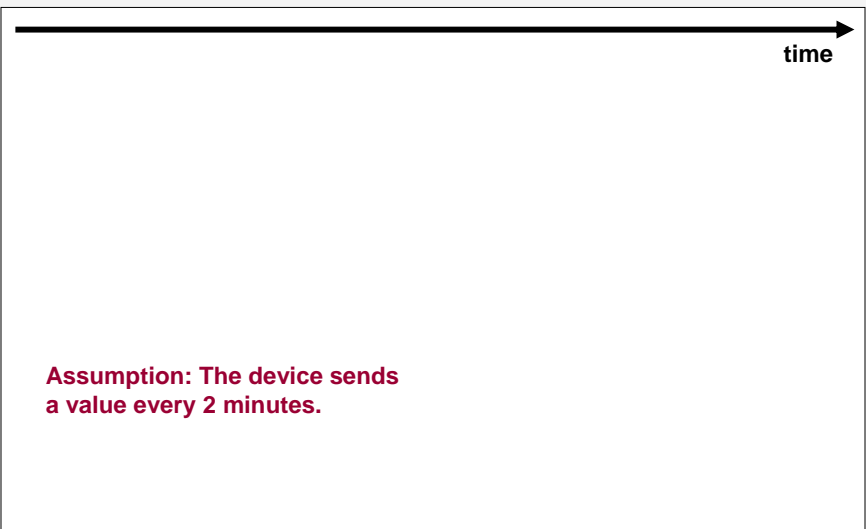
Evaluation Specification 10



The graph area is a large rectangle with a horizontal axis at the top. The axis is a solid black line with an arrowhead pointing to the right, labeled "time". The rest of the rectangle is empty, intended for a graphical profile definition.

Notify if the wind speed is repeatedly (3 times) over 100 km/h and this is followed by a malfunction of the wind speed measurement device located at Mystery Creek. Check this in 2007.

Evaluation Specification 11



The graph area is a large rectangle with a horizontal axis at the top. The axis is a solid black line with an arrowhead pointing to the right, labeled "time". The rest of the rectangle is empty, intended for a graphical profile definition.

Assumption: The device sends a value every 2 minutes.

Questionnaire
- User Study of a Graphical Profile Definition Language –

1 Please describe your general impression of using the graphical profile definition language:

2 I found using the graphical profile definition language...

(Please tick the appropriate option)

Not intuitive very intuitive

Not easy very easy

Not satisfying very satisfying (i.e. annoying vs. fun)

Please explain your answers:

3 How experienced are you in using queries or alerting?

(e.g. google or digital libraries)

Not experienced very experienced**4 How often do you use queries?**

Less than once per year 2-3 times per year Every 2-3 months Once per month 2-3 per month Once per week More than once per week

5 Is English your first language
 Yes No
6 Give your age on 1st January 2007**7 Which gender are you?****8 What is your profession?**

(If you are a student/academic staff please include your area, e.g. mathematics)

Appendix D. Single-user Profile Software Interface Study: Participant Workbook



User Study of an Interface for Profile Specification for Alerting Systems

Researcher Doris Jung

Participant Workbook

Session Number:

Date and Time:

Evaluation Task Order:

The University of Waikato School of Computing and Mathematical Sciences

Research Participant's Bill of Rights

The following is a list of your rights if you participate in a research project organised within the School of Computing and Mathematical Sciences at the University of Waikato.

As a research participant, you have the right:

- To be treated with respect and dignity in every phase of the research.
- To be fully and clearly informed of all aspects of the research prior to becoming involved in it.
- To enter into clear, informed, and written agreement with the researcher prior to becoming involved in the activity. You should sense NO pressure, explicit or otherwise, to sign this contract.
- To choose explicitly whether or not you will become involved in the research under the clearly stated provision that refusal to participate or the choice to withdraw during the activity can be made at any time without penalty to you.
- To be treated with honesty, integrity, openness, and straightforwardness in all phases of the research, including a guarantee that you will not unknowingly be deceived during the course of the research.
- To receive something in return for your time and energy.
- To demand proof that an independent and competent ethical review of human rights and protections associated with the research has been successfully completed.
- To demand complete personal confidentiality and privacy in any reports of the research unless you have explicitly negotiated otherwise.
- To expect that your personal welfare is protected and promoted in all phases of the research, including knowing that no harm will come to you.
- To be informed of the results of the research study in a language you understand.
- To be offered an range of research studies or experiences from which to select, if the research is part of fulfilling your educational or employment goals.

The contents of this bill were prepared by the University of Calgary who examined all of the relevant Ethical Standards from the Canadian Psychological Association's Code of Ethics for Psychologists, 1991 and rewrote these to be of relevance to research participants.

Descriptions of the CPA Ethical Code and the CPA Ethical Standards relevant to each of these rights are available at <http://www.cpa.ca/ethics2000.html> and <http://www.psych.ucalgary.ca/Research/ethics/bill/billcode.html> if you would like to examine them.

The complete CPA Ethical Code can be found in Canadian Psychological Association "*Companion manual for the Canadian Code of Ethics for Psychologists*" (1992).

The University of Waikato School of Computing and Mathematical Sciences

Research Consent Form

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

Research Project Title

User Study of an Interface for Profile Specification for Alerting Systems.
(PhD Project Title: Realisation and Representation of Collaborative Profiles.)

Researcher

Doris Jung.

Experiment Purpose

This study will evaluate the usability of an Interface for Profile Specification for Alerting Systems.

Participant Recruitment and Selection

Undergraduate and graduate students from the University of Waikato are being recruited for this study. Academic and non-academic staff members of any department at the University of Waikato will also be recruited for the study.

Procedure

This session will require about 1:15 hours of your time.

You will be given a tutorial. This will be followed twice by a sequence of training tasks followed by evaluation tasks. One task sequence will ask you to specify solutions to given descriptions with the help of the interface. The other task sequence will present you specifications of information needs with the help of the interface. You will be asked to describe verbally and/or on paper what these specifications seem to symbolise. Please think aloud while solving task sequences – describing how you try to solve the tasks.

Afterwards we will have a brief discussion about how you experienced solving the tasks and you will be asked to fill in a short questionnaire about your background.

Data Collection

A questionnaire will be used to collect background information. As you are solving the task sequences and think aloud the interviewer will take notes. The researcher also will use your task solving notes and record your specification results for the evaluation of the study.

Data Archiving/Destruction

Data will be kept securely stored by the researcher.

Confidentiality

Confidentiality and participant anonymity will be strictly maintained. All information gathered will be used for statistical analysis only and no names or other identifying characteristics will be stated in the final or any other reports.

Likelihood of Discomfort

There is no likelihood of discomfort or risk associated with participation.

Researcher

Doris Jung. She is working on her doctorate in the Computer Science Department at the University of Waikato. This study will contribute to her research on the realisation and representation of collaborative profiles. Her supervisor is Associate Professor Steve Jones.

Doris can be contacted in room G2.27 of the School of Computer and Mathematical Sciences building at the University of Waikato. Her phone number is 838 4547 and her email address is d.jung@cs.waikato.ac.nz.

Finding out about Results

The Participants can find out the results of the study by contacting the researcher after 1/3/2008.

Agreement

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive you legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to not answer specific items or questions in interviews or on questionnaires. You are free to withdraw from the study at any time without penalty. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact the researcher.

Participant

Date

Investigator/Witness

Date

A copy of this consent form has been given to you to keep for your records and reference.

**User Study of
an Interface for Profile Specification
for Alerting Systems**

Researcher: Doris Jung
Supervisor: Steve Jones

Tutorial

User Study of
an Interface for Profile Specification for
Alerting Systems

These tasks will be shown in the interface.

Tutorial 1

Example of Weather Observation

Temperature =	13 °C
Wind Speed =	32 km/h
Wind Direction =	W
Rainfall (last hr) =	0.2 mm
Humidity =	74 %
Pressure =	992 hPa

Tutorial 2

Notify if wind stronger than 50 km/h.

wind
>
50 km/h

Each **condition** is presented by a (grey) **box**.

Tutorial 3

Notify if temperature is different from 0 °C.

temperature
≠
0 °C

Tutorial 4

Notify if humidity is higher than 60 % and the temperature is more than 30 °C.

humidity > 60 %
temperature > 30 °C

If all of several conditions have to be fulfilled, place boxes **adjacent beneath** each other.

Tutorial 5

Notify if humidity is more than 65 % or the temperature is above 25 °C or the pressure is higher than 992 hPa.

humidity
>
65 %

temperature
>
25 °C

pressure
>
992 hPa

If one of several conditions has to be fulfilled, place boxes with a gap beneath each other.

Tutorial 6

Notify if humidity is higher than 60 % and the temperature is more than 30 °C, both happening within 2 minutes.

humidity
>
60 %

temperature
>
30 °C

2:00 min
↔

An annotated arrow gives a duration. The desired conditions have to be fulfilled within the given time.

Tutorial 7

Notify if the wind repeatedly is stronger than 60 km/h, i.e. the wind is stronger than 60 km/h for at least three times in an hour.

time

wind speed > 60 km/h	wind speed > 60 km/h	wind speed > 60 km/h
-------------------------------	-------------------------------	-------------------------------

1 hour

If you want to select **one piece of information out of a series of repetition** of that information, place boxes **adjacent next to** each other. This notifies you about the occurrence of the last condition box.

Tutorial 8

Notify if the wind comes from the South and this is followed by a wind speed of at least 100 km/h within maximally 1 minute.

time

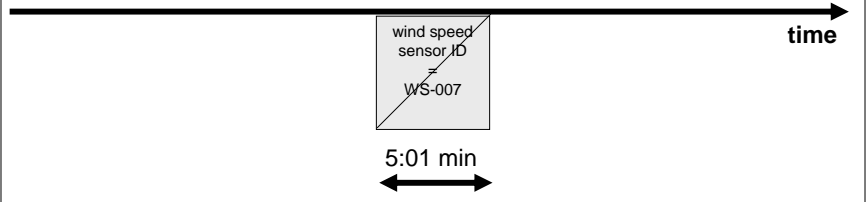
wind direction = South	wind speed >= 100 km/h
---------------------------------	---------------------------------

1 minute

If you are only interested in an alert about a condition if some other condition has been fulfilled previous to that, use a **sequence** of boxes by placing them **adjacent next to each other**.

Tutorial 9

Notify if the wind speed sensor with the identification WS-007 potentially has been blown off the roof.



wind speed
sensor ID
WS-007

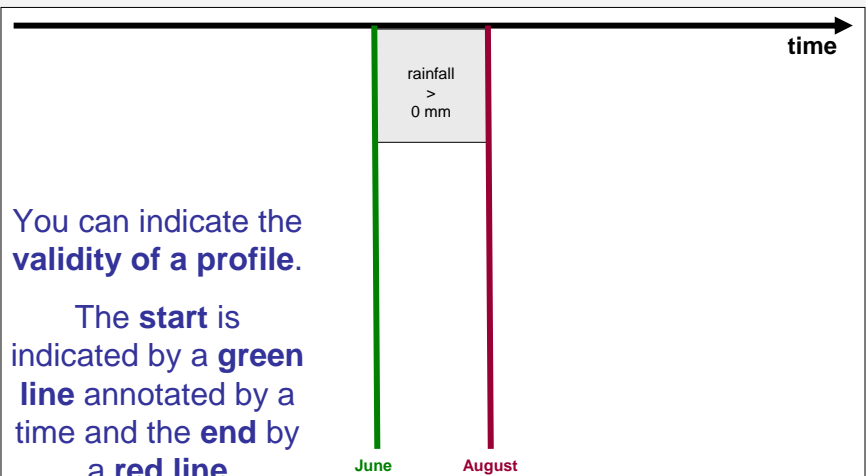
5:01 min

Assumption: Sensor WS-007 sends a value every 5 minutes.

If you are interested in **information that does not occur**, **cross out the box** with that information. You need to know how often information would be send normally in order to determine the **duration** you require.

Tutorial 10

Notify if any rainfall has been registered. Check this only from June to August.
(During these month it is likely that there will be black ice on the desert road and someone has to be sent out to check the road.)



rainfall
>
0 mm

June August

You can indicate the **validity of a profile**.

The **start** is indicated by a **green line** annotated by a time and the **end** by a **red line** annotated by a time.

Tutorial 11

Since 1999, Helen Clark, during the week, has to walk to the Beehive. She does not like to be rained on. Therefore notify when there is any rain on those days between 8:30 and 8:35. Then Helen knows to take along her umbrella!

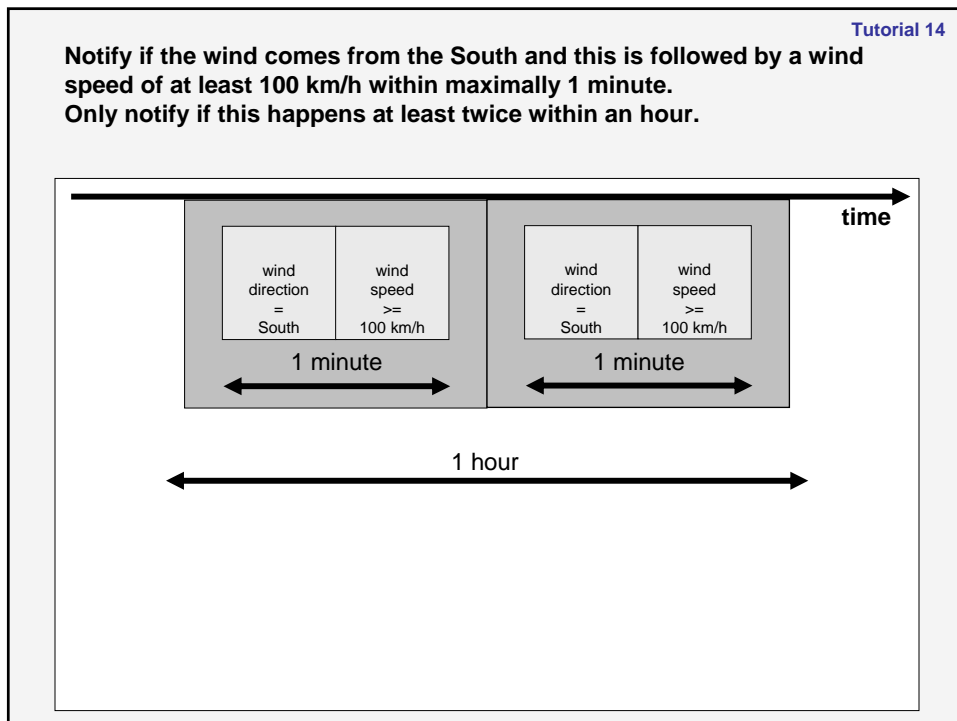
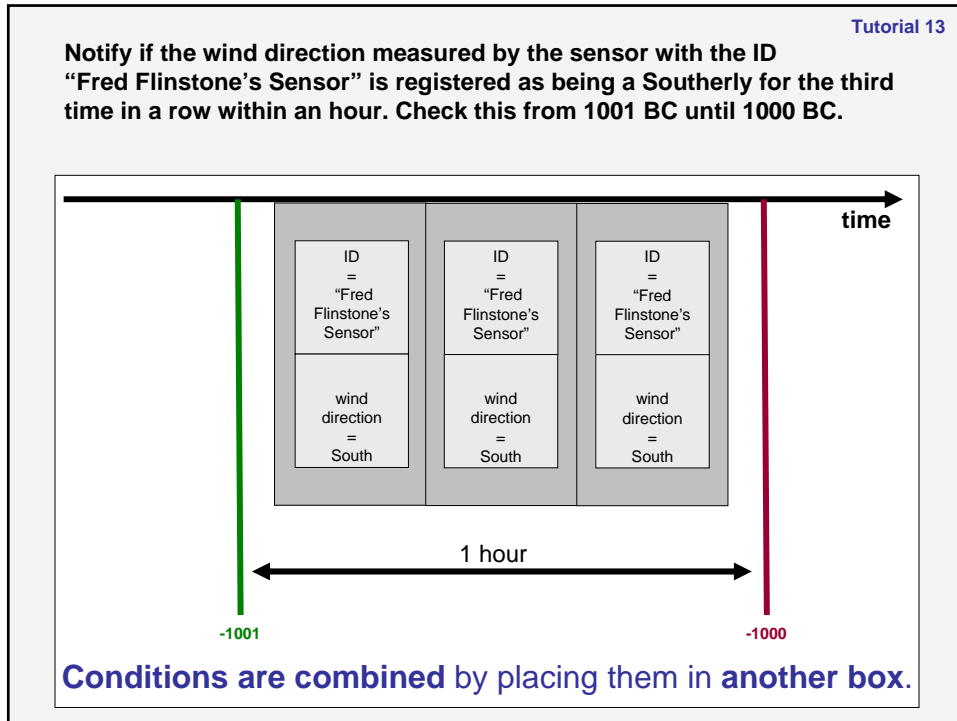
The diagram shows a horizontal timeline labeled 'time'. A vertical green line marks the start at 8:30 on Monday, 1999. A vertical red line marks the end at 8:35 on Friday. A grey box between these lines contains the text 'rainfall > 0 mm'. Below the green line, the text '1999 Monday 8:30' is written in green. Below the red line, the text 'until stopped Friday 8:35' is written in red.

You can combine several validity times. Simply annotate with several times.

Tutorial 12

Notify if the humidity exceeds 70 % or if the temperature is higher than 30 °C. Also notify if the temperature is higher than 25 °C and this is followed by rainfall. Test this only during the day (6:00 - 23:00).

The diagram shows a horizontal timeline labeled 'time'. A vertical green line marks the start at 6:00 and a vertical red line marks the end at 23:00. Three grey boxes are stacked vertically between these lines. The top box contains 'humidity > 70 %'. The middle box contains 'temperature > 30 °C'. The bottom box is split into two columns: the left column contains 'temperature > 25 °C' and the right column contains 'rainfall > 0 mm'. Below the green line, the text '6:00' is written in green. Below the red line, the text '23:00' is written in red.



**User Study of
an Interface for Profile Specification
for Alerting Systems**

Researcher: Doris Jung
Supervisor: Steve Jones

**Interpretation Tasks
Training**

User Study of
an Interface for Profile Specification for
Alerting Systems

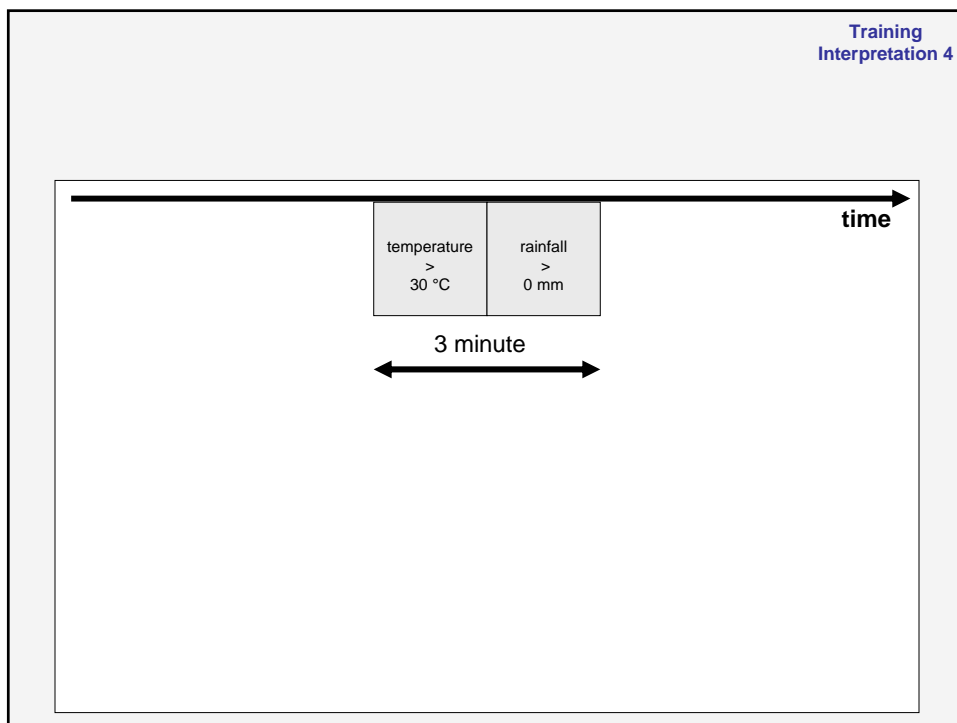
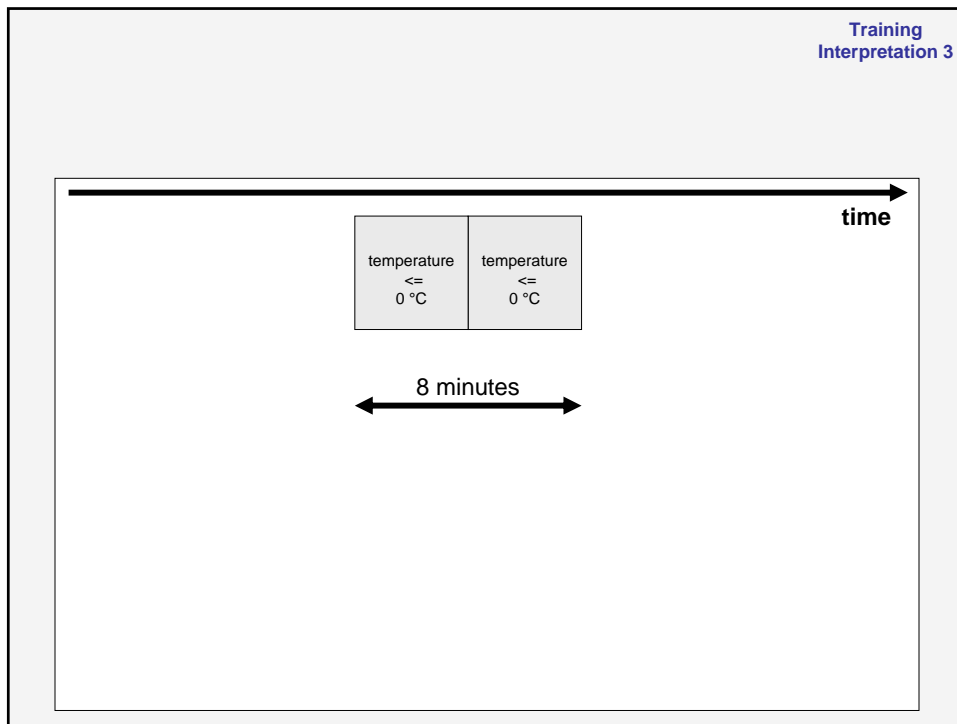
These tasks will be shown in the interface.

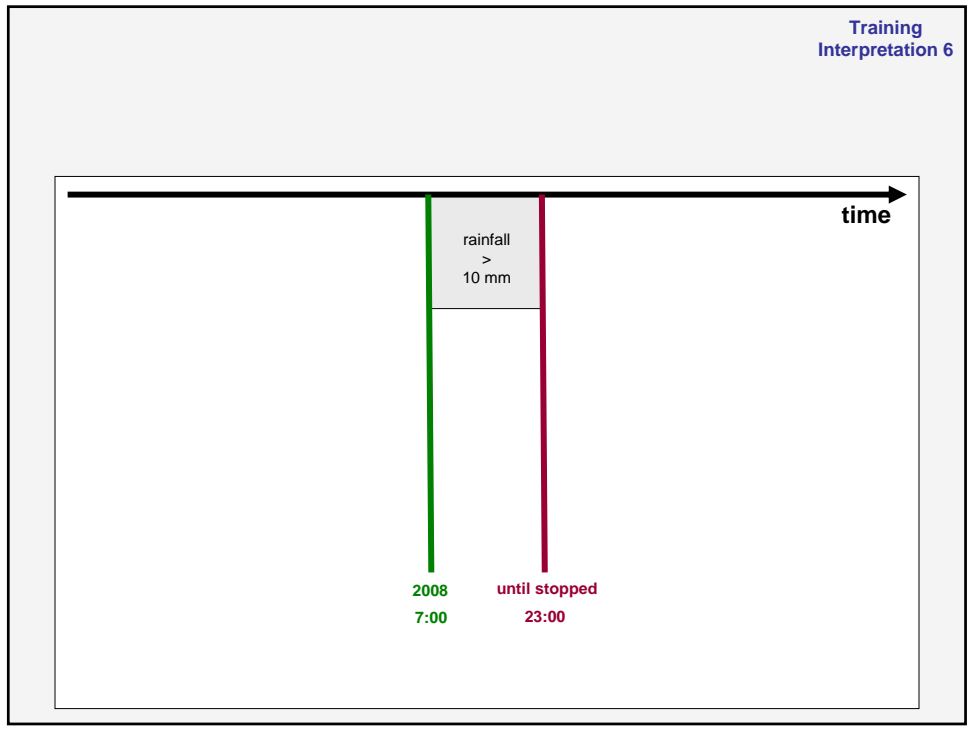
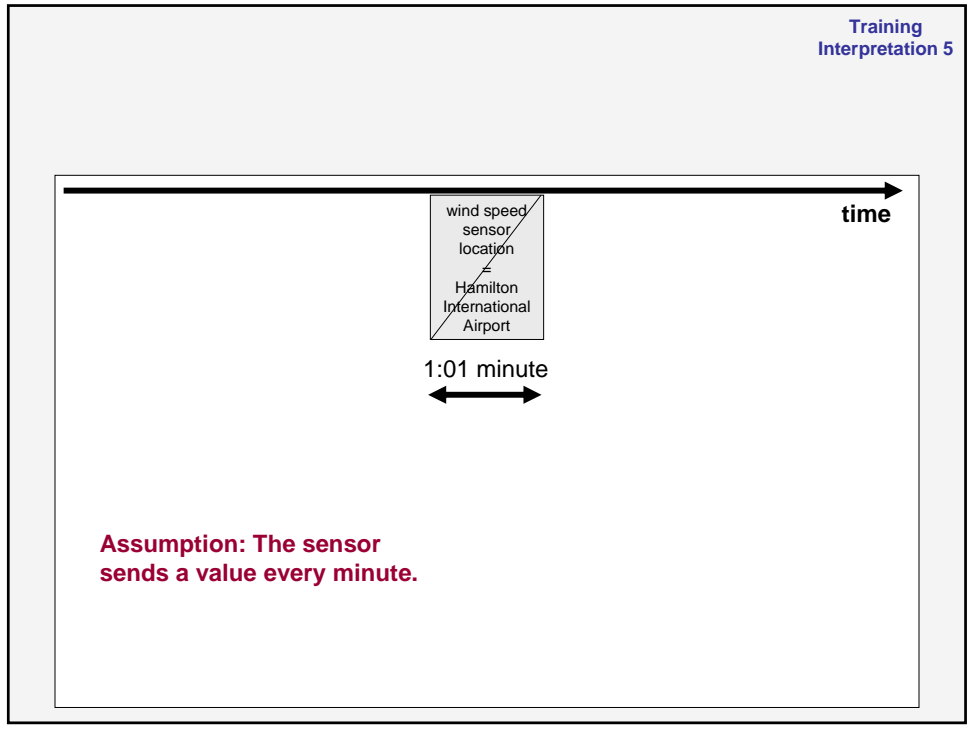
**Training
Interpretation 1**

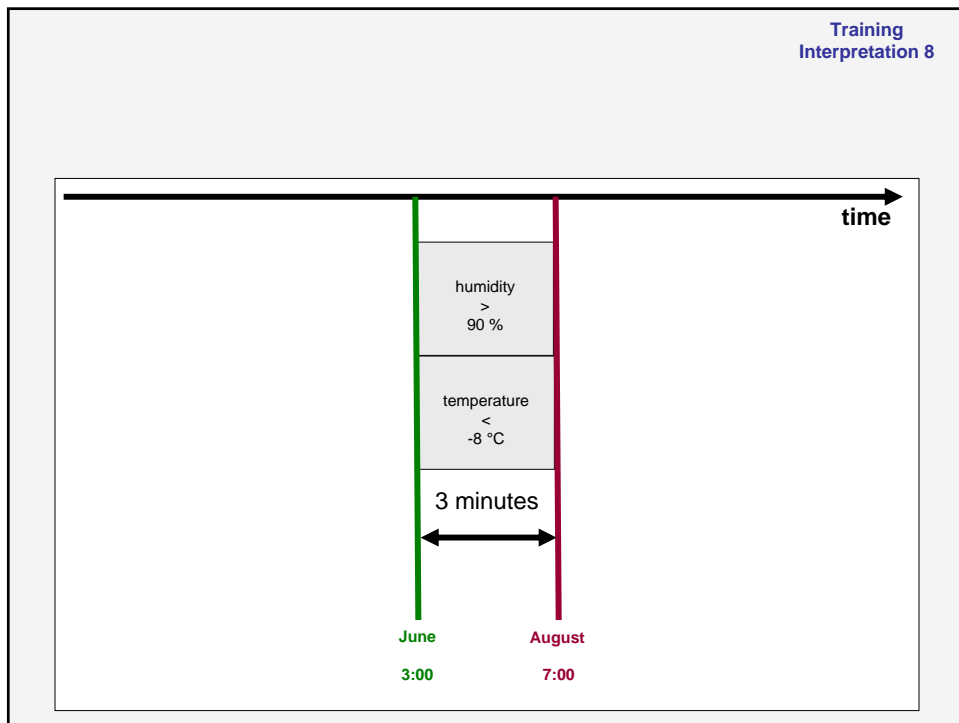
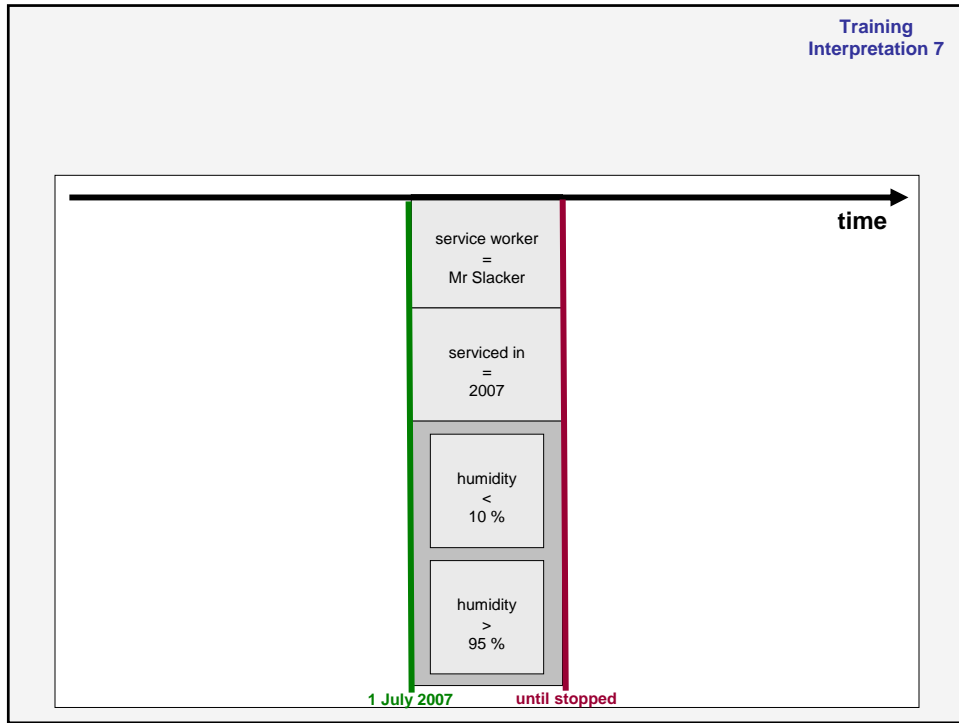
	temperature > 10 °C
	temperature < 30 °C

**Training
Interpretation 2**

	temperature < 0 °C
	temperature > 40 °C
	pressure < 800 hPa







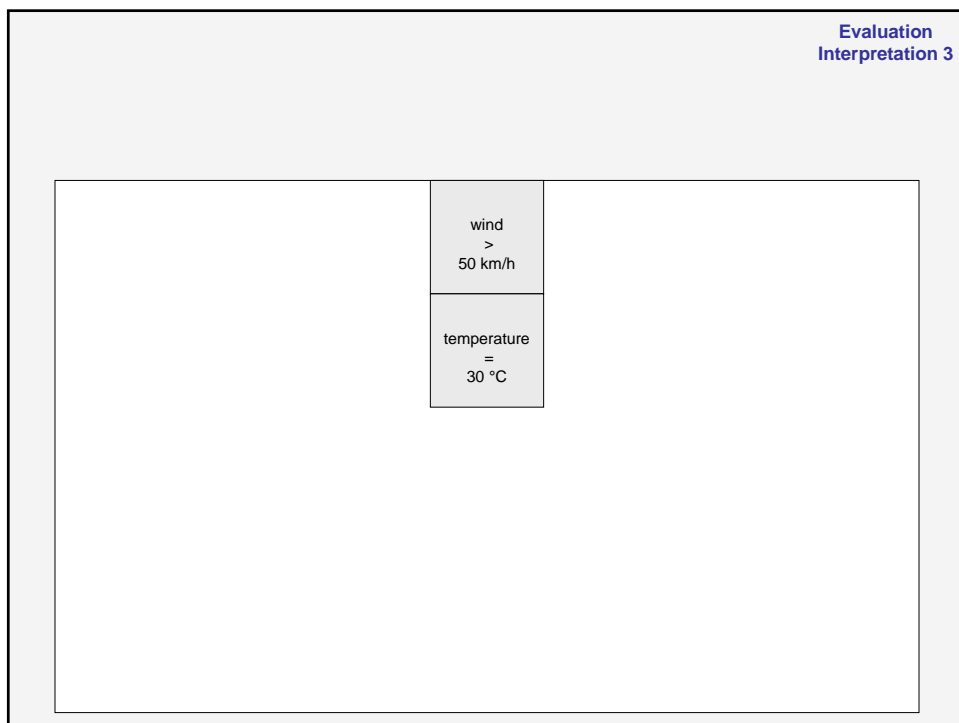
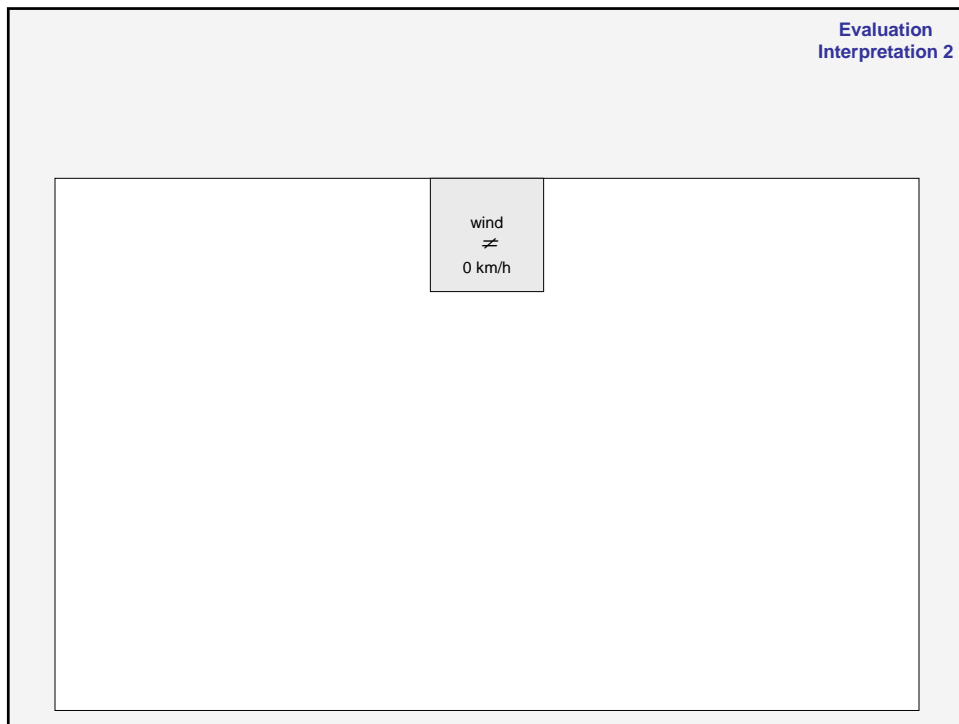
Interpretation Tasks Evaluation

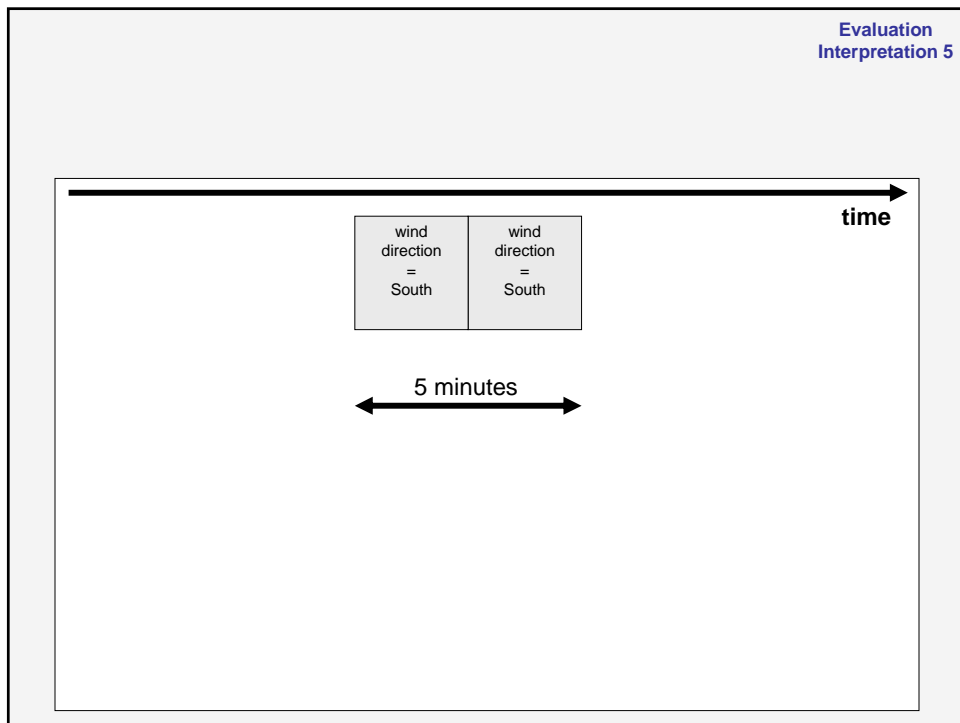
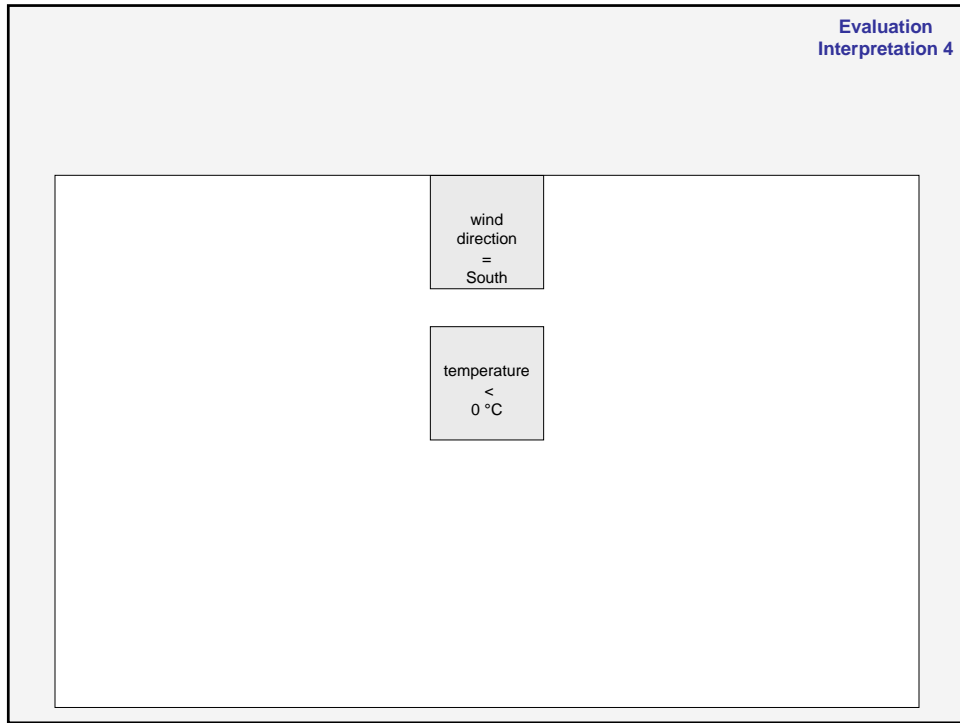
User Study of
an Interface for Profile Specification for
Alerting Systems

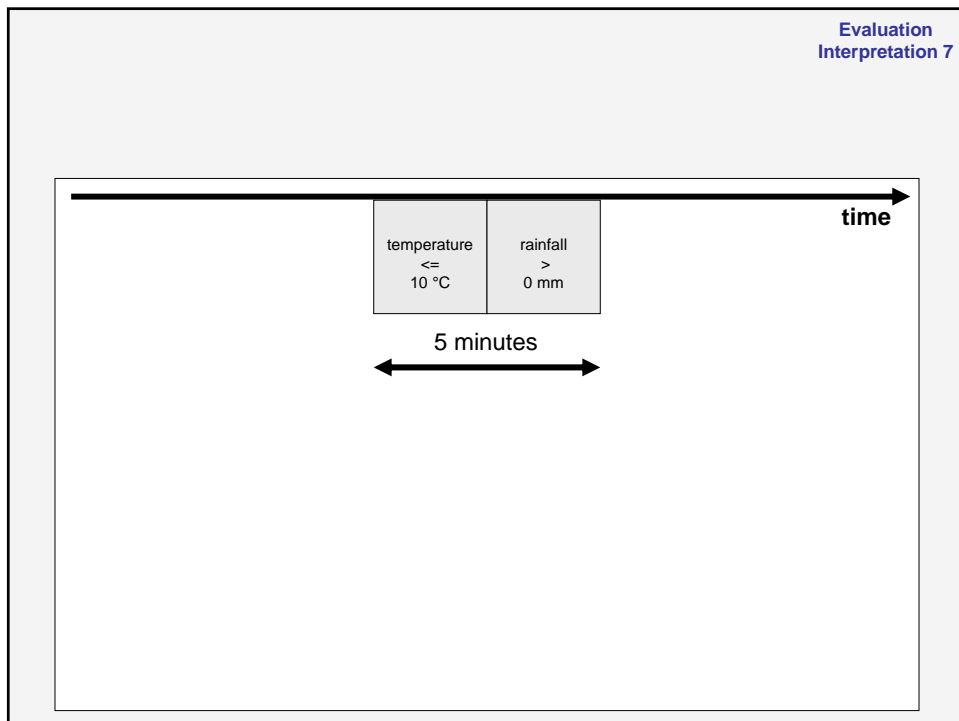
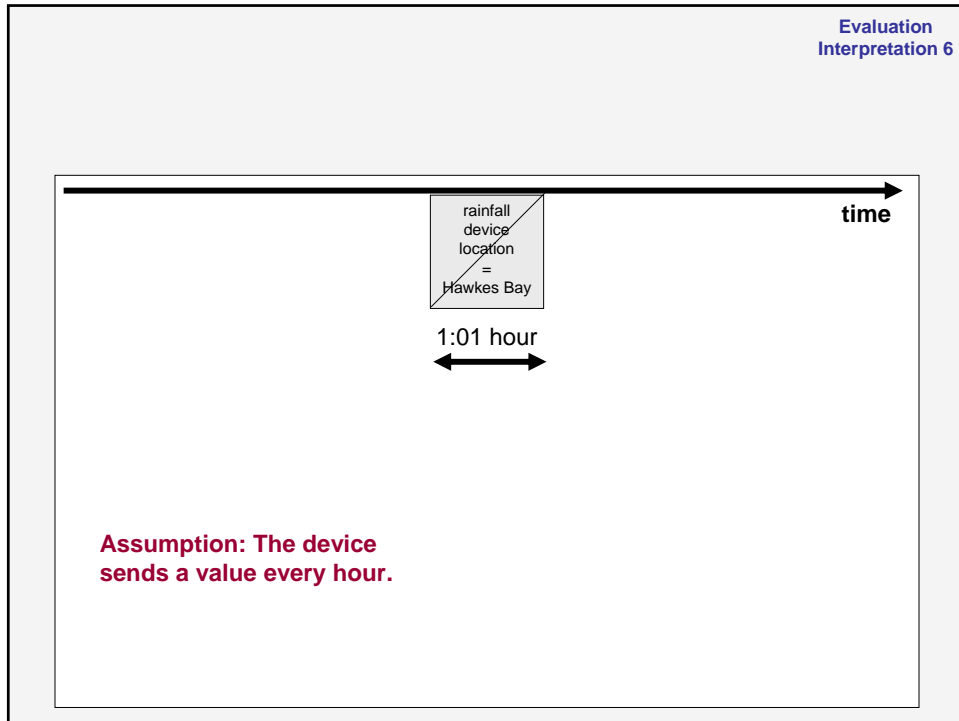
These tasks will be shown in the interface.

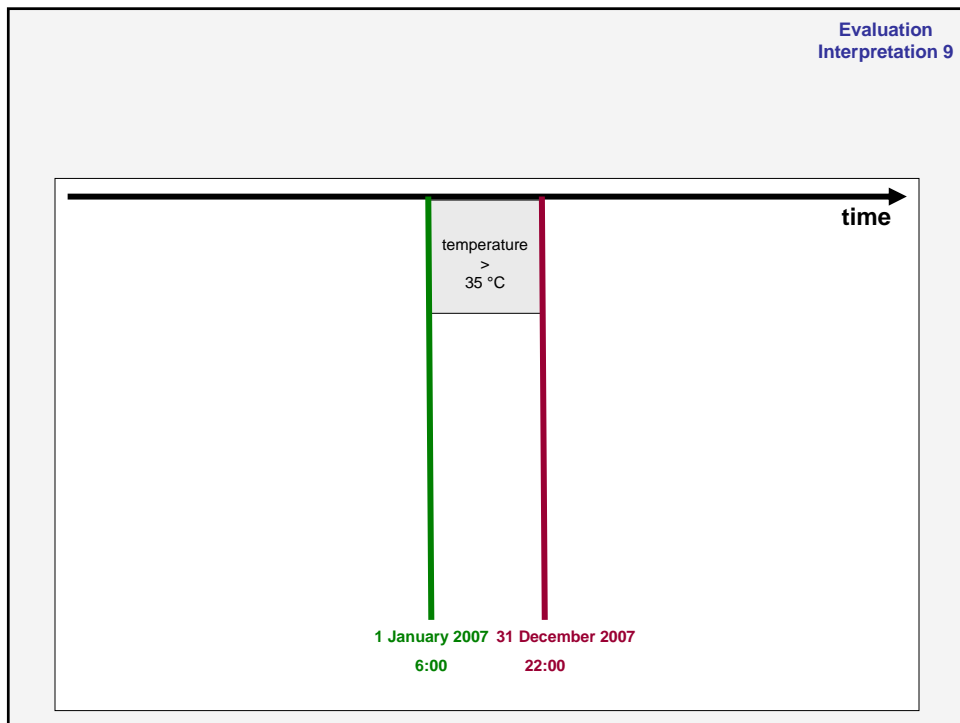
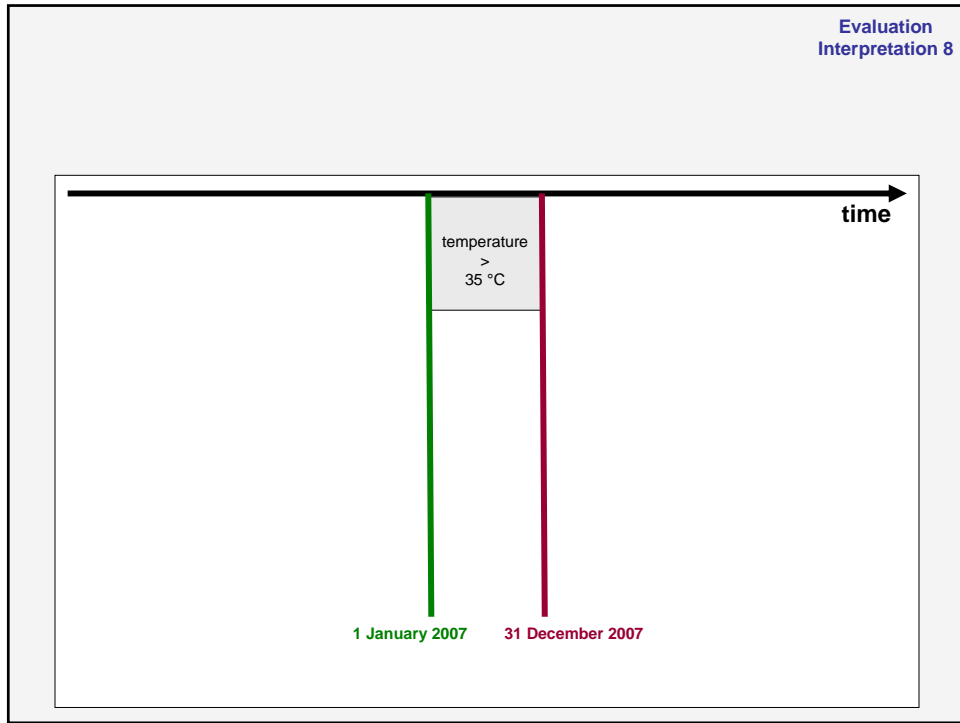
Evaluation
Interpretation 1

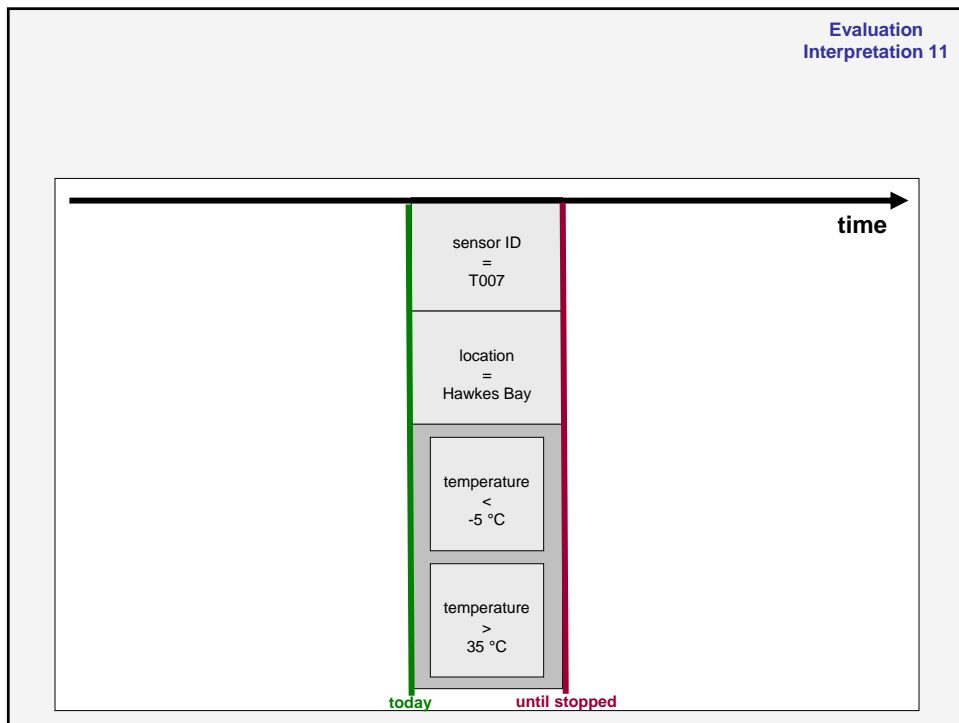
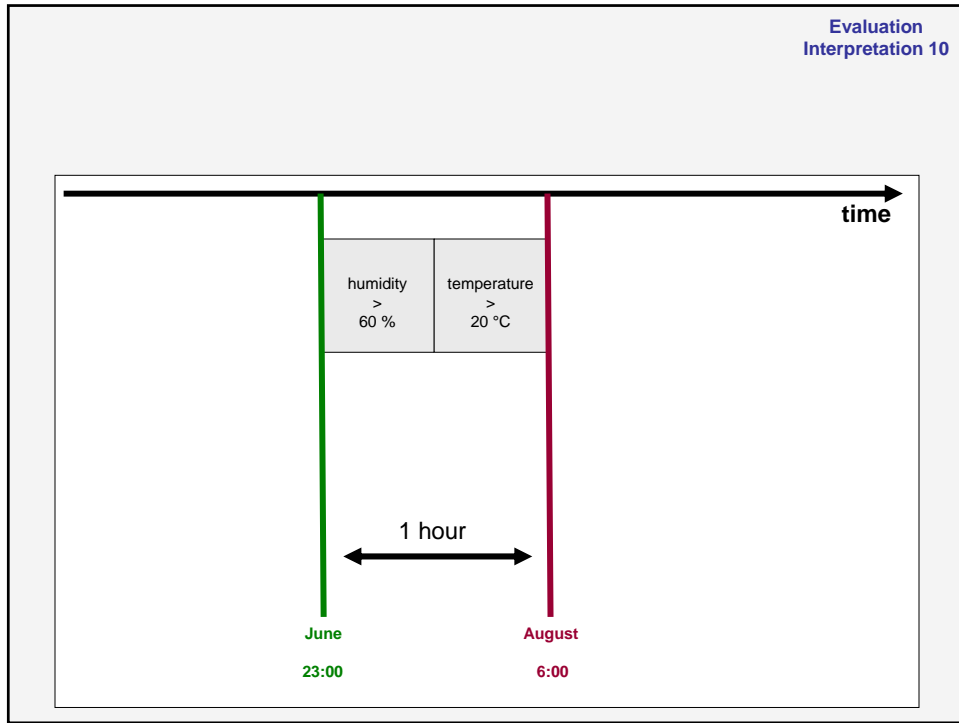
wind
direction
=
North











Specification Tasks

User Study of an Interface for Profile Specification for Alerting Systems

These tasks will be given to the subjects on
paper.

Specification Tasks Training

1. Notify if it is raining and the temperature is colder than 20°C.
2. Notify if there is rainfall of above 5 mm/h or if the humidity is above 70%.
3. Notify if there is a heat wave, i.e. the temperature is above 40°C at least three times in 2 weeks.
4. Notify if the wind direction is North and this is followed by a wind speed of at least 150km/h within maximally 30 seconds.
5. Notify if the wind direction sensor with the identification "WD-00" potentially is broken.
Assumption: Sensor WD-00 sends a value every 5 minutes.
6. Check whether the temperature in 2008 and 2009 ever falls below -10 °C. To save resources, test this only during winter (June - August).
7. Notify if the temperature is smaller than 0 °C. Also notify if the temperature is smaller than 3 °C and this is followed by rainfall. Test this only during winter (June-August).
8. Notify if on three occasions (3 separate months) we have more than 50mm of rainfall in Christchurch in a month. We are interested in this during 2007 only.

Specification Tasks Evaluation

1. Notify if the pressure is above 1000 hPa.
2. Notify if any wind has been registered.
3. Notify if there a breeze (i.e. some wind, it does not matter how strong) coming from the East.
4. Notify if the wind direction is either from the South or from the North.
5. Notify if it is raining. However, make sure that this was not only one short shower, but has been registered more than once within 10 minutes.
6. Notify if the wind speed measurement device located at Mystery Creek is malfunctioning.
Assumption: The device sends a value every 2 minutes.
7. Notify if the wind speed is at least 100 km/h and this is followed by rainfall within 1 minute.
8. Check whether it ever rains more than 50 mm/h (August-September).
9. Check whether in summer (December - February) it ever rains more than 50 mm/h during the day (6:00 – 22:00).
10. Notify if the wind “goes wild”. I.e. it comes from every direction (only consider the four main directions) at least once in an hour. Check this only in summer (December - February) and only during the day (7:00 to 23:00).
11. Notify if within 5 minutes the wind speed is repeatedly (3 times) over 100km/hour, and this is followed by a malfunction of the wind-speed measurement device located at Mystery Creek. Check this in 2007. Assumption: The device sends a value every 2 minutes.



A – Interface

- 1 Please describe your general impression of using the interface for the specification of information needs:**

Please answer the following questions in relation to the tasks that you have just completed. Tick the reply that best represents your response.

- 2 How intuitive did you find using the interface?**

not intuitive very intuitive

- 3 How easy did you find using the interface?**

not easy very easy

- 4 How satisfied are you with using the interface? (I.e. was it annoying or fun?)**

not satisfied very satisfied



- 5 How much mental and perceptual activity was required (e.g. thinking, deciding, remembering, looking, searching, etc.)?**

little a lot

- 6 How much physical activity was required (e.g., selecting, dragging, scrolling, etc.)?**

little a lot

- 7 Overall, to what extent did you become frustrated whilst carrying out the tasks? (I.e. how insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?)**

not at all very

- 8 How successful do you think you were in properly completing the tasks?**

not at all very

- 9 Do you have any comments regarding the above questions?**



B – Computing Background

10 How experienced are you in using queries or alerting (e.g. using Google or digital libraries)?

not experienced very experienced

11 How often do you use queries/alerts?

less than once once per 2-3 per month once per 2-3 per week more than 2-3
per month month week per week

12 How experienced are you in using computers?

not experienced very experienced

13 How often do you use computers?

less than once once per 2-3 per month once per 2-3 per week more than 2-3
per month month week per week



C – Personal Background

14 Is English your first language?

Yes No

15 Give your age on 1st January 2008.

16 Which gender are you?

17 What is your profession?

If you are a student/academic staff please include your area, e.g. mathematics and whether you are an undergrad or grad/postgrad student.

Appendix E. Collaborative Profile Software Interface Study: Participant Workbook



User Study of an Interface for Collaborative Profile Specification for Alerting Systems

Researcher Doris Jung

Participant Workbook

Session Number:

Date and Time:

The University of Waikato School of Computing and Mathematical Sciences

Research Participant's Bill of Rights

The following is a list of your rights if you participate in a research project organised within the School of Computing and Mathematical Sciences at the University of Waikato.

As a research participant, you have the right:

- To be treated with respect and dignity in every phase of the research.
- To be fully and clearly informed of all aspects of the research prior to becoming involved in it.
- To enter into clear, informed, and written agreement with the researcher prior to becoming involved in the activity. You should sense NO pressure, explicit or otherwise, to sign this contract.
- To choose explicitly whether or not you will become involved in the research under the clearly stated provision that refusal to participate or the choice to withdraw during the activity can be made at any time without penalty to you.
- To be treated with honesty, integrity, openness, and straightforwardness in all phases of the research, including a guarantee that you will not unknowingly be deceived during the course of the research.
- To receive something in return for your time and energy.
- To demand proof that an independent and competent ethical review of human rights and protections associated with the research has been successfully completed.
- To demand complete personal confidentiality and privacy in any reports of the research unless you have explicitly negotiated otherwise.
- To expect that your personal welfare is protected and promoted in all phases of the research, including knowing that no harm will come to you.
- To be informed of the results of the research study in a language you understand.
- To be offered an range of research studies or experiences from which to select, if the research is part of fulfilling your educational or employment goals.

The contents of this bill were prepared by the University of Calgary who examined all of the relevant Ethical Standards from the Canadian Psychological Association's Code of Ethics for Psychologists, 1991 and rewrote these to be of relevance to research participants.

Descriptions of the CPA Ethical Code and the CPA Ethical Standards relevant to each of these rights are available at <http://www.cpa.ca/ethics2000.html> and <http://www.psych.ucalgary.ca/Research/ethics/bill/billcode.html> if you would like to examine them.

The complete CPA Ethical Code can be found in Canadian Psychological Association "*Companion manual for the Canadian Code of Ethics for Psychologists*" (1992).

The University of Waikato School of Computing and Mathematical Sciences

Research Consent Form

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

Research Project Title

User Study of an Interface for Collaborative Profile Specification for Alerting Systems.

Researcher

Doris Jung.

Experiment Purpose

This study will evaluate the usability of an Interface for the Collaborative Profile Specification for Alerting Systems.

Participant Recruitment and Selection

Undergraduate and graduate students from the University of Waikato are being recruited for this study. Academic and non-academic staff members of some departments at the University of Waikato might also be recruited for the study. Moreover, nursing students from Wintec as well as other health care providers will be approached and asked if they would be willing to participate in the study.

Procedure

This session will require about 2:00 hours of your time.

You will be introduced to the material of the study in a tutorial which will be followed by a training phase in which you can explore the software to be tested by solving tasks that will be given to you in a natural language description on paper. This will be followed by another introductory phase that presents further features of the software to you. You can explore these features in the subsequent training phase in which you will be asked to solve further tasks that (this time) are given to you by the program. Following to that you will be given further tasks by the program in order to test the usability of the software

Please think aloud while solving the given tasks – describing how you try to solve them.

Afterwards we will have a brief discussion about how you experienced solving the tasks and you will be asked to fill in a short questionnaire about your background.

Data Collection

A questionnaire will be used to collect background information. As you are solving the task sequences and think aloud the interviewer will take notes. The researcher also will record your specification results for the evaluation of the study and record the session.

Data Archiving/Destruction

Data will be kept securely stored by the researcher.

Confidentiality

Confidentiality and participant anonymity will be strictly maintained. All information gathered will be used for statistical analysis only and no names or other identifying characteristics will be stated in the final or any other reports.

Likelihood of Discomfort

There is no likelihood of discomfort or risk associated with participation.

Researcher

Doris Jung. She is working on her doctorate in the Computer Science Department at the University of Waikato. This study will contribute to her research on the realisation and representation of collaborative profiles. Her supervisor is Associate Professor Steve Jones. Doris can be contacted in room G2.27 of the School of Computer and Mathematical Sciences building at the University of Waikato. Her phone number is 838 4547 and her email address is d.jung@cs.waikato.ac.nz.

Finding out about Results

The Participants can find out the results of the study by contacting the researcher after 1/10/2008.

Agreement

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to not answer specific items or questions in interviews or on questionnaires. You are free to withdraw from the study at any time without penalty. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact the researcher.

Participant

Date

Investigator/Witness

Date

A copy of this consent form has been given to you to keep for your records and reference.



Online Tutorial Part 1: Traditional Profiles



Training Part 1: Traditional Profiles

- 1) Notify if patient Miller has a different treatment provider than Dr. Excellent.

- 2) Notify if there is reference material on the topic of hypertension or on the topic of blood pressure.

- 3) Notify if a patient's temperature is measured as critical, i.e. the temperature is above 40 °C for at least three times within 2 minutes.

- 4) Notify if a patient's pulse falls below 50 BPM and then, within the next 30 seconds, the oxygen saturation (SO₂) drops below 94%.

5) Notify if the pulse-oximeter with the Sensor ID PO-00 might be broken.

The pulse-oximeter normally sends data every minute.

6) Notify if patient Miller shows any symptoms of sunburn between 1 January 2009 and 31 December 2010.

This only needs to be tested between 8:00 – 20:00.

7) Notify if a patient's temperature is below 35 °C.

Also notify if the oxygen saturation is less than 94% and this is followed by a pulse greater than 90 BPM.

Test this only in the morning (4:00 – 8:00).

8) Notify if patient Miller is diagnosed with headaches for at least 3 times in a month.

We are interested in this during 2009 only.



Online Tutorial Part 2: Collaborative Profiles



Training Part 2: Collaborative Profiles Owner Tasks

- 1) An insurance company pays a premium to clients if their medical costs per month are low at least three times in a year. An administrator (**you**) sets up a template for monitoring who is eligible for this premium.

This template offers the possibility to check for a “given” patient whether the overall cost per month is below a “certain” (e.g. somewhere between 0NZ\$ and 100NZ\$) threshold for at least three times in a year.

This template is used by the **medical advisor**.

- 2) An insurance company (**you**) wants to know if a patient has been promoted (position = Manager) and within a “relevant time” of this (e.g. between 1 and 6 months) that patient’s blood pressure has risen significantly (systole greater than 200 mmHg or diastole greater than 130 mmHg).

The exact relevant time will be filled in by a **medical advisor**.

3) In a hospital a doctor (**you**) wants to be notified if a patient has abnormal fluctuations in his blood pressure. For this he wants to monitor the following sequence of blood pressure readings:

Systole above 180 mmHg,
then diastole below 60 mmHg,
then systole above 180 mmHg,
then diastole below 60 mmHg.

All of this should happen within 10 minutes.
The data should be monitored during the doctor's shift.

However, the doctor (**you**) does not yet know what shift he will be assigned to, i.e. when the data should be monitored. So an **administrator** has to update this information. However, the doctor never works overnight, so it will be sometime between 6:00 and 21:00.

4) A parent (**you**) wants to get notified if the blood pressure parameters of their child, Peter, are critical.

She knows that his systole may be 150 mmHg but no higher. She knows that his diastole may be 100 mmHg and no higher.

However, she does not know if already one extreme value is critical or if it only matters if both of them are elevated. A **doctor** will help her with this decision.

5) A nurse (**you**) specifies to be notified if Mr. Miller's blood pressure is too high several times in one week.

Too high in his case means a systole of above 180 mmHg

or

a diastole of above 120 mmHg.

The nurse only wants to be notified if this happens "a critical amount of times" per week.

How often is considered critical would be provided by a **doctor**. You can offer him the range of 2 to 10 repetitions.

6) A doctor (**you**) wants to create a profile for his patients to be supplied with electronic reference material.

A notification should be sent if there is new material on heart disease.

The patient wants to be notified if the above holds **and** if other relevant limitations hold. The patient should be able to add these limitations later. Thus, enable the **patient** to extend the profile.



Training Part 2: Collaborative Profiles Refiner Tasks

1) Mr. Miller

2) 1 month

3) 8:00 – 16:00

4) It is critical as soon as one of them occurs.

5) You want to be notified as soon as it is too high at least 3 times.

6) For this example assume that you generally have a very high expertise in heart disease. However, as soon as it comes to alternative medicine your expertise is only low.



Evaluation: Collaborative Profiles Owner Tasks

- 1) A nurse (**you**) prepares a template for an intensive care unit for emergency situations.

The template specifies a vague profile that monitors if either the systole is below a “certain threshold” (offer a range of 80-100mmHg) or if the diastole is below a “certain” threshold (offer a range of 40 – 60 mmHg) for “some” patient.

The template will be used by **doctors**.

2) The nurse (**you**) wants to be notified if Mr. Miller has cancelled three appointments within a “certain period of time”, which might become critical.

The **doctor** knows further details about the duration of this period of time and will specify this at a later point in time. (Offer a range between 1 and 6 months.)

3) A doctor (**you**) defines a profile for detecting high blood pressure during the day.

He wants to be notified if the systole is above 130 mmHg and the diastole is above 80 mmHg.

The doctor leaves it open as to when the day begins and ends, which is to be defined by the **patient** himself. However, he knows for certain that the patient will not get up before 5:00 and will go to bed at 23:00 at the latest.

4) A nurse (**you**) knows that the sequence of two certain symptoms indicates danger of a heart attack.

However, she can't remember which order of these symptoms indicates the critical condition – was it A and then B, or was it B and then A?? She will have to ask the **doctor** for help.

5) A patient's cholesterol level is measured each week...

A nurse (**you**) knows that it is unhealthy for patient Miller to have a cholesterol level higher than 250 mg/dl. It is not critical as a one-off occurrence. However, if it happens repeatedly in a month, she wants to be notified because his treatment has to be changed.

The nurse is unsure "how often in a month" the level may be increased without any major risk for the patient's health. She gives this job to the **doctor**. She offers him values between 2 and 10.

6) An insurance wants to know if at least three times in a month there are unexpected events for Mr. Miller.

The administrator (**you**) prepares a template that can be extended later on.

This template says that the above should hold **and** that further information (of whatever sort) has to hold in order for the insurance to be notified.

A medical advisor will add this extension later.



Evaluation: Collaborative Profiles Refiner Tasks

- 1) A blood pressure no lower than 90 mmHg for the systole and no lower than 50 mmHg for the diastole
- 2) 1 month
- 3) 7:00 – 23:00
- 4) B then A
- 5) Notify if it occurs 5 times or more.
- 6) Any cost that is higher than 500 NZ\$ and that is not of type Ophthalmologist is an unexpectedly high cost for Mr. Miller.



A – Interface

- 1 Please describe your general impression of using the interface for the collaborative specification of information needs:**

Please answer the following questions in relation to the tasks that you have just completed. Circle the reply that best represents your response.

- 2 How intuitive did you find using the interface?**

not intuitive 1 2 3 4 5 very intuitive

- 3 How easy did you find using the interface?**

not easy 1 2 3 4 5 very easy

- 4 How satisfied were you with using the interface?**

not satisfied 1 2 3 4 5 very satisfied



5 How much mental and perceptual activity (e.g. thinking, deciding, remembering, looking, searching, etc.) was required using the interface - irrespective of what task it was?

little 1 2 3 4 5 a lot

6 How much physical activity was required (e.g. selecting, dragging, scrolling, etc.)?

little 1 2 3 4 5 a lot

7 Overall, to what extent did you become frustrated whilst carrying out the tasks?

not at all 1 2 3 4 5 very

8 How successful do you think you were in properly completing the tasks?

not at all 1 2 3 4 5 very

9 Do you have any comments regarding the above questions?



B – Computing Background

10 How experienced are you in using queries (e.g. using Google)?

not experienced 1 2 3 4 5 very experienced

11 How often do you use queries?

1	2	3	4	5	6
once per month or less	2-3 per month	once per week	2-3 per week	more than 2-3 per week	daily

12 How experienced are you in setting up alerts?

not experienced 1 2 3 4 5 very experienced

13 How often do you use (i.e. set up) alerts?

1	2	3	4	5	6
once per month or less	2-3 per month	once per week	2-3 per week	more than 2-3 per week	daily

**14 How experienced are you in using computers?**

not experienced 1 2 3 4 5 very experienced

15 How often do you use computers?

1 2 3 4 5 6

once per 2-3 per month once per 2-3 per week more than 2-3 daily
month or less week per week

C – Personal Background**16 Is English your first language?**

Yes No

17 Give your age on 1st January 2008.**18 Which gender are you?****19 What is your profession?**

If you are a student/academic staff please include your area, e.g. mathematics and whether you are an undergrad or grad/postgrad student.

