

Manuel Rudolph

# Generation of Usable Policy Administration Points for Security and Privacy



Editor-in-Chief: Prof. Dr. Dieter Rombach  
Editorial Board: Prof. Dr. Peter Liggesmeyer  
Prof. Dr. Frank Bomarius

FRAUNHOFER VERLAG

# **PhD Theses in Experimental Software Engineering**

Volume 68

Editor-in-Chief: Prof. Dr. Dieter Rombach

Editorial Board: Prof. Dr. Frank Bomarius  
Prof. Dr. Peter Liggesmeyer  
Prof. Dr. Dieter Rombach

Manuel Rudolph

## **Generation of Usable Policy Administration Points for Security and Privacy**

Fraunhofer Verlag

Zugl.: Kaiserslautern, TU, Diss., 2019

Printing:  
Mediendienstleistungen des  
Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart

Printed on acid-free and chlorine-free bleached paper.

All rights reserved; no part of this publication may be translated, reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. The quotation of those designations in whatever way does not imply the conclusion that the use of those designations is legal without the consent of the owner of the trademark.

© by **Fraunhofer Verlag**, 2020  
ISBN (Print): 978-3-8396-1579-9  
Fraunhofer-Informationszentrum Raum und Bau IRB  
Postfach 800469, 70504 Stuttgart  
Nobelstraße 12, 70569 Stuttgart  
Telefon +49 711 970-2500  
Telefax +49 711 970-2508  
E-Mail [verlag@fraunhofer.de](mailto:verlag@fraunhofer.de)  
URL <http://verlag.fraunhofer.de>

# **Generation of Usable Policy Administration Points for Security and Privacy**

Vom Fachbereich Informatik  
der Technischen Universität Kaiserslautern  
zur Verleihung des akademischen Grades

**Doktor der Ingenieurwissenschaften (Dr.-Ing.)**

genehmigte Dissertation  
von

**Manuel Rudolph, M. Sc.**

Fraunhofer-Institut für Experimentelles Software Engineering IESE  
(Fraunhofer IESE)  
Technische Universität Kaiserslautern

Berichterstatter:

Prof. Dr. Dr. h.c. Dieter Rombach  
Prof. Dr. Alexander Pretschner

Dekan:

Prof. Dr. Stefan Deßloch

Datum der wissenschaftlichen Aussprache:

11.12.2019





To Sabrina and Emily



# Acknowledgements

I would like to express my gratitude to many people who supported me in the last years while doing my PhD.

First, I would like to thank my family and friends who have helped me in many ways over the last few years while I was doing my doctoral thesis. My special thanks go to my wife Sabrina, who has continuously supported and encouraged me with great patience to pursue my PhD. I would like to thank my parents Monika and Werner for making my carefree education possible and for constantly believing in my goals. Finally, I would like to thank my daughter Emily, who strongly motivated me to finish this work fast.

I thank Prof. Dieter Rombach and Prof. Alexander Pretschner for their valuable advice and guidance during this thesis. Especially their insightful questions and feedback, which provided different perspectives on the thesis, improved it in many ways. Many thanks also go to my division head Dr. Jörg Dörr for continuously giving me feedback and many valuable hints for improvement.

During the thesis research, I received much support from my colleagues at Fraunhofer IESE. I am very grateful to my colleague Svenja Polst for her advice regarding psychological aspects and support on the experiment in this thesis. My work also much benefited from countless discussions with my esteemed colleagues Christian Jung, Denis Feth, Cornelius Moucha, Raj Shah and Anne Hess. I thank the students Kevin Schmitt, Christian Vollat and Thorsten Reinberger-Eyer who performed parts of the implementation work and literature research. Finally yet importantly, I would like to thank Reinhard Schwarz for proofreading this piece of work.



# Abstract

Users want to gain more self-determination in the form of self-responsible definition and control of their security and privacy demands. To this end, they can use so-called Policy Administration Points (PAPs) for the specification of security and privacy policies. However, users face usability problems with existing tools. PAPs provide different specification paradigms, which determine the specification process for the task of policy specification including the level of expressiveness and guidance that the user is getting during the specification.

This thesis addresses the topic of automated creation of usable Policy Administration Points. First, it focusses on the mapping of specification paradigms to user groups for increasing the usability by means of effectiveness, efficiency and satisfaction. Second, we propose a method for the automated creation of PAPs. This includes a method for eliciting security and privacy policy templates from an application domain, a policy template model for formalizing these policy templates as well as a PAP generation framework for the automated creation of policy specification interfaces within PAPs based on an instance of the policy template model.

We first present two case studies that reveal improvement potential for our contributions and next explain another two case studies that validate our results. More specifically, we confirm with these case studies the effectiveness of the policy template elicitation method, the completeness of the policy template model and the feasibility of the policy specification interface generation with our PAP generation framework. Finally, we confirm in an experiment that selecting the most appropriate specification paradigm for a user significantly increases the usability of a PAP for the task of policy specification.



# Table of Contents

|   |              |
|---|--------------|
| <b>Acknowledgements .....</b>   | <b>v</b>     |
| <b>Abstract .....</b>   | <b>vii</b>   |
| <b>List of Figures.....</b>   | <b>xiii</b>  |
| <b>List of Tables .....</b>   | <b>xxi</b>   |
| <b>List of Abbreviations .....</b>  | <b>xxiii</b> |
| <b>1 Introduction .....</b>   | <b>1</b>     |
| 1.1 Motivation .....  | 1            |
| 1.2 Problem Derivation Surveys .....  | 3            |
| 1.2.1 PDS1: »SECCRIT« Survey .....  | 4            |
| 1.2.2 PDS2: »Museum Pfalzgalerie Kaiserslautern« Survey....                       | 6            |
| 1.2.3 PDS3: Survey in the context of the policy specification<br>experiment ..... | 9            |
| 1.2.4 Summary and Conclusion .....  | 11           |
| 1.3 Problem Statement .....   | 12           |
| 1.4 Contribution .....  | 16           |
| 1.5 Hypotheses .....  | 19           |
| 1.5.1 Hypotheses for RQ1: Usability of Specification<br>Paradigms.....            | 19           |
| 1.5.2 Hypotheses for RQ2: Elicitation .....                                       | 23           |
| 1.5.3 Hypotheses for RQ3: Formalization .....                                     | 25           |
| 1.5.4 Hypotheses for RQ4: Automation .....  | 26           |
| 1.6 Research Approach .....   | 27           |
| 1.7 Assumptions and Limitations.....  | 28           |
| 1.8 Outline.....  | 29           |
| <b>2 Foundations and Related Work .....</b>                                       | <b>31</b>    |
| 2.1 Research Approach .....   | 31           |
| 2.2 Elicitation of Security and Privacy Requirements .....                        | 31           |
| 2.3 Policy Models and Languages.....  | 36           |
| 2.4 Usable Security and Privacy Policy Specification .....                        | 40           |
| 2.5 Existing PAPs and Derived Specification Paradigms.....                        | 45           |
| 2.5.1 Security and Privacy Specification Approaches and<br>Tools.....             | 45           |
| 2.5.2 Overview of Derived Specification Paradigms .....                           | 51           |
| 2.6 User Behavior.....  | 54           |
| 2.6.1 Intension Models .....  | 54           |
| 2.6.2 User Type Models .....  | 55           |
| 2.7 Summary and Conclusion .....  | 57           |



|          |  |            |
|----------|--|------------|
| <b>3</b> | <b>Policy Template Elicitation Method .....</b>                                      | <b>61</b>  |
| 3.1      | Research Approach .....  | 62         |
| 3.2      | Method Overview .....  | 64         |
| 3.3      | Step 1: Information Retrieval .....  | 66         |
| 3.4      | Step 2: Workshop Preparation.....  | 68         |
| 3.5      | Step 3: Execution of Elicitation Workshop .....                                      | 71         |
| 3.6      | Step 4: Documentation of Workshop Results .....                                      | 75         |
| 3.7      | Step 5: Policy Template Derivation and Validation .....                              | 78         |
| 3.8      | Summary and Conclusion.....  | 81         |
| <b>4</b> | <b>Policy Template Model.....</b>  | <b>83</b>  |
| 4.1      | Research Approach .....  | 84         |
| 4.2      | Overview of Policy Template Model.....   | 85         |
| 4.3      | Domain Sub-model .....   | 87         |
| 4.4      | Security and Privacy Sub-model.....  | 90         |
| 4.5      | Template Sub-model .....   | 90         |
| 4.6      | Specification-Level Template Sub-model.....  | 93         |
| 4.7      | Implementation-Level Template Sub-model.....   | 95         |
| 4.8      | Specification Paradigm Projection Sub-model .....                                    | 98         |
| 4.9      | Example .....  | 100        |
| 4.10     | Summary and Conclusion.....  | 108        |
| <b>5</b> | <b>PAP Generation Framework .....</b>  | <b>111</b> |
| 5.1      | Research Approach .....  | 112        |
| 5.2      | Reference Architecture.....  | 114        |
| 5.2.1    | Architectural Overview.....  | 114        |
| 5.2.2    | Concept for Embedding a Generated Policy<br>Specification Interface into a PAP ..... | 124        |
| 5.3      | Specification Paradigms.....   | 124        |
| 5.3.1    | Selection of Specification Paradigms .....   | 124        |
| 5.3.2    | Specification Paradigm Algorithms.....   | 127        |
| 5.4      | Reference Implementation .....   | 134        |
| 5.5      | Summary and Conclusion.....  | 140        |
| <b>6</b> | <b>Mapping Users to Specification Paradigms.....</b>                                 | <b>143</b> |
| 6.1      | Research Approach .....  | 143        |
| 6.2      | Mapping Specification Paradigms to Users .....                                       | 144        |
| 6.2.1    | User Intention Model .....   | 145        |
| 6.2.2    | Example for Barriers of a PAP.....   | 148        |
| 6.2.3    | Matching Specification Paradigms to Users .....                                      | 149        |
| 6.3      | Mapping Specification Paradigms to Personas.....                                     | 150        |
| 6.3.1    | Selection of Persona Model.....  | 151        |
| 6.3.2    | Mapping the Specification Paradigms to the<br>Personas of Dupree.....                | 151        |
| 6.4      | Summary and Conclusion.....  | 153        |
| <b>7</b> | <b>Method for Usable PAP Generation .....</b>  | <b>155</b> |
| 7.1      | Research Approach .....  | 155        |
| 7.2      | Method Overview .....  | 156        |

|          |   |            |
|----------|---|------------|
| 7.3      | Step 1: Policy Template Elicitation .....               | 158        |
| 7.4      | Step 2: Instantiation of Policy Template Model .....    | 159        |
| 7.5      | Step 3: Instantiation of PAP Generation Framework ..... | 161        |
| 7.6      | Step 4: Specification Paradigm Selection .....          | 162        |
| 7.7      | Step 5: Specification of Policy with PAP .....          | 163        |
| 7.8      | Summary and Conclusion .....                            | 164        |
| <b>8</b> | <b>Evaluation for Improvements .....</b>                | <b>165</b> |
| 8.1      | Research Approach .....                                 | 165        |
| 8.2      | Case Study: Software Cluster Project »SINNODIUM« .....  | 165        |
| 8.2.1    | Project Summary .....                                   | 166        |
| 8.2.2    | Design and Execution .....                              | 166        |
| 8.2.3    | Results .....   | 168        |
| 8.2.4    | Observations and Lessons Learned .....                  | 171        |
| 8.2.5    | Threats to validity .....                               | 172        |
| 8.2.6    | Summary .....   | 173        |
| 8.3      | Case Study: European Project »SECCRIT« .....            | 174        |
| 8.3.1    | Project Summary .....                                   | 174        |
| 8.3.2    | Design and Execution .....                              | 175        |
| 8.3.3    | Results .....   | 176        |
| 8.3.4    | Observations and Lessons Learned .....                  | 178        |
| 8.3.5    | Threats to validity .....                               | 180        |
| 8.3.6    | Summary .....   | 181        |
| 8.4      | Summary and Conclusion .....                            | 182        |
| <b>9</b> | <b>Evaluation for Validation .....</b>                  | <b>185</b> |
| 9.1      | Research Approach .....                                 | 185        |
| 9.2      | Case Study: Software Campus Project »BeSure« .....      | 186        |
| 9.2.1    | Project Summary .....                                   | 186        |
| 9.2.2    | Design and Execution .....                              | 187        |
| 9.2.3    | Results .....   | 190        |
| 9.2.4    | Observations and Lessons Learned .....                  | 194        |
| 9.2.5    | Threats to validity .....                               | 198        |
| 9.2.6    | Summary .....   | 199        |
| 9.3      | Case Study: »Digital Villages« .....                    | 200        |
| 9.3.1    | Project Summary .....                                   | 200        |
| 9.3.2    | Design and Execution .....                              | 201        |
| 9.3.3    | Results .....   | 203        |
| 9.3.4    | Observations and Lessons Learned .....                  | 207        |
| 9.3.5    | Threats to validity .....                               | 208        |
| 9.3.6    | Summary .....   | 209        |
| 9.4      | Policy Specification Experiment .....                   | 210        |
| 9.4.1    | Design and Execution .....                              | 211        |
| 9.4.2    | Data Analysis and Results .....                         | 217        |
| 9.4.3    | Discussion .....  | 235        |
| 9.4.4    | Threats to validity .....                               | 238        |
| 9.4.5    | Summary and Conclusion .....                            | 239        |
| 9.5      | Summary and Conclusions .....                           | 241        |

|  |            |
|--|------------|
| <b>10 Summary and Future Work .....</b>                      | <b>249</b> |
| 10.1 Methodological and Technological Contributions.....     | 250        |
| 10.2 Empirical Contributions .....                           | 251        |
| 10.3 Validation Results.....                                 | 253        |
| 10.4 Open Issues and Future Work.....                        | 256        |
| <b>References .....</b>                                      | <b>261</b> |
| <b>Appendix A Security Policy Template Elicitation .....</b> | <b>269</b> |
| A.1 Elicitation Techniques.....                              | 269        |
| A.2 Documentation Techniques.....                            | 275        |
| A.3 Validation Techniques .....                              | 278        |
| A.4 Prioritization Techniques .....                          | 280        |
| A.5 Generic Attacker Roles, Threats and Countermeasures..... | 281        |
| <b>Appendix B PAP Generation Framework.....</b>              | <b>285</b> |
| B.1 XML Schema for Policy Vocabularies .....                 | 285        |
| <b>Appendix C The Personas of the Dupree Model.....</b>      | <b>291</b> |
| <b>Appendix D Case Study: »SECCRIT« .....</b>                | <b>295</b> |
| D.1 Excerpt of »SECCRIT« Study Results .....                 | 295        |
| D.2 Example of Policy Template in Policy Vocabulary .....    | 301        |
| <b>Appendix E Case Study: »BeSure« .....</b>                 | <b>305</b> |
| E.1 Excerpt of »BeSure« Study Results .....                  | 305        |
| <b>Appendix F Case Study: »Digital Villages« .....</b>       | <b>309</b> |
| F.1 Excerpt of »Digital Villages« Study Results.....         | 309        |
| <b>Appendix G Policy Specification Experiment.....</b>       | <b>311</b> |
| G.1 Invitation Email .....                                   | 311        |
| G.2 Experiment Handout .....                                 | 312        |
| G.3 Screenshots of Experiment .....                          | 314        |
| G.4 Sample Solution .....                                    | 334        |
| G.5 Detailed Results of Statistical Analyses.....            | 337        |
| G.6 Raw Data .....   | 351        |
| <b>Lebenslauf .....</b>                                      | <b>353</b> |

# List of Figures

|            |   |    |
|------------|---|----|
| Figure 1:  | Overview of the Problem Derivation Surveys .....  | 3  |
| Figure 2:  | SECCRIT Survey Question 1 – »Do You Think That End Users Should Be Enabled to Specify Their Own Security Policies for Protecting Their Data in Cloud Services?« ..... | 5  |
| Figure 3:  | SECCRIT Survey Question 2 – »Do You Think That Usability Issues Are a Major Concern Regarding End Users Specifying Their Own Security Policies?« .....                | 6  |
| Figure 4:  | MPK Survey Question 1 – »How Often Do You Check Your Security and Privacy Settings?« .....  | 7  |
| Figure 5:  | MPK Survey Question 2 – »Why Don't You Use Security and Privacy Settings More Often?« .....   | 8  |
| Figure 6:  | Experiment Survey Question 1 – »How Often Do You Update the Security and Privacy Settings of Each Web Service on Average?« .....                                      | 10 |
| Figure 7:  | Experiment Survey Question 2 – »What Keeps You from Updating Your Security and Privacy Settings More Often?« .....  | 10 |
| Figure 8:  | Overview of Survey Results.....   | 12 |
| Figure 9:  | Relation of Practical Problems, Scientific Problem and Research Questions.....  | 16 |
| Figure 10: | Relation between Practical Problems, Scientific Problem, Research Questions and Contributions.....  | 18 |
| Figure 11: | The Empirical Contributions Mapped to the Evaluations for Improvement and Validation .....  | 27 |
| Figure 12: | Relation between Contributions, Hypotheses and Case Studies and the Experiment.....   | 28 |
| Figure 13: | Relation between Practical and Scientific Problems and Case Studies, the Experiment and the Hypotheses .....  | 28 |
| Figure 14: | Dupree's Persona Model .....  | 56 |
| Figure 15: | Research Approach for the Policy Template Elicitation Method.....   | 63 |
| Figure 16: | Policy Template Elicitation Method.....   | 65 |
| Figure 17: | Examples of Elicited Assets, Threats and Countermeasures .....  | 72 |
| Figure 18: | Exemplary Result of the Asset Elicitation .....   | 75 |
| Figure 19: | Research Approach for the Policy Template Model.....  | 84 |
| Figure 20: | Policy Template Model .....   | 87 |
| Figure 21: | Domain Sub-model .....  | 88 |
| Figure 22: | Security and Privacy Sub-model.....   | 89 |

|  |     |
|--|-----|
| Figure 23: Meta Model - Model - Instance .....   | 91  |
| Figure 24: Template Sub-model.....   | 92  |
| Figure 25: Specification-Level Template Sub-model.....   | 94  |
| Figure 26: Implementation-Level Template Sub-model.....  | 95  |
| Figure 27: Specification Paradigm Projection Sub-model .....   | 97  |
| Figure 28: Excerpt of the Policy Template Model Showing the<br>Interplay of SLP and ILP Elements and the Relation of<br>a Policy to Domain Elements .....                              | 101 |
| Figure 29: Generated Specification Interface for Exemplary SLPT<br>Implementing the Specification Paradigm »Template<br>Instantiation« .....   | 103 |
| Figure 30: Exemplary Instantiated Policy in the IND <sup>2</sup> UCE Policy<br>Language .....  | 105 |
| Figure 31: Excerpt of the Policy Template Model Showing the<br>Interplay between SLPTs and the Elements for Defining<br>Projection Rules for Different Specification Paradigms .....   | 106 |
| Figure 32: Generated Specification Interface Implementing the<br>Specification Paradigm »Default Policies«, which Shows<br>the Specified Projection Rules for the Exemplary SLPT ..... | 106 |
| Figure 33: Generated Specification Interface Implementing the<br>Specification Paradigm »Wizard«, which Shows one<br>Wizard Page for the Exemplary SLPT .....                          | 107 |
| Figure 34: Research Approach for the PAP Generation Framework ...  | 112 |
| Figure 35: Model-View-Controller Concept in the PAP Generation<br>Framework.....   | 114 |
| Figure 36: Inheritance Relation between Model, Presenter and<br>Controller Layers .....  | 116 |
| Figure 37: Relation of Elements between Layers .....   | 117 |
| Figure 38: Interfaces for View Elements of the View Layer .....  | 118 |
| Figure 39: Interfaces for Presentation Elements of the Presentation<br>Layer .....   | 120 |
| Figure 40: Interfaces and Elements of the Controller Layer .....   | 122 |
| Figure 41: Concept for Embedding a Generated Policy Specification<br>Interface into a PAP.....   | 123 |
| Figure 42: Selection of Specification Paradigms .....  | 125 |
| Figure 43: Mockup of Specification Paradigm »Default Policies«.....  | 128 |
| Figure 44: Mockup of Specification Paradigm »Security Levels« .....  | 129 |
| Figure 45: Mockup of Specification Paradigm »Wizard« .....   | 130 |
| Figure 46: Mockup of Specification Paradigm »Template<br>Instantiation« .....  | 133 |
| Figure 47: Current Modules in the Reference Implementation of the<br>PAP Generation Framework .....  | 135 |
| Figure 48: Injection of Presentation Elements at runtime .....   | 138 |

|  |     |
|--|-----|
| Figure 49: Policy Editor in UI Framework »JavaFX« that Embeds a PAP and Supports Policy Management Functionality .....                     | 138 |
| Figure 50: Exemplary PAP Using View Module »JavaFX«, Policy Vocabulary »CS4« and Presentation Module »Template Instantiation« .....        | 138 |
| Figure 51: ILP in MYDATA Policy Language Version 4.0 Generated by the PAP in UI Framework »JavaFX« .....                                   | 139 |
| Figure 52: Example PAP using View Module »JavaFX«, Policy Vocabulary »CS4« and Presentation Module »Wizard« ....                           | 139 |
| Figure 53: Example PAP using View Module »JavaFX«, Policy Vocabulary »CS4« and Presentation Module »Default Policies« .....                | 140 |
| Figure 54: Example PAP using View Module »JavaFX«, Policy Vocabulary »CS4« and Presentation Module »Security Levels« .....                 | 140 |
| Figure 55: User Intention Model .....  | 145 |
| Figure 56: User Type and Persona Models .....  | 150 |
| Figure 57: Assumed Matching of our Specification Paradigms to the Personas of Dupree for Best Usability .....                              | 152 |
| Figure 58: Research Approach for the Method for Usable PAP Generation .....  | 156 |
| Figure 59: Customization Decisions for a PAP at Development Time and Runtime .....   | 157 |
| Figure 60: Overview of the Method for Usable PAP Generation .....  | 158 |
| Figure 61: Exemplary PAP Using View Module »Android«, Policy Vocabulary »SINNODIUM« and Presentation Module »Template Instantiation« ..... | 170 |
| Figure 62: ILP in IND <sup>2</sup> UCE Policy Language Version 1.1 Generated by the Android PAP .....                                      | 171 |
| Figure 63: Second Version of the Policy Template Elicitation Method  | 175 |
| Figure 64: Example PAP Using View Module »Android«, Policy Vocabulary »BeSure« and Presentation Module »Template Instantiation« .....      | 193 |
| Figure 65: Security Knowledge to Persona Mapping .....   | 220 |
| Figure 66: Security Motivation to Persona Mapping .....  | 220 |
| Figure 67: Boxplot Diagram of the Participants' Age .....  | 221 |
| Figure 68: Ratio of Mistakes Made by Personas per Paradigm to All Decisions .....  | 224 |
| Figure 69: Time Needed in Seconds to Complete all Six Tasks with a Specification Paradigm per Persona .....                                | 228 |
| Figure 70: Participant's Satisfaction with Specification Paradigms .....   | 231 |
| Figure 71: Participant's Satisfaction with Specification Paradigms per Persona .....   | 231 |

|  |     |
|--|-----|
| Figure 72: Goal Tree - OR Decomposition .....  | 277 |
| Figure 73: Goal Tree - AND Decomposition .....   | 277 |
| Figure 74: Character Traits for Persona »Fundamentalist« .....   | 291 |
| Figure 75: Character Traits for Persona »Amateur«.....   | 292 |
| Figure 76: Character Traits for Persona »Marginally Concerned« .....   | 292 |
| Figure 77: Character Traits for Persona »Lazy Expert«.....   | 293 |
| Figure 78: Character Traits for Persona »Technician«.....  | 294 |
| Figure 79: Example PAP Using View Module »Swing«, Policy<br>Vocabulary »SECCRIT« and Presentation Module<br>»Template Instantiations« .....                            | 298 |
| Figure 80: ILP in IND <sup>2</sup> UCE Policy Language Version 1.1 Generated<br>by PAP in UI Framework »Swing« .....   | 298 |
| Figure 81: Example PAP Using View Module »Swing«, Policy<br>Vocabulary »SECCRIT« and Presentation Module<br>»Default Policies«.....                                    | 299 |
| Figure 82: Example PAP Using View Module »Android«, Policy<br>Vocabulary »SECCRIT« and Presentation Module<br>»Template Instantiations« .....                          | 299 |
| Figure 83: ILP in IND <sup>2</sup> UCE Policy Language Version 1.1 Generated<br>by PAP in UI Framework »Android« .....   | 300 |
| Figure 84: Example PAP Using View Module »Android«, Policy<br>Vocabulary »SECCRIT« and Presentation Module<br>»Default Policies«.....                                  | 300 |
| Figure 85: Example PAP Using a Preliminary Version of the View<br>Module »Web«, the Policy Vocabulary »SECCRIT« and<br>the Presentation Module »Default Policies«..... | 301 |
| Figure 86: Example PAP Using View Module »Web«, Policy<br>Vocabulary »Digital Villages« and Presentation Module<br>»Template Instantiation« .....                      | 309 |
| Figure 87: Example PAP Using View Module »Web«, Policy<br>Vocabulary »Digital Villages« and Presentation Module<br>»Default Policies«.....                             | 309 |
| Figure 88: Example PAP Using View Module »Web«, Policy<br>Vocabulary »Digital Villages« and Presentation<br>Module »Wizard« .....                                      | 310 |
| Figure 89: Example PAP Using View Module »Web«, Policy<br>Vocabulary »Digital Villages« and Presentation Module<br>»Security Levels«.....                              | 310 |
| Figure 90: Policy Specification Experiment - Handout Page 1 .....  | 312 |
| Figure 91: Policy Specification Experiment - Handout Page 2 .....  | 313 |
| Figure 92: Screenshot - Language Selection.....  | 314 |
| Figure 93: Screenshot - Login Page.....  | 314 |
| Figure 94: Screenshot - Demographic Questions.....   | 315 |



|   |     |
|---|-----|
| Figure 95: Screenshot - Relation to Fraunhofer IESE .....   | 315 |
| Figure 96: Screenshot - Relation to Fraunhofer IESE .....   | 316 |
| Figure 97: Screenshot - Motivation Question .....   | 316 |
| Figure 98: Screenshot - Persona Fundamentalist .....  | 317 |
| Figure 99: Screenshot - Persona Amateur .....   | 317 |
| Figure 100: Screenshot - Persona Marginally Concerned.....  | 318 |
| Figure 101: Screenshot - Persona Lazy Expert .....  | 318 |
| Figure 102: Screenshot - Persona Technician .....   | 319 |
| Figure 103: Screenshot - Persona Confirmation .....   | 319 |
| Figure 104: Screenshot - Scenario.....  | 320 |
| Figure 105: Screenshot - Specification Explanation .....  | 320 |
| Figure 106: Screenshot - Specification Type: Template 1.....  | 321 |
| Figure 107: Screenshot - Specification Type: Template 2.....  | 321 |
| Figure 108: Screenshot - Specification Type: Template 3.....  | 322 |
| Figure 109: Screenshot - Specification Type: Template 4.....  | 322 |
| Figure 110: Screenshot - Specification Type: Template 5.....  | 323 |
| Figure 111: Screenshot - Specification Type: Template 6.....  | 323 |
| Figure 112: Screenshot - Specification Type: Template Confirmation ..   | 324 |
| Figure 113: Screenshot - Specification Type Rating .....  | 324 |
| Figure 114: Screenshot - Specification Type: Default Policies 1.....  | 325 |
| Figure 115: Screenshot - Specification Type: Default Policies 2.....  | 325 |
| Figure 116: Screenshot - Specification Type: Default Policies 3.....  | 326 |
| Figure 117: Screenshot - Specification Type: Default Policies 4.....  | 326 |
| Figure 118: Screenshot - Specification Type: Default Policies 5.....  | 327 |
| Figure 119: Screenshot - Specification Type: Default Policies 6.....  | 327 |
| Figure 120: Screenshot - Specification Type: Wizard 1 .....   | 328 |
| Figure 121: Screenshot - Specification Type: Wizard 2 .....   | 328 |
| Figure 122: Screenshot - Specification Type: Wizard 3 .....   | 329 |
| Figure 123: Screenshot - Specification Type: Wizard 4 .....   | 329 |
| Figure 124: Screenshot - Specification Type: Wizard 5 .....   | 330 |
| Figure 125: Screenshot - Specification Type: Wizard 6 .....   | 330 |
| Figure 126: Screenshot - Specification Type: Wizard 7 .....   | 331 |
| Figure 127: Screenshot - Specification Type: Wizard 8 .....   | 331 |
| Figure 128: Screenshot - Specification Type: Privacy Levels .....   | 332 |
| Figure 129: Screenshot - Specification Type Preference Ordering .....   | 333 |
| Figure 130: Screenshot - Identification with Scenario and Persona.....  | 333 |
| Figure 131: Screenshot - Final Page and Scores .....  | 334 |
| Figure 132: Kruskal-Wallis-Test on Influence of Specification<br>Paradigms on Conducted Mistakes with Pairwise<br>Comparison of Specification Paradigms (Q1.1.1)..... | 337 |



|   |     |
|---|-----|
| Figure 133: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Marginally Concerned (Q1.1.2) .....   | 337 |
| Figure 134: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Amateurs with Pairwise Comparison of Specification Paradigms (Q1.1.2) .....                     | 338 |
| Figure 135: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Lazy Experts with Pairwise Comparison of Specification Paradigms (Q1.1.2) .....                 | 338 |
| Figure 136: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Technician with Pairwise Comparison of Specification Paradigms (Q1.1.2) .....                   | 339 |
| Figure 137: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Fundamentalists with Pairwise Comparison of Specification Paradigms (Q1.1.2) .....              | 339 |
| Figure 138: Kruskal-Wallis-Test on Influence of Persona Selection on Conducted Mistakes with Pairwise Comparison of Specification Paradigms (Q1.1.3) .....  | 340 |
| Figure 139: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness (Q1.2.1) .....                          | 340 |
| Figure 140: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Marginally Concerned (Q1.2.2) ..... | 341 |
| Figure 141: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Amateurs (Q1.2.2)....               | 341 |
| Figure 142: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Lazy Experts (Q1.2.2) .....         | 342 |
| Figure 143: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Technicians (Q1.2.2) .....          | 342 |
| Figure 144: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Fundamentalists (Q1.2.2) .....      | 343 |

|   |     |
|---|-----|
| Figure 145: Cross Tables including Fisher's Exact-Test on Influence of Persona on Correct Self-Evaluation regarding Objective Correctness (Q1.2.3) .....                    | 343 |
| Figure 146: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time with Pairwise Comparison of Specification Paradigms (Q1.3.1) .....                   | 344 |
| Figure 147: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Marginally Concerned (Q1.3.2) .....  | 344 |
| Figure 148: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Amateurs with Pairwise Comparison of Specification Paradigms (Q1.3.2) .....      | 345 |
| Figure 149: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Lazy Experts with Pairwise Comparison of Specification Paradigms (Q1.3.2) .....  | 345 |
| Figure 150: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Technicians with Pairwise Comparison of Specification Paradigms (Q1.3.2) .....   | 346 |
| Figure 151: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Fundamentalists (Q1.3.2) .....   | 346 |
| Figure 152: Kruskal-Wallis-Test on Influence of Personas on Needed Time (Q1.3.3) .....  | 347 |
| Figure 153: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction with Pairwise Comparison of Specification Paradigms (Q1.4.1) .....                  | 347 |
| Figure 154: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Marginally Concerned (Q1.4.2) .....   | 348 |
| Figure 155: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Amateurs with Pairwise Comparison of Specification Paradigms (Q1.4.2) .....     | 348 |
| Figure 156: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Lazy Experts with Pairwise Comparison of Specification Paradigms (Q1.4.2) ..... | 349 |
| Figure 157: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Technicians (Q1.4.2) .....  | 349 |
| Figure 158: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Fundamentalists (Q1.4.2) ...  | 350 |
| Figure 159: Kruskal-Wallis-Test on Influence of Personas on Satisfaction (Q1.4.3) .....   | 350 |



# List of Tables

|           |  |     |
|-----------|--|-----|
| Table 1:  | List of PAPs from Academia and Practice and Their Used Specification Paradigms .....             | 53  |
| Table 2:  | Tabular Documentation of Assets .....  | 76  |
| Table 3:  | Tabular Documentation of Threats .....   | 76  |
| Table 4:  | Tabular Documentation of Countermeasures .....   | 76  |
| Table 5:  | Exemplary Documented Asset .....   | 77  |
| Table 6:  | Exemplary Documented Threat .....  | 77  |
| Table 7:  | Exemplary Documented Countermeasures for a Threat .....  | 77  |
| Table 8:  | Tabular Notation of a Policy Template .....  | 79  |
| Table 9:  | Exemplary Policy Template .....  | 81  |
| Table 10: | Exemplary Policy Template »Access to Financial Data in Different Situations« .....               | 100 |
| Table 11: | Examples for Mapping of View Elements with UI Framework Components .....                         | 136 |
| Table 12: | Barrier Categories as Discrepancies between User Requirements and User Resources .....           | 147 |
| Table 13: | Potential Barriers for Users of the Twitter PAP .....  | 148 |
| Table 14: | Required user resources of the selected specification paradigms .....                            | 149 |
| Table 15: | Documented Asset »Financial Data of Client« .....  | 168 |
| Table 16: | Documented Threat »Data Theft of Financial Data for Creation of Tax CD« .....                    | 169 |
| Table 17: | Policy Template »Mass Retrieval of Data« .....   | 170 |
| Table 18: | Asset »Communication Data« .....   | 190 |
| Table 19: | Threats for Asset »Communication Data« .....   | 191 |
| Table 20: | Countermeasures for Threat »T4: Unintentional Sending of Hidden, Sensitive Information« .....    | 191 |
| Table 21: | Policy Template »Secure Email Sending« .....   | 192 |
| Table 22: | Lists of Elicited Use Cases, Assets and User Roles .....   | 203 |
| Table 23: | Mapping of Use Cases (X-Axis), Assets (Y-Axis) and User Roles (Numbers in Cells) .....           | 204 |
| Table 24: | 3-6-5 Sheet for Threat Elicitation of Use Case »Exchanging« (Dmg: Damage; Pb: Probability) ..... | 205 |
| Table 25: | Identified Countermeasures for Use Case »Exchanging« .....                                       | 205 |
| Table 26: | Derived Example Policies for Use Case »Exchanging« .....   | 206 |
| Table 27: | Exemplary Policy Template »DorfFunk: Help Requests and Offers« .....                             | 207 |

|           |   |     |
|-----------|---|-----|
| Table 28: | Personas Chosen by Participants of the Experiment.....  | 219 |
| Table 29: | Mistakes per Paradigm .....   | 221 |
| Table 30: | Participants with 100 Percent Objective Correctness .....   | 222 |
| Table 31: | Participants per Personas Making Zero Mistakes per<br>Paradigm .....                                | 222 |
| Table 32: | Ratio of Mistakes Made by Personas per Paradigm to All<br>Decisions.....                            | 223 |
| Table 33: | Perceived Correctness per Specification Paradigm .....  | 225 |
| Table 34: | Accuracy of Perceived Correctness (Correct Positive (P)<br>and Negative (N) Self-Evaluations) ..... | 226 |
| Table 35: | Mean Time in Minutes of Specification with Different<br>Specification Paradigms.....                | 228 |
| Table 36: | Satisfaction with Specification Paradigms for Personas<br>(SD: Standard Deviation) .....            | 230 |
| Table 37: | Selection of Elicitation Techniques.....  | 275 |
| Table 38: | Goal Description Template .....   | 276 |
| Table 39: | Stakeholder Description Template .....  | 277 |
| Table 40: | Documented Asset »Critical Service« .....   | 295 |
| Table 41: | Policy Template »Critical VM Migration« .....   | 296 |
| Table 42: | Asset »Job Data« .....  | 305 |
| Table 43: | Threats for Asset »Job Data« .....  | 305 |
| Table 44: | Asset »Public Data« .....   | 306 |
| Table 45: | Threats for Asset »Public Data« .....   | 306 |
| Table 46: | Countermeasures for Threat »T5: Intentional<br>Tampering« .....                                     | 307 |
| Table 47: | Countermeasures for Threat »T6: Unencrypted Sending<br>of Confidential Emails« .....                | 307 |
| Table 48: | Policy Template »Secure Email Receiving« .....  | 307 |

## List of Abbreviations

|       |   |
|-------|---|
| C     | Contribution                              |
| CS    | Case Study                                |
| E     | Experiment                                |
| H     | Hypothesis                                |
| ID    | Identifier                                |
| ILP   | Implementation-level Policy               |
| ILPT  | Implementation-level Policy Template      |
| IT    | Information Technology                    |
| M     | Metric                                    |
| PAP   | Policy Administration Point               |
| PDS   | Problem Derivation Survey                 |
| PP    | Practical Problem                         |
| Q     | Question                                  |
| RE    | Requirements Engineering                  |
| RQ    | Research Question                         |
| SLP   | Specification-level Policy                |
| SLPT  | Specification-level Policy Template       |
| UI    | User Interface                            |
| UML   | Unified Modeling Language                 |
| VM    | Virtual Machine                           |
| XACML | eXtensible Access Control Markup Language |



# 1 Introduction

We start this thesis with a short motivation of the topic. Next in this chapter, we present three surveys that underpin the practical relevance of this work followed by a detailed description and refinement of the addressed practical and scientific problems. We formulate hypotheses to measure the benefits of our approach and present our concrete contribution. The chapter ends with a description of our research approach, assumptions and limitations as well as an outline of the research work carried out in support of this thesis.

## 1.1 Motivation

Since the beginning of the Internet age, users have been increasingly sharing personal and sensitive data with online services and other users. In 2018, every minute users worldwide conducted 3,877,140 searches on Google, shared 2,083,333 snaps on Snapchat, posted 49,380 photos on Instagram and 6,940 users got matched on Tinder in the same period [1]. These numbers are increasing year after year. The companies behind those online services collect, store, analyze, reuse and partially resell these data. As most of those services are free for users, the companies build their main business model on data analytics, data reselling or personalized advertisement.

The prevalent data-centric business models make it increasingly complicated for users to understand and control the use of personal data by third parties. Therefore, users become increasingly afraid of data misuse, and their need for a better protection of their privacy raises.

In Germany, only a minority of four percent think that they still have complete control over the information they provide online and 45 percent feel they lost control. Sixty-eight percent of the latter are concerned about not having the complete control over their personal data [2]. A majority of users are uncomfortable with the way in which Internet companies use their data for their business [2, 3]. »There is now a level of uncertainty regarding data. People are beginning to express their mistrust in businesses, particularly in the technology space«, says, for example, Siân John from Symantec [3]. Especially in social networks, the majority of users also want to restrict the usage of their data, for example, the audience of shared data [2]. According to the state of the art, for example, Cranor and Buchler [4] demand user decision making when it comes to the configuration of security features.



Thus, users want and should gain more self-determination in the form of self-responsible definition and control of their security and privacy demands for personal data shared with online services. Therefore, they want to specify security and privacy policies that are then enforced by the IT system.

**Definition: Policy**

A policy is a set of requirements and/or their implementation.

**Definition: Specification-level Policy**

A specification-level policy (SLP) is a policy that has been specified by a human.

**Definition: Implementation-level Policy**

An implementation-level policy (ILP) is a policy that can be directly used to implement requirements. The implementation can be done technically (e.g., by a software system) or organizationally.

**Definition: Security Policy**

In the context of this thesis, we define a security policy to be a policy that targets at the protection of data and systems.

**Definition: Privacy Policy**

In the context of this thesis, we define a privacy policy to be a policy that targets at the protection person rights and personal data.

When we talk about policies in the context of this thesis, we refer to security and privacy policies. Many online services recognize the user concerns and provide tools to users for configuring measures for their online accounts (security policies) and for controlling the use of their uploaded data (privacy policies). These tools are called »policy editors«, »security and privacy settings« or »Policy Administration Points«. We use the latter academic term for such tools in the context of this thesis.

**Definition: Policy Administration Point (PAP)**

A policy administration point is a tool with which users can specify their requirements in order to (manually or automatically) produce one or more ILPs<sup>1</sup>.

---

<sup>1</sup> This definition extends the one from the XACML standard, which defines a PAP as »the system entity that creates a policy or policy set« [5]. In the context of this work, we explicitly define that a user is creating a policy with a PAP.

A user specifies his security and privacy demands in a PAP. As we defined, those are specification-level policies. SLPs specified by non-expert users are oftentimes natural-language representations of their own demands and describe which assets need to be protected. These SLPs typically lack details about the implementation of the demands, as users oftentimes do not have the necessary background knowledge and skills. The PAP produces implementation-level policies as output that describe how the user's demands have to be enforced, for example, technically by a security system. We focus on PAPs for non-experts in the context of this work. However, there also exist PAPs for expert users that allow the specification of policies with concrete enforcement instructions. In this case, the specification-level policy (specified by the user) equals the implementation-level policy (to be enforced by the system).

Despite the users' concerns of having the possibility to use PAPs as described above, a study in the field of social network shows that 42 percent of the users have never tried to change their security and privacy settings [2]. When asked why they did not change their settings, users replied that they did not consider it necessary, or did not know how to do it. Thus, even if the need arises to specify security and privacy policies for Internet services, many users do not do it. As studies in the past revealed [6–8], one major reason is that users have usability problems when using PAPs. Therefore, usability issues need to be considered when developing a PAP for the specification of security and privacy policies [9–11]. We empirically substantiate the usability issues with PAPs in the next section.

## 1.2 Problem Derivation Surveys

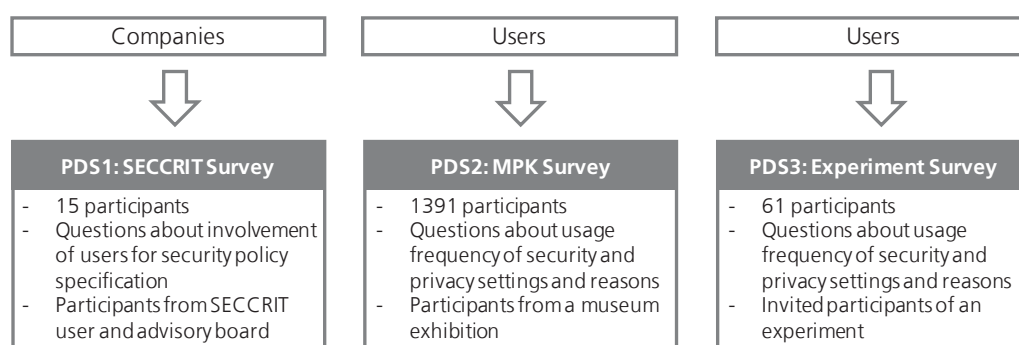


Figure 1: Overview of the Problem Derivation Surveys

Above, we claim that users struggle with the specification of their own security and privacy policies, because they face usability issues with existing PAPs. In addition, we assume that companies generally want to provide PAPs to users. Both assumptions motivate the entire work presented in this thesis. That is why we substantiate these assumptions with evidence before deriving concrete practical and scientific problems. Therefore, we present the results of three problem derivation surveys

(PDS1–3) in the following, which we conducted with companies acting as PAP providers and PAP users. We present an overview in Figure 1.

### **1.2.1 PDS1: »SECCRIT« Survey**

The goal of the first survey was the determination of the relevance of end user involvement in the policy specification process for industry. The survey was conducted in the context of the European research project »SECCRIT« (Secure Cloud Computing for Critical Infrastructure IT). The project aimed to improve IT security, trustworthiness, and assurance in the area of cloud-computing for critical infrastructure IT. One key contribution of the project were usability improvements for the specification of security policies for critical infrastructure IT. We present more information about this project in Section 8.3.1.

#### ***Setup and Execution***

We conducted this survey in 2014 in order to elicit new and confirm already known needs regarding cloud security. To this end, we designed an online questionnaire. In total, the survey consisted of 15 questions. The survey was rolled out to the user and advisory board of the »SECCRIT« project, whose members represented companies potentially eligible as PAP providers. At the time of our survey execution, 46 companies settled in the domain of critical infrastructure or cloud provisioning took part in the user and advisory board. In total, we sent the questionnaire to 60 persons from those companies. Participation was voluntary and the participants could skip questions if they felt unable or unwilling to answer them. We anonymized participants and aggregated all results so that individual results could not be attributed to the participant or its institution. The following two questions of the survey are relevant for this thesis:

- Do you think that end users should be enabled to specify their own security policies for protecting their data in cloud services?
- Do you think that usability issues are a major concern regarding end users specifying their own security policies?

A complete documentation of the survey can be found in [12].

#### ***Results***

Nineteen of the 60 invited persons started the survey and partially answered the questionnaire. Fifteen participants finished the questionnaire.

Most participants (12 out of 14; one participant skipped the first question) stated that users should be able to specify security policies on their own in order to protect their own data in cloud services (see Figure 2). However, five of them doubted that users are capable of specifying security policies. Two participants would rather deny end users the option to specify security policies as this would jeopardize security. Not a single participant stated that there is no need for end users to specify security policies on their own.

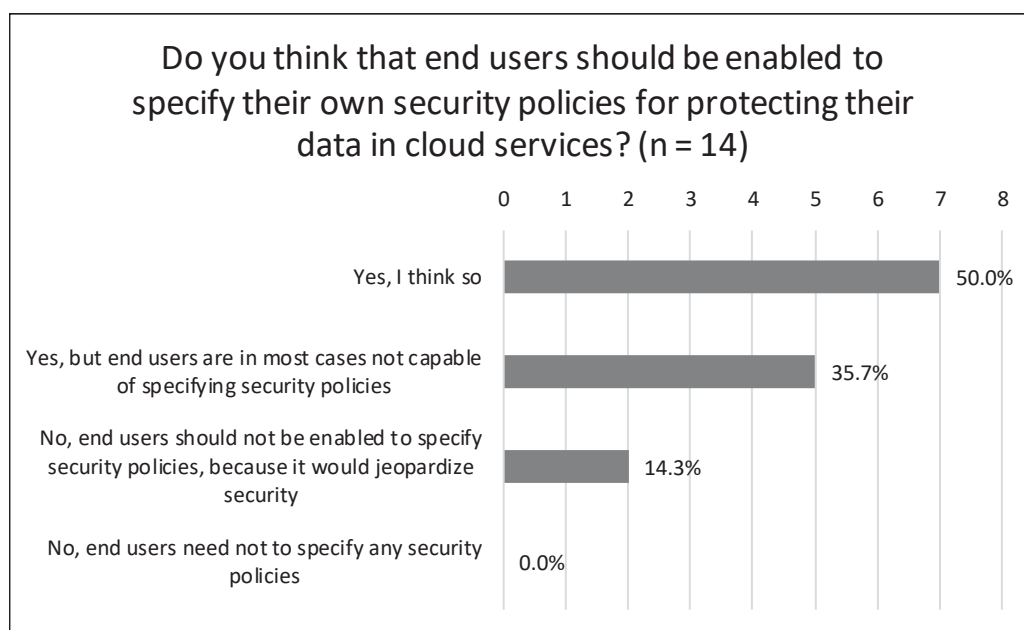


Figure 2:

SECCRIT Survey Question 1 – »Do You Think That End Users Should Be Enabled to Specify Their Own Security Policies for Protecting Their Data in Cloud Services?«

The opinions of the participants whether usability issues are a major concern for users with respect to security policy specification diverged as shown in Figure 3. Four participants voted for usability as a major concern. Eight participants were uncertain but two of them tended to »yes« and two of them to »no«. Three participants did not see usability as a major concern.

### **Summary and Conclusion**

This excerpt of the »SECCRIT« survey shows that industry predominantly supports the participation of users in the specification process of security policies. However, some participants feared security threats resulting from this participation. Therefore, when providing PAPs for users, a strong focus must be set on the objective correctness of the security policies specified by users. The participating companies seem to be undecided about whether usability issues are a major concern for users when specifying security policies.

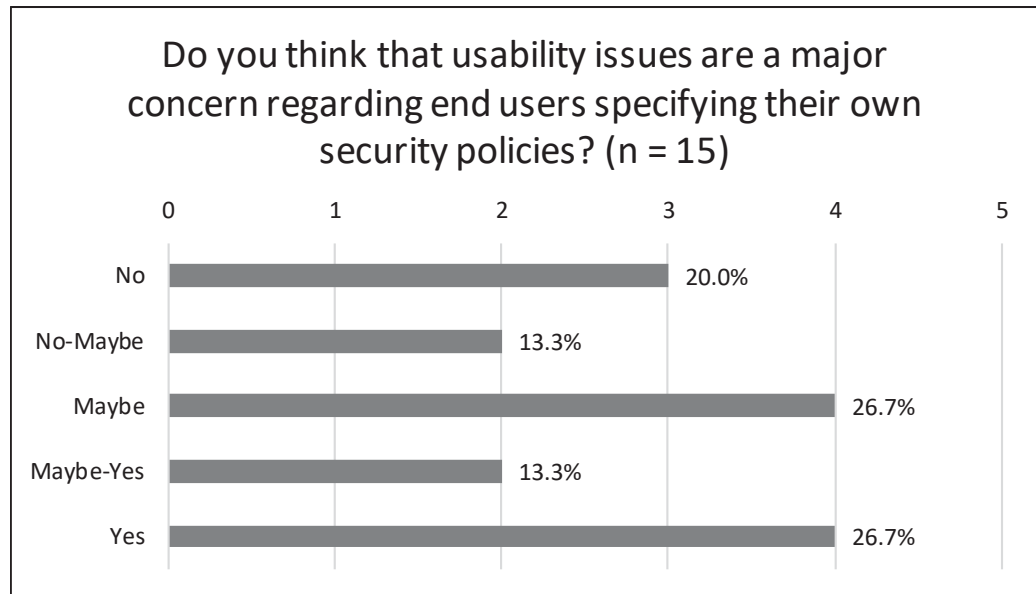


Figure 3: SECCRIT Survey Question 2 – »Do You Think That Usability Issues Are a Major Concern Regarding End Users Specifying Their Own Security Policies?«

### 1.2.2 PDS2: »Museum Pfalzgalerie Kaiserslautern« Survey

The goal of the second survey was to elicit the behavior of users regarding security and privacy policy specification in PAPs of online services. This includes their frequency of specifying security and privacy policies. In addition, the surveyed reasons why users do not perform this task more regularly. We conducted this survey as part of an exhibition from September 2018 until February 2019 in the museum »Pfalzgalerie« located in Kaiserslautern, Germany. The topic of the exhibition was »Without Key and Lock – Chances and Risks of Big Data«.

#### ***Setup and Execution***

The survey was part of an interactive security awareness quiz that we created as an exhibit. We installed instances of the quiz on eight touch screen monitors that stood next to each other. We offered the interactive exhibit in German language. It included nine questions. Five questions were quiz question challenging the security and privacy knowledge of the participants; the other four were survey questions to capture the spectrum of opinions from the entire population. We did not inform the participants about their participation in a survey. The mixture of quiz and survey has the disadvantage that we cannot rule out an influence of the quiz part on the survey results. On the other hand, a quiz has the advantage that it attracts the visitors' attention and motivates them to finish the survey.

We relate two of the survey questions directly to the work in this thesis. The first one is: »How often do you check your security and privacy settings?«. We provided multiple options for answering (see Figure 4) with

a single choice. During the design of the experiment, we defined that checking the security and privacy settings »multiple times per year« or »before every usage« are acceptable frequencies. This means that, in our opinion, less frequent checking of the settings poses a threat to security and privacy. In these cases, the frequency of checking should be increased. Therefore, we asked all participants who chose one of the other options »at most once a year«, »only directly after registration« or »never« for their reasons for only rarely checking security and privacy settings: »Why don't you use security and privacy settings more often?«. We gave multiple-choice answers (see Figure 5).

We provided only single and multiple choice answers, because the exhibit did not allow the input of text. Visitors did not need to identify themselves before starting the survey. Thus, it was anonymous. Hence, we did not prevent multiple participations. In addition, we did not supervise participants when filling in the survey. Thus, we do not know whether participants were influenced by others during the participation.

Due to the anonymous setting, we did not elicit demographic data from the participants. However, we assume that visitors of the museum exhibition represent a wide range of different people. Thus, we believe that this survey covered a cross section of society—i.e., it also included people without any special expertise in security and privacy. The cross section was important to us to get a holistic picture of users of PAPs of online services.

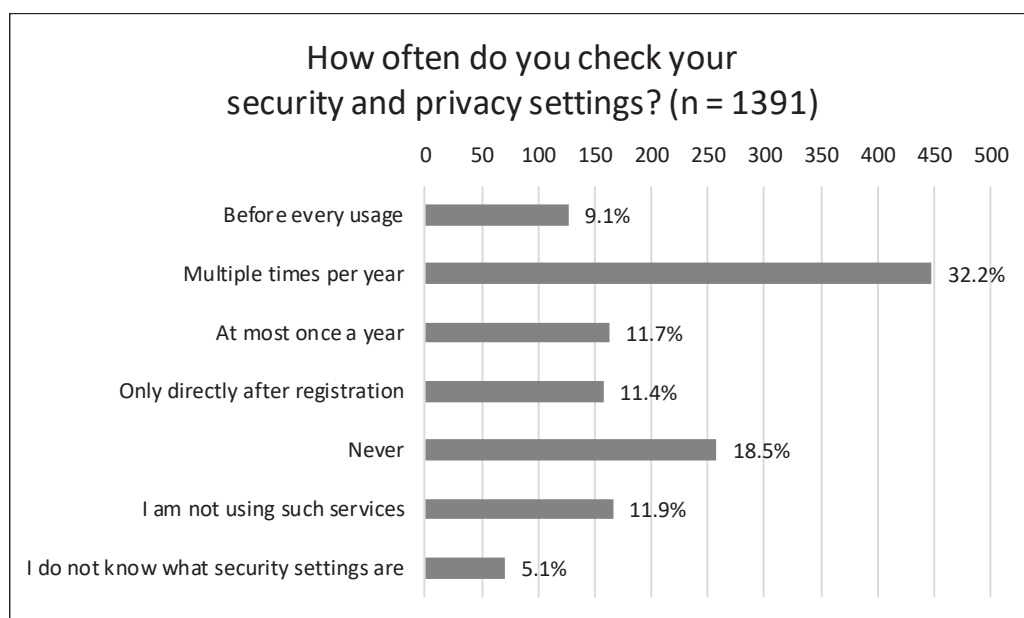


Figure 4:

MPK Survey Question 1 – »How Often Do You Check Your Security and Privacy Settings?«

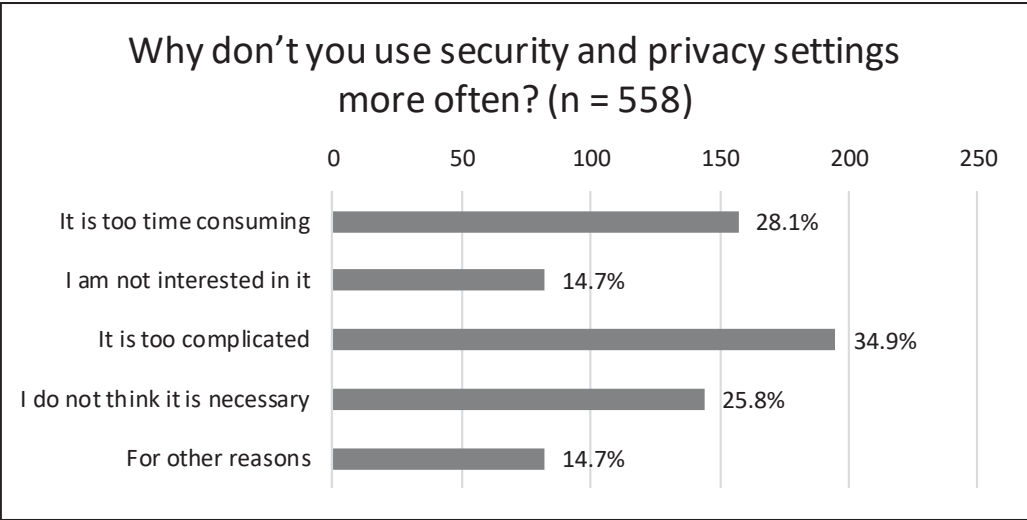


Figure 5: MPK Survey Question 2 – »Why Don't You Use Security and Privacy Settings More Often?«

**Results**

In total, 1,391 participants answered the first question: »How often do you check your security and privacy settings?« The results are presented in Figure 4. The figure shows that 9.1 percent of all participants claim to check their security and privacy settings of online services before each use. A significant share of 32.2 percent check them multiple times per year, 11.7 percent at most once a year and 11.4 percent only once directly after registration. Regarding the remaining answers, 18.5 percent of the 1,391 participants state that they never check their security and privacy settings, 11.9 percent said that they do not use such services and 5.1 percent answered that they do not know what security settings are.

All participants checking their security and privacy settings too infrequently were asked the second question: »Why don't you use security and privacy settings more often?« The survey tool informed the participants that they could select multiple answers. Figure 5 shows the results. Of the 558 participants, 34.9 percent stated that these PAPs are too complicated, and 28.1 percent answered that checking the security and privacy settings is too time consuming. Of all respondents, 25.8 percent said that they do not think that checking the security and privacy settings is necessary, and 14.7 percent were not interested in the settings. Moreover, 14.7 percent stated to have other reasons.

**Summary and Conclusion**

The answers to these two questions indicate that many users use PAPs for checking security and privacy settings, but most of them only sporadically. About 40 percent of the participants check security and privacy settings once a year or less. The main reason for such infrequent use is that they

perceive this task as unnecessary or as too complicated and time consuming. The latter two reasons indicate usability issues.

The too time-consuming use of PAPs indicates that users experience a lack of efficiency. Users also answered that they experience the task of checking security and privacy settings as too complicated. This indicates a limited satisfaction with the tools. In addition, it leads to the assumption that users fear incorrect specifications, which would result in a lack of effectiveness of the PAP causing incorrect specified security and privacy policies. In summary, the survey shows that users have problems using PAPs that we can explain with usability issues.

### **1.2.3 PDS3: Survey in the context of the policy specification experiment**

The goal of the third survey was to confirm the findings of the MPK survey described in the previous section. In particular, it aimed to elicit the frequency in which users use PAPs for checking their security and privacy settings and reasons why they do not perform this task more frequently. We conducted this survey as part of a larger experiment, which is part of the overall evaluation of this work (see Section 9.4). We sent the invitations to the online experiment on February 7, 2018 and accepted participations for 14 days. We collected the results on February 22, 2018.

#### ***Setup and Execution***

The survey was part of a security and policy specification experiment. We designed the experiment as an online experiment accessible via a browser. We invited all participants and prevented multiple participations by the same participant. The experiment was offered in German or English language.

We asked all participants two questions, similar to the ones of the MPK survey:

- »How often do you update the security and privacy settings of each web services on average?« (six single-choice answering options; see Figure 6)
- »What keeps you from updating your security and privacy settings more often?« (ten multiple-choice answering options; possibility to name other reasons; see Figure 7)

The first question was designed as single-choice, the second one as multiple-choice with the option to name additional reasons.



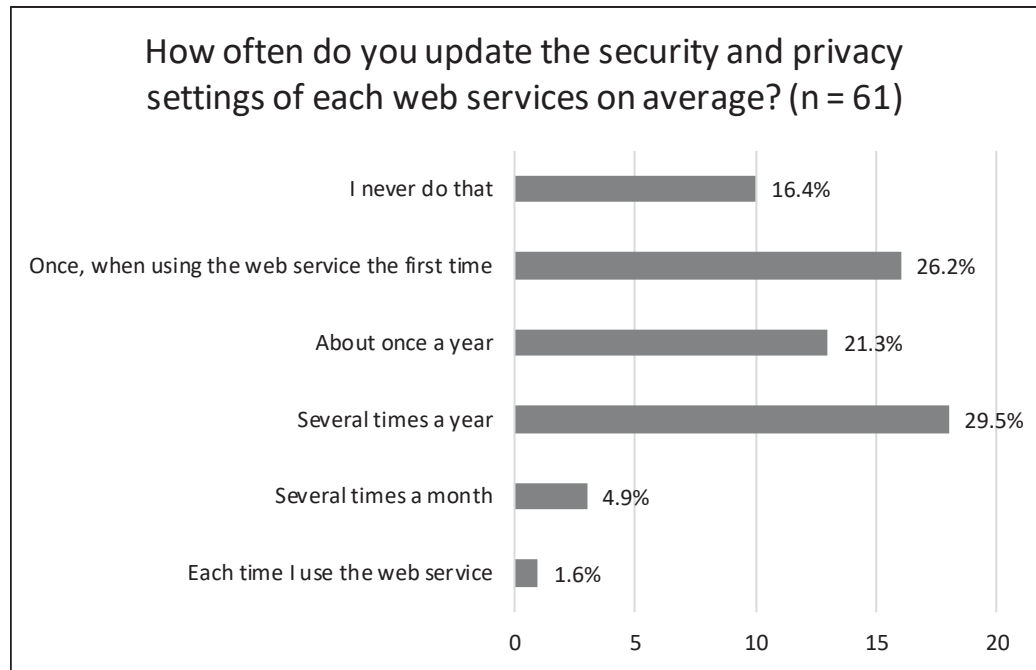


Figure 6: Experiment Survey Question 1 – »How Often Do You Update the Security and Privacy Settings of Each Web Service on Average?«

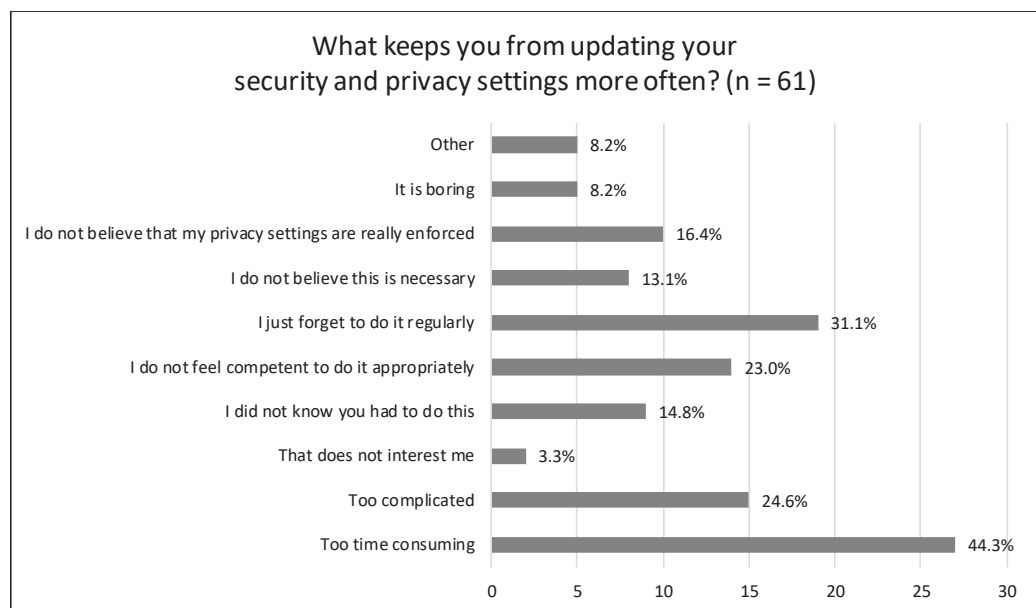


Figure 7: Experiment Survey Question 2 – »What Keeps You from Updating Your Security and Privacy Settings More Often?«

## Results

In total, 61 participants finished in the experiment and answered both questions. We present further information about participants' demographic data in Section 9.4.2. Of all participants, as we show in Figure 6, 16.4 percent never update their security and privacy settings of web services, and 26.2 percent do it only once when using the web service the first time. All other participants update the settings more frequently. Among them, 21.3 percent update the settings about once a year, 29.5

percent several times a year, 4.9 percent several times a month and 1.6 percent on each use of the web service.

All participants answered the second question. The top four reasons were named as »too time consuming« (44.3%), »I just forget to do it regularly« (31.1%), »too complicated« (24.6%) and »I do not feel competent to do it appropriately«. Further responses are shown in Figure 7.

### ***Summary and Conclusion***

The answers to those two questions show that more than 60 percent of all participants update security and privacy settings only once a year or less frequently. The main reasons are that users perceive the task of checking security and privacy settings in PAPs as too time-consuming or too complicated, that they do not feel competent enough to do it or that they just forget to do it. All these reasons indicate usability issues. There seems to be a lack of efficiency of these available tools. Users perceive the necessary time for using security and privacy settings as too high. In addition, users seem to fear the incorrect use of such tools and, thus, too low effectiveness, as they perceive the tools as too complicated and feel not competent enough to use them.

#### **1.2.4 Summary and Conclusion**

The three surveys revealed that industry recognizes the need to bring users into the loop of specifying policies with PAPs. However, a significant portion of users uses PAPs too infrequently. Participants stated that PAPs are too complicated and too time-consuming and users do not feel competent enough for using them. These and other reasons indicate usability problems with existing PAPs.

Usability can be defined as »the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use« [13]. If the use of a tool is too time-consuming, then the user does not accept the efficiency of the tool and its specification process. The unacceptability of the perceived complexity and the refusal of the use of the tool indicate that users are dissatisfied with the tool. Furthermore, the high complexity for the users and their fear not being competent enough may also negatively affect the effectiveness of the tool, which means that settings are potentially set incorrectly.

In summary, after analyzing the results of the surveys, we see the need to provide PAPs for specifying security and privacy policies to users, but current PAPs and their specification processes lack usability. We show an overview of the survey results in Figure 8.

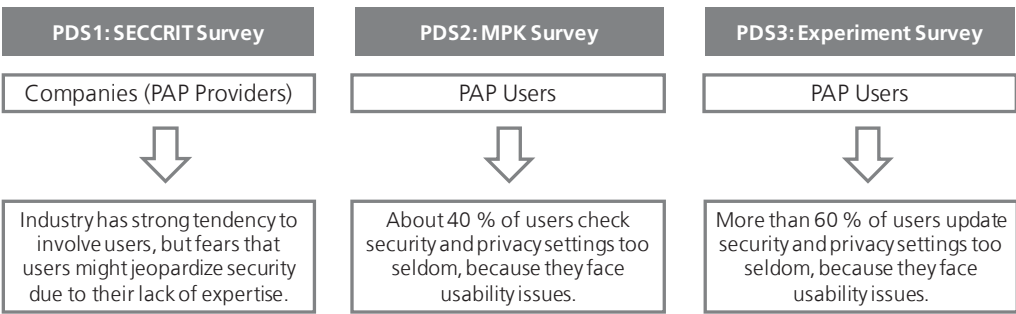


Figure 8: Overview of Survey Results

The results of the survey also indicate a lack of security and privacy awareness among users. However, we do not consider this issue in this thesis.

### 1.3 Problem Statement

In the context of this thesis, we use the ISO 9241 definition of usability [13]:

**Definition: Usability**

Usability is the »extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use« [13].

The usability issues identified in the surveys can be categorized in effectiveness, efficiency and satisfaction issues.

**Definition: Objective Correctness of Policies**

A policy is objectively correct if it expresses the security or privacy demand of the user who specified the policy.

**Definition: Perceived Correctness of Policies**

A policy is perceived correct by the user if the user is confident that his specified policy expresses his actual security or privacy demand.

The use of a PAP is effective for a user if he can specify objectively correct policies and if he can correctly self-evaluate the objective correctness of the specified policies (perceived correctness). The PAP is efficient for a user, if the time required for specifying a policy is acceptable to the user. The PAP is satisfying for a user if he enjoys the experience of using it. Several studies [2, 3, 11] and our problem evidence surveys (see Sections 1.2.2 and 1.2.3) show that users face usability issues when using state-of-the-art PAPs. For example, Zhao et. al [11] surveyed existing approaches

for privacy policy specification and found that »their tool developments have largely failed to deliver a user-friendly interface« [11]. Thus, we formulate the following first practical problem to be addressed in this thesis:

**Practical Problem 1 (PP1) – Usability Issues of Users**

Users have usability issues (too low effectiveness, efficiency and satisfaction) when specifying policies with existing Policy Administration Points.

In practice, users can configure their security and privacy policies (e.g., for online services) in several ways. The user interfaces for policy specification in PAPs differ, for example, in their specification process, their expressiveness and their guidance. We call these different types of policy specification interfaces and processes »specification paradigms«.

**Definition: Specification Paradigm**

A specification paradigm defines the specification process and the user interface in a PAP for the task of policy specification including the expressiveness and the guidance that the user receives during the specification.

Specification paradigms are generic concepts that describe the functionality of the user interface of a PAP. This includes the expressiveness, which defines the granularity on which the user can influence the policy specification. On the one hand, the user may freely assemble fine-grained policy elements to the demanded policy. On the other hand, the user may select from pre-defined policies or sets of policies, which are provided by security or privacy experts. Specification paradigms also describe the guidance, which defines, for example, the provision of additional information (e.g., explaining elements of policies or their effects) or the segmentation of the policy specification process in small and ordered steps. Specification paradigms do not define the content, thus, the policies that can be potentially specified with the PAP, and do not consider domain-specific concepts.

Most online services provide in their PAP exactly one specification paradigm for all users. However, users have different capabilities and motivations for using PAPs [14, 15]. Studies have shown that personalization to the user can increase the usability of, for example e-commerce websites [16] or security mechanisms [17]. One possible personalization of PAPs is the user-specific selection of a specification paradigm based on the user's capabilities and motivation. We assume that this user-specific selection has a positive effect on the usability of a PAP. We expect that users can achieve faster and more effective specifications

and have a more satisfying experience when using the most appropriate specification paradigm. We know from literature that the optimization of user interfaces of PAPs can improve their usability and positively influence the security and privacy specification experience for users [10, 18]. Literature also states, that security, privacy and usability need to be considered together [9]. However, there have not been any studies conducted so far that investigate usability effects of different specification paradigms of PAPs on users. We want to clarify this issue by answering the following research question in this thesis:

**Research Question 1 (RQ1) – Usability of Specification Paradigms**

Which specification paradigm fits best to the user of a PAP in order to support effective, efficient and satisfying policy specification?

As pointed out in the first problem derivation survey (see Section 1.2.1), industry is generally willing to provide PAPs to users. However, their development requires effort. We know from literature that companies are only willing to invest this effort if they have an incentive [19]. Especially, if companies wanted to provide usable PAPs with multiple supported specification paradigms to users, their development effort would be multiplied potentially. This leads to our second practical problem:

**Practical Problem 2 (PP2) – Specification Paradigms increase Effort**

Creation of usable PAPs with multiple supported specification paradigms increases development effort

One common approach for the reduction of development effort is automation, for example, by generating source code automatically. Thus, with respect to the development effort it would be beneficial to generate the user interfaces for policy specification according to the specification paradigms in PAPs. So far, no such method has been proposed in the literature. Thus, we identify the following scientific problem for this thesis:

**Scientific Problem (SP) – No Automation for PAP Creation**

No method for generating usable PAPs with multiple supported specification paradigms exists

Further open research questions exist that we want to answer for solving the scientific problem. First, the mapping of users to specification paradigms, as we addressed in RQ1, needs to be researched.

Furthermore, PAPs allow the users to specify security or privacy policies within a specific application domain.

**Definition: Application Domain**

We define an application domain as the target area for a PAP, in which the policies specified by the PAP are applied. An application domain contains stakeholders with security and privacy demands for data and systems that need to be protected with policies. An application domain can range from a single application or web service to an organization or industry branch.

Ideally, the specification boundaries of the PAP within which the user can express his security and privacy preferences should be aligned to the application domain. These specification boundaries differ for different specification paradigms as they provide different specification processes and expressiveness during the specification. Thus, when industry is providing a new PAP, the security and privacy demands that should be reflected in the PAP need to be elicited from the application domain (e.g., from the various stakeholders). However, there is no established method for eliciting the security and privacy demands and other relevant information from an application domain that are needed for providing PAPs with multiple supported specification paradigms. Thus, we derive the following research question:

**Research Question 2 (RQ2) – Elicitation**

How can we efficiently and effectively elicit all relevant information from an application domain that is necessary for providing usable PAPs with multiple supported specification paradigms?

As multiple examples in academia [20–22] show, the automated generation of application code can be supported by models. Thus, we need a model to describe security and privacy demands from an application domain in such a way that they can be used for the generation of policy specification interfaces for multiple specification paradigms in PAPs. According to our definition, a PAP must be usable by a human, for example by displaying the specification options in a human-understandable representation (SLP). In addition, a PAP must be capable of creating policies in a machine-understandable format as the output after specification (ILP). However, experts may understand the machine-understandable format and specify policies accordingly. We do not consider this in our work and focus on non-expert users, which may need a human-understandable policy format such as natural language. We require that our model must be capable of describing policies as both human-understandable and machine-understandable representations. Furthermore, rules for the transformation of human-understandable policy instances into machine-understandable equivalents must be definable. However, current research lacks methods for modelling security and privacy demands in such a way that usable policy specification

interfaces in PAPs with multiple supported specification paradigms can be generated from them. In addition, the modelling of security demands is requested in the literature as a major requirement for the development of high assurance systems [23]. This leads to the third research question:

### Research Question 3 (RQ3) – Formalization

How can we classify and represent security and privacy demands with a model so that they can be used for the generation of usable PAPs with multiple specification paradigms?

Finally, we want to use the elicited and modelled security and privacy demands to automate the creation of usable PAPs. However, we are lacking a technology for the generation of policy specification interfaces in PAPs that represent multiple specification paradigms. Code generation can replace manual development and reduce implementation effort, especially if PAPs with multiple supported specification paradigms need to be created. Thus, we can derive the fourth research question:

### Research Question 4 (RQ4) – Automation

How can we automate the creation of policy specification interfaces in PAPs that represent multiple specification paradigms?

We expect to solve the two practical problems if we can provide a method for generating usable PAPs with multiple supported specification paradigms for users. Therefore, we want to answer the four research questions. We summarize the practical problems, the scientific problem and the research questions addressed in this thesis in Figure 9.

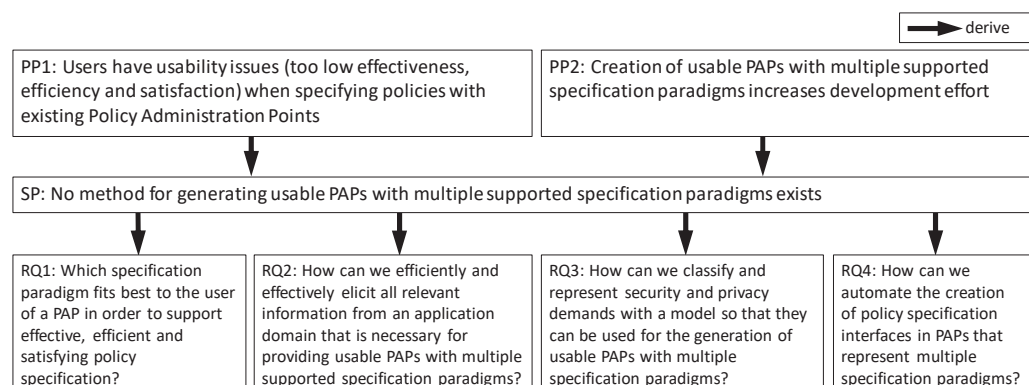


Figure 9: Relation of Practical Problems, Scientific Problem and Research Questions

## 1.4 Contribution

In order to enable flexible user-specific customization of policy specification tools for enhanced usability, we aim at providing an



automated creation process for domain-specific, multi-paradigm PAPs. The usability improvements of the PAP are addressed by the selection of an appropriate specification paradigm and the use of application domain-specific terminology. Our solution idea is the **method for usable PAP generation**. This approach contains four contributions:

- **Contribution 1 (C1) – User to Specification Paradigm Mapping:** A broad body of research exists on the usability improvement of security and privacy systems [6, 7, 10, 11, 18, 19, 24–41]. However, the literature is lacking studies on the effect of different specification paradigms on the usability of a PAP. One relevant problem is the identification of the adequate expressiveness of a PAP for a given type of user [25]. We address this gap by providing guidance for selecting the appropriate specification paradigms for (types of) users in terms of usability (effectiveness, efficiency and satisfaction) based on empirical data.
- **Contribution 2 (C2) – Policy Template Elicitation Method:** The state of the art regarding the elicitation of security and privacy requirements reveals a lot of methods and approaches for the elicitation of security and privacy related requirements [23, 42–50], policies [51–55] and risks [56–60]. We could not identify a systematic approach for eliciting policy templates directly from the stakeholders of an application domain in the literature. Thus, we devise a method for eliciting security and privacy demands from an application domain using state of the art RE techniques. The output of the method are policy templates aligned to the terminology of the users and the domain. Such a template can be instantiated as a concrete policy that reflects a security and privacy demand.
- **Contribution 3 (C3) – Policy Template Model:** We identified several very specific models in the state of the art that explain security and privacy principles and concepts [61–73] and revealed model-driven approaches for the refinement and generation of machine-understandable policies [74–76]. However, the literature lacks a generic model for modelling security and privacy demands in the form of policy templates that is capable of building the baseline for the automation of the PAP creation. We provide such a model, which contains all information necessary for the PAP generation framework to generate a PAP that supports multiple specification paradigms.
- **Contribution 4 (C4) – PAP Generation Framework (Concept and Implementation):** We contribute the PAP generation framework for the automation of the PAP creation. The framework is capable of generating user interfaces for policy specification, which can implement multiple specification paradigms, at runtime. Our framework includes generation algorithms for four state of the art specification paradigms. The specification options on the user



interfaces and the transformation rules for producing machine-understandable policies stem from an instance of the policy template model. The PAP can be configured at runtime to use this specific instance. We did not find a comparable approach in the literature for automating the creation of policy specification interfaces like our PAP generation framework.

- **Contribution 5 (C5) – Method for Usable PAP Generation:** The method for usable PAP generation combines the previous four contributions to a comprehensive approach for generating usable PAPs with multiple supported specification paradigms, as requested in the scientific problem. We could not find a comparable method in the literature.

We illustrate the relations of the contributions to the research questions in Figure 10.

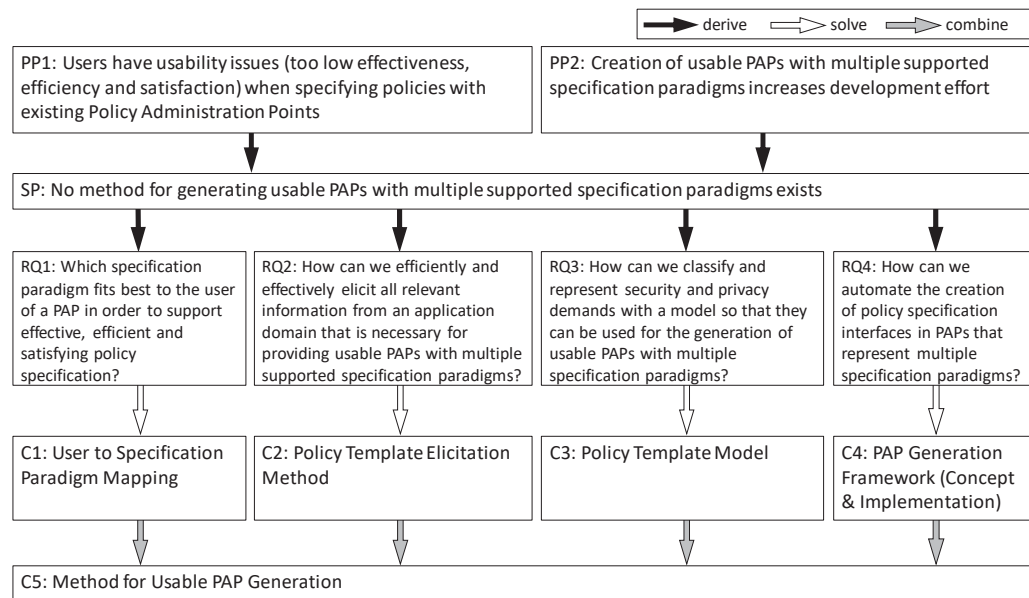


Figure 10: Relation between Practical Problems, Scientific Problem, Research Questions and Contributions

In addition to our methodical and technical contributions, we provide empirical contributions regarding the validation of our results comprising four case studies and one experiment:

- In two **case studies**, we applied our contributions in industrial settings. These case studies aimed at improving the completeness and correctness of the elicited information with the policy template elicitation method and the completeness of the policy template model. The case studies also explored the general feasibility of automating the PAP creation with the PAP generation framework. In the studies, we demonstrate the applicability of the method for usable PAP generation in two different application domains.

- In addition, we validated our contributions in our end-to-end **evaluation** of the overall method, which we split into two parts:
  - We applied the method for usable PAP generation in two more **case studies** to confirm our previous results.
  - We conducted a **policy specification experiment** in which we assessed the usability improvements of a multi-paradigm PAP in terms of increasing effectiveness, efficiency and satisfaction when providing the best of four specification paradigms to different user groups (personas).

In the next section, we formulate hypotheses and map them to the contributions and research questions for a systematic evaluation of our contributions.

## 1.5 Hypotheses

We presented our problem statement and contributions in the preceding sections. We formulated four research questions and proposed contributions for answering these questions. To justify our contributions in the evaluation, we present hypotheses for each research question. Our hypotheses concern four aspects:

- The usability of generated PAPs
- The correctness, completeness and user satisfaction with respect to the policy template elicitation method
- The completeness of the policy template model
- The feasibility of the automation of PAP creation.

### 1.5.1 Hypotheses for RQ1: Usability of Specification Paradigms

According to RQ1, we want to determine how to map suitable specification paradigms to users in order to increase the usability of PAPs.

#### **Hypothesis 1 (H1): Usability of PAP**

The usability of a PAP with the best matching specification paradigm is 30% higher than with the worst matching specification paradigm.

According to the ISO 9241 definition [13], the quality »usability« can be split into the sub-qualities effectiveness, efficiency and satisfaction. Thus, for measuring usability improvements, we subdivide H1 into different sub hypotheses and measure the effect of different specification paradigms on these sub-qualities independently. We expect that the effects of

different specification paradigms on the sub-qualities vary for different users. Therefore, we compare these effects for different user groups, which are represented by so-called personas. We present more details about personas and the chosen persona model in Section 6.3.

If we want to improve the specification of security and privacy policies, we need to consider two different types of effectiveness. On the one hand, users need to specify objectively correct policies with PAPs. The policy, that is the output of the PAP, must express the security or privacy demand of the user. We call this the objective effectiveness of the policy specification. On the other hand, the user must be confident that the resulting policy is objectively correct. We call this the perceived effectiveness of the policy specification. Therefore, we split the effectiveness in two hypotheses regarding objective and perceived effectiveness. The hypothesis regarding the objective effectiveness of different specification paradigms is:

**Hypothesis 1.1 (H1.1): Objective effectiveness of PAP**

H1.1<sub>A</sub>: On average, users make at least 30% fewer mistakes with a PAP when comparing the best matching specification paradigm to the worst matching specification paradigm.

H1.1<sub>O</sub>: Users cannot achieve 30% fewer mistakes with a PAP when comparing the best matching specification paradigm to the worst matching specification paradigm.

| GQM for H1.1           |   |
|------------------------|---|
| <b>Object</b>          | Analyze the <i>specification paradigms</i>  |
| <b>Purpose</b>         | for the purpose of <i>comparison</i>  |
| <b>Focus</b>           | with respect to <i>objective effectiveness</i>  |
| <b>Stakeholder</b>     | from the viewpoint of <i>users</i> and <i>personas</i>  |
| <b>Context Factors</b> | in the context of <i>an experiment</i> .  |
| <b>Question Q1.1.1</b> | Can the optimal mapping of specification paradigms of PAPs to users reduce the number of specification mistakes at least by 30%?  |
| <b>Metric M1.1.1</b>   | Ratio of percentage mistakes with best specification paradigm to percentage mistakes with worst specification paradigm.           |
| <b>Question Q1.1.2</b> | Is the optimal mapping of specification paradigms reducing the number of specification mistakes for each persona by at least 30%? |
| <b>Metric M1.1.2</b>   | Percentage of personas for which the optimal mapping reduces the specification mistakes by 30 percent.                            |

|                        |  |
|------------------------|--|
| <b>Question Q1.1.3</b> | Does the persona selection influence the objective effectiveness when using the different specification paradigms? |
| <b>Metric M1.1.3</b>   | Significance of influence of selected persona on made mistakes with specification paradigms.                       |

We formulate the following hypothesis with respect to the perceived effectiveness of the policy specification with different specification paradigms:

### **Hypothesis 1.2 (H1.2): Perceived effectiveness of PAP**

H1.2<sub>A</sub>: On average, the users' self-evaluation regarding objective policy correctness (perceived correctness) when specifying policies with a PAP has at least a 30% higher accuracy when comparing the best matching specification paradigm to the worst matching specification paradigm.

H1.2<sub>B</sub>: Users cannot achieve a 30% higher accuracy regarding the self-evaluation of objective policy correctness with a PAP when comparing the best matching specification paradigm to the worst matching specification paradigm.

| <b>GQM for H1.2</b>    |  |
|------------------------|--|
| <b>Object</b>          | Analyze the <i>specification paradigms</i>   |
| <b>Purpose</b>         | for the purpose of <i>comparison</i>   |
| <b>Focus</b>           | with respect to <i>perceived effectiveness</i>   |
| <b>Stakeholder</b>     | from the viewpoint of <i>users</i> and <i>personas</i>   |
| <b>Context Factors</b> | in the context of <i>an experiment</i> .   |
| <b>Question Q1.2.1</b> | Can the optimal mapping of specification paradigms of PAPs to users increase the accuracy of the correct self-evaluation regarding objectively correct specified policies by at least 30%? |
| <b>Metric M1.2.1</b>   | Ratio of correct positive estimations plus correct negative estimations to all estimations   |
| <b>Question Q1.2.2</b> | Does the optimal mapping of specification paradigms increase the accuracy of estimations regarding objectively correct specified policies for each persona by at least 30%?                |
| <b>Metric M1.2.2</b>   | Percentage of personas for which the optimal mapping increases the accuracy of estimations regarding objectively correct specified policies by at least 30 percent.                        |
| <b>Question Q1.2.3</b> | Does the persona selection influence the perceived correctness?  |

|                      |  |
|----------------------|--|
| <b>Metric M1.2.3</b> | Significance of influence of selected persona on perceived correctness with specification paradigms. |
|----------------------|--|

Additionally, we investigate the specification time that users need with different specification paradigms to specify policies, as seen in the following hypothesis:

### **Hypothesis 1.3 (H1.3): Efficiency of PAP**

H1.3<sub>A</sub>: On average, users are specifying policies at least 30% faster when specifying policies with a PAP comparing the best matching specification paradigm to the worst matching specification paradigm.

H1.3<sub>0</sub>: Users cannot achieve 30% faster specifications with a PAP when comparing the best matching specification paradigm to the worst matching specification paradigm.

| <b>GQM for H1.3</b>    |   |
|------------------------|---|
| <b>Object</b>          | Analyze the <i>specification paradigms</i>  |
| <b>Purpose</b>         | for the purpose of <i>comparison</i>  |
| <b>Focus</b>           | with respect to <i>efficiency</i>   |
| <b>Stakeholder</b>     | from the viewpoint of <i>users</i> and <i>personas</i>  |
| <b>Context Factors</b> | in the context of <i>an experiment</i> .  |
| <b>Question Q1.3.1</b> | Can the optimal mapping of specification paradigms of PAPs to users decrease the time needed to specify policies by at least 30%? |
| <b>Metric M1.3.1</b>   | Time necessary to specify policy.   |
| <b>Question Q1.3.2</b> | Is the optimal mapping of specification paradigms for decreasing the time needed to specify policies valid for all personas?      |
| <b>Metric M1.3.2</b>   | Percentage of personas for which the optimal mapping decreases the time needed to specify policies by at least 30 percent.        |
| <b>Question Q1.3.3</b> | Does the persona selection influence the time needed to specify policies?   |
| <b>Metric M1.3.3</b>   | Significance of influence of selected persona on the time needed to specify policies with specification paradigms.                |

The last considered sub-quality of usability is user satisfaction. We investigate the user satisfaction when using the different specification paradigms with the following hypothesis:

**Hypothesis 1.4 (H1.4): Satisfaction with PAP**

H1.4<sub>A</sub>: On average, the user satisfaction during a policy specification with a PAP when using the best matching specification paradigm is significantly higher than with the worst matching specification paradigm.

H1.4<sub>0</sub>: We cannot achieve a significantly higher user satisfaction with a PAP when comparing the best matching specification paradigm to the worst matching specification paradigm.

| <b>GQM for H1.4</b>    |   |
|------------------------|---|
| <b>Object</b>          | Analyze the <i>specification paradigms</i>  |
| <b>Purpose</b>         | for the purpose of <i>comparison</i>  |
| <b>Focus</b>           | with respect to <i>satisfaction</i>   |
| <b>Stakeholder</b>     | from the viewpoint of <i>users</i> and <i>personas</i>  |
| <b>Context Factors</b> | in the context of <i>an experiment</i> .  |
| <b>Question Q1.4.1</b> | Can the optimal mapping of specification paradigms of PAPs to users significantly increase the satisfaction experienced by users during the policy specification? |
| <b>Metric M1.4.1a</b>  | Significance for the increase of users' satisfaction of specification paradigms measured with a rating on a scale from 1 (unsatisfied) to 5 (satisfied).          |
| <b>Metric M1.4.2b</b>  | Ranking of specification paradigms  |
| <b>Question Q1.4.2</b> | Is the optimal mapping of specification paradigms for increasing the satisfaction experienced by users during the policy specification valid for all personas?    |
| <b>Metric M1.4.2</b>   | Percentage of personas for which the optimal mapping significantly increases the user satisfaction.   |
| <b>Question Q1.4.3</b> | Does the persona selection influence the satisfaction with specification paradigms?   |
| <b>Metric M1.4.3</b>   | Significance of influence of selected persona on the satisfaction with specification paradigms.   |

**1.5.2 Hypotheses for RQ2: Elicitation**

All relevant information from an application domain that is necessary for providing PAPs with multiple supported specification paradigms must be elicited. It is important that the information is complete so that users can specify all their security and privacy demands using the PAP. Thus, we formulate the following hypothesis regarding the completeness of the elicited policy templates with the policy template elicitation method:

**Hypothesis 2 (H2): Completeness of elicited information**

H2<sub>A</sub>: On average, the elicited policy templates cover at least 90% of the security and privacy demands from an application domain.

H2<sub>0</sub>: We cannot elicit policy templates that cover at least 90% of the security and privacy demands from an application domain.

| GQM for H2             |  |
|------------------------|--|
| <b>Object</b>          | Analyze the <i>policy template elicitation method</i>  |
| <b>Purpose</b>         | for the purpose of <i>evaluation</i>   |
| <b>Focus</b>           | with respect to <i>completeness</i>  |
| <b>Stakeholder</b>     | from the viewpoint of <i>expert</i>  |
| <b>Context Factors</b> | in the context of <i>four case studies</i> .   |
| <b>Question Q2.1</b>   | Is the policy template elicitation method capable of eliciting 90% of all necessary policy templates for the application domain? |
| <b>Metric M2.1</b>     | Ratio of elicited policy templates to all required policy templates in the application domain.                                   |

In addition, the content of the elicited policy templates must correctly represent the security and privacy demands from the application domain:

**Hypothesis 3 (H3): Correctness of elicited information**

H3<sub>A</sub>: On average, at least 90% of the elicited policy templates correctly represent the security and privacy demands from an application domain.

H3<sub>0</sub>: We cannot elicit policy templates from which at least 90% correctly represent the security and privacy demands from an application domain.

| GQM for H3             |  |
|------------------------|--|
| <b>Object</b>          | Analyze the <i>policy template elicitation method</i>  |
| <b>Purpose</b>         | for the purpose of <i>evaluation</i>   |
| <b>Focus</b>           | with respect to <i>correctness of elicited policy templates</i>  |
| <b>Stakeholder</b>     | from the viewpoint of <i>expert</i>  |
| <b>Context Factors</b> | in the context of <i>four case studies</i> .   |
| <b>Question Q3.1</b>   | Is the policy template elicitation method capable of eliciting correct policy templates that cover the security and privacy demands from the application domain? |
| <b>Metric M3.1</b>     | Ratio of correctly elicited policy templates to all elicited policy templates.   |

The participants of elicitation workshops in which the policy template elicitation method is used shall have a positive experience:

**Hypothesis 4 (H4): User acceptance of elicitation method**

H4<sub>A</sub>: At least 90% of the participants feel comfortable with the policy template elicitation method.

H4<sub>0</sub>: Less than 90% of the participants feel comfortable with the policy template elicitation method.

| GQM for H4             |   |
|------------------------|---|
| <b>Object</b>          | Analyze the <i>policy template elicitation method</i>   |
| <b>Purpose</b>         | for the purpose of <i>evaluation</i>  |
| <b>Focus</b>           | with respect to <i>user acceptance</i>  |
| <b>Stakeholder</b>     | from the viewpoint of <i>users</i>  |
| <b>Context Factors</b> | in the context of <i>four case studies</i> .  |
| <b>Question Q4.1</b>   | Do users rate a workshop in which the policy template elicitation method is applied as a positive experience?   |
| <b>Metric M4.1</b>     | Ratio of users who rate the participation in a workshop in which the policy template elicitation method is applied as a positive experience to the total number of participants |

### 1.5.3 Hypotheses for RQ3: Formalization

The following hypothesis for RQ3 describes the desired capability of the policy template model to model security and privacy demands completely in the form of policy templates:

**Hypothesis 5 (H5): Completeness of policy template model**

H5<sub>A</sub>: On average, the policy template model can model at least 90% of the elicited security and privacy demands from an application domain in the form of policy templates.

H5<sub>0</sub>: We cannot model at least 90% of the elicited security and privacy demands from an application domain in form of policy templates using the policy template model.



| GQM for H5             |  |
|------------------------|--|
| <b>Object</b>          | Analyze the <i>policy template model</i>   |
| <b>Purpose</b>         | for the purpose of <i>characterization</i>   |
| <b>Focus</b>           | with respect to <i>completeness</i>  |
| <b>Stakeholder</b>     | from the viewpoint of <i>expert</i>  |
| <b>Context Factors</b> | in the context of <i>four case studies</i> .   |
| <b>Question Q5.1</b>   | Is the policy template model capable to represent more than 90 percent of the elicited security and privacy demands in the form of policy templates? |
| <b>Metric M5.1</b>     | Number of policy templates modeled in the policy template model divided by number of policy templates elicited in the case studies                   |

#### 1.5.4 Hypotheses for RQ4: Automation

Finally, the feasibility of automation in the PAP creation process based on an instance of the policy template model is to be investigated by the following hypothesis:

##### Hypothesis 6 (H6): Feasibility of automation of PAP creation

H6<sub>A</sub>: With respect to our case studies, 100 percent of the user interfaces for the policy specification that implement different specification paradigms can be generated in PAPs during runtime.

H6<sub>0</sub>: With respect to our case studies, less than 100 percent of the user interfaces for the policy specification that implement different specification paradigms can be generated in PAPs during runtime.

| GQM for H6             |   |
|------------------------|---|
| <b>Object</b>          | Analyze the <i>PAP generation framework</i>   |
| <b>Purpose</b>         | for the purpose of <i>evaluation</i>  |
| <b>Focus</b>           | with respect to <i>automation</i>   |
| <b>Stakeholder</b>     | from the viewpoint of <i>expert</i>   |
| <b>Context Factors</b> | in the context of <i>four case studies</i> .  |
| <b>Question Q6.1</b>   | Is the process of user interface creation for the task of policy specification automatable for multiple specification paradigms and UI frameworks?                        |
| <b>Metric M6.1</b>     | Ratio of supported specification paradigms to all tested specification paradigms.   |
| <b>Metric M6.2</b>     | Ratio of UI frameworks for which policy specification interfaces in PAPs can be generated to all tested UI frameworks based on different policy template model instances. |

## 1.6 Research Approach

In the following, we describe the scientific approach for realizing the five contributions C1 to C5 and for answering the four respective research questions RQ1 to RQ4.

In general, we chose an iterative exploration and improvement process. First, we identified existing PAPs in practice and academia to derive specification paradigms in common use. Next, we surveyed the state of the art for approaches regarding user type models, elicitation approaches for security and privacy requirements and policies, security and privacy models and usable security and privacy specification. Based on the gained insights, we devised the first versions of our contributions.

We applied each version of our aforementioned contributions in two case studies for eliciting their improvement potential. After each case study, we improved the contributions accordingly. After finalizing the contributions, we validated their quality with respect to the research questions and hypotheses of this thesis in two more case studies and one experiment. Figure 11 summarizes our empirical contributions.

| Evaluation for Improvement   | Evaluation for Validation   |
|--|---|
| <ul style="list-style-type: none"> <li>- »Sinnodium« case study</li> <li>- »SECCRIT« case study</li> </ul> | <ul style="list-style-type: none"> <li>- »BeSure« case study</li> <li>- »Digital Villages« case study</li> <li>- Policy specification experiment</li> </ul> |

Figure 11: The Empirical Contributions Mapped to the Evaluations for Improvement and Validation

We show the relation between the contributions, hypotheses, case studies and the experiment in Figure 12. Figure 13 summarizes the evaluation plan of our research approach with respect to the practical and scientific problems.

Last, we aimed to receive feedback for informal validation of our contributions from the academic communities. Therefore, we presented our contributions and the evaluation results at various workshops and conferences [77–84].

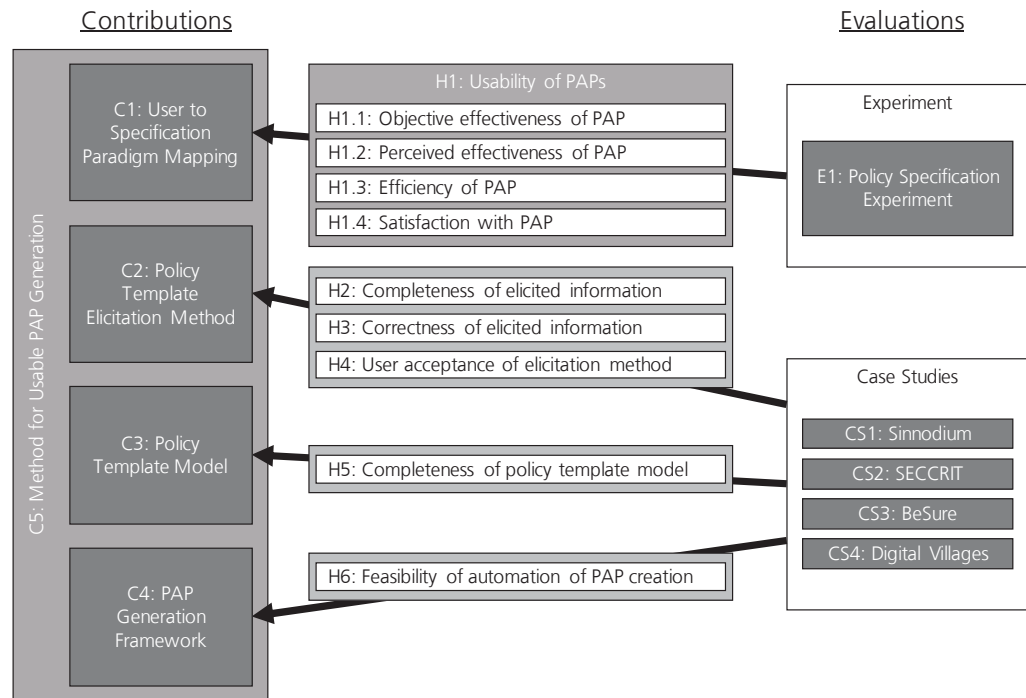


Figure 12: Relation between Contributions, Hypotheses and Case Studies and the Experiment

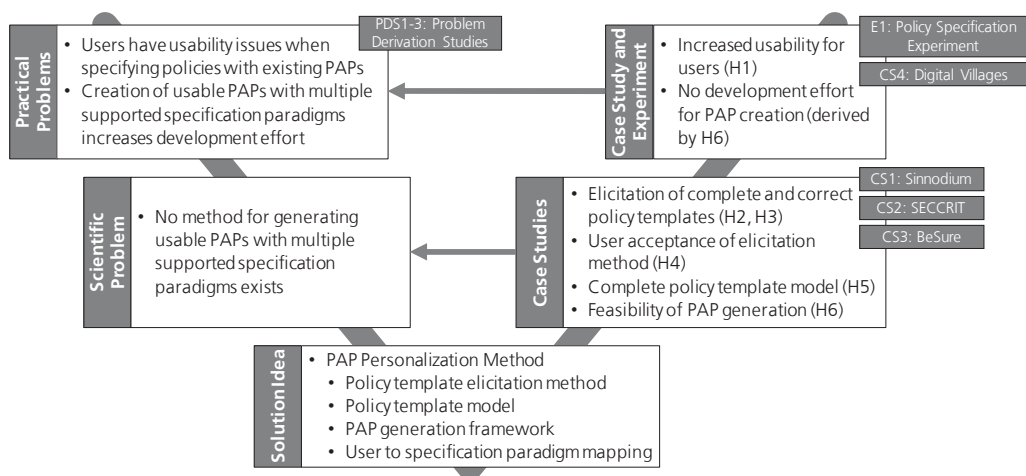


Figure 13: Relation between Practical and Scientific Problems and Case Studies, the Experiment and the Hypotheses

## 1.7 Assumptions and Limitations

The following assumptions and limitations apply to this thesis:

- In this thesis, we do not address missing security and privacy awareness that users may have according to our problem derivation studies.

- We do not propose a new method for transforming specification level natural language security policies into machine-understandable policies in this thesis. Appropriate approaches already exist (e.g., [76]). However, we provide the option for defining respective transformation rules in our policy template model.
- We do not address the topics of policy conflicts or policy negotiation in this thesis.
- Our method for usable PAP generation is lacking an evaluated approach for the specification of projection rules for multiple specification paradigms and transformation rules for generating machine-understandable policies from specified human-understandable policies. This is currently an expert-based task.
- We selected four specification paradigms from the state of the art and practice that strongly vary in the dimensions of expressiveness and guidance. Therefore, we assume to represent the spectrum of available specification paradigms well with our selection. However, we do not know whether other specification paradigms provide better results with respect to usability.
- The PAP generation framework is currently not supporting or using the domain sub-model or the security and privacy sub-model of the policy template model. Further research is required to integrate this information into the policy specification interfaces of PAPs.

## 1.8 Outline

This thesis is structured into ten chapters as follows:

In Chapter 2, we surveyed the state of the art with respect to elicitation approaches for security and privacy requirements, security and privacy models, usable security and privacy specification and models for user behavior. In addition, we present existing PAPs from practice and academia and derive specification paradigms.

In Chapter 3, we present our policy template elicitation method, which we use for retrieving policy templates from stakeholders of an application domain. After a method overview, we describe the five steps of the method in detail: information retrieval, workshop preparation, execution of elicitation workshop, documentation of workshop results as well as policy template derivation and validation.

In Chapter 4, we describe our policy template model including the various sub-models. These include the representation of specification-level policy templates, the definition of transformation rules for generating implementation-level policies and the specification of projection rules to allow the support for multiple specification paradigms.

In Chapter 5, we present the PAP generation framework for automating the creation of usable policy specification interfaces in PAPs that support multiple specification interfaces. We explain the architecture of the framework and explain how developers can integrate it in existing software. Next, we discuss the selection of the four supported specification paradigms and explain the generation algorithms for each paradigm. Finally, we present our reference implementation of the PAP generation framework.

In Chapter 6, we explain the theoretical work behind our user to specification paradigm mapping. We first present a user intention model that explains barriers that users may have when using a PAP. We argue that users differ in their capabilities. Therefore, we select a user type model for clustering users of PAPs into groups for better mapping individual users to specification paradigms.

In Chapter 7, we combine the previous four contributions to the method for usable PAP generation and explain their joint application.

In Chapter 8, we describe the two industrial case studies »SINNODIUM« and »SECCRIT« that we used for gaining improvement potential for the iterative enhancement of our contributions. For both case studies, we present a project summary, explain the design and execution, show the results, discuss our observations and lessons learned as well as explain how we addressed threats to validity.

In Chapter 9, we describe how we validated our contributions with two more case studies and one experiment. One of the case studies was conducted in an industrial setting. For each case study, we give a project summary, explain the design and execution, present the results, discuss our observations and lessons learned as well as explain how we addressed threats to validity. For the experiment, we first describe the setup and execution. Next, we explain how we analyzed the results, which are presented afterwards. In addition, we discuss the meaning of our results. Finally, we discuss threats to validity and our conclusions. For both the case studies and the experiment, we discuss how the results address our research goals by confirming or refusing our hypotheses.

In Chapter 10, we summarize the results of this thesis and our conclusions. Furthermore, we discuss open issues as well as potential future work.

## 2 Foundations and Related Work

In this chapter, we provide an overview of relevant related work with respect to the research questions investigated in this thesis. We discuss relevant approaches and identify gaps in the body of research to which we contribute with our work. In addition, we present the foundation for the selection of specification paradigms and the persona model.

We structure the remainder of this chapter as follows. We first explain our research approach in Section 2.1. The state of the art in the elicitation of security and privacy requirements is presented in Section 2.2 (related to RQ2), followed by the relevant work on existing security and privacy models in Section 2.3 (related to RQ3). In Section 2.4, we discuss the body of research into usable security and privacy policy specification (related to RQ1 and RQ4). In Section 2.5, we present existing PAPs from state of the art and practice, and we derive and compare specification paradigms (related to RQ4). Finally, we present related work in the field of user behavior in Section 2.6. Section 2.7 summarizes this chapter and our conclusions.

### 2.1 Research Approach

The foundations and related work presented in this chapter has been obtained through continuous and iterative literature research. We conducted individual literature surveys with respect to the four research questions. In addition, we have continuously expanded our knowledge of the state of research through supervised student research projects and diploma theses, advice from colleagues, conference visits and discussions with specialist colleagues.

### 2.2 Elicitation of Security and Privacy Requirements

We claim in RQ2 that we want to devise an approach for eliciting all relevant information from an application domain that is necessary for providing PAPs with multiple supported specification paradigms. Thus, we need to elicit security and privacy demands systematically from stakeholders of the application domain. This field is well researched, and a plethora of well-evaluated approaches already exists. Hence, we decided to reuse existing approaches in our policy template elicitation method. In the following, we describe the related work in the areas of »elicitation techniques« and »security and risk assessment«.

Several types of elicitation techniques have been proposed in research and industry that are relevant for our work. We first identify »general requirements elicitation techniques«. Next, we present more specific »elicitation techniques for security and privacy requirements«, which are already adapted to the area of security and privacy. Finally, we further narrow the focus down and discuss »elicitation techniques for policies«.

### ***General Requirements Elicitation Techniques***

In general, the entire spectrum of established requirements engineering techniques proposed in the literature can be applied to elicit assets, use cases, threats and countermeasures in our policy template elicitation method. Examples are brainstorming [85], domain analysis [86], interviews [87], questionnaires [88] and task analysis [89]. Pohl [90], Pohl and Rupp [91], Rupp [92], Zowghi and Coulin [86] as well as Zhang [93] survey existing requirement engineering techniques. In our method, we recommend the use of some of these techniques based on our experience. However, the suitability of a technique depends on various factors, such as goal, application domain, available time and group size of participants in a workshop. Thus, other techniques may fit better in other applications of our method. In Appendix A.1, we present a selection of requirements engineering techniques in more detail.

One important aspect for the elicitation of requirements is the selection of stakeholders. Alexander [94] proposes a stakeholder analysis approach. In Alexander's terminology, a stakeholder is an individual person or a legal entity. Stakeholders can be clustered; they represent one or more roles. He describes the onion model approach for relating roles to the target systems. TORE [95] and the approach by Mitchell et al. [96] classify and map stakeholders to specific characteristics. Cameron et al. [97] describe a method for analyzing stakeholders' needs and prioritize them regarding their value to others. Although their method was developed to identify stakeholders related to space exploration, it can be adapted and applied to different application domains such as security policies. Stakeholder identification and analysis methods are used in the second step of the policy template elicitation method (see Section 3.4).

### ***Elicitation Techniques for Security and Privacy Requirements***

In addition to the generic requirement elicitation techniques, several approaches specifically devised for the domain of security and privacy have been proposed in research.

Dörr [42, 44] proposes a method for eliciting non-functional requirements including security requirements. The elicitation is built on experience-based quality models that describe general characteristics of quality attributes, including metrics to measure the quality attributes and means



to achieve them. The quality models are tailored to the specific needs of each project during application. The method also offers guidance for the elicitation by providing checklists and prioritization questionnaires and for the documentation of the results.

Alexander [43] presents misuse case modelling to describe abnormal use of assets. He defines a misuse case to be »simply a use case from the point of view of an actor hostile to the system under design« [43]. The misuse cases are expected to reveal security and safety problems that could have caused systems failures or higher development effort.

Olzak [45] describes a high-level threat modeling approach for guided and practical conduction of threat analysis within a business environment. His approach comprises six steps: identification of critical assets, decomposition of the system, identification of possible points of attack, threat identification, categorization and prioritization of threats and threat mitigation. Threat identification is mainly based on the systematic analysis of software development artifacts such as UML diagrams. In contrast to this approach, we aim at eliciting security and privacy demands based on assets, threats and countermeasures that are elicited from stakeholders of the application domain. However, the steps of our method closely resemble the steps of Olzak's method.

The framework by Haley et al. [46] supports an asset-based elicitation and analysis of security requirements. The approach contains similar process steps as proposed by Olzak and by us in our policy template elicitation method: Assets are identified, and based on those, threats are derived. Finally, security requirements for preventing these threats are defined as security-related system constraints.

Van Lamsweerde et al. [23] published an extension for the KAOS approach that identifies threats by systematically analyzing anti-goals of a system. To this end, they first define process requirements for a security requirements engineering process to achieve results with high assurance. Next, they present a process for the elicitation, modeling and analysis of security requirements based on their requirements. Among the proposed requirements for security requirement elicitation, they demand the seamless integration of security requirements elicitation into the system engineering process and they require the introduction of formalization.

Deng et al. [47] developed the LINDDUN method as a »comprehensive framework to model privacy threats in software-based systems [47]«. The method supports the elicitation and fulfillment of privacy requirements and therefore clusters privacy threats into the following categories: linkability, identifiability, nonrepudiation, detectability, information disclosure, content unawareness and consent/policy noncompliance. The method guides experts in the identification of threats and their mapping



on system elements. To this end, discovered threats are mapped to data flow diagrams in order to identify misuse case scenarios and to elicit privacy requirements.

Mead et al. [48] propose Security Quality Requirements Engineering (SQUARE) as a systematic process for eliciting security requirements. The SQUARE process puts emphasis on a recommended sequence of process steps, but it does not propose any specific elicitation technique.

Fletcher and Liu [49] propose a structured object-oriented approach for security requirements analysis. They develop a context object diagram for cyber-physical systems (CPSs), which is a high-level representation of the CPS that shows interactions between the target system and external objects. Using this representation, use cases are specified to identify the main functionalities of the CPS. In order to identify potential threats to the CPS correctly, each use case is decomposed using so-called activity swim lanes to reveal the detailed activities performed by an actor to achieve that associated task.

Phan et al. [50] describe a security engineering process for web-service applications, which may also be applicable for more generic scenarios. The process is based on conventional software development processes. At the same time, it takes into account the additional aspects relating to security requirements. Their approach is called SOABSE (service-oriented architecture business security engineering).

### ***Elicitation Techniques for Policies***

Few approaches were proposed in research regarding the systematic elicitation of policies. Existing proposals address different levels of policies: specification-level policies, such as organizational company policies, and implementation-level policies, such as XACML policies.

As part of the SPARCLE workbench, Karat et al. [52] present an approach for eliciting concrete privacy demands from company representatives. They use questionnaires and semi-structured interview techniques for retrieving information from the stakeholders. However, their specific elicitation of privacy demands is not very structured, but based on open questions and on discussions about use cases. They also do not compare the applicability of different requirements engineering techniques to the elicitation process.

Callele and Wnuk [53] present techniques for crafting corporate policies based on traditional requirements engineering. They conclude that interviews, brainstorming and survey techniques have been effective in gathering the information needed to support the development of a corporate intellectual property (IP) policy. According to the authors, these

techniques can also be used for other types of policies. In order to draw this conclusion, a case study was carried out in a company in the information and communications technology sector, which confirmed their findings. Requirements engineering techniques were successfully applied to the task of IP policy generation, resulting in the creation of a corporate IP policy. This IP policy was positively validated by the senior management of the company.

Sainan and Yu [54] investigated how to integrate requirements for access control policies into the analysis phase of the system development process. They describe how functional requirements of the system are elicited. By modeling the functional requirements, they obtain the access control requirements, and they model security requirements by extending the UML notation.

Hibshi et al. [51] evaluate the effect of different security levels on users by letting them rate different instantiations of so-called security vignette templates. In a study, the researchers instantiated the templates for creating concrete security requirements in order to assess the user perception. Those vignette templates are a simplistic equivalent of our policy templates. In our terminology, they correspond to policy templates with only text and selectable text elements. In contrast to Hibshi et al., however, we aim to provide policy templates to users so that they can instantiate them as concrete security and privacy policies that reflect their own demands.

In an approach proposed by Oladimeji et al. [55], security requirements are analyzed based on UML diagrams. The authors relate security concerns to functional models and record their findings in a soft goal interdependency graph. Security policies are modeled using UML.

The security controls suggested in the NIST publications 800-30 [98], 800-37 [99], and 800-53 [100] can be seen as a kind of policy templates. However, these security controls are described in a rather generic way and need to be adapted to the application domain. This includes the adjustment of terminology. With the application domain-specific elicitation of policy templates in our approach, we directly handle the adaption during the elicitation process. Nevertheless, especially the NIST 800-53 catalog can provide valuable input for the elicitation process.

### ***Security and Risk Assessment***

The elicitation of security requirements is closely related to security and risk assessment because security needs and security risks are two complementary concepts. Accordingly, security and risk assessment methods can provide a suitable basis for policy elicitation.

Security-related checklists, guidelines, control question and catalogs help to gain a systematic overview of the relevant security issues of a given application domain. Examples are the NIST special publication 800-53 [100], the German »IT Grundschutz Kompendium« [101], the Common Criteria [102], or the ISO/IEC 27001 standard [103]. Most of the security requirements proposed in such standards are quite generic and not tailored to a specific application domain. However, they can be used as input during the elicitation of policy templates in our method. In addition to those generic standards, domain-specific equivalents exist, for example, for the domain of cloud computing [104].

Risk assessment approaches and their documentation can also provide valuable input for the elicitation of policy templates. Behnia et al. [56] and Busby et al. [57] give overviews of risk assessment approaches.

The OCTAVE Allegro method (Operationally Critical Threat, Asset and Vulnerability Evaluation) [58] can be used for identifying and managing information security risks. It focuses primarily on information assets as we also do in our approach. In OCTAVE Allegro, risk elicitation workshops are carried out. However, the method does aim to derive policy templates from the elicited information, and the use of different RE techniques is not considered.

CORAS [59, 60] is a model-based method for performing security risk analysis. Similar to OCTAVE, CORAS does not focus on eliciting policy templates, but on getting a complete list of risks and corresponding treatments to address them. In CORAS, an expert uses the CORAS tool for modelling security risks. There is no explicit elicitation workshop described in OCTAVE as in our approach.

### ***Summary***

In sum, the literature provides multiple approaches for the elicitation of security and privacy requirements. This includes approaches for the elicitation of security and privacy related requirements [23, 42–50], policies [51–55] and risks [56–60]. We could not identify a systematic approach for eliciting policy templates directly from the stakeholders of an application domain in the literature. Thus, we devise a method for eliciting security and privacy demands from an application domain using state of the art RE techniques.

## **2.3 Policy Models and Languages**

We claim in RQ3 that we want to devise a model for formalizing security and privacy demands. We surveyed the state of the art in order to identify

existing security and privacy related models that can contribute to this task.

In the past, numerous security models have been proposed that describe security requirements for systems, such as the models by Bell and LaPadula [61], Biba [62], Landwehr et al. [63] and Lampson [64]. These models typically describe exactly one security property, such as confidentiality in the Bell-LaPadula model.

Similarly, many access control models were proposed in the literature. Today, the most commonly used access control model is the role-based access control (RBAC), introduced by Ferraiolo et al [65]. The basic idea is to control the access to resources based on user roles instead of individual users, and to assign one or more roles to each user. The authors show that RBAC is a policy rich mechanism and that its configuration reflects organizational policies, which allows RBAC to be adaptable to any organizational structure and any way of conducting business.

Joshi et al. [66] describe a generalized temporal extension of role based access control, called generalized temporal role based access control (GTRBAC). Because of its generality, GTRBAC can be used for defining a diverse set of access control policies. Moreover, it simplifies authorization administration in large enterprises. The authors provide a framework that augments the GTRBAC model with XML to support policy enforcement in a heterogeneous, distributed environment. X-GTRBAC is a policy specification language that provides compact representation of access control policies for a generic computer-integrated enterprise, while allowing content-based and context-aware access control.

At the SACMAT conference in 2008, Alturi and Ferraiolo raised two questions concerning access control models:

- Is it possible for a unifying access control meta-model to be developed given the large diversity and types of existent access control policies?
- What practical good would such a meta-model serve?

The answers to these questions are relevant for us as we try to find a meta-model for security and privacy policies and templates in general based on previous research work.

Barker [105] tries to answer both questions. He addresses the fundamental concepts an access control meta-model has to have. More specifically, he describes access control in general in relation to the primitive notions of categories, relationships among categories and relationships between categories and principals. Classification types used in access control, such as classification by role, user attributes and

clearance, are particular instances of the more general class of categories. Principals include any elements that may access a resource in a system to which access must be controlled. Furthermore, Barker's model describes the semantics of the relevant relations. Moreover, Barker states that having a shared conception of an access control meta-model is important for reducing the burden on policy administrators when it comes to representing application-specific access control requirements. Identifying a common access control model is also desirable because it allows various general syntaxes to be developed in terms of the generic model. However, Barker's model is restricted to access control and does not fit our requirements for a policy template model.

Leitner et al. [67] focus on Process-Aware Information Systems (PAIS). They present a unified security policy data model based on responsibilities, permissions and constraints to cover structural as well as operational aspects of processes. They claim that security policies and processes must be designed separately from each other and that the relation between them should be expressed by an explicit mapping, which avoids side effects by changing either business processes or security policies. Moreover, the separation simplifies »consistency checks and enforcement of the security policies on the one side and evolution of processes and associated policies on the other« [67].

Choi et al. [68] address the security issues that cloud computing environments face. The classic access control models used nowadays (mainly RBAC) cannot provide dynamic access control, since they do not include context-aware elements. The main reason why RBAC is not sufficient is that in cloud computing, access permissions of service providers and users differ. In order to address this issue, Choi et al. propose a new access control model based on context-reasoning with ontologies (Onto-ACM) for dynamic access control. Onto-ACM is a semantic analysis model that can address differences in the granted permissions between service providers and users.

Haguouche and Jarir [69] also address the problem of heterogeneity of access-control models. In practice, many different access-control models (languages, types of enforcement) may possibly interact with each other, for example, in cross-organizational collaboration. The authors describe different access-control models and derive an abstraction in order to define a generic model that expresses general authorization rules and represents the access-control entities using a high-level access control model.

Another common access control approach is the attribute-based access control (ABAC) model. Ed-Daibouni et al. [70] point out that ABAC has drawbacks in terms of privacy-aware concerns. They present an extended

ABAC model, called privacy-aware ABAC model (PA-ABAC) that addresses these issues.

Caramujo et al. [71] propose and discuss the RSL-IL4Privacy language, a structured language format for the specification and documentation of privacy policies with multiple representations. The language partially uses natural language in order to allow users to specify their privacy policies. The approach aims to formalize existing natural language policies of, for example, social media platforms in order to be able to compare them more easily.

The aforementioned models are mainly concerned with access control. However, with the advent of a data-driven economy it has become increasingly important not only to protect access to data but also to be able to control what happens after granting access. Usage control is a generalization of access control: Apart from access permissions, it also regulates the subsequent data usage after initial access has been granted. [106].

Sandhu and Park [72, 106, 107] propose an usage control model (UCON), which is an attribute-based access control model that evaluates access conditions not only before granting it (the so-called pre-access phase) but also during the ongoing access phases. The UCON model is comprehensive enough to cover traditional access control models and to provide protection of system resources in a collaborative and dynamic environment.

Jürjens [73] proposes UMLsec as an extension of the Unified Modelling Language (UML). UMLsec enables a security expert to specify security-related information formally in a system design with UML diagrams. UMLsec can express security constraints that provide criteria for the security evaluation of a system design.

Basin et al. [74] argue that security models can be used for the precise documentation of security requirements and for the generation of code, for example, for generating completely preconfigured security infrastructures. However, the focus of their work is on the enforcement of security in software system, not on the specification of security and privacy demands by users with a PAP.

In another work, Basin et al. [108] provide an approach for the model-driven generation of security-aware graphical user interfaces. They aim for limiting the visible actions of users in a software user interface based on the currently active policies. They generate the user interface based on information that was modeled by security and GUI designers. However, they do not support the generation of policy specification user interfaces.



Neisse and Dörr [75] present an approach for the specification of usage control policies and their refinement from specification-level policies into implementation-level policies with a meta model for policy specification at different layers of abstraction. They specify a software system with the Interaction System Design Language (ISDL) and enrich it with usage control policies. However, they do not consider the policy specification by users of the system.

Kumari [76] also proposes an approach for the model-driven refinement of specification-level policies into implementation-level policies with a policy derivation framework. She proposes to specify the domain-specific formal semantics of actions by instantiating a meta model supporting the hierarchical refinement of actions. In her work, users can specify specification level policies with a template-based PAP. The resulting policies are automatically refined into machine-understandable equivalents. However, Kumari is investigating neither the process for template elicitation from an application domain nor the usability of the PAP nor the generation of usable PAPs.

Policy languages support the specification of security or privacy demands during the runtime of a system in a machine-understandable format (implementation level). Various policy languages for the specification of machine-understandable policies have been proposed. De Coi and Olmedilla [109] give an overview on implementation-level policy languages. We do not discuss them in detail, as they are outside the focus of our work. We merely extend our policy template model with models of XML-based policy languages to support the transformation of SLPs into ILPSs. However, the specification of those transformation rules is not researched in this work and is, thus, in its infancy. In our reference implementation, we use the IND<sup>2</sup>UCE policy language [110] or the MYDATA policy language [111], respectively, both of which are based on the policy language for distributed usage control by Hilty et al. [112].

In sum, we identified multiple models in literature that explain security and privacy principles and concepts [61–73] and model-driven approaches for the refinement and generation of machine-understandable policies [74–76]. The area of modelling security and privacy demands in form of policy templates is not yet covered in the state of the art. We elaborate such a model that builds the baseline for the automation of the PAP creation.

## **2.4 Usable Security and Privacy Policy Specification**

As stated in the introduction, we aim to increase the usability of PAPs (compare RQ1 and RQ4). Several approaches with a similar objective have been proposed. In the following, we present work that contributes to

improving the usability of PAPs and other security and privacy-related software during their creation.

Whitten and Tygar [27] state that the design principles required to achieve usable security are significantly different from those of general consumer software. They tested their hypothesis in a case study where they evaluated the usability of PGP 5.0. They conclude that »a body of public work on usability evaluation in a security context would be extremely valuable, and will almost certainly have to come from research sources, since software developers are not eager to make public the usability flaws they find in their own products« [27]. Later, Cheng et al. [28] as well as Ruoti et al. [29] independently found that the usability of email encryption with PGP did not significantly improve in later versions of the software.

In [24], Whitten shows how the usability patterns »safe staging« and »metaphor tailoring« improve the usability of security software, in her case an application for email encryption with PGP. The safe staging approach proposes the step-by-step activation of security functionality. This means that less experienced users start with a limited functionality of the software and after the fulfillment of specific conditions (e.g., an application time threshold, a skill test or an explicit activation by the user) more features are enabled and introduced to the more experienced users. Whitten evaluated the step-by-step increase of the expressiveness of her tool. Metaphor tailoring proposes to use known symbols as metaphors for representing concepts of security.

A major problem with security policy configuration is that interfaces are often designed poorly and that usability aspects are not considered properly. To better address the security and usability co-design issues, Cranor and Garfinkel [30] propose concepts and processes for making security software more usable.

A more generic approach to usable security is presented by Zurko [31]. She examines the human and social aspects of IT security and notes that there is a difference between understanding and effectively using security controls. Zurko observes that computer systems (and therefore security mechanisms) are often too complex for users to understand. However, in order to use security effectively, she argues that users do not need to understand every detail of the implementation. The utility of the security mechanism helps users, not the knowledge of their functionality. Zurko proposes to make insecure options less attractive and harder to select, so that less experienced users do not activate insecure security settings by accident. She also states that usable security should be used as an instrument for marketing.

Kuo et al. [32] consider the configuration of secure 802.11 networks. They note that today's configuration interfaces often fail to consider how



people interact with technology and that the configuration task is a dysfunctional conversation. Among other measures, they propose the following three design principles: Developers of security configuration tools should not expect any technical knowledge or expertise from the users. In addition, the effort of users using the tool should be minimized. Finally, they emphasize the importance of letting users have a positive user experience when configuring security aspects.

Reeder et al. [33] discovered open challenges in the task of specifying security and privacy policies with PAPs. They therefore made the following suggestions:

- Ensure that users understand the relations between protectable objects and the terminology.
- Provide a clear and consistent terminology to the user.
- Communicate and enforce a clear structure of the policies and their specification process.
- Explain the default policies (security and privacy by default) to the user.
- Discover and prevent policy conflicts.

We address the first three challenges in our work by eliciting information directly from stakeholders of an application domain and by using their terminology when deriving policy templates. In addition, we provide different specification processes and policy structures in the form of group-specific specification paradigms, which we map to the respective users to increase usability.

Vania et al. [34] report on experimental evaluations of improving the usability of policy specifications with SPARCLE by assisting the users in the specification process. For example, the authors propose to add syntax highlighting to a natural language policy specification interface. In the experiment, their hypothesis was that highlighting would help users to learn how to write policies. During the experiment, the users said that they liked the new feature, but the effectiveness of policy specification did not increase.

Kuo [35] demonstrates that security in communication can be enhanced significantly by reducing the impact of user errors. To this end, several design strategies are introduced and the applicability of these strategies for secure communication is discussed. One major aspect is that developers should »make realistic assumptions of user knowledge and human behavior« [35]. We agree that user errors must be prevented. However, we focus on the increase of objective effectiveness by matching appropriate specification paradigms.

Lampson [19] identified reasons for the lacking usability in security solutions. Mainly, he argues, usability is poor because vendors have little incentives for spending effort to make security solutions more usable. According to Lampson, the lack of incentive is ultimately caused by the many users who do not care about security, mainly because they do not understand the potential monetary loss from security incidents. In addition, Lampson requests more simple models of security that users can understand.

Johnson et al. [25] proposed guidelines for solving remaining challenges for security and privacy policy authoring interfaces in addition to those recommendations presented by Reeder et al. [33]. They propose an appropriate limitation of expressiveness in PAPs to communicate risks and threats to the user as well as to provide access to metadata. We follow this proposition, as we provide specification paradigms with different levels of expressiveness.

In addition, Johnson et al. [10] positively evaluated the use of policy templates for the process of policy specification for non-experts. Based on their experimental results, they recommended to use such templates. We accept this advice and propose the specification paradigm »template instantiation«.

Fang and LeFevre [26] propose an active-learning privacy wizard for social networking sites in order minimize the configuration effort for users. At a high level, the wizard solicits a limited amount of user input and other information already visible to the user. Using this information the wizard infers a privacy-preference model describing the user's personal privacy preferences. This model is used to configure the user's detailed privacy settings automatically. We also propose the specification paradigm »wizard«.

Zhao et al. [11] surveyed existing approaches for the specification of privacy policies by the user. They found that existing PAPs fail to deliver a user-friendly interface. They argue that one major reason is the tool designers' lack of understanding of the user group, whose available mental resources do not match the required resources of the tool. This mismatch causes usability issues for the user, which makes the tool appear complicated and error-prone.

Morisset and Sanchez [36] propose a user-friendly tool for the visualization and handling of large numbers of attribute-based access control policies. They aim to reduce the cognitive load of the user by applying the circle packing visualization technique to the task of policy visualization. This technique first displays an overview of the policies, and then the user can zoom in to see more details of specific policies. They positively evaluated their approach in an experiment in which users were asked to perform

changes on existing policies. They found that the approach is accepted by users in terms of satisfaction. Moreover, users can perform fast with their tool. However, they did not evaluate the effectiveness of their tool, that is, the correctness of the specified or modified policies, as we did in our experiments.

Narouei et al. [37] consider the challenge of time consuming and error-prone retrieval of access control policies from documents with an automated extraction process based on semantic role labeling. They use the high-level requirements specification documents in unrestricted natural language that most organizations have to extract access control policies with their approach. However, they do not involve the user in the policy specification process, and they do not provide a corresponding PAP.

Gerl and Prey [38] present a personal privacy policy user interface, which uses the »Visual Information Seeking Mantra« [39] design principle. This principle proposes a step-wise refinement of displayed data from an overview to a detailed view. They mainly focus on the presentation of privacy policies in a human-understandable format, and they evaluate their approach in comparison to state-of-the-practice privacy policies. Their approach provides different specification paradigms on one user interface. More specifically, the user creates a coarse-grained specification in a specification paradigm with low expressiveness and then switches to a specification paradigm with more expressiveness to flesh out the details of the policy. We currently do not support such switching between specification paradigms at runtime; however, this is part of our future work.

A significant body of work assesses the usability of privacy settings in online social networks, such as Facebook.

Strater and Lipford [6] and Lipford et al. [18] examined how privacy settings in social networks (Facebook) are used. These authors identify two main problems: Users do not understand the accessibility of personal information by others; thus, before they can define meaningful privacy policies they need to learn what to disclose and what to protect. In addition, many users only change the default specification of their privacy settings if something bad happens, such as a privacy breach. Therefore, the authors propose to use more restrictive default policies but note that this will not help users to understand their privacy settings and may not accurately reflect the users' actual privacy needs. Moreover, they argue that users need to be made aware of what information is shared with whom, which can be achieved with improved interfaces that make privacy settings as simple as possible and include them into regular profile modification (instead of locating them on separate pages).

Boyd and Hargittai [40] investigated reasons why users do not configure their privacy settings in Facebook. They found that both frequency and type of Facebook use as well as Internet skills affect the user habits regarding the specification of privacy settings. The authors also carried out research to improve the usability of privacy settings and policy specification. Similar to Boyd and Hargittai, we reason in our user intention model that the knowledge and skill level of the users influences their behavior.

Liu et al. [7] as well as Madejski et al. [41] investigated the discrepancy between desired and actual privacy settings in Facebook. Both author groups confirmed that users were seemingly unable to specify their privacy settings in Facebook correctly. They studied the actual sharing intentions of the users in order to identify violations in the users' privacy settings. They identified severe mismatches between users' intention and actual settings. Madejski et al. also found that a majority of users cannot or will not fix those mismatches.

To summarize, a broad body of research exists on the usability improvement of security and privacy systems [6, 7, 10, 11, 18, 19, 24–41]. However, we identified a gap in the literature with respect to studies that investigate the effect of different specification paradigms on the usability of a PAP. We address this gap by providing guidance for selecting the appropriate specification paradigms for (types of) users in terms of usability (effectiveness, efficiency and satisfaction) based on empirical data.

## **2.5 Existing PAPs and Derived Specification Paradigms**

A specification paradigm defines the specification process in a PAP for the task of policy specification. Thus, every PAP has to implement at least one specification paradigm. Multiple PAPs exist in practice and in academia. We derived multiple specification paradigms from the state of the art and practice. Therefore, we first identified and analyzed existing PAPs in section 2.5.1. Next, we survey proposed paradigms derived from existing specification approaches in Section 2.5.2.

### **2.5.1 Security and Privacy Specification Approaches and Tools**

PAPs used for the specification of security and privacy demands can be found in various application domains. They are present in tools used on a daily basis, such as social media networks or Internet browsers. Other PAPs, such as the ones found in commercial tools, are mainly used by experts for the administration of security and privacy demands for other users in a community. We clustered the PAPs we identified into the following four categories:

- Security and privacy settings in online services
- Security and privacy settings in browsers
- Security and privacy settings in commercial tools
- Tools and prototypes for security and privacy policy specification from academia

The following subsections describe the relevant PAPs that we found in the literature and in the field.

### ***Tools and Prototypes for Security and Privacy Policy Specification from Academia***

A lot of research went into the specification of privacy and security settings by security experts in the form of machine-understandable policies. Even if the focus of our work is to enable non-experts to specify privacy demands in natural language, the interface concepts for machine-understandable policies can be transferred to natural language interfaces for privacy policy specification. Some concepts are introduced in this section.

PERMIS [113] is a generic RBAC-based (Role-Based Access Control) authorization infrastructure developed at the University of Kent, UK. PERMIS policies are created with the »Policy Editor« or the »Policy Wizard«. These tools target expert users and system administrators, respectively. The policy wizard uses a policy specification paradigm with multiple sequential small specification steps. It asks supportive questions to guide the user through the specification process. The policy editor provides the user a template-based approach for the policy specification. Specified policies are generated on the fly as XML clauses and displayed to the user. Both tools can be attached to local LDAP systems to facilitate specification. They use a generic terminology that might be incomprehensible to non-experts. PERMIS supports the conversion of policies to the policy languages XACML or OWL/RDF.

KAoS [114–116] is a policy and domain services framework. It contains the KAoS Policy Administration Tool (KPAT), which is a policy editor for the specification of OWL-DL or DAML policies. It was designed to provide policy specification capabilities for administrators that do not require intensive training. The KPAT editor is driven by the ontologies of the computational environment and the application context loaded into it. Policy templates are provided for instantiation, which are based on the ontology and presented as hypertext templates forming natural English sentences. Specified policies are automatically transformed into machine-understandable equivalents. In addition, a policy wizard is provided, which divides the decisions to be made by the user into several small, well-

explained steps. To limit the decisions that need to be made by the user, KPAT also provides customization options for creating specific policy editor instances tailored to an individual application-domain.

Karat et al. [33, 52, 117] propose a tool named SPARCLE that allows users to enter their security demands in natural language or in a structured natural language-based format. SPARCLE can transform the structured format into machine-understandable policies. The authors worked out several key usability challenges that need to be mastered to improve the policy specification process. Among others, the used terminology must be clearly defined and structured, and it must be used consistently. Default rules that apply if no other security policy is specified must always be described. Rule conflicts must be detected and explained to the policy creator. However, Karat et al. do not use and compare different specification paradigms.

Fang and LeFevre [26] propose an active learning wizard that enables users to set their own privacy policies by making regular, brief decisions on whether or not to share a particular data item with an entity. The authors chose an iterative learning approach to minimize the difficulties of users in making holistic decisions on the privacy of their own data. Their privacy wizard instantiates a privacy-preference model describing the user's privacy preferences. This model instance is then used to configure the user's privacy settings automatically. They aim at limiting the amount of user input as much as possible in order to relieve the user of the specification burden.

The Hades Java Applet Permission Editor [118] was developed at TU Hamburg. It allows the specification of security settings for the Java VM. It provides a text editor in which policies can be specified or changed directly in plain text according to a given grammar. The user can add a permission block that can be used as text-based template. Besides that, Hades does not provide any specification support, help functionality or GUI.

Inglesant et al. [119] present a method and tool for transforming access control policies into machine-understandable policies. The policy creator needs to specify his security demands in a controlled natural language format. The controlled language consists of simple sentence templates with variables that can be instantiated.

MotOrBAC [120, 121] is an open source policy editor based on the OrBAC model developed by Telecom Bretagne. It provides various different forms and options to create and manage OrBAC policies. Furthermore, it supports the simulation of policies and access requests.



The UMU-XACML-Editor [122] is a policy editor developed at the University of Marcia (UMU). It provides a template-based graphical user interface for building XACML policies. The specification process is strongly aligned to the XML representation of the policy, which requires XACML expertise from users. The tool provides schema validation and checks for missing parameters.

Stepien et al. [8] argue that early XACML editors, such as the UMU-XACML editor, require expert knowledge of users in order to be usable. They propose an XACML editor for non-experts, which is based on a natural language notation of XACML. Users can specify policies by formulating access control policies in a structured language format. Stepien et al. see positive effects in basing the specification on natural language.

Vollat [123] discusses the applicability of various usability patterns to PAPs and to the policy specification process in general. He applies usability patterns to various PAP prototypes and evaluates their effect on users, for example, by evaluating the tools with AttrakDiff tests. To carry out his studies, he implemented a template-based policy editor and a policy wizard.

Verlaenen et al. [124] present a policy ontology with a generic policy model, which can be extended to a specific policy language. They aim to bridge the gap between general-purpose and domain specific policy languages. XML was chosen as the base language for their policies, but since XML is not suited for non-experts, they propose a template approach on the specification level.

Reeder et al. [125] note that most of the time, policies are displayed as a list of rules; therefore, interactions between policies cannot be properly portrayed. It is up to the user to determine rule interactions. While experienced administrators might invest the time to learn and use complex user interfaces, novice and occasional users will not. To address the need for better PAPs, the authors introduce a new model (called Expendable Grids) for displaying and editing policies. In a user study, they show that using their interface for authoring file permissions is superior to the Windows XP native file permissions interface. The Expendable Grid interface allows users to complete tasks more accurately and faster than does the Windows file permission interface.

Conti et al. [126] developed a prototype of a privacy PAP for the healthcare domain. This PAP provides two different template-based policy specification user interfaces: The first is a simple one for non-experts with a very limited expressiveness; the second provides a higher expressiveness for experts.

Kumari [76] proposes a prototype of a PAP for privacy policies in a social network scenario. With her tool, users can define specification-level policies at two levels of granularity. She shows that inexperienced users can select from predefined privacy policies whereas advanced users can use a template-based specification. However, Kumari mainly focuses on the model-based transformation of specification-level policies to implementation-level policies.

Villarreal et al. [127] propose privacy tokens as a mechanism for privacy specification by users. They developed a system with which users can specify generic privacy policies that can be handed over to different online services for enforcement. They provide a list of predefined privacy profiles, and the users select one of them or specify an individual profile. Token customization is realized by predefined privacy policies that the user can select for creating an individual privacy profile.

### ***Security and Privacy Settings in Online Platforms***

Besides academic approaches, many domain-specific PAPs exist in practice. For many users, security and privacy are important issues when sharing personal data online, for example, in social networks and Internet platforms. Therefore, many companies provide their users interfaces for setting their security and privacy preferences.

Facebook [128] allows users to specify their privacy settings in a very fine-grained manner. Facebook supports several security settings, where users, in addition to standard security features such as password control, can limit the visibility of their information. Settings are supposed to be usable by non-expert users. Consequently, users receive a lot of support during specification, such as explanations, examples or simulations of the effects of the specified policies. For example, users can see their profiles from the perspective of other users. Facebook uses a mixture of different concepts for the various specification options: They use template-based specifications, small specification wizards and predefined policies that can be enabled or disabled by the user. Facebook updates the corresponding user interfaces regularly. In the past, studies revealed usability problems with Facebook's privacy settings. For example, users expected in some cases a different behavior from their specified privacy policies [7]. Lipford et. al demonstrated how usability improvements in Facebook's privacy settings can influence privacy management of users in a positive way [18].

Google [129] provides web interfaces for the specification of privacy and security policies for the one's personal Google account. For the privacy settings in online accounts, the company has introduced a privacy check wizard that guides the user through multiple pages to configure the use of personal information by Google services and third parties. In addition,



Google provides a set of categorized and predefined policies that can be enabled or disabled by the user.

Twitter [130] provides security and privacy settings as predefined policies. In some cases, the user can select the most suitable option from a list of policies. In all other cases, the user can only enable or disable predefined policies.

### ***Security and Privacy Settings in Internet Browsers***

Users use Internet browsers to access online services. In doing so, they share personal data with services in the Internet and with other service users. We analyzed four of the most popular browsers [131] regarding security and privacy settings.

Google Chrome (Version 64) [132], Mozilla Firefox [133] and Microsoft Edge (Version 41) [134] offer configuration options for security and privacy settings. In most cases, they offer users predefined policies in two different ways: Either they provide individual, independent policies that the user can enable or disable, or they provide a policy list from which the user can choose. Those two ways mainly differ in the number of specification options.

Microsoft Internet Explorer (Version 11) [135] uses a security level approach for setting the coarse-grained security settings. The security levels are named »Medium«, »Medium-high« and »High«. By default, each level denotes a predefined set of configuration settings. Optionally, users can customize these default profiles by enabling or disabling the predefined policies according to their individual preferences.

### ***Security Settings in Commercial Tools***

There exist a variety of other software containing security and privacy settings to configure the corresponding data protection measures. We highlight some examples in the following.

Microsoft's Local Group Policy Editor [136] of the Windows operating system (e.g., Windows 10) offers a variety of settings (e.g., firewall settings, password policies, startup/shutdown scripts) for Windows environments. The editor provides extensive specification support, such as explanations and examples. It uses template-based specification, small specification wizards, predefined policies to select from and specific security settings that can be enabled or disabled.

Using Windows File Permissions [136] of the Windows operating system, users can restrict the access to individual files and folders (e.g., in the network or for other users). Although file permissions are mainly used by

experienced users, non-experts may also get in touch with these settings. Windows divides the settings into a simple standard screen and an advanced detail view. In the simple view, fixed permissions (e.g., read or write) can be granted for users. The advanced settings offer more options for assigning file permissions to entities. Both views are structured as templates that the user can instantiate as concrete permissions. However, inexperienced users will hardly understand the terminology (e.g., the difference between modify and write or the meaning of object type).

IBM P3P (Platform for Privacy Preferences) is a technical platform to provide data protection information, which is mainly supported by Internet Explorer. With the IBM policy editor [137], website administrators can specify protection information policies. Policies can be created from scratch or from templates, and users navigate through a tree structure to specify their data usage preferences. The policy editor provides views for XML and HTML, and it supports error checking and recommendations. Extensive documentation is available.

The Policy Design Tool [138] by IBM is an Eclipse-based tool to model and analyze high-level security requirements and to specify templates and XACML policies. It contains a PDP that allows the simulation of access requests.

The Identity Server by WSO2 [139] is an open source identity and entitlement server. It supports the specification of XACML policies. The user interface is a web-based application providing dynamic forms; it mainly targets system administrators, that is, experts. It offers specification support to the extent that variables can be chosen from lists or drop-down menus. The identity server is an expert tool that requires deep knowledge about the system.

## 2.5.2 Overview of Derived Specification Paradigms

The PAPs described in the previous subsection differ in their underlying specification paradigm. During policy specification, the PAPs request different input in different ways, and they provide different expressiveness and different levels of guidance to the user. We derived the following specification paradigms from existing PAPs:

- **Template Instantiation:** The user can instantiate the desired privacy setting by adjusting selection options in a template-based interface. Usually, templates offer multiple decision options and thus allow a fine-grained specification of one's personal security and privacy demands. The templates can be domain-specific or generic. The user can choose the specification order on his own.

- **Wizard:** The user can instantiate privacy settings based on a template-based interface, where the specification process is subdivided into several small steps. The user cannot decide on the specification order. The specification process is usually well guided.
- **Default Policies:** The user can select from multiple predefined privacy policies per topic. The expressiveness in the specification is therefore limited.
- **Security and Privacy Levels:** The user can select a level of security and privacy that contains a predefined set of default privacy policies without offering customization possibilities per policy.
- **On/off Switches:** The user enables or disables one or more predefined policies without any customization options per policy. This paradigm is a specialization of the paradigm »default policies« allowing only the activation or deactivation of exactly one privacy policy per topic.
- **Text-based Specification:** The user enters plain text security and privacy policies into a tool. The grammar of the text is given by either natural language, a controlled natural language or a policy language. The text input and the corresponding policy output can be on the level of specification and implementation policies.
- **Grid-based Specification:** The user maps individual assets and policies with a PAP based on a grid layout.
- **By Design:** Security by design and privacy by design without any customization options are not a specification paradigm, but one way how service developers can handle security and privacy settings. Users do not set anything by themselves, but have to rely on the default security and privacy configuration. Many smaller online services, such as web shops, do not provide options to their users for configuring personal security and privacy preferences.

Table 1 summarizes the PAPs found in the state of the art and state of the practice and maps the derived specification paradigms to those PAPs. In addition, the table shows whether the PAPs allow the specification of policies on the human-understandable level (specification-level policy – SLP) or on the machine-understandable level (implementation-level policy – ILP).

We use a selection of the specification paradigms derived from literature in our PAP generation framework in order to create respective PAPs. We explore and empirically substantiate the mapping of specification paradigms to users to increase the usability of the PAP (in terms of effectiveness, efficiency and user satisfaction).

Table 1: List of PAPs from Academia and Practice and Their Used Specification Paradigms

| PAP in SotA and SotP          | Specification Paradigms |               |                        |        |                  |                 |                 |                          |                          |
|-------------------------------|-------------------------|---------------|------------------------|--------|------------------|-----------------|-----------------|--------------------------|--------------------------|
|                               | Academic/Practice       | ILP/SLP level | Template Instantiation | Wizard | Default Policies | On/off Switches | Security Levels | Text-based Specification | Grid-based Specification |
| Conti et al.                  | A                       | SLP           | X                      |        |                  |                 |                 |                          |                          |
| Facebook Privacy Settings     | P                       | SLP           | X                      |        | X                | X               |                 |                          |                          |
| Fang and LeFevre Wizard       | A                       | SLP           |                        | X      |                  |                 |                 |                          |                          |
| Google Chrome                 | P                       | SLP           |                        |        | X                | X               |                 |                          |                          |
| Google Privacy Dashboard      | P                       | SLP           |                        | X      |                  | X               |                 |                          |                          |
| Hades Java Policy Editor      | A                       | ILP           |                        |        |                  |                 |                 | X                        |                          |
| IBM P3P Policy Editor         | P                       | SLP           | X                      |        |                  |                 |                 |                          |                          |
| IBM Policy Design Tool        | P                       | SLP/ILP       | X                      |        |                  |                 |                 |                          |                          |
| Inglesant et al.              | A                       | SLP           |                        |        |                  |                 |                 | X                        |                          |
| KPAT                          | A                       | SLP/ILP       | X                      | X      |                  |                 |                 |                          |                          |
| Kumari                        | A                       | SLP           | X                      |        |                  |                 |                 |                          |                          |
| Microsoft Edge                | P                       | SLP           |                        |        | X                | X               |                 |                          |                          |
| Microsoft Internet Explorer   | P                       | SLP           |                        |        | X                |                 | X               |                          |                          |
| MotOrBAC Editor               | A                       | ILP           | X                      |        |                  |                 |                 |                          |                          |
| Mozilla Firefox               | P                       | SLP           |                        |        | X                | X               |                 |                          |                          |
| PERMIS                        | A                       | ILP           | X                      | X      |                  |                 |                 |                          |                          |
| Reeder et al.                 | A                       | SLP           |                        |        |                  |                 |                 |                          | X                        |
| SPARCLE tool                  | A                       | SLP           | X                      |        |                  |                 |                 | X                        |                          |
| Stepien et al. XACML editor   | A                       | SLP           |                        |        |                  |                 |                 | X                        |                          |
| Twitter                       | P                       | SLP           |                        |        | X                | X               |                 |                          |                          |
| UMU-XACML-Editor              | A                       | ILP           |                        |        |                  |                 |                 |                          |                          |
| Verlaenen et al.              | A                       | SLP/ILP       | X                      |        |                  |                 |                 |                          |                          |
| Villarreal et al.             | A                       | SLP           |                        |        |                  | X               | X               |                          |                          |
| Vollat's Usable Policy Editor | A                       | SLP           | X                      | X      |                  |                 |                 |                          |                          |
| Windows File Permissions      | P                       | SLP           | X                      |        |                  |                 |                 |                          |                          |
| Windows Group Policy Editor   | P                       | SLP           | X                      | X      | X                | X               |                 |                          |                          |
| WSO2 Identity Server          | P                       | SLP           | X                      |        |                  |                 |                 |                          |                          |

## 2.6 User Behavior

In order to be capable of mapping specification paradigms to users, we need to understand users and their capabilities better. Therefore, we explored the literature about user intention models and user type models.

### 2.6.1 Intension Models

Theories and models that try to explain human behavior, but are not specialized in security or privacy, inspired our intention model (compare Section 6.2.1). A key element in the model is the user's »intention«. Psychological models often distinguish intention from behavior. For instance, in the theory of planned behavior (TPB) by Ajzen [140], intention and behavior are distinct elements. In that theory, however, intention is equivalent to what we call »motivation«, and the element »perceived behavioral control« is part of our element »barriers«. According to TPB, perceived behavior control influences the behavior and what the author calls »intention«. Thus, the TPB is included in our user intention model, but our model integrates the user requirements of PAPs and offers therefore a more detailed view on barriers (perceived behavioral control). The TPB also has an element called »attitude«. In our model, a positive attitude towards PAPs is a prerequisite for the application of our model.

The interrelation of barriers and motivation is part of the behavioral model for persuasive design by Fogg [141]. Fogg presents the interrelation of motivation and simplicity factors, which are barriers but positively formulated, in a graph. The graphical representation illustrates that high motivation can lead to the performance of a behavior even if there are barriers and that low motivation can lead to the performance of a behavior when the barriers are low.

The element »need for privacy« was inspired by the well-known hierarchy of needs by Maslow [142]. According to Maslow, persons are dominated and their behavior is organized by unsatisfied needs only. We consider the need for privacy to be a subset of Maslow's need for safety and security. In our model, we assume that users whose need for security and privacy is satisfied will not take action to improve their security and privacy.

The Privacy Paradox describes the dichotomy between the need for privacy and the actual behavior of users with respect to taking privacy-related actions. Kokolakis et al. surveyed the state of the art regarding the privacy paradox [143]. They outline multiple explanations for this phenomenon. However, they do not address the concept of barriers users must master for specifying policies. We consider barriers in our user intention model in Section 6.2.1.

## 2.6.2 User Type Models

Each user has different characteristics, capabilities and resources. This leads us to the assumption that different specification paradigms are likely to fit differently well to a certain user with respect to usability. To explore the relationship between suitable specification paradigms and user types, we explored related work regarding user type models.

Many generic user type models exist in psychology that cluster users into categories. Each category explains the character traits and behavior of a certain user type. Those methods describe human traits and behavior in general, that is, they are not tied to a particular situation or domain. Examples are the Big Five personality traits [144], Keirsey's Temperaments [145] and the Myers-Briggs Type Indicators [146].

Besides these generic user type models, other work relates to the use of computers and the character traits relevant for security and privacy decisions.

Westin conducted around 30 privacy surveys for classifying users [147]. In most of his privacy surveys, he clusters the users into three categories based on their privacy concerns: Fundamentalist (high concern), Pragmatist (medium concern), and Unconcerned (low concern). However, Westin's approach is controversially discussed in the literature. For example, Urban and Hoofnagel [148] argue that Westin's work is neglecting the importance of knowledge or available information about privacy practices and domain specific business processes.

The approach »Concern for Information Privacy (CFIP)« of Smith et al. [149] measures the privacy concern of a person as a numerical value. This value is calculated based on fifteen statements about privacy, which the person rates on a 7-point Likert scale. The scenarios of CFIP are kept quite abstract and do not directly relate to online services that collect and process user data.

Malhotra et al. propose their approach »Internet Users' Information Privacy Concerns (IUIPC)« [150], which extends the existing work of Smith (CFIP). IUIPC reflects the concerns of Internet users about information privacy with a special focus on the individuals' perception of fairness in the context of data privacy.

The Information Seeking Preferences by Morton and Sasse [151] are an approach to cluster users into the five groups: information controllers, security concerned, benefit seekers, crowd followers and organizational assurance seekers. The categorization is based on the ranking of 40 privacy related statements. Their approach aims to support companies in

providing services that are better adopted by users in terms of privacy behavior.

Dupree et al. proposed a model with five privacy personas [14]. They reacted to the criticism of Westin's privacy indexes by considering both motivation (concern) and knowledge. The persona model was built upon empirical data. Dupree derived the five personas from personal interviews with 32 university-related digital natives, who had an average age of 26.3 with a standard deviation of 5.9. The five personas can be differentiated according to two attributes of the user: the user's knowledge of security and privacy and the user's motivation to spend effort on privacy and security protection. The personas also describe the handling of personal data in the Internet age and the general need for security in the IT sector.

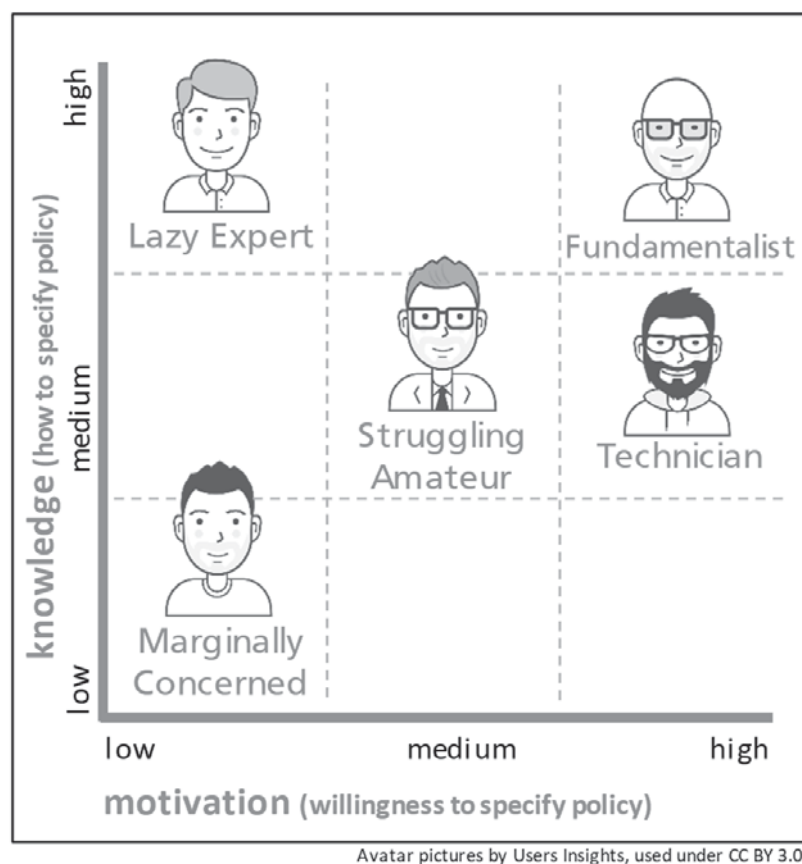


Figure 14: Dupree's Persona Model

Dupree's model distinguishes users by their motivation (willingness to specify privacy settings) and their knowledge of how to specify appropriate privacy settings. The five personas are (see Figure 14):

- Marginally Concerned: Low knowledge and low motivation
- Amateur: Medium knowledge and medium motivation
- Technician: Medium knowledge and high motivation



- Lazy Expert: High knowledge and low motivation
- Fundamentalist: High knowledge and high motivation

The personas are described in detail in Appendix C.

### **Summary**

In sum, we explored multiple user intension models [140–143] and user group models [14, 144–151] in the literature. We use the user intension models from the state of the art for the creation of an extended user intension model that explains the discrepancy between the users' demands for security and privacy protection and the reality of the users ignoring their interaction options, thus, the use of a PAP. We use our model to identify barriers for users to use PAPs and relate them to usability issues that users face with PAPs. We select one of the user group models for matching different specification paradigms to user groups to evaluate the potential usability increase when selecting the best matching specification paradigm.

## **2.7 Summary and Conclusion**

In this chapter, we described and discussed the foundation of our research and related work. We presented the state of the art in the fields of »elicitation of security and privacy requirements«, »policy models and languages« and »usable security and privacy policy specification«. In addition, we built the foundation for this work with respect to the following two aspects: We collected existing PAPs from literature and practice to derive specification paradigms, and we presented the state of the art in the field of user behavior including user type models.

A broad body of research exists on the usability improvement of security and privacy systems. Many approaches evaluate and improve the usability of PAPs or security software in general (e.g., [6, 18, 24, 25]). Some of them propose explicit specification paradigms to be used, such as the »template instantiation« by Johnson et al. [10] and by Hibshi et al. [51] or the »wizard« by Fang and LeFevre [26]. However, the effect of different specification paradigms on the usability of a PAP has not been studied intensively so far. One key problem is to identify the adequate expressiveness of a PAP for a given type of user [25]. We address this gap with the **User to Specification Paradigm Mapping**, which is **Contribution 1 (C1)** of this thesis. We analyze the effect of specification paradigms on users with respect to usability in order to give recommendations for specification paradigm selection. Kuo [35] states that realistic assumptions about user knowledge must be considered. Boyd [40] sees knowledge as an important requirement for users to specify privacy settings. We address this issue by analyzing the usability of



specification paradigms with respect to user groups (personas) clustered according to knowledge and motivation.

The state of the art regarding the elicitation of security and privacy requirements reveals a lot of methods and approaches for the elicitation of security and privacy related requirements [23, 42–50], policies [51–55] and risks [56–60]. All these methods aim to elicit concrete security or privacy requirements, threats and risks in the early software development phases. However, we did not identify a systematic approach for eliciting policy templates directly from the stakeholders of an application domain using state of the art RE techniques. In contrast to previous work, we aim at eliciting policy templates whose concrete instances (security or privacy policies) are specified by users at runtime. Thus, we do not want to enforce one static set of policies for a system, but allow users to adapt the policies to their individual security and privacy demands at runtime. Thus, we propose the **Policy Template Elicitation Method** as **Contribution 2 (C2)** of this thesis. To this end, we reuse and combine existing, proven concepts and techniques from the state of the art in our method (e.g., [85–93]). We elicit and derive example policies during the elicitation of policy templates by identifying assets, use cases, threats and countermeasures. Similar process steps are proposed, for example, by Haley et al. [46] for their security requirements elicitation, by Olzak [45] for his threat modelling approach and by Cranor and Garfinkel [30] in their secure system design. Multiple RE techniques have been proposed for the elicitation of security and privacy requirements and policies. For example, Karat et al. [52] use questionnaires and semi-structured interviews. Callele and Wnuk [53] confirm that interview, brainstorming and survey techniques to be applicable techniques. We decided to use group dynamics and selected RE techniques that can be used in workshops with varying group sizes.

We identified several very specific models in the state of the art that explain security and privacy principles and concepts [61–73]. In addition, several model-driven approaches for the refinement and generation of machine-understandable policies have been proposed [74–76]. None of the identified models and model-driven approaches is a generic model for modelling security and privacy demands in the form of policy templates that is capable of building the baseline for the automation of the PAP creation. We propose the **Policy Template Model** as **Contribution 3 (C3)** of this thesis. We decided to develop a more generic model for formalizing policy templates including their projection on different specification paradigms and their transformation into implementation-level policies. To the best of our knowledge, such a model does not yet exist.

Lampson [19] noted a missing incentive for vendors to spend effort on improving the usability of security solutions. Having security and privacy

demands formalized as policy templates, the next logical step is the provision of these templates to users as GUIs in PAPs. In order to limit the implementation effort, we propose the (semi-)automatic generation of such PAPs with our **PAP Generation Framework**, which is **Contribution 4 (C4)** of this thesis. We did not find a comparable approach in the literature for automating the creation of policy specification interfaces like our PAP generation framework. However, we identified several specification paradigms in the state of the art and practice, which we apply in our framework.

In summary, we combine the aforementioned four contributions to a comprehensive method for automating the creation of policy specification interfaces representing multiple specification paradigms in PAPs. We call this approach the **Method for Usable PAP Generation**, which is **Contribution 5 (C5)** of this thesis. We could not find a comparable method in the literature. However, we align parts of our contributions to existing approaches or reuse existing work as described above.



### 3 Policy Template Elicitation Method

The specification of policies can be challenging, especially for users inexperienced in security and privacy. One problem for users of PAPs can be an inadequate expressiveness of the specification options provided by the PAP [25]. Depending on the application domain in which a PAP is used for specifying policies, different security and privacy demands may exist.

To reduce the expressiveness of a PAP, it can be tailored to a given application domain by only offering relevant specification options to the user. One way to provide specification options with limited expressiveness are policy templates.

**Definition: Policy Template**

A policy template is a pattern formulated in a policy language that can be instantiated as a concrete policy.

Either the policy language can have a machine-understandable format and grammar, or it can be a natural language. Compared to the specification of policies from scratch, the instantiation of such a template at runtime with a PAP is easier and less error-prone. Johnson et al. evaluated policy templates to increase the usability of policy specification for users [10]. Natural-language policy templates can still be difficult to use if, for example, unknown terminology is used. This observation suggests that templates should be drafted according to the stakeholders' preferences in the domain of application in which a PAP is to be used.

In this chapter, we present a method for eliciting policy templates from an application domain, which represents Contribution 2 of this thesis (see Section 1.4). The overall goal of this method is to elicit all available information from the application domain that is needed for the instantiation of the policy template model.

We structure this chapter as follows. We explain the research approach for the policy template elicitation method in Section 3.1. In Section 3.2, we present an overview of the method. The five main steps of the method are presented in the following sections: the information retrieval in Section 3.3, the workshop preparation in Section 3.4, the conduction of the elicitation workshop in Section 3.5, the documentation of the workshop results in Section 3.6 and the derivation and validation of policy templates in Section 3.7. We summarize and conclude this chapter in Section 3.8.

### 3.1 Research Approach

We developed the policy template elicitation method in an iterative process. We devised three versions of the method and applied each version in a case study. We used the observations and lessons learned from the first two case studies for improving the method. We validated the application of the final version in two more case studies, and we assessed the quality of the results of our method in an experiment.

Our method is aligned to existing methods for security requirements elicitation and risk assessment from the literature. Similar to other approaches [30, 45, 46], we first elicit assets, threats for these assets and countermeasures for mitigating or preventing these threats. From this information and other collected documents, we derive policy templates.

Existing approaches for gathering security and privacy requirements focus on the elicitation of general security requirements [23, 47] or on risk assessment [98–100]. Mellado surveyed existing work [152]. However, the existing work does not cover the elicitation of policy templates for a specific application domain directly from stakeholders.

In the first version of our method, the method expert (the person executing the policy template elicitation method) elicited all information solely from existing documentation and discussions with stakeholders of the application domain without a structured process. However, we learned that we could derive policy templates from assets, threats, countermeasures and example policies. Based on this insight, we defined the first version of the policy template notation format to reflect these key ingredients and applied it in a case study. We used this first version of the policy template elicitation method in the »SINNODIUM« case study. At the end of the study, we carried out several interview sessions between 2013 and 2014 to ask the domain and technology experts from the company »vwd«, our application partner in the study, for improvement suggestions regarding the template and the elicitation process. The feedback obtained led to an improved second version of our method.

In the second version of the policy template elicitation method, the method expert created an initial list of assets, threats and countermeasures drawn from existing documentation and derived policy templates. Next, domain and technology experts from the companies »Amaris«, »ETRA«, »Mirasys« and »OTE« were asked to validate and improve these initial policy templates. We let those experts present their policy templates in a workshop on April 3, 2014. In this workshop, we further improved the policy templates based on the expert’s feedback and suggestions. In contrast to the first version, we integrated a workshop for the cooperative elaboration of policy templates into our method. However, the initial assets, threats and countermeasures were still elicited

by the method expert. However, this may strongly bias the method result, the policy templates.

To avoid these biases in the final version of the method, we decided to elicit assets, threats and countermeasures directly from stakeholders of the application domain in a workshop. We identified key stakeholders that need to be involved in the workshop for information elicitation and result validation. We created a coarse framework of our method, containing the following three steps: preparation of elicitation, conduction of elicitation and derivation and validation of policy templates.

For the elicitation of assets, threats and countermeasures from stakeholders in the workshop, we decided to use existing requirements engineering techniques, because there already exist a plethora of established and well-tested methods. Thus, we surveyed potential RE techniques in the literature. We mapped their characteristics on our requirements for each step of the policy template elicitation method. Finally, we selected appropriate techniques per step with the support of a requirements engineering expert.

We executed the »BeSure« case study for confirming the feasibility and user acceptance of the method as well as the completeness and correctness of the results. On April 14, 2015, we conducted a workshop together with experts from the company »DATEV« to validate our new version. The workshop consisted of three elicitation rounds for assets, threats and countermeasures. We used different RE techniques for each elicitation round. In contrast to the previous versions of the method, we elicited information directly from the stakeholders and used established RE techniques for this task.

To confirm the feasibility and user acceptance of our approach, we reapplied this final version of the policy template elicitation method in the case study »Digital Villages« with experts of »Fraunhofer IESE«. To this end, we conducted an elicitation workshop on July 7, 2017. We partially tested different RE techniques for the elicitation.

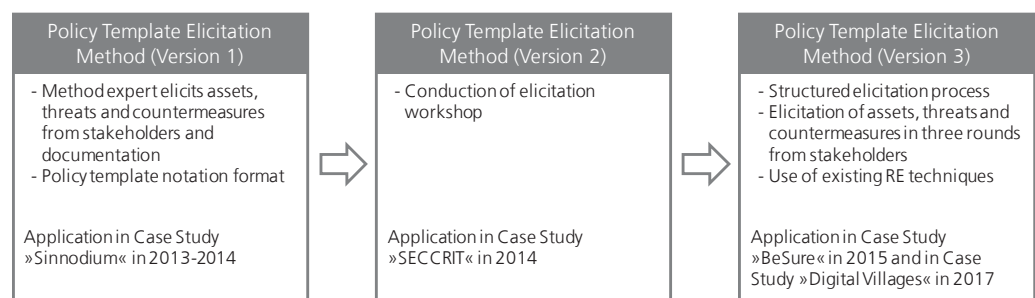


Figure 15: Research Approach for the Policy Template Elicitation Method

Figure 15 summarizes the improvements for the three versions of the policy template elicitation method and their relation to the case studies.

In this thesis, we document the policy template elicitation method as a process with five steps. In comparison, we described this method with only three steps in [80]. By splitting the three steps into five steps, we aimed to achieve a more structured description of the method in this thesis including the involved roles and the necessary input and output per step. We are confident that these changes do not influence the evaluation results. In addition, we extended the policy template notation format by variable types, maximal and minimal values for numerical variables, minimal and maximal numbers of selectable values in selections as well as the conjunction type (»AND« and »OR«) of selections. However, this optional extension has not been evaluated in the case studies described here.

We identified the following requirements for our policy template elicitation method:

- **Req\_Elicitation\_Derivation-of-Policy-Templates:** We require the method to produce policy templates as its major output. We want to use policy templates as the basic concept as they have been evaluated to be usable for ordinary users in the literature [10] and meet a major challenge for policy specification: the appropriate limitation of expressiveness [25].
- **Req\_Elicitation\_Application-Domain:** The policy templates elicited with the proposed method must reflect security and privacy demands of users from the application domain, in which the elicitation took place. The focus on the application domain limits the expressiveness of the policy templates as required by Johnson et al. [25].
- **Req\_Elicitation\_Understandable-Terminology:** Policy templates have to be understood, instantiated and managed not only by software engineers, but also by end users as required by Reeder et al. [33]. As we cannot expect deep technical knowledge from typical end users, we require policy templates to use understandable terminology for users.

## 3.2 Method Overview

In this section, we present our entire process for the systematic elicitation of policy templates from an application domain. We call this process the policy template elicitation method. A method expert is executing the method in an application domain. The method consists of five steps as shown in Figure 16:

- **Step 1 – Information Retrieval:** The method expert contacts a contact person from the application domain. He identifies relevant stakeholders to be involved in the elicitation of policy templates and other useful information sources.
- **Step 2 – Workshop Preparation:** The method expert prepares a workshop for eliciting information from stakeholders of the application domain. He defines the goals and constraints of the elicitation together with the contact person. In addition, the method expert gains a basic, high-level understanding of the application domain, which supports the preparation of the elicitation workshop. Based on the information already collected, exemplary assets, threats and countermeasures are elaborated. A list of workshop participants is finalized.
- **Step 3 – Execution of Elicitation workshop:** The method expert conducts a workshop with the stakeholders to extract relevant information. First, the assets of the application domain and typical use cases for them are elicited. Next, threats with respect to these assets are identified. Finally, potential countermeasures are determined.
- **Step 4 – Documentation of Workshop Results:** The method expert documents all results from the workshop. In addition, he derives exemplary policies from the elicited information by combining assets, threats and countermeasures.
- **Step 5 – Derivation and validation of policy templates:** The method expert derives policy templates from the workshop results. A validation of the results with users from the application domains concludes the method execution.

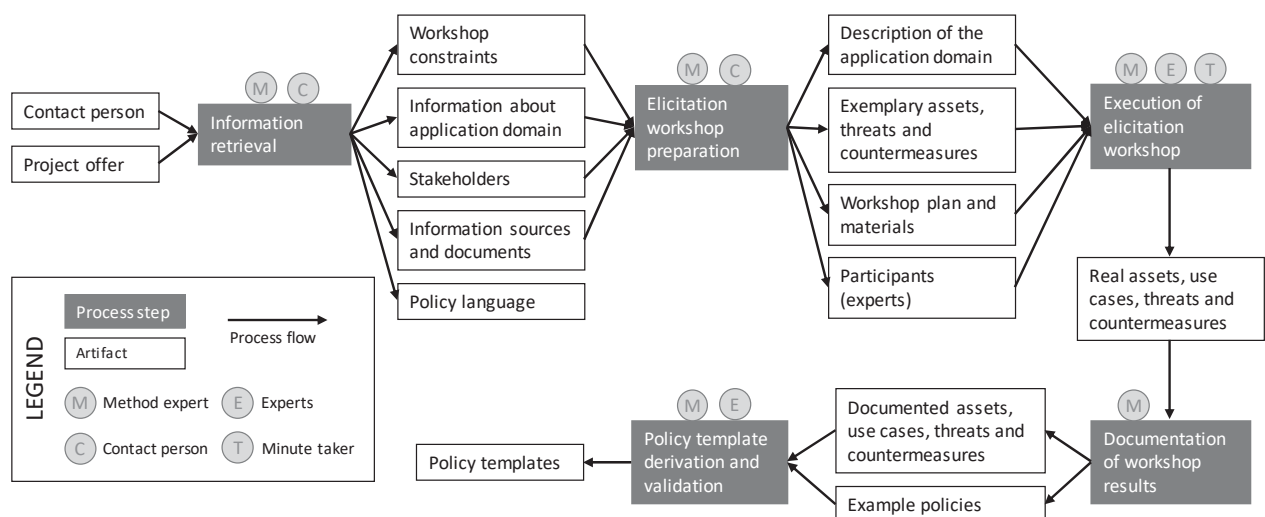


Figure 16: Policy Template Elicitation Method

The five steps are explained in more detail in the following sections. For each step, we explain the roles involved, the input and output of the step, the execution of the step and the RE techniques being used.



### 3.3 Step 1: Information Retrieval

The goal of the first step is the retrieval of information for the preparation of the elicitation workshop in Step 2.

#### *Roles Involved*

- The **method expert** executes the entire policy template elicitation method. In the first step, he collects available information about the application domain. To this end, he contacts a contact person from the application domain.
- The **contact person** is a domain expert from the application domain. He should have an in-depth knowledge of the application domain, including knowledge of typical use cases and relevant stakeholders. Multiple contact persons may exist.

#### *Input*

One mandatory input is the contact information of at least one **contact person**. Further optional inputs exist. For example, an official **project offer** or a similar document may exist that summarizes the key expectations of the customer about the elicitation workshop results or the creation of usable PAPs.

#### *Output*

The goal of the first phase is the retrieval of relevant information for the elicitation workshop. Therefore, the **constraints for the workshop** need to be identified and agreed with the contact person. The method expert collects **information about the application domain**. In addition, together with the contact person the method expert must create a list of relevant **stakeholders**. Moreover, the method expert identifies a list of other relevant **information sources and documents**. If a PAP shall produce ILPs for an existing security or privacy system within the application domain, the method expert needs to identify the used **policy language**.

#### *Process Description*

First, the method expert needs to obtain a basic understanding of the general goal of the elicitation. Therefore, he performs a document analysis on existing documents, such as the project offer, to discover the customer's goals and potential constraints for the elicitation.

Second, the method expert sets up the baseline for the elicitation workshop. To this end, the method expert elicits the following information from the contact person(s):

- **Information about the application domain:** The method expert lets the contact person explain the key characteristics of the application domain and an overview of assets, use cases, already known threats and countermeasures in place. In addition, an overview of the technical system where the assets are used and for which later on policies should be specified is provided. This includes the **policy language** used for policy enforcement in the technical system.
- **Workshop constraints:** The method expert clarifies constraints with respect to the elicitation workshop with the contact person such as the date and the duration of the workshop and the number of participants.
- **Stakeholders:** The method expert identifies relevant stakeholders of the application domain for participation in the workshop. This may include domain experts (who know assets and potential threats in the application domain), technology experts for the target system where the policies will be enforced, security and privacy experts (e.g., security officers), legal experts (who understand the applicable legal regulations), asset owners and typical users of the target system and the assets in the application domain. The contact person must deliver contact details for these stakeholders. Stakeholder description templates may be used for documentation.
- **Further information sources:** The contact person must provide existing documentation to the method expert for the preparation of the workshop. This includes relevant regulations, laws and guidelines as well as a technical description of the target system in which the PAP should be integrated and for which users shall specify policies.

Finally, the method expert documents the workshop baseline.

### ***Recommended Requirements Engineering Techniques***

We recommend that the method expert conducts a »semi-structured interview« with the contact person. This interview can be performed via a personal meeting or a phone call. If three or more contact persons exist, a short »workshop« should be conducted. If it is difficult to organize an interview or a workshop (due to limited availability of contact persons), a »questionnaire« could be prepared and sent to the contact persons. However, we have not elaborated such a questionnaire, yet. See Appendix A.1 for further elicitation techniques.

To extract information from existing documentation, the method expert can use the RE technique »document analysis«. Several RE techniques

exist for the structured documentation of the workshop baseline. Recommended formats are, for example, »goal trees«, »goal description templates« and »stakeholder description templates«. See Appendix A.2 for further documentation techniques.

### ***Example***

The method expert Mr. White receives the request for eliciting policy templates in the application domain of data protection for a mobile app in the area of financial advisory. The contact person of the customer is Mrs. Black. In an initial phone call, Mr. White and Mrs. Black discuss details about the information elicitation. Mrs. Black explains that the application domain is a mobile app with which financial advisors of a bank can access financial data of bank clients on business trips and in direct consultations at the client's home. Mr. White and Mrs. Black identify relevant stakeholders to be users of the app (bank clients and financial advisors), security, technology and domain experts of the bank and legal experts that understand the assets, which are subject to regulations of the BaFin (German Federal Financial Supervisory Authority). Mrs. Black provides a list of potential participants. Mrs. Black can organize a half-day workshop with one representative of each stakeholder role at the customer's place. She also notes that the instantiated policies shall be enforced in their mobile app and in their backend. Therefore, Mr. White and Mrs. Black agree on using the MYDATA policy language.

## **3.4 Step 2: Workshop Preparation**

The goal of the second method step is the preparation of the elicitation workshop in step 3. This includes the validation of documentation created by the method expert, yet.

### ***Roles Involved***

- The **method expert** prepares the elicitation workshop.
- The **contact person** reviews already documented goals and constraints for the workshop, exemplary assets, threats and countermeasures as well as other gathered information.

### ***Input***

The method expert uses the **constraints for the workshop** and the list of **stakeholders** for the workshop preparation. The method extracts information from the **description of the application domain** and other relevant **information sources and documents**.

## ***Output***

The method expert devises a high-level **description of the application domain** containing typical scenarios and use cases. In addition, he derives a list of **exemplary assets, threats and countermeasures** of the application domain from existing documentation for triggering workshop participants. The method expert creates a **workshop plan and material** including date, location, agenda with fixed time-slots, introductory slide show, workshop material for creativity methods and a list of **participants**.

## ***Process Description***

First, the method expert analyzes the existing documentation including notes from the interview with the contact person. Using this information, he devises a high-level description of the application domain containing typical scenarios and use cases. A description of typical scenarios within the application domain can narrow down the scope of the elicitation, and it helps the method expert to prepare the elicitation workshop.

Second, the expert identifies initial exemplary assets in the application domain. An asset can be a digital document containing sensitive information or any other file or resource that is valuable for at least one stakeholder in the application domain. As this asset has a value, others might be interested to steal, manipulate or destroy it, which is a threat for this asset. Countermeasures must be taken in order to prevent or mitigate the threat. A policy describes a security or privacy rule for applying a countermeasure to protect an asset against a threat. A list of exemplary assets, threats and countermeasures for the application domain may be used during the workshop to trigger introverted or uncreative participants.

Third, the method expert creates a workshop plan with respect to the constraints, which are, for example, limitations in duration of the workshop, the room in which the workshop takes place and the number and roles of participants. The method expert selects elicitation techniques for the different elicitation rounds (see Section 3.5) of the workshop and makes an agenda with fixed time-slots for each elicitation round. During the workshop, new assets and threats may come into the mind of participants in the second or third elicitation round, respectively. The method expert needs to decide how to integrate these additional information. Either, there is an integration session after each elicitation round or the workshop is planned iteratively so that the three rounds are executed multiple times until no further input is given by the participants.

Fourth, the method expert prepares the material necessary for the execution of the selected elicitation techniques and prepares an introductory slideshow, which explains the agenda and workshop process to the participants.

Fifth, the method expert selects the participants from the list of potential participants provided by the contact person. We recommend that the method expert selects one representative of each stakeholder role: domain expert, technology expert, security expert, asset owner and a typical user. In total, the number of workshop participants is recommended to be between five and ten [91].

Last, the method expert sends all material to the contact person for validation. The contact person verifies the correctness of the information provided in the material. Finally, the contact person invites the participants to the workshop.

### ***Recommended Requirements Engineering Techniques***

The method expert performs a document analysis based on available information sources. The contact person can use validation techniques for the information review. A description of different validation techniques can be found in Appendix A.3.

### ***Example***

The method expert Mr. White prepares the elicitation workshop. First, he concretizes the description of the application domain based on the information retrieved during the phone call with Mrs. Black. In addition, he picks some exemplary assets, threats and countermeasures of the application domain. The key assets are the financial data of bank clients. These must be protected in use cases inside the bank, on business trips and in consultations at the client's home. Mrs. Black named a potential attacker to be a hacker that wants to steal and sell information about high-value clients of public life such as politicians. Potential threats are the loss or theft of the mobile device or the accidental display of financial data of a wrong bank client. Exemplary countermeasures may be the automatic increase of security measures for the mobile device outside the bank (e.g., password-based screen lock) or the context-aware permission to access client data based on the current position of the mobile device (e.g., only access to data of client Mrs. Orange at the home of Mrs. Orange).

In addition, the method expert prepares the workshop agenda and material. Mr. White selects »brainstorming on cards« as the technique for asset as well as threat elicitation. He therefore prepares colored cards for the different information types (assets, data owners, policy authors, use cases and relevant regulations and laws, threats, attackers, existing documentation on threats, prioritization). He further selects the »6-3-5 method« for the countermeasure elicitation and prepares respective 6-3-5 sheets for 6 countermeasures per each of three top threats of one asset.

Mr. White sets the agenda to 20 minutes introduction, 1 hour per each elicitation round including discussion, two breaks of 10 minutes each between the elicitation rounds and a 20 minutes final discussion with feedback collection.

Finally, the method expert Mr. White selects the appropriate participants that Mrs. Black shall invite to the workshop.

### 3.5 Step 3: Execution of Elicitation Workshop

The goal of the third step is the elicitation of assets, use cases, threats and countermeasures from stakeholders of the application domain.

#### ***Roles Involved***

- The **method expert** moderates the elicitation workshop.
- The **participants** actively contribute to the elicitation workshop and provide information about the application domain.
- A **minute taker** documents all information revealed by the participants vocally or written on workshop material.

#### ***Input***

The method expert moderates the workshop according to the **plan** and with the **material** prepared in the previous step. In case of uncreative participants, the method expert can provide examples from the prepared list of exemplary **assets, threats and countermeasures** of the application domain. The invited **participants** attend the elicitation workshop.

#### ***Output***

The major outputs of the elicitation workshop are **real assets, use cases, threats and countermeasures** of the application domain and their relation among each other. The minute taker captures this information in a photo protocol and in the written documentation. We demonstrate examples of assets, threats and countermeasures in Figure 17.

#### ***Process Description***

The objective of the workshop is to elicit and document relevant assets (i.e., valuable domain objects to be secured), threats (i.e., intentional or unintentional actions harming security or privacy), and countermeasures (i.e., actions to prevent or mitigate threats). This information is required to derive policy templates.

We propose to elicit the information in three elicitation rounds using appropriate elicitation techniques. These elicitation rounds are also used in other method known from literature [30, 45, 46].

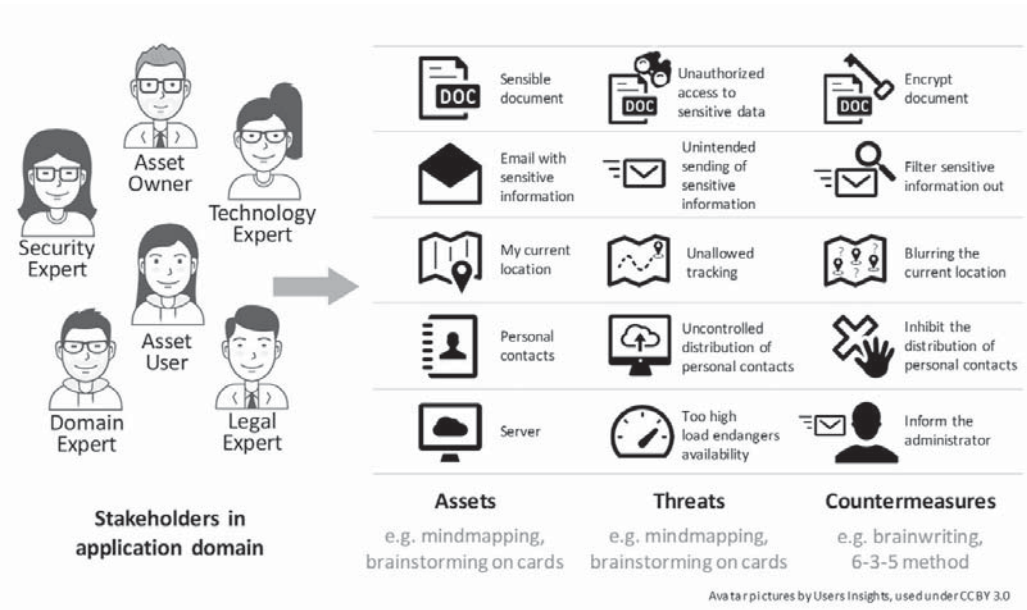


Figure 17: Examples of Elicited Assets, Threats and Countermeasures

**Round 1 – Assets and Use Cases:**

The first elicitation task is the identification of assets of the application domain and their properties. Properties include information about the owner, monetary value, and sensitivity of the assets, applicable laws and regulations as well as typical use cases and the users that want to use and to protect the assets (i.e. policy authors). The method expert asks all participants to share assets and their properties with the entire group of participants. The way of communication depends on the used elicitation technique. For example, if brainstorming on cards is applied, a participant writes an asset and properties on respective cards and explains them to the other participants before pinning the cards to a board. Discussions are welcome. However, the method expert needs to stop discussions that do not contribute to the goal of the workshop. The method expert clusters the cards on the board according to similar categories of assets or similar properties.

In the second elicitation round, threats for the assets are identified. Due to time constraints, it might be necessary to exclude some assets from further investigation. Therefore, all assets are prioritized after elicitation. We suggest using the two ordinal scales »monetary value of asset« and »sensitivity of asset« to support the prioritization:



- Monetary value of asset:
  - low (€)
  - medium (€€)
  - high (€€€)
- Sensitivity of asset:
  - public
  - internal use only
  - highly confidential

Each workshop participant has to estimate these properties. The median value of all votes is used. We use the top prioritized assets for the threat elicitation.

## **Round 2 – Threats:**

We want to identify relevant threats of the application domain. Threats can be elicited either per asset or per use case (if the use case has a list of relevant assets assigned). It is beneficial to elicit properties of each threat including relevant attackers and existing documentation (e.g., risk assessment documents). We propose to prioritize the threats as well. Therefore, after elicitation, all threats are prioritized using the two ordinal scales »severity of the potential damage caused by threat« and »probability of threat occurrence« to facilitate prioritization. Each workshop participant has to estimate the severity and the probability of each threat. The median value of all votes is used. We recommend combining these two properties into a single risk value according to literature [45, 153]. The method export should carefully consider whether the typical scale with the values high, medium and low is appropriate. Some users can hardly differentiate between the values and their meaning is up to interpretation [154]. Instead, more understandable and easily differentiable values per scale could be used, such as:

- Severity of the potential damage caused by threat:
  - irrelevant
  - costly
  - existence-threatening
- Probability of the threat occurrence:
  - almost impossible
  - likely
  - permanently



### **Round 3 – Countermeasures:**

In the final elicitation round, countermeasures for preventing, mitigating or at least detecting attacks are collected from the participants. Typically, multiple countermeasures exist for each threat. We know that this list of countermeasures is most probably incomplete and that suggested countermeasures may not sufficiently mitigate the threats. Thus, the method expert needs to assess and extend elicited information during documentation and policy template derivation.

### **The end of the workshop:**

The method expert asks for feedback, especially regarding the used RE techniques. This feedback can be used to build up an experience base with respect to the feasibility of the applied RE methods.

After the workshop, the method expert takes photos of all boards and workshop material. The results of the workshop are provided as a photo protocol. All workshop material is collected and archived.

### ***Recommended Requirements Engineering Techniques***

We propose to use »brainstorming on cards« or »mind mapping« as techniques for the identification and elicitation of assets. For the elicitation of threats, we recommend »brainstorming on cards«, »mind mapping«, »brainstorming paradox«, »6-3-5 method«, »change of perspective« or »attack trees«. All these techniques fit to the challenge of eliciting threats. However, we did not evaluate all of them. For the elicitation of countermeasures, we suggest to use a »brain writing« method (e.g., 6-3-5 Method), as the brain writing forms are well suited to efficiently collect a variety of countermeasures. We base our suggestions about elicitation techniques on our own experiences. However, the scientific literature also confirms the feasibility of the techniques »interview«, »brainstorming« and other »survey techniques« for the elicitation of corporate policies [53]. See Appendix A.1 for further information about the elicitation techniques.

For the prioritization of assets and threats, we recommend using the techniques »ranking« or »top-ten technique«. See Appendix A.4 for further information about the prioritization techniques.

### ***Example***

The method expert Mr. White conducts the elicitation workshop as planned. The participating stakeholders reveal a variety of real assets, use cases, threats and countermeasures. The minute taker Mrs. Red takes

photos of each pin board. The resulting photo of an asset on the pin board could look like Figure 18.

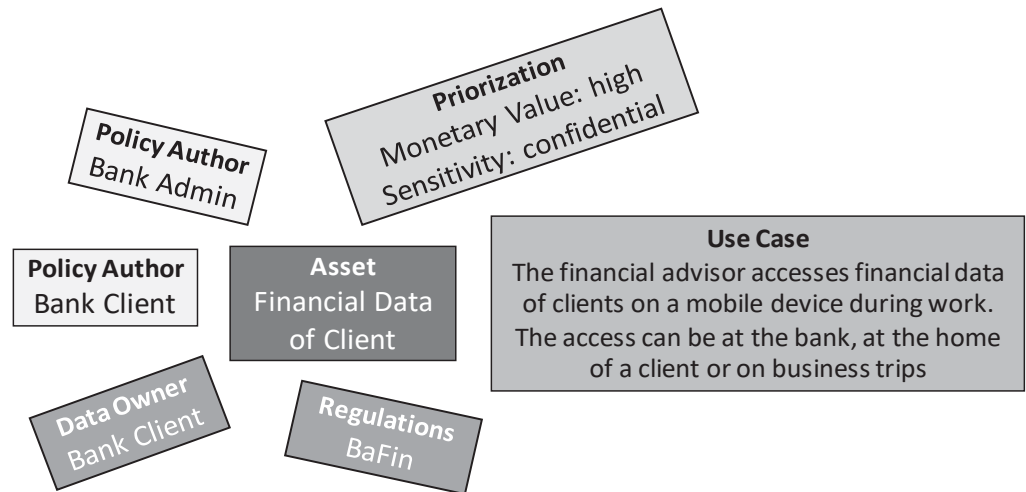


Figure 18: Exemplary Result of the Asset Elicitation

### 3.6 Step 4: Documentation of Workshop Results

The goal of the fourth step is the documentation of the elicited assets, use cases, threats and countermeasures. With the documented information, the method expert derives policy templates in the fifth and last step of the policy template elicitation method.

#### ***Roles Involved***

- The **method expert** documents the workshop results.

#### ***Input***

The method expert documents the information elicited in the workshop. The elicited **assets, use cases, threats and countermeasures** of the application domain from the workshop material, the photo protocol and the documentation from the minute taker are used.

#### ***Output***

The method expert produces **documented assets, use cases, threats and countermeasures** of the application domain. In addition, he combines the elicited information into **example policies**.

#### ***Process Description***

First, the method expert documents the assets, use cases, threats and countermeasures. We propose tabular templates for their documentation

as shown for assets in Table 2, for threats in Table 3 and for countermeasures in Table 4.

After documenting the assets, use cases, threats and countermeasures, the method expert combines the retrieved information into exemplary security and privacy policies. For each countermeasure, the method expert formulates an example policy. The method expert may rephrase the elicited information to harmonize terminology. An exemplary policy is: »If a financial advisor wants to access financial data of a client and is neither in the bank nor in an appointment at the client's home, access is prohibited and an error message is displayed.«

Table 2: Tabular Documentation of Assets

|                           |  |
|---------------------------|--|
| Asset ID                  | Identifier of the asset  |
| Asset                     | Name of the asset  |
| Data Owner                | Owner of the asset   |
| Example Use Case          | Exemplary use case that describes the use of the asset in the application domain   |
| Policy Authors            | Potential policy authors that would want to protect the asset  |
| Prioritization Properties | Prioritization with the two scales »monetary value of asset« and »sensitivity of asset«                                  |
| Legal Regulations         | Relevant legal laws and regulations that need to be considered for the creation of example policies and policy templates |

Table 3: Tabular Documentation of Threats

|                        |   |
|------------------------|---|
| Threat ID              | Identifier of the threat  |
| Related Use Case       | Use case that is affected by the threat   |
| Related Asset          | Asset that is affected by the threat  |
| Attackers              | Person causing the threat   |
| Threat                 | Natural language description of the threat including the prioritization scales »probability« and »damage« |
| Existing Documentation | References to existing documentation about the threat   |

Table 4: Tabular Documentation of Countermeasures

|   |
|---|
| Countermeasures for threat:<br>T1 Data theft of financial data after stealing mobile device |
| Countermeasure description in natural language.   |
| Countermeasure description in natural language.   |
| Countermeasure description in natural language.   |
| Countermeasure description in natural language.   |

## Example

The method expert Mr. White documents the elicited assets, use cases, threats and countermeasures using the proposed templates. Table 5 shows an example of an identified asset.

Table 5: Exemplary Documented Asset

|                           |   |
|---------------------------|---|
| Asset ID                  | A1  |
| Asset                     | Financial data of client  |
| Data Owner                | Client  |
| Example Use Case          | The financial advisor accesses financial data of clients on a mobile device during work. The access can be at the bank, at the home of a client or on business trips. |
| Policy Authors            | Bank administrator  |
| Prioritization Properties | Monetary value of asset: high (€€)<br>Sensitivity of asset: confidential  |
| Legal Regulations         | Regulations of BaFin  |

Table 6 shows an exemplary threat for the asset in Table 5.

Table 6: Exemplary Documented Threat

|                        |  |
|------------------------|--|
| Threat ID              | T1   |
| Related Use Case       | UC1: Financial advisor works outside the bank  |
| Related Asset          | A1: Financial data of client   |
| Attackers              | Data thief   |
| Threat                 | Data theft of financial data after stealing mobile device <ul style="list-style-type: none"> <li>probability: likely (medium)</li> <li>damage: existence-threatening (high)</li> </ul> |
| Existing Documentation | not available  |

Table 7 shows an excerpt of countermeasures identified for the threat in Table 6.

Table 7: Exemplary Documented Countermeasures for a Threat

|   |
|---|
| Countermeasures for threat:<br>T1 Data theft of financial data after stealing mobile device |
| Allow access to client's financial data only in bank or at client's home.                   |
| Deny access to client's financial data on business trips.                                   |
| Let client authenticate before access on financial data outside the bank.                   |
| Inform supervisor on denied access request on financial data.                               |

Mr. White derives example policies from the elicited assets, use cases, threats and countermeasures, such as:

- If the financial advisor is about to access financial data of client Mrs. Orange at the home of Mrs. Orange, access is granted.
- If the financial advisor is about to access financial data of client Mrs. Orange at the home of Mrs. Orange, Mrs. Orange needs to enter her PIN before access is granted.
- If the financial advisor is about to access financial data of client Mrs. Orange on the business trip, access is denied.
- If the access or the financial advisor to client data is inhibited, inform the supervisor of the financial advisor about this access attempt.

### 3.7 Step 5: Policy Template Derivation and Validation

The goal of the fifth and final step of the policy template elicitation method is to derive and validate policy templates.

#### ***Roles Involved***

- The **method expert** derives policy templates
- **Experts** from the application domain validate the policy templates

#### ***Input***

The method experts uses the **documented assets, use cases, threats, countermeasures and example policies** for deriving policy templates.

#### ***Output***

The final output of the method is a list of **policy templates**.

#### ***Process Description***

First, the method expert derives policy templates. To this end, he uses the assets, use cases, threats, countermeasures and example policies elicited and elaborated in the previous steps. Example policies with identical meanings must be unified, and overlapping policies can be generalized to initial template prototypes. After this derivation step, multiple templates may implement the same countermeasure or protect the same asset. Next, the method expert refines the templates by adding branches and parameters. A parameter is a variable part of the template that is assigned during instantiation. Parameter types can be text (e.g., an email address for notifications), numbers (e.g., number of letters in a password), or a predefined list of values (e.g., active directory entries). A branch is a selectable part of the policy template. The method expert defines rules for the selection of the branches (How many branches may be selected? Are

selected branches combined with an »AND« or an »OR« as the conjunction?). Eventually, each documented example policy must be instantiable by using one of the derived policy templates. The method expert extends the policy templates with respect to additional reasonable countermeasures. He identifies those countermeasures mentioned in existing documentation (e.g., risk assessment documentation or relevant guidelines for security and privacy). Additionally, the method expert adds countermeasures known from his experience in this task. Finally, the method experts validates whether all example policies can be instantiated from the derived policy templates. Table 8 shows the structure of a policy template.

Table 8: Tabular Notation of a Policy Template

| ID                     | Policy Template Name        | Asset   | Target System                                     | Policy Author   |
|------------------------|-----------------------------|---|---|---|
| ID                     | The name of policy template | The asset for which policies can be instantiated  | The system on which the policies will be enforced | The users, which will use the policy template for policy instantiation in a PAP |
| Policy Template Syntax |                             | The syntax of the policy template described with the policy template notation format          |   |   |
| Description            |                             | Natural language description of the policy template   |   |   |
| Threat                 |                             | Related threat(s) for the asset that can be mitigated or prevented with instantiated policies |   |   |
| Security/Privacy Goals |                             | Relevant security and privacy goals   |   |   |
| Example Instantiation  |                             | Exemplary policy instantiated from this policy template                                       |   |   |

Second, experts from the application domain review the policy templates with respect to quality characteristics such as correctness and completeness. The validation of the security policy templates is a manual task. The method expert asks the participants of the workshop to confirm correctness and completeness of the derived policy templates and the terminology used in the templates. Especially the completeness should be confirmed or supported by the stakeholders of the application domain as the derivation of policy templates from the example policies can be incomplete. In case of mistakes, inconsistencies or missing information, the method expert consolidates the reviews and adjusts the policy templates accordingly. Next, the method expert asks the experts to validate them again. This iteration ends when all policy templates are considered correct and complete.

### ***Policy Template Notation Format for Specification-level Policies***

A policy template is a blueprint of a security or privacy policy that is not completely instantiated. PAPs provide policy templates for instantiation.

Thus, during the specification, a user fills in the variable parts of a policy template in such a way that a complete policy results.

We developed a notation format with a simple grammar for specification-level policy templates. The method expert can use this notation format to describe specification-level policy templates in documents. The format is composed of natural language statements that we can concatenate with the following grammatical elements:

- We write natural language statements in plain text. An example is:
  - »The deliverer may not access my complete address.«
- Variables are surrounded by angle brackets. The name of the variable is written between the angle brackets. Variables with the same name are bound, that is, they are two instances of the same variable. By default, a variable can contain any text value like a String variable in Java. If other variable types are required, the type can be defined after the variable name with a leading colon symbol. Available types are string, integer, float, boolean and date. The values for the two number types integer and float can be further restricted by a minimum and a maximum value notated with a comma-separated list within parentheses. The »\*« symbol expresses an infinite maximum value. An example is:
  - »<actor:string> may not see my complete address and <actor:string> may get access to the first <phoneDigits:integer> digits of my phone number and to the first <creditCardDigits:integer(2,14)> digits of my credit card number.«
- We denote selectable texts by surrounding square brackets and separate selectable items by pipe symbols. The selectable items may only contain text. An examples is:
  - »The deliverer [may not see my complete address | may get access to my phone number].«
- We denote selections by surrounding square brackets and the trailing notation of the conjunction and quantifiers. The selectable items are separated by pipe symbols. If no conjunction and no quantifier is specified, the selection is exclusive by default. The selection is limited to one instance of each element. The conjunction can be specified as the first value within parentheses after the closing square bracket. The available conjunctions are »AND« and »OR«. The second and third value, separated by commas, specify the minimum and maximum allowed number of selected items. For simplification, we added the two quantifiers »+« (at least one selection item) and »\*« (any number of selection item) that can replace the minimum or maximum value and the conjunction, which is set to »AND«. Examples are:

- »The deliverer [may not see my complete address|may get access to my phone number|may get access to my credit card number](AND,1,2).«
- »The deliverer [may not see my complete address|may get access to my phone number|may get access to my credit card number]+.«
- »<actor:string> is informed if somebody accesses [contact details | file <filename:string>].«

In addition to the notation format, we enriched the policy template with additional information. We give each policy template a unique identifier and a descriptive name. Each template references an asset, a target system, and security and privacy goals. A policy template structure is shown in Table 8.

### Example

The method expert defines policy templates based on the elicited information and the derived example policies. For the specification of the policy template syntax, Mr. White combines similar example policies and transforms the diverging parts of those example policies as variable parts in the policy template. We present an exemplary policy template in Table 9.

Table 9: Exemplary Policy Template

| ID                     | Policy Template Name                             | Asset  | Target System       | Policy Author                     |
|------------------------|--|--|---------------------|-----------------------------------|
| PT1                    | Access to financial data in different situations | Financial data of bank client  | Mobile advisory app | Bank administrator or bank client |
| Policy Template Syntax |  | If the financial advisor is about to access financial data of <client:string> [inside the bank on a business trip at the home of <client:string>], then [allow access allow access after successful authentication by <client:string> inhibit access]. |                     |                                   |
| Description            |  | The access of financial advisors to financial data of bank clients need to be restricted in different situations for different clients.  |                     |                                   |
| Threat                 |  | Unintended access to financial data of bank clients  |                     |                                   |
| Security/Privacy Goals |  | Confidentiality  |                     |                                   |
| Example Instantiation  |  | If the financial advisor is about to access financial data of client Mrs. Orange at the home of Mrs. Orange, allow access.   |                     |                                   |

## 3.8 Summary and Conclusion

In this chapter, we presented the policy template elicitation method. Below, we briefly address the fulfillment of the requirements for the policy template elicitation method (as stated in Section 3.1):



- **Req\_Elicitation\_Derivation-of-Policy-Templates:** Our method uses information from stakeholders of an application domain elicited in a workshop for deriving actual policy templates. The policy templates limit the expressiveness of the policy specification if provided in a PAP.
- **Req\_Elicitation\_Application-Domain:** As the method builds upon an elicitation workshop with representative stakeholders from the application domain, we are confident that the resulting policy templates reflect the security and privacy demands of the application domain.
- **Req\_Elicitation\_Understandable-Terminology:** We elicit assets, threats and countermeasures from stakeholders of the application domain in a workshop. Therefore, all elicited information is formulated by the stakeholders in the terminology that is typically used in the application domain. Thus, if the method expert carefully avoids changing the terminology during the derivation of policy templates, the templates reflect the domain-specific terminology. Moreover, stakeholders are requested to validate the final templates with respect to terminology.

Overall, we created a method for deriving policy templates from information elicited in an application domain by representative stakeholders. The elicited information contains assets, use cases, threats and countermeasures typical for the application domain. The resulting policy templates can be used for tailoring a PAP to the application domain. A user can instantiate policy templates in order to express his personal security and privacy demands. We elicit the information in a workshop. We partially use established RE techniques for elicitation, documentation, prioritization and validation of information.

## 4 Policy Template Model

The security and privacy demands of different users in an application domain are often very diverse. Users want to use PAPs to specify and enforce policies that express their individual security and privacy demands when using a system. The method expert must therefore collect the security needs of users from an application domain and turn them into configurable security policies.

To formalize the security and privacy demands of an application domain, an appropriate model for the specification of policy templates is required. Our policy template model, which represents Contribution 3 of this thesis (see Section 1.4), constitutes the foundation for the instantiation of security and privacy policies. The idea behind our policy model is to describe real world security and privacy demands in the form of threats and corresponding countermeasures within an application domain and to derive security policy templates from the countermeasures. Policy templates are specified in a human-understandable format, on the specification level. A user can instantiate such a policy template as a human-understandable policy, that is, a specification-level policy (SLP). In addition, we want to support the transformation of an SLP into a machine-understandable representation of this policy, that is, an implementation-level policy (ILP).

In addition, we want to support different specification paradigms for the user interfaces for policy specification in PAPs. All specification paradigms differ in their expressiveness with which the user can specify policies and their guidance the user receives during the specification process. In our approach, all supported specification paradigms are based on different presentations of and interactions with the policy templates. Thus, users can instantiate policies from policy templates or select from already instantiated policies.

In summary, an instance of the policy template model must contain all information that is necessary for generating policy specification interfaces in PAPs with multiple supported specification paradigms and for supporting the generation of ILPs. We call the instance of a policy model that is used in a PAP a policy vocabulary:

**Definition: Policy Vocabulary**

A policy vocabulary is a configuration for the PAP generation framework that is based on an instance of the policy template model.

This chapter describes the policy template model, which is the link between the policy template elicitation method and the generation of policy specification interfaces in PAPs with multiple supported specification paradigms. We divide the policy template model into several sub-models, which are interwoven, each serving a specific purpose. We explain our research approach in Section 4.1. Section 4.2 gives an overview of the sub-models. Sections 4.3 to 4.8 present the individual sub-models in detail. We provide an example in Section 4.9. In Section 4.10, we summarize and conclude the chapter.

### 4.1 Research Approach

We developed the policy template model in an iterative way. First, we elicited key requirements and created the initial version of the policy template model. We supported the specification of security policy templates on the specification level. In addition, we enabled the definition of generation rules for generating ILPs from instantiated policy templates. We applied this first version in the industrial case study »SINNODIUM«.

In the second version, we extended the policy template model by an application domain model that describes the relation between entities in the application domain and policy templates. Threats and countermeasures act as intermediate elements. In addition, we added support for the specification paradigm »default policies« by providing model elements for the specification of pre-defined policy template instantiations. Moreover, we improved the definition of transformation rules for the ILP generation.

Finally, we devised version 3 of our model, which additionally provides two additional specification paradigms: »security levels« and »wizard«.

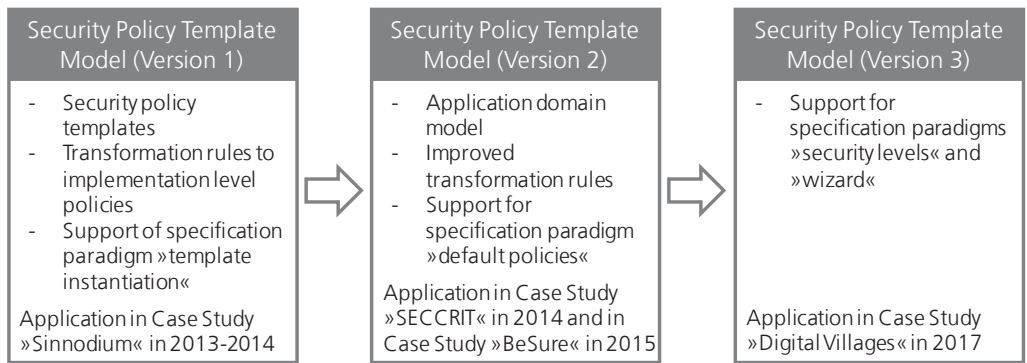


Figure 19: Research Approach for the Policy Template Model

We derived several requirements from the state of the art that our policy template model must meet:

- **Req\_Model\_Domain-Independence:** A key requirement for our model is domain independence. That is, the model should be applicable in any application domain. As security and privacy demands of stakeholders and therefore requirements for security and privacy solutions differ in different application domains, the model must not be limited to a specific domain. The policy templates and the supported policy language for technically enforcing ILPs must be exchangeable. The domain model must cover all entities and relationships that are relevant for the derivation of policy templates. This includes, but is not limited to organizational structures, use cases, assets, threats and countermeasures. The ability to address multiple application domains is essential to limit the expressiveness of the policy templates as required by Johnson et al. [25].
- **Req\_Model\_Understandable\_Templates:** The model must support the specification of human-understandable policy templates (SLP templates) as required in the literature [10, 33]. We want to support policy templates on the specification level that is written in natural language.
- **Req\_ILP\_Generation:** The model must support the specification of generation rules for the generation of an ILP from an SLP instantiated from a policy template. Kumari [76] states that ideally, policies should be specified on the specification level and then be transformed into the implementation level specification. Therefore, we support the transformation of SLPs into ILPs. However, we only support the manual specification of transformation rules. Research into more sophisticated and automated ILP transformation is not part of this thesis, but several approaches have been described in the literature (e.g., [75, 76]).
- **Req\_Model\_Specification-Paradigm-Projection:** The model must support the definition of projection rules for representing the policy templates with different specification paradigms on the policy specification interface of a PAP. This requirement stems from our ambition to support multiple specification paradigms in PAPs, as described in contribution C1.

## 4.2 Overview of Policy Template Model

The policy template model bridges the gap between an application domain and the technical implementation of a PAP. It explains the relationships between entities and their actions in the application domain, assets, corresponding threats and countermeasures. The method expert can instantiate the policy template model to describe the relevant policy templates for an application domain and to define necessary information for the automated generation of policy specification interfaces in PAPs with multiple supported specification paradigms.

Our model consists of six connected sub-models. We first give an overview of the sub-models and then describe each of the sub-models in detail in subsequent sections. We define the following six sub-models:

- The **domain sub-model** can be used to describe the application domain in which the PAP is about to be applied. The model describes the relevant relations among domain objects. An instance of this model highlights relevant stakeholders and other entities in this application domain that perform actions on assets that need to be protected. Further details can be found in Section 4.3.
- The **security and privacy sub-model** describes how threats and countermeasures relate to actions from the domain sub-model and to the policies from the template sub-model. More details are presented in Section 4.4.
- The **template sub-model** describes the relationship between the policy templates and concrete policy instances on the two abstraction levels, the specification level and the implementation level. The specification level reflects natural language descriptions of the policies whereas the implementation level considers machine-understandable representations of the policies, for example in XML notation. We present the template sub-model in Section 4.5.
- The **specification-level template sub-model** allows the method expert to create policy templates on the specification level with several template elements. The aim is to provide policy templates in natural language. The policies resulting from instantiated templates describe concrete countermeasures for preventing or mitigating a threat on an asset. However, they lack information about how this countermeasure is technically enforced. Further details can be found in Section 4.6.
- The **implementation-level template sub-model** allows the method expert to create policy templates on the implementation level in a machine-understandable format. In addition to the specification-level policy templates, information about the technical enforcement of the resulting policies is included. The instantiation of an implementation-level policy template is linked to the instantiation of the corresponding specification-level policy template. Therefore, transformation rules can be specified by the method expert. This means that when the user creates a specification-level policy, a corresponding implementation-level policy is automatically generated. The resulting policies describe how the system enforces the security demand, but most users will probably not understand this representation of the policy. More details are presented in Section 4.7.
- Using a PAP, users can specify a set of security policies in many different ways. We call these different approaches specification paradigms. The **specification paradigm projection sub-model**

describes how the different specification paradigms are linked with the specification-level policy templates. The sub-model contains all information necessary for the representation of the policy templates in policy specification interfaces of the PAP with multiple supported specification paradigms. We present the specification paradigm projection in Section 4.8.

We show an overview of the sub-models and their dependencies in Figure 20. All model diagrams use UML syntax, and we created them with Enterprise Architect.

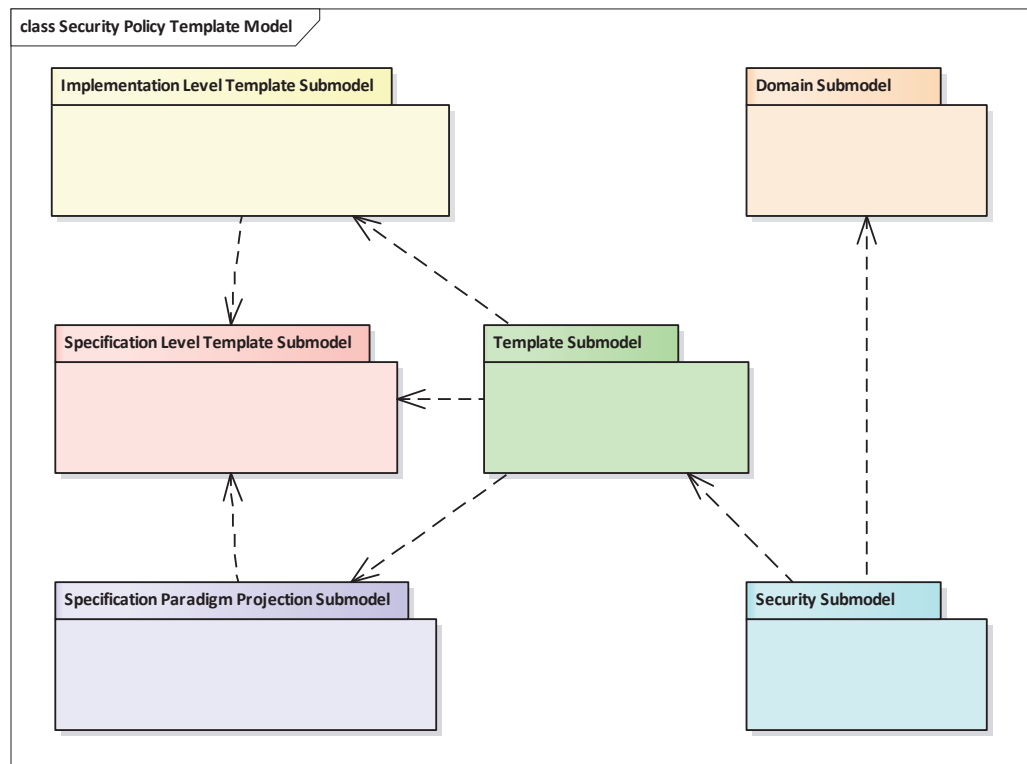


Figure 20: Policy Template Model

The method expert instantiates the security policy template as a policy vocabulary as part of the method for usable PAP creation. Details can be found in Section 7.4.

### 4.3 Domain Sub-model

For systematically collecting and understanding the threats on assets and potential countermeasures within an application domain, we must first identify the relevant elements in this domain. Therefore, the domain sub-model primarily describes entities in the application domain and their actions that affect assets. The domain model was designed to be very generic so that it can describe multiple application domains.

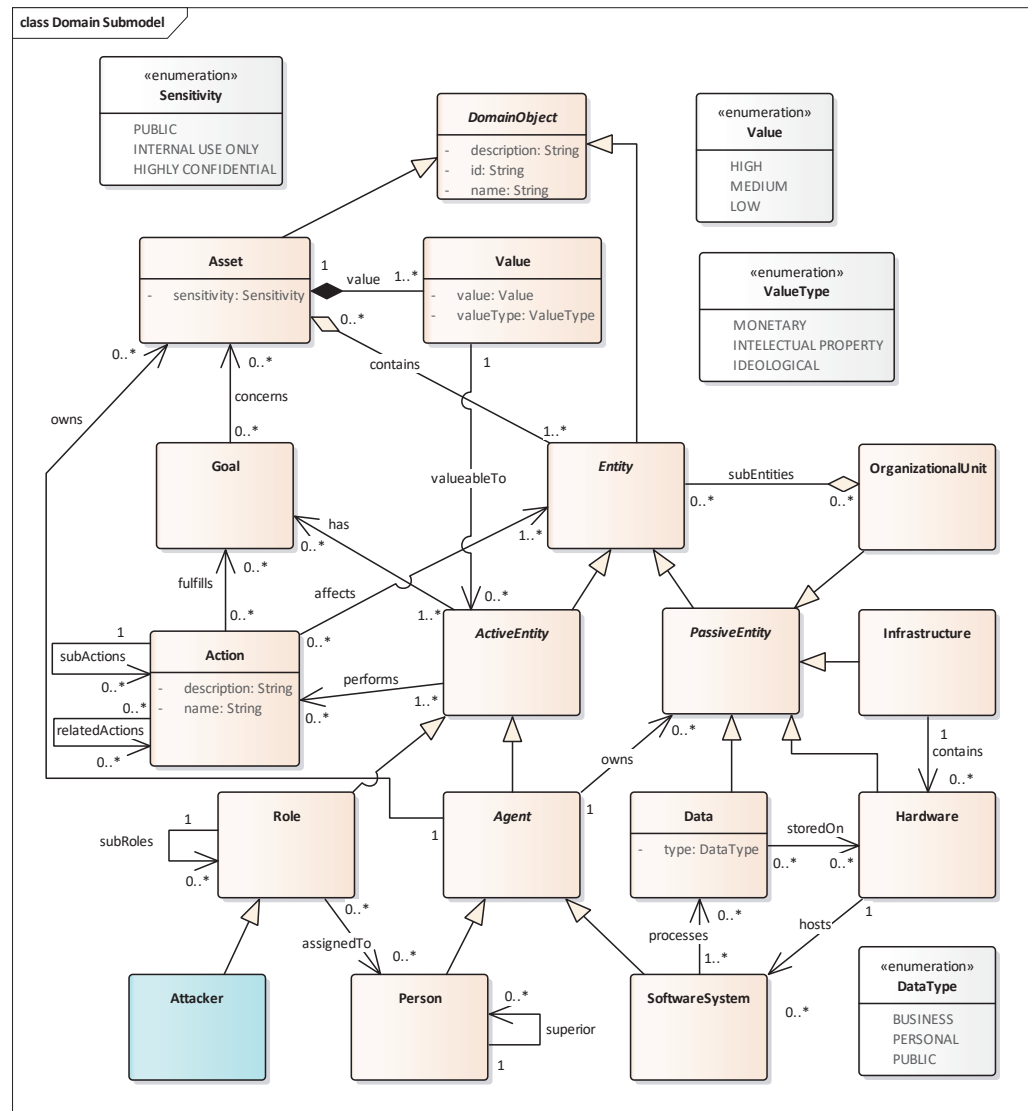


Figure 21: Domain Sub-model

Figure 21 depicts the domain model. The domain model describes entities, their relations and the actions they perform. A domain can have one or more entities, which can be either active or passive. Active entities have one or more intrinsic goals they want to achieve. Therefore, they can trigger multiple actions affecting different active or passive entities to fulfill certain goals. Active entities can be persons or software systems, both are generalized to agents. Agents can own assets and other passive entities in the application domain. Roles can be assigned to persons, which means that a role can also act as an active entity and perform actions in the application domain. A very special role is the attacker, which is the generic representation for a person performing a malicious action that represents a threat. Passive entities are those entities that do not trigger actions by themselves, but that can be affected by actions. Examples are organizational units, infrastructural objects, hardware or data. Data is stored on hardware systems and processed by software that runs on

hardware, which in turn is part of a certain infrastructure (e.g., a building). Types of Data include personal data, business data and public data.

All entities can be part of an asset. An asset is in addition of value to a particular active entity. Therefore, assets have a particular sensitivity ranging from public to highly confidential and a monetary value that is measurable in a currency or can have an ideational value to an active entity. As it is very hard to determine concrete numbers for monetary or ideational values, we rate asset values on a three step ordinal scale from low over medium up to high value. Both sensitivity and value have an impact on the diligence in which threats to an asset need to be elicited and prevented or mitigated.

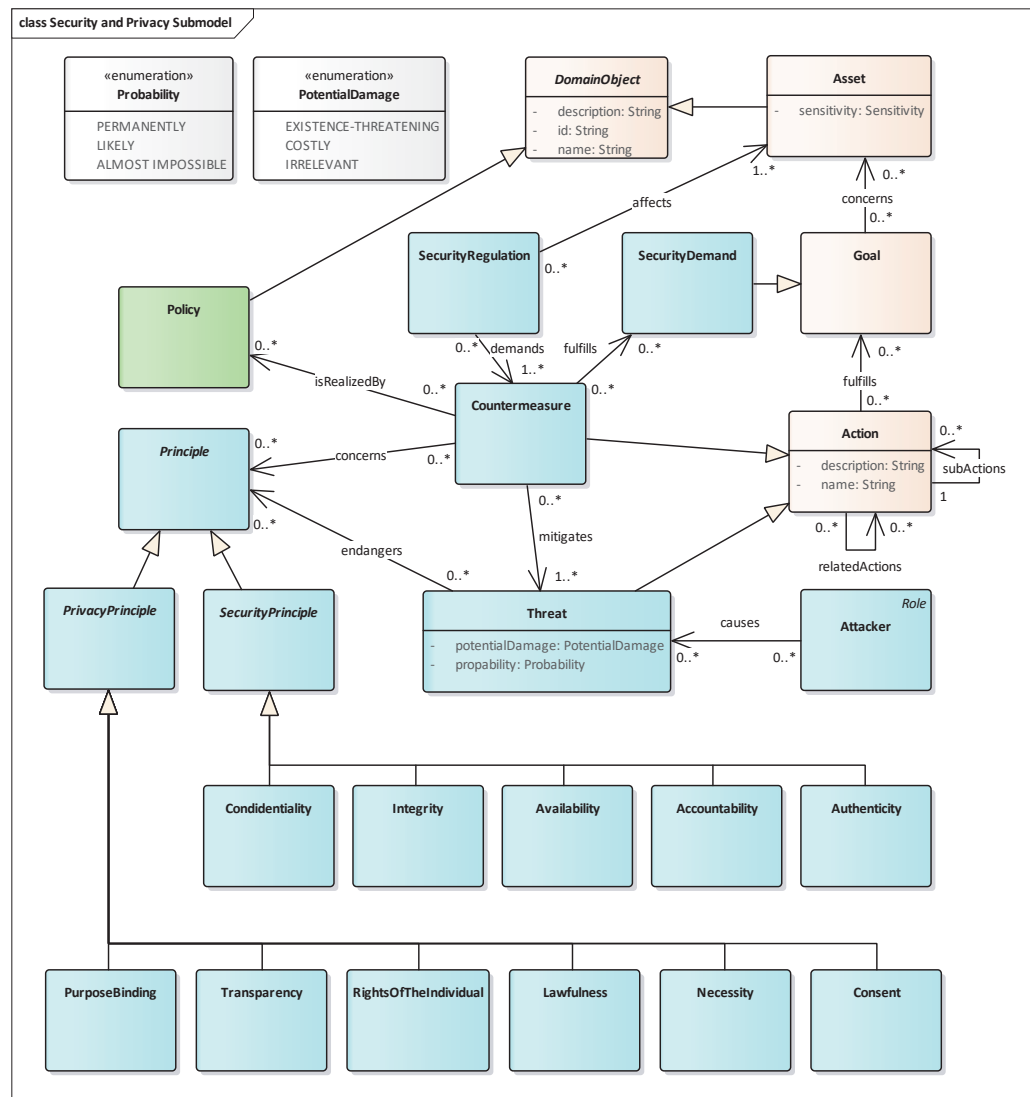


Figure 22: Security and Privacy Sub-model



## 4.4 Security and Privacy Sub-model

The security and privacy sub-model describes the threats that can occur in the application domain and the corresponding countermeasures. We present the sub-model in Figure 22.

Threats are a specialization of actions from the domain model; actually, they denote malicious actions performed by an attacker. The attacker can be, for example, a malicious hacker, a script kiddie or an intentional or accidental attacker from inside the organization. Threats can occur with a certain probability and cause a certain amount of damage. Since both parameters are difficult to determine exactly, we have opted for a 3-point value scales (severity of the potential damage: irrelevant – costly – existence-threatening; probability of the threat: almost impossible – likely – permanently). Threats must be prevented or at least mitigated with suitable countermeasures. With the rating on those scales, the method expert in cooperation with security experts can assess the urgency of the technical implementation of the countermeasure for each threat during the policy elicitation phase.

Countermeasures are also a specialization of actions in the application domain. Threats jeopardize basic security and privacy principles, which can be protected with countermeasures. We use the IT security principles according to the ISO 27000 standard [155], which are confidentiality, integrity, availability, accountability and authenticity. As privacy principles, we use the purpose binding, transparency, rights of the individual, lawfulness, necessity and consent by design from the ENISA report »Security and Data Protection by Design« [156].

## 4.5 Template Sub-model

The template sub-model describes the relationship between the two levels of abstraction of a security policy and between security policy templates and concrete security policies. We support users in expressing their own security and privacy demands. We have recognized in our own project experience and in reviewing the state of the art that security needs differ from user to user. This demands the possibility of individual customization of the security policies. Templates of policy are required to enable customization. To this end, we elicit policy templates for the application domain with the policy template elicitation method.

Kumari and Pretschner [157] and Neisse et al. [158] distinguish two levels of abstraction for policies: specification-level policies (SLPs) and implementation-level policies (ILPs). SLPs describe security demands in a format that is easy to understand for non-experts. We defined SLPs to be specified by a human and chose natural language as the specification-level

format. The focus of the specification-level is clearly on the explanation of the security and privacy requirements of a user in the application domain. An SLP might therefore lack details about the implementation of the user demand. An ILP describes the concrete technical implementation of the demands so that it is interpretable and enforceable by the security or privacy system. Thus, the policy must exist in a format that can be executed by a machine, such as a policy language in XML notation. To transform an SLP into an ILP, concrete transformation rules are required. These rules must contain the information for the technical enforcement of the policy, which is missing on the abstract specification-level. We have opted for individual transformation rules specified by the method expert.

We define policy templates in an instance of the policy template model, both on the specification level and on the implementation level. As described above, the policy templates also need to be specified on two abstraction levels. Thus, for each specification-level policy template (SLPT), we also define an implementation-level policy template (ILPT). A user can instantiate a specification-level policy template as a concrete SLP in natural language. However, the security system requires an equivalent machine-understandable ILP for enforcement. Therefore, the expert refines the SLPT into an ILPT and links the ILPT to the corresponding SLPT. This linkage facilitates the transformation of a specification-level policy into an implementation-level policy based on the respective templates. Figure 23 illustrates the relationship between templates and policies on the different abstraction levels.

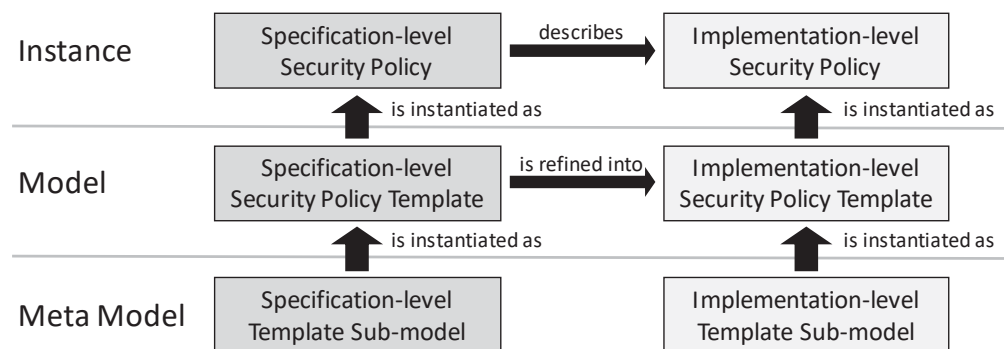


Figure 23: Meta Model - Model - Instance

Generic transformation rules or automated transformations are not in the focus of this work. A more sophisticated approach for the derivation of implementation level policies, such as the one proposed by Kumari [76, 157, 159], could improve the model and the elicitation method (see Chapter 3).

Details of both abstraction levels of policy templates and the transformation rules are described in the subsequent sections. In Section 4.6, the sub-model for creating specification-level policy templates

is presented. Section 4.7 explains the creation of implementation-level policy templates and their linkage to the specification-level.

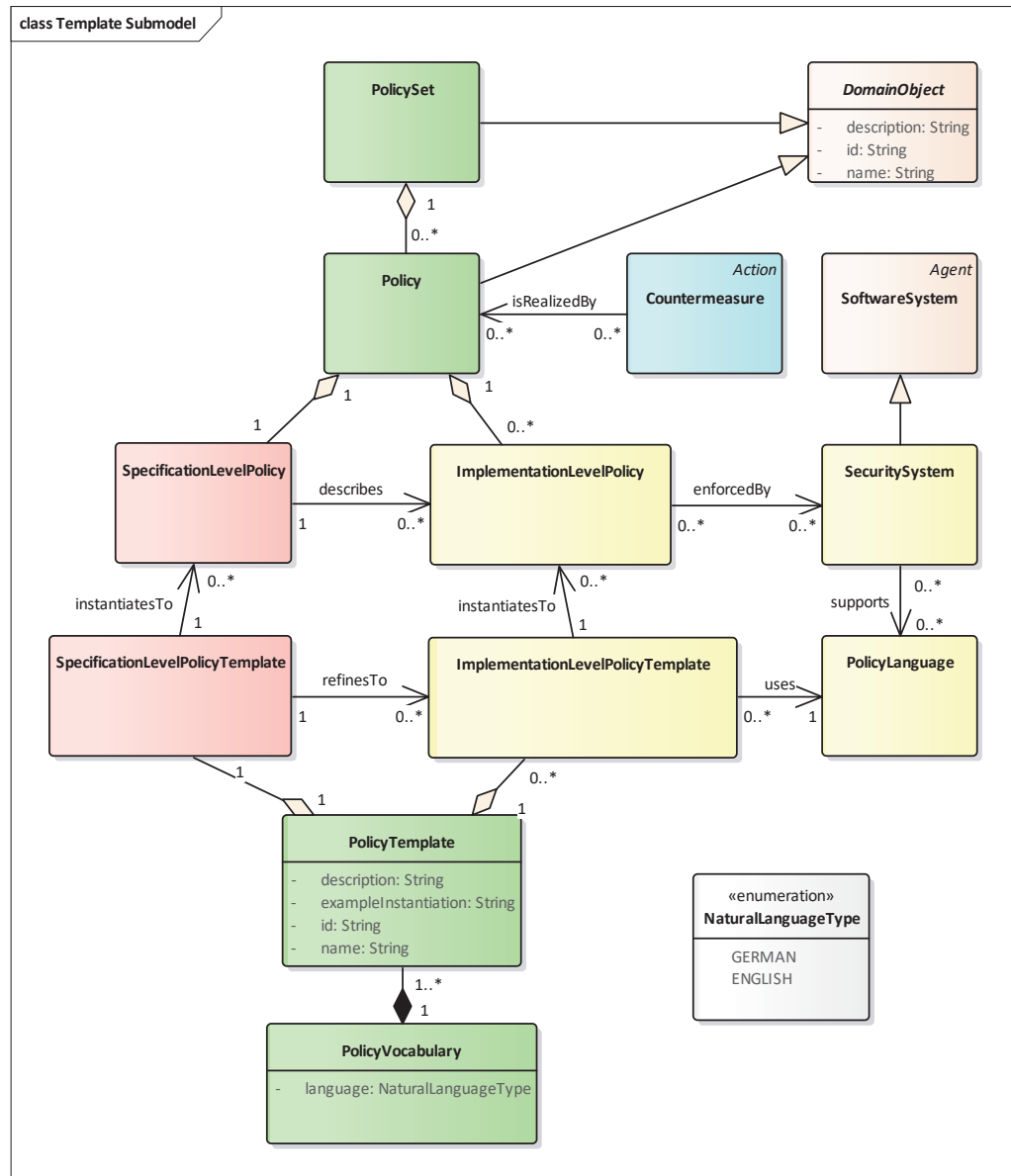


Figure 24: Template Sub-model

Figure 24 depicts our template sub-model. A policy in our model is part of a policy set and realizes a countermeasure of the security and privacy sub-model. Each policy contains exactly one SLP, which describes the countermeasure in natural language. In addition, multiple ILPs can exist as part of a policy. A security system (implying the privacy functionality) in our model is the software component that executes ILPs in order to enforce countermeasures technically. Different security systems require different machine-understandable notation, such as the policy languages of IND<sup>2</sup>UCE [110] or XACML [5]. Thus, if multiple security systems shall be supported, multiple equivalent ILPs must be specified. Each SLP is an instantiation of a specification-level policy template, and each ILP is an

instantiation of an implementation-level policy template. An ILPT uses one specific policy language to formulate machine-understandable instructions that represent the natural language description on the specification-level. A policy template in our model is a container for exactly one SLPT and multiple ILPTs refining the SLPT. A policy vocabulary is a container for multiple policy templates within an application domain. We also use this term for the complete instantiation of a policy template model, because the policy vocabulary is the root element of the policy template model. All specification-level policies within a policy vocabulary must use the same language (e.g., English or German).

## 4.6 Specification-Level Template Sub-model

The specification-level template sub-model (see Figure 25) refines the SLPT from the template sub-model. An SLPT consists of multiple elements, which we call SLPT elements (Specification-Level Policy Template Elements). The SLPT elements are building blocks for constructing policy templates. The method expert can use them to define the specification boundaries in which users can instantiate the templates to create policies that meet their individual security and privacy demands. We have defined the following SLPT elements, which reflect the elements of the policy template notation format introduced in Section 3.7:

- **Text:** This element corresponds to a text block that is to explain part of the policy template to the user. The text is defined by the method expert and cannot be changed by the user. In addition, text elements can be used to complement the remaining elements in such a way that complete and comprehensible natural language sentences are created. In the user interface, this element can be realized as a text viewer.
- **Selection:** This element offers the user a selection of different paths in the template. Each path is represented by an element group. In an element group, all SLPT elements can be used to refine this path. We limit the instantiation options so that a user may only select each element group once per instantiation. The method expert must specify the minimum and maximum number of paths that the user may select when instantiating the template. The selection is optional if no path has to be selected. The selection is mandatory if at least one path has to be selected. The selection is exclusive if exactly one path has to be selected. In the user interface, exclusive selections can be implemented as radio buttons and other selections as check boxes.
- **Variable:** This element allows text input by the user. This allows the user to individualize the template during instantiation. The method expert can typify the variable to restrict user input. Possible types are full text (string), integers, floating point numbers (float) and date/time specifications. For numeric input types, a minimum value and a

maximum value can be defined. On the graphical user interface, this element can be implemented as an input field.

- **Selectable text:** This element is a specialization of the variable. The method expert provides a list of possible variable values (**variable choices**) from which the user must choose the appropriate one. Note that a variable choice contains both a natural language value and a machine-understandable equivalent. For example, a user name can be displayed on the user interface, but an appropriate user id can be used at the implementation level. This element can be implemented as a drop-down box on the graphical user interface.

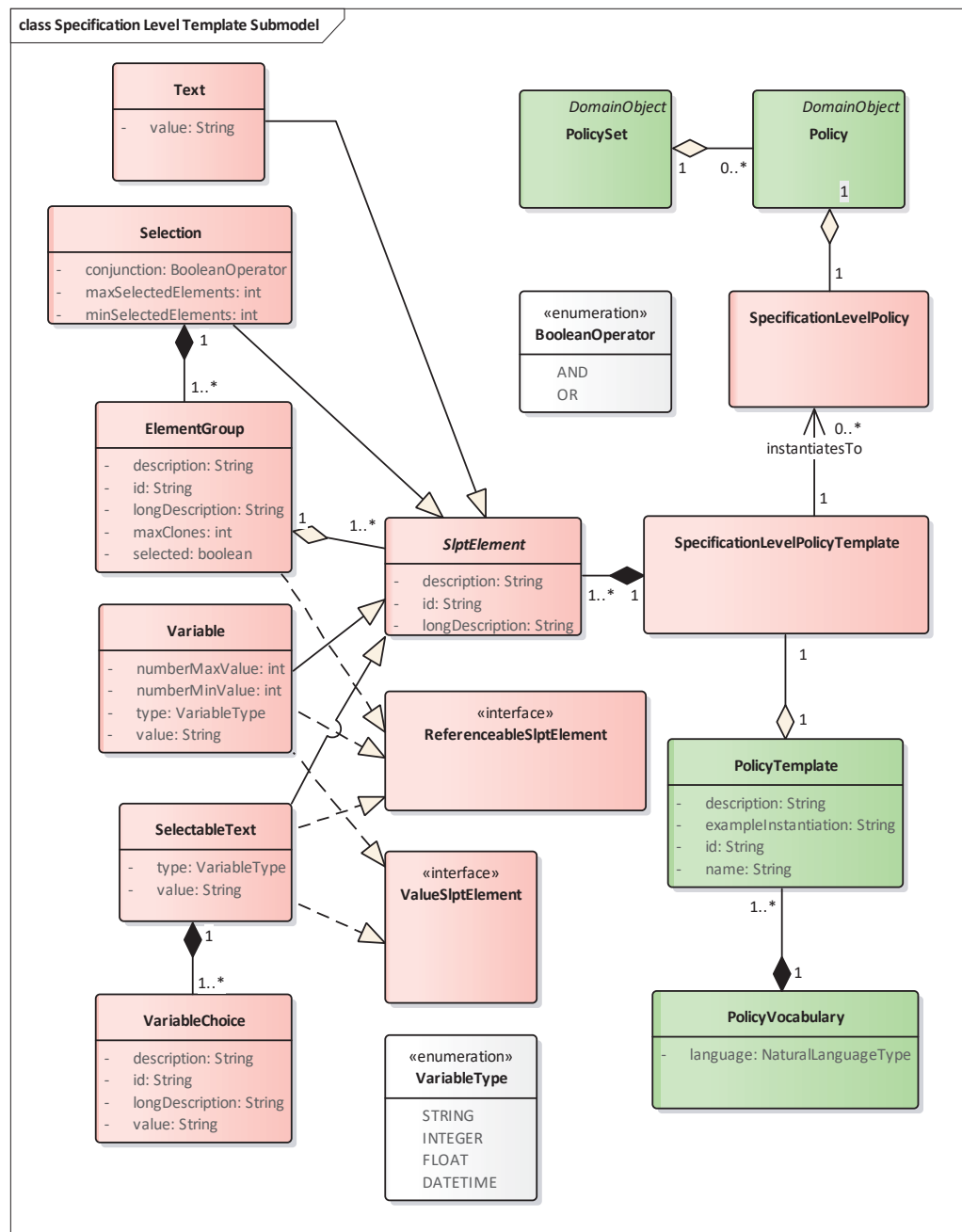


Figure 25: Specification-Level Template Sub-model

For linking the SLPTs with the corresponding ILPTs (described in Section 4.7) and with the specification paradigms (described in Section 4.8) two interfaces were defined, with which different SLPT elements are grouped:

- **ValueSlptElement:** All SLPT elements, for which a value specified by the user can be read, implement this interface. This means that the specified values can be read out from an SLP (i.e., an instantiated SLPT) and injected when an ILP is generated.
- **ReferenceableSlptElement:** When configuring the different specification paradigms, the method expert can pre-instantiate parts of an SLPT or set different values. To do this, the corresponding SLPT elements must be referenceable, which is ensured by this interface.

## 4.7 Implementation-Level Template Sub-model

The implementation level template sub-model (see Figure 26) refines the implementation-level policy template of the template sub-model.

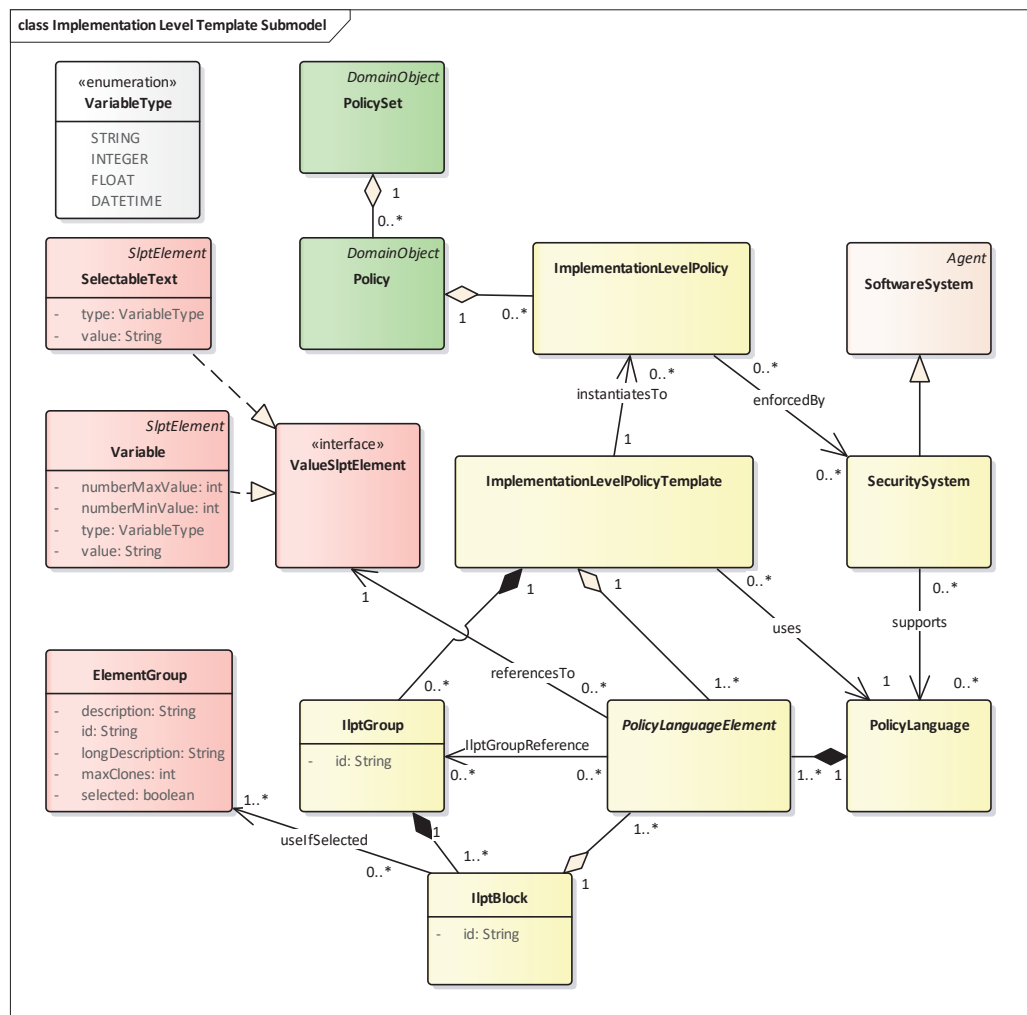


Figure 26: Implementation-Level Template Sub-model

An ILP describes a user's demand for privacy or security in a machine-understandable format so that a security system can enforce it. Since a security system generally supports only one specific policy language, the implementation-level policy must be formulated in this policy language. In our approach, the user instantiates an SLP from an SLPT. If this SLP is to be technically enforced, an ILP must be created analogously. Since many users do not understand machine-understandable policy languages and cannot specify ILPs on their own, we decided to generate ILPs from SLPs with defined transformation rules. Therefore, the method expert develops an implementation-level policy template analogous to the specification-level policy template. This ILPT is linked to the instantiation options of the specification-level policy template so that an ILP can be generated directly by the PAP when a SLP is instantiated by a user. The method expert must ensure that each instantiation of the linked policy templates leads to equivalent policies at both levels of abstraction.

An ILPT consists of elements of the chosen machine-understandable policy language, which we call policy language elements. We have limited our approach to XML-based policy languages. Other policy languages with other notation formats are conceivable, but have not been examined and are therefore currently not supported.

The method expert specifies a template structure similar to the one on the specification level. Immutable policy language elements are modeled directly as child elements of the ILPT. The transformation rules consist of the following parts:

- The method expert can associate an attribute value of a concrete policy language element with a ValueSlptElement, which is an element that contains a value entered or selected by the user. This value is used as the attribute value of the policy language element when generating the ILP. Examples of values defined by the user are user ids («Only Tom may access my data»), email addresses («If my data is used, send a message to mail@mail.com») or the number of permitted data uses («The advertising provider may use my data only three times»).
- The method expert can model the selection paths defined at specification level. He can specify blocks of policy language elements, which he inserts into the ILPT depending on the element groups selected by the user at the specification level. To do this, he links an ILPT block to an element group. If the user selects this element group at the specification level, the linked ILPT block is added to the implementation-level policy. The method expert groups ILPT blocks into ILPT groups. ILPT groups are part of an ILPT. The method expert can reference an ILPT group in a policy language element in the ILPT. When instantiating the ILPT as a concrete implementation-level policy, he inserts all selected ILPT blocks of the ILPT group as child elements.

In this thesis, we do not prove that an ILPT can be specified for every possible SLPT. Natural language can express any need for security and privacy. SLPTs use natural language. ILPTs use a policy language. This language is limited to a certain vocabulary and can cover a set of security and privacy demands. Therefore, it would be necessary to prove that the policy language is complete in terms of the specification of all conceivable security and privacy demands. This is not the focus of this work and can be part of future work.

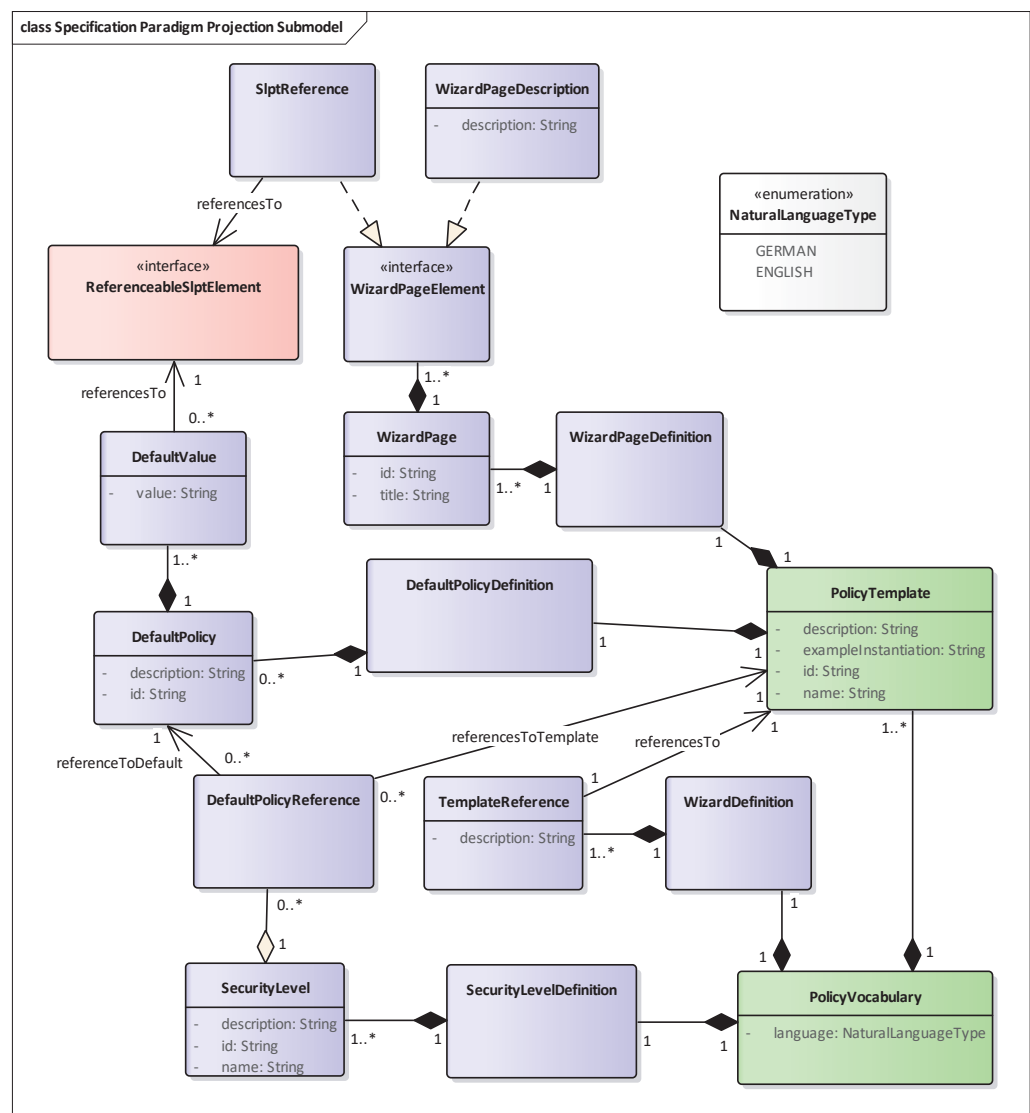


Figure 27: Specification Paradigm Projection Sub-model



## 4.8 Specification Paradigm Projection Sub-model

We present multiple specification paradigms in the context of this thesis in the Sections 2.5 and 5.2.2. These specification paradigms differ in the graphical representation of the user interface and in the interaction process between the user and the security system.

Our policy template model and the PAP generation framework currently support four different specification paradigms. The method expert defines projections rules to configure these four specification paradigms when instantiating the specification paradigm projection sub-model (see Figure 27). Since this configuration differs for all specification paradigms, the available paradigms are explained below:

- **Template Instantiation:** When instantiating the specification-level template sub-model, the method expert specifies policy templates. The individual elements of the policy template correspond to graphical elements on the user interface. When using the specification paradigm »template instantiation«, a PAP displays the elements on the graphical user interface exactly as specified in the policy template. The user can instantiate a concrete policy from the policy template by selecting paths in selections and text modules in selectable texts and by entering values in variables. The method expert does not need to specify any projection rules for this specification paradigm.
- **Default Policies:** The specification paradigm »default policies« limits the specification options for the user when instantiating the policy templates. Thus, the user can select per policy template the default policy that best matches his personal security and privacy needs. The method expert specifies a default policy element from the model for each pre-instantiated default policy per policy template. In this default policy element, he creates a default value element for each modifiable element in the SLPT. All SLPT elements that implement the »ReferenceableSlptElement« interface can be referenced via their id. For variables, the variable element of the SLPT template itself is referenced and the specified default value is used during instantiation. With selectable texts, the selectable text element of the SLPT template itself is referenced and the id of the desired variable choice is specified as the value. With selections, the paths to be selected are referenced individually in the policy template, that is, the element groups. The value specifies whether they are selected or not (true/false). The method expert can specify any number of default policies per policy template. On the user interface, the user selects the desired default policies per template. However, the user cannot modify default policies.
- **Security Levels:** The specification paradigm »security levels« severely limits the specification options of the users. They can choose exactly

one of several security or privacy levels. For brevity, we only talk about security levels in the following, but the same principles apply to privacy levels as well. The method expert specifies the security levels as part of the policy vocabulary. A security level consists of a name, a description and a set of references to default policies. This means that the method expert provides a selection of different lists of default policies (i.e., pre-instantiated policy templates) to the user by specifying security levels. A security level may reference any number of default policies per policy template.

- **Wizard:** For the specification paradigm »wizard«, the user instantiates the policy templates in a fixed order. The method expert defines this order in the policy vocabulary using a list of template references. In addition, the method expert can divide the instantiation of each policy template into several smaller specification steps. To this end, the method expert specifies a list of steps for each policy template, which we call wizard pages. The method expert models a wizard page with text blocks (WizardPageDescription elements) and modifiable SLPT elements. The text blocks used in the original specification-level policy templates cannot be reused, since they are designed to create a cloze text with correct natural grammar together with all modifiable elements. The method expert uses the text blocks on the wizard pages to describe only the part of the policy template that can be configured by the user on the current page. In addition, further explanatory text blocks can be added so that the user can better understand the effect of his decisions. Each wizard page has an individual title, which is displayed on the graphical user interface.

Table 10: Exemplary Policy Template »Access to Financial Data in Different Situations«

| ID                     | Policy Template Name                             | Asset   | Target System       | Policy Author      |
|------------------------|--|---|---------------------|--------------------|
| PT8                    | Access to financial data in different situations | Financial data of bank client   | Mobile advisory app | Bank administrator |
| Policy Template Syntax |  | If a financial advisor is about to access financial data of clients [inside the bank on a business trip in home office] and already accessed <numberOfDataRecords:integer(1,*)> data records [this hour today this week], then [allow access allow access and inform <recipient:string> inhibit access inhibit access and inform <recipient:string>]. |                     |                    |
| Description            |  | The access of financial advisors to financial data of bank clients need to be restricted in different situations. Especially the number of data records per period needs to be controllable.  |                     |                    |
| Threat                 |  | Unintended access to financial data of bank clients   |                     |                    |
| Security/Privacy Goals |  | Confidentiality   |                     |                    |
| Example Instantiation  |  | If a financial advisor is about to access financial data of clients in home office and already accessed 100 data records today, then inhibit access and inform mrs.black@bank.de.   |                     |                    |

## 4.9 Example

The method expert derived the policy template »access to financial data in different situations« shown in Table 10 with the policy template elicitation method. The model expert uses this information to instantiate the policy template model in order to create a policy vocabulary. Figure 28 shows an excerpt of the policy template model including the relevant elements for the modeling of the exemplary policy template.



Listing 1: Exemplary SLPT

```

<slpt>
  <text id="pt8_t1" value="If a financial advisor is about to access
    financial data of clients "/>
  <selection id="pt8_situation" minSelectedElements="1"
    maxSelectedElements="1">
    <elementGroup id="pt8_situation_inside_bank">
      <text value="inside the bank"/>
    </elementGroup>
    <elementGroup id="pt8_situation_business_trip">
      <text value="on a business trip"/>
    </elementGroup>
    <elementGroup id="pt8_situation_home_office">
      <text value="in home office"/>
    </elementGroup>
  </selection>
  <text value=" and already accessed " id="pt8_t2"/>
  <variable id="pt8_records" type="integer" description="number of data
    records" numberMinValue="1"/>
  <text value=" data records " id="pt8_t3"/>
  <selectableText id="pt8_period" type="string">
    <variableChoice id="pt8_period_hour" description="this hour"
      value="thisHour"/>
    <variableChoice id="pt8_period_day" description="today" value="today"/>
    <variableChoice id="pt8_period_week" description="this week"
      value="thisWeek"/>
  </selectableText>
  <text value=", then " id="pt8_t4"/>
  <selection id="pt8_reaction" minSelectedElements="1"
    maxSelectedElements="1">
    <elementGroup id="pt8_reaction_allow">
      <text value="allow access"/>
    </elementGroup>
    <elementGroup id="pt8_reaction_allow_inform">
      <text value="allow access and inform "/>
      <variable id="pt8_notification_recipient_allow" type="string"
        description="email address"/>
    </elementGroup>
    <elementGroup id="pt8_reaction_inhibit">
      <text value="inhibit access"/>
    </elementGroup>
    <elementGroup id="pt8_reaction_inhibit_inform">
      <text value="inhibit access and inform "/>
      <variable id="pt8_notification_recipient_inhibit" type="string"
        description="email address"/>
    </elementGroup>
  </selection>
</slpt>

```

We present an exemplary generated user interface implementing the specification paradigm »template instantiation« in Figure 29. We explain the modelling of projection rules to support additional specification paradigms below. In the current version of the policy template model, we neither model direct relations between SLPT elements and actions of the application domain nor refine these actions into more fine-grained elements. We can add these extensions as part of future work, in order to

support automation regarding the creation of SLPTs, ILPTs and transformation rules. Therefore, we can reuse existing work from the literature such as the work of Kumari [76]. Currently, the specification of SLPTs, ILPTs and respective transformation rules is a manual task.

The figure shows a web-based specification interface titled "Access to financial data in different situations". It prompts the user to "Please refine the policy template". The main condition is "If a financial advisor is about to access financial data of clients". Below this, there are three radio button options for the situation: "inside the bank" (selected), "on a business trip", and "in home office". A second condition is "and already accessed" followed by a text input "number of data records", the word "data records", and a dropdown menu currently set to "this hour". This is followed by the word "then". Below the "then" clause, there are four radio button options for the action: "allow access" (selected), "allow access and inform" (with an "email address" input field), "inhibit access", and "inhibit access and inform" (with an "email address" input field). At the bottom of the form is a "Generate Policy" button.

Figure 29: Generated Specification Interface for Exemplary SLPT Implementing the Specification Paradigm »Template Instantiation«

Next, the model expert specifies the ILPT in the desired policy language. To this end, he first defines the invariable skeleton of the ILPT using policy language elements. The ILPT (using the IND<sup>2</sup>UCE policy language) that belongs to the exemplary SLPT is shown in Listing 2.

Listing 2: Exemplary ILPT

```
<ilpt>
  <ind2ucePolicy>
    <policy id="urn:policy:phdTest:access_records" description="">
      <ind2uce:mechanism event='urn:action:phdTest:access_records'>
        <ind2uce:if ilptGroupReference="pt8_ilp_reaction">
          <ind2uce:and ilptGroupReference="pt8_ilp_situation">
            <ind2uce:greater>
              <ind2uce:count>
                <ind2uce:eventOccurrence event='urn:action:phdTest:access_records'>
                  <parameter:string name='user'>
                    <event:string eventParameter='user' default='' />
                  </parameter:string>
                </ind2uce:eventOccurrence>
                <ind2uce:when fixedTime='$ref:pt8_period' />
              </ind2uce:count>
              <constant:number value='$ref:pt8_records' />
            </ind2uce:greater>
          </ind2uce:and>
        </ind2uce:if>
```

```

    </ind2uce:mechanism>
  </policy>
</ind2ucePolicy>
</ilpt>

```

In addition, the model expert defines two types of variable parts:

- **Variable values:** References to variables or selectable text elements of the SLPT can be inserted into the values of attributes of policy language elements. During the instantiation of the ILPT, these values are replaced by the respective values entered by the user in the specification interface.
- **ILPTGroupReferences:** Variable parts of the ILPT can be inserted into the model based on the selection of element groups during the instantiation of an SLPT. To this end, the model experts assigns a reference to an ILPT group to a policy language element. In addition, he specifies these variable parts as ILPT blocks as part of an ILPT group. Each ILPT block is linked to an element group. If an element group of the SLPT is selected on the specification interface, the corresponding ILPT block is inserted as a child of the policy language element that has the reference to the ILPT group assigned. Listing 3 shows the ILPT group for the selection of the current situation of the financial advisor in the exemplary policy template.

Listing 3: Exemplary ILPT Group

```

<ilptGroup id="pt8_ilp_situation">
  <ilptBlock id="pt8_ilp_situation_inside_bank"
    use="pt8_situation_inside_bank">
    <pip:boolean method='urn:info:phdTest:insideBank' default='false' />
  </ilptBlock>
  <ilptBlock id="pt8_ilp_situation_business_trip"
    use="pt8_situation_business_trip">
    <pip:boolean method='urn:info:phdTest:businessTrip' default='false' />
  </ilptBlock>
  <ilptBlock id="pt8_ilp_situation_home_office"
    use="pt8_situation_home_office">
    <pip:boolean method='urn:info:phdTest:homeOffice' default='false' />
  </ilptBlock>
</ilptGroup>

```

Using these two types of variable parts, the model expert can define transformation rules for a policy template that allows the generation of an ILP based on an SLP. We present an example of a generated ILP in Figure 30.

```

⊕ <policy id='urn:policy:phdTest:access_records191753553' description='If a financial advisor is about to access
financial data of clients in home office and already accessed 100 data records today, then inhibit access and inform
mrs.black@bank.de.'>
  ⊕ <mechanism event='urn:action:phdTest:access_records'>
    ⊕ <if>
      ⊕ <and>
        ⊕ <greater>
          ⊕ <count>
            ⊕ <eventOccurrence event='urn:action:phdTest:access_records'>
              ⊕ <parameter:string name='user'>
                <<event:string eventParameter='user' default=''>/>
              </parameter:string>
            </eventOccurrence>
            <when fixedTime='today'>/>
          </count>
          <<constant:number value='100'>/>
        </greater>
        <<pip:boolean method='urn:info:phdTest:homeOffice' default='false'>/>
      </and>
      ⊕ <then>
        <<inhibit/>
        ⊕ <execute action='urn:action:phdTest:send_mail'>
          <<parameter:string name='recipient' value='mrs.black@bank.de'>/>
          ⊕ <parameter:string name='message'>
            ⊕ <concat>
              <<constant:string value='User '>/>
              <<event:string eventParameter='user' default=''>/>
              <<constant:string value=' requested too much data items.'>/>
            </concat>
          </parameter:string>
        </execute>
      </then>
    </if>
  </mechanism>
</policy>

```

⬆Top

Figure 30: Exemplary Instantiated Policy in the IND<sup>2</sup>UCE Policy Language

Finally, the model expert can define projection rules to support multiple specification paradigms. Figure 31 shows an excerpt of the policy template model including the relevant elements for the definition of projection rules for different supported specification paradigms for a policy template. Our model currently supports the specification paradigms »template instantiation«, »default policies«, »security levels« and »wizard«.



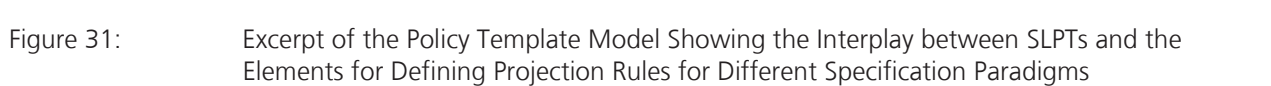


Figure 32: Generated Specification Interface Implementing the Specification Paradigm »Default Policies«, which Shows the Specified Projection Rules for the Exemplary SLPT

To support the specification paradigm »default policies«, the model expert specifies a set of default instantiations for each SLPT. To define a default instantiation, he sets the values of each variable element of the SLPT. We present an example in Listing 4. During the generation of a specification interface that uses the specification paradigm »default policies«, the SLPT is instantiated according to the default instantiations and these instances are shown on the user interface. We present an exemplary specification interface in Figure 32. The values of the specified default instantiations are also used for the generation of ILPs.

Listing 4: Exemplary Projection Rules for the Specification Paradigm »Default Policies«

```
<defaultInstantiations>
  <defaultInstantiation id="pt8_default1" description="If a financial
    advisor is about to access financial data of clients in home office
    and already accessed 100 data records today, then inhibit access and
    inform mrs.black@bank.de.">
    <defaultValue ref="pt8_situation_inside_bank" value="false"/>
    <defaultValue ref="pt8_situation_business_trip" value="false"/>
    <defaultValue ref="pt8_situation_home_office" value="true"/>
    <defaultValue ref="pt8_records" value="100"/>
    <defaultValue ref="pt8_period" value="pt8_period_day"/>
    <defaultValue ref="pt8_reaction_allow" value="false"/>
    <defaultValue ref="pt8_reaction_allow_inform" value="false"/>
    <defaultValue ref="pt8_reaction_inhibit" value="false"/>
    <defaultValue ref="pt8_reaction_inhibit_inform" value="true"/>
    <defaultValue ref="pt8_notification_recipient_inhibit"
      value="mrs.black@bank.de"/>
  </defaultInstantiation>
</defaultInstantiations>
```

To support the specification paradigm »wizard«, the model expert specifies for each SLPT a set of small specification steps, which we call wizard pages. For each wizard page, he freely orchestrates text blocks (SLPT descriptions) and references to the variable parts of the SLPT. Next, he sets the order of the wizard pages per paradigm. Listing 5 shows exemplary projection rules. We present a screenshot of a generated wizard page in Figure 33.

Page Number 3/3: Reaction

Do you want to allow or inhibit data record access after the threshold is reached? And do you want to inform somebody about the situation?

☐ allow access

☒ allow access and inform

☐ inhibit access

☐ inhibit access and inform

Back Next

Generate Policy

Figure 33: Generated Specification Interface Implementing the Specification Paradigm »Wizard«, which Shows one Wizard Page for the Exemplary SLPT

Listing 5: Exemplary Projection Rules for the Specification Paradigm »Wizard«

```
<wizardPageDetails>
  <page id="pt8_page1" title="Current Situation">
    <slptDescription description="In which situation do you want to restrict
      data access by financial advisors?"/>
    <slptReference ref="pt8_situation"/>
  </page>
  <page id="pt8_page2" title="Number of Accessed Data Records">
    <slptDescription description="How many data records are allowed?"/>
    <slptReference ref="pt8_records"/>
    <slptDescription description="In which period are the data records
      allowed?"/>
    <slptReference ref="pt8_period"/>
  </page>
  <page id="pt8_page3" title="Reaction">
    <slptDescription description="Do you want to allow or inhibit data
      record access after the threshold is reached? And do you want to
      inform somebody about the situation?"/>
    <slptReference ref="pt8_reaction"/>
  </page>
</wizardPageDetails>
```

To support the specification paradigm »security levels«, the method experts defines overarching sets of default instantiations. Each set, which represents a security level, may reference at most one default instantiation per policy template.

## 4.10 Summary and Conclusion

In this chapter, we presented the policy template model. Below, we briefly review the fulfillment of the requirements for the policy template model (as stated in Section 4.1):

- **Req\_Model\_Domain-Independence:** The security policy template model was designed for being applicable to different application domains. To address this requirement, we introduced the generic domain sub-model. It allows the method expert to describe actions and corresponding actors in an application domain, declare malicious actions as threats and compensating actions as countermeasures. Threats and countermeasures are linked to basic security and privacy principles. The method expert specifies policy templates and links them to related threats and countermeasures.
- **Req\_Model\_Understandable\_Templates:** The specification-level template sub-model allows the specification of policies templates in natural language. The method expert can use several template elements for the creation of SLPTs. This sub-model is compatible with the policy template notation format presented in Section 3.7.
- **Req\_ILP\_Generation:** The implementation-level template sub-model and its linkage to the specification-level template sub-model facilitates

the specification of transformation rules for the ILP generation after the instantiation of an SLPT by the user.

- **Req\_Model\_Specification-Paradigm-Projection:** The model does not allow the specification of generic projection rules. Consequently, the model must be extended for each new specification paradigm that we want to support. We currently provide four specification paradigms: »template instantiation«, »default policies«, »security levels« and »wizard«.

To conclude, a policy vocabulary, which is an instance of the presented policy template model, is capable of describing policy templates on two abstraction levels (SLPT and ILPT). This allows the instantiation of natural language policy specifications and their transformation into concrete, machine-enforceable security and privacy policies. These instantiated policies realize countermeasures against threats to the application domain, which can be described within a model instantiation as well.



## 5 PAP Generation Framework

Our goal is to provide PAPs that suit the individual needs of the user. As developing PAPs is effort-consuming, we aim for an automated solution. We want to automate the user interface development for policy specification so that these interfaces can be generated at runtime.

Supporting the use of different policy vocabularies allows the PAP generation framework to cover different application domains. Our framework is a modular toolkit that supports different variation points. We provide different specification paradigms to allow the personalization of the PAP to individual user groups. In addition, the framework realizes the operability in different user interface frameworks (e.g., Java desktop applications, Android apps or web applications) and the translation of specified SLPs into ILPs using different policy languages. These variation points provide the necessary adaptability for using PAPs in different application domains.

In this chapter, we present a PAP generation framework, which represents Contribution 4 of this thesis (see Section 1.4). This framework provides an automated generation of policy specification interfaces in PAPs with multiple supported specification paradigms based on a policy vocabulary.

The PAP generation framework presented in this work is a concept. We describe the necessary components, their functionality and their interrelation, and we present our reference implementation of the PAP generation framework. With our concept, a PAP generation framework for other system environments can be implemented with limited effort.

The remainder of this chapter is structured as follows. The research approach is explained in Section 5.1. Section 5.2 provides the key concepts of the PAP generation framework including an overview of the framework in Section 5.2.1 and a concept for embedding a PAP in existing software in Section 5.2.2. The selection of the supported specification paradigms and their generation algorithms are presented in Section 5.3. Section 5.4 describes our reference implementation of the PAP generation framework and shows some exemplary PAPs. Section 5.5 concludes this chapter.

## 5.1 Research Approach

We developed the PAP generation framework in an iterative process. We developed three versions of the framework and applied each version in a case study (see Figure 34). We used the observations and lessons learned from the applications in the case studies for improving and extending the framework.

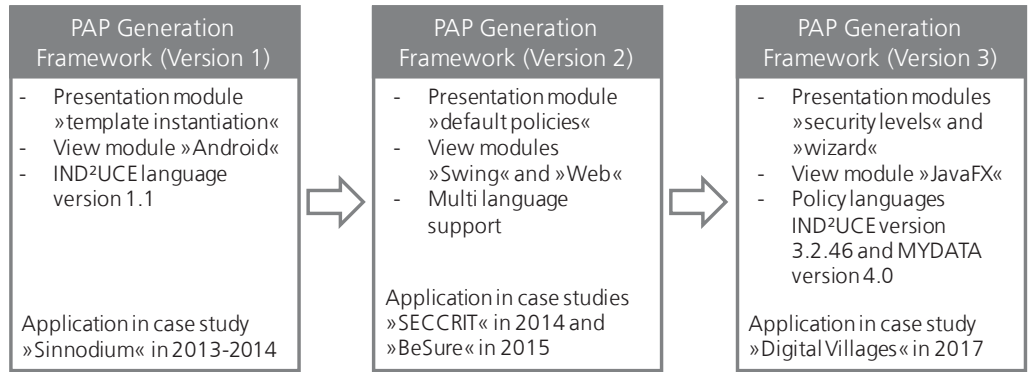


Figure 34: Research Approach for the PAP Generation Framework

We first built a prototype of the PAP generation framework in the context of the »SINNODIUM« case study (see Section 8.2). It supported the specification paradigm »template instantiation«, the user interface (UI) framework »Android«, the machine-understandable language »IND²UCE policy language version 1.1« and was tested in one application domain. The resulting Android PAP was used in a project demonstrator. It let users specify human-understandable policies through template instantiations and transformed these specifications into machine-understandable policies that were enforced in the demonstrator.

Next, we conceptualized the modular character of the framework and added more modules. The second version additionally supported the specification paradigm »default policies« as presentation modules to show the use of multiple paradigms in one PAP. We also added view modules for the user interface (UI) frameworks »Swing« for Java applications and »Web« for web applications to demonstrate the application across multiple UI technologies and user devices. In addition, we implemented support for multiple languages. Standardized terms that are not defined in the policy vocabulary, such as filling words and default description texts, were available in English and German. We applied this version of the PAP generation framework in the »SECCRIT« case study, where we created a PAP with generated user interfaces for policy specification in the application domain of »Cloud systems in critical infrastructures«. We also used this version for the case study »BeSure«.

Finally, we extended the framework with more modules and features. We added support for the UI framework »JavaFX« and created presentation

modules for the specification paradigms »security levels« and »wizard«. We supported the policy languages IND<sup>2</sup>UCE version 3.2.46 and MYDATA policy language version 4.0. This version of the PAP generation framework was used in the end-to-end evaluation of this thesis. The evaluation included the case study »Digital Villages« and the policy specification experiment.

We identified the following requirements for our PAP generation framework from the state of the art:

- **Req\_Framework\_UI-Generation:** The framework must support the automated generation of specification interfaces in a PAP using a policy vocabulary as input. The resulting user interface inside a PAP can be used by a user to specify own security or privacy demands within the specification options provided by the specification paradigm. Automation is mandatory as companies are not willing to spend effort in the development of usable PAPs if no incentive exists, as stated by Lampson [19].
- **Req\_Framework\_Modularity:** The framework needs to be extensible. Therefore, we propose a modular architecture. A module on an architectural layer must be replaceable by another module, e.g., for supporting different specification paradigms or UI frameworks. The modularity is required as it allows the support of multiple specification paradigms to be used in the same PAP as required by Contribution C1 as well as the limitation of development effort, which PAP vendors are not willing to spend, as noted by Lampson [19]).
- **Req\_Framework\_Policy-Templates:** Generated policy specification interfaces in PAPs need to provide all policy templates on the user interface that are modeled in a policy vocabulary. This requirement stems from Hypothesis H2, which requires completeness for elicited policy templates.
- **Req\_Framework\_Specification-Paradigms:** To provide personalization to different user groups, generated policy specification interfaces in PAPs need to support multiple specification paradigms. We require the specification paradigm selection at runtime. Thus, a single PAP must support all specification paradigms so that either the expert can configure the default paradigm for each user group or the user itself can select his preferred specification paradigm on PAP startup.
- **Req\_Framework\_Multi-User-Interface-Frameworks:** Because policy specification affects users of web applications in browsers as well as users of mobile devices or traditional applications, the PAP framework must not be limited to a single UI framework. Support of multiple UI frameworks limits the development effort, which companies are not willing to spend, as noted by Lampson [19].



- **Req\_Framework\_Policy-Transformation:** Kumari [76] recommends the specification of policies on the specification level. Therefore, the generated PAPs must be capable of transforming specified policies (SLPs) into machine-understandable equivalents (ILPs). Regarding ILPs, the support of multiple XML-based policy languages is required to increase the applicability of the framework in practice.

## 5.2 Reference Architecture

Section 5.2.1 describes the architecture of the framework. In Section 5.2.2, we show how to embed the framework into existing software components.

### 5.2.1 Architectural Overview

To achieve the required flexibility of the framework, we chose a modular architecture. Our framework architecture is based on an adapted model-view-controller design pattern [160], as shown in Figure 35.

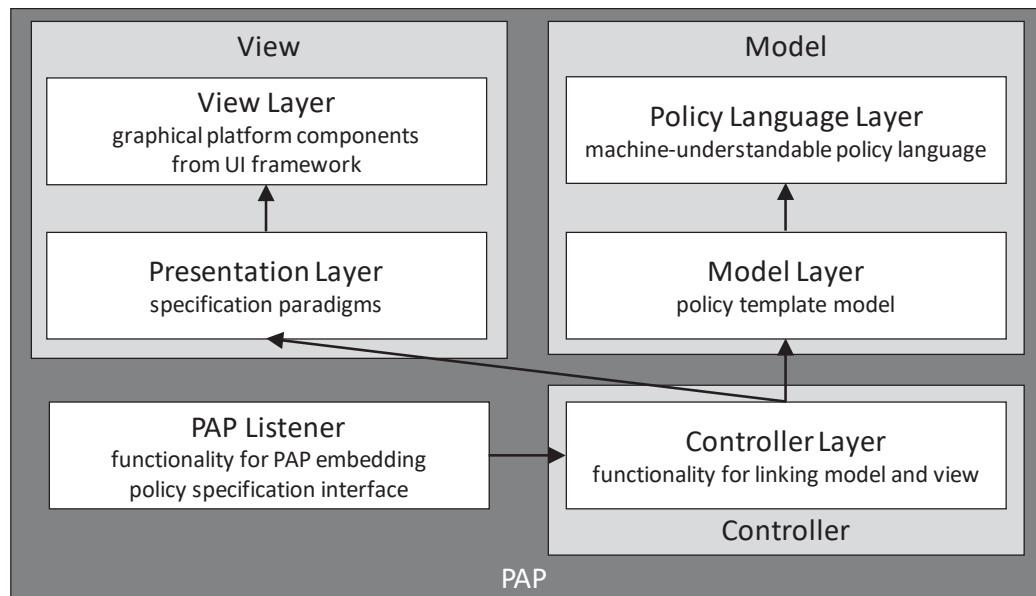


Figure 35: Model-View-Controller Concept in the PAP Generation Framework

We chose this design pattern as it clearly separates data and graphical user interface. This separation is important, as the basic requirement for our framework is to have replaceable modules on several layers. Our architecture comprises five layers, which reflect the variation points listed in the previous section:

- **View Layer:** The view layer encapsulates typical user interface elements of an UI framework. These include elements for user

interaction, information display, user interface styling and layout organization. View elements are, for example, buttons or text fields.

- **Presentation Layer:** The presentation layer provides modules that implement the different specification paradigms by assembling elements from the view layer. The user interface of a specification paradigm is described as a generation algorithm. This algorithm assembles the available view elements in such a way that the user interface corresponds to the specification paradigm. In addition, the algorithm describes how the content of the policy vocabulary is presented with corresponding view elements on the user interface so that the user can specify policies. We provide a presentation module for each specification paradigm.
- **Policy Language Layer:** The policy language layer contains modules for the PAP output. For example, modules for machine-understandable policy languages such as the IND<sup>2</sup>UCE policy language can be provided.
- **Model Layer:** The model layer represents the policy template model. Thus, an instance of the policy template model is a module of this layer. It contains policy templates and all necessary information for the user interface generation. If the PAP is expected to produce machine-understandable policies as an output, the instance of the model must include transformation rules from SLPTs to ILPTs for the selected policy language module.
- **Controller Layer:** The controller layer initiates the user interface creation, and it organizes the information exchange among the model and view layers

In addition to the five layers, we defined a **PAP listener** as an interface the PAP must implement that integrates the generated policy specification interfaces. All layers and the PAP listener are described in detail in the following subsections. We defined interfaces for each layer to facilitate the exchangeability of framework modules.

### ***Relation between Layers***

The elements on the different layers of the PAP generation framework need to interact with each other in order to provide a fully functional PAP. According to the chosen model-view-controller approach, the content from the model layer need to be transferred to the presentation layer by the controller layer.

The policy template model provides a structure for policy templates. A policy template contains an SLP template for the human-understandable representation of a policy template as well as an ILP template for the machine-understandable representation. The SLP template itself contains

several SLPT elements for modelling the template (see Section 4.6). The SLP templates shall be offered to the user on the user interface using a specific specification paradigm. Accordingly, the SLPT elements need to be provided on the presentation level in order to be used for the implementation of specification paradigms. Thus, the structure of the policy vocabulary model including the policy templates is the baseline for our reference architecture. For each of these model elements, we also provide a respective controller and presenter element. We chose an inheritance relation between presenter, model and controller layer as shown in Figure 36.

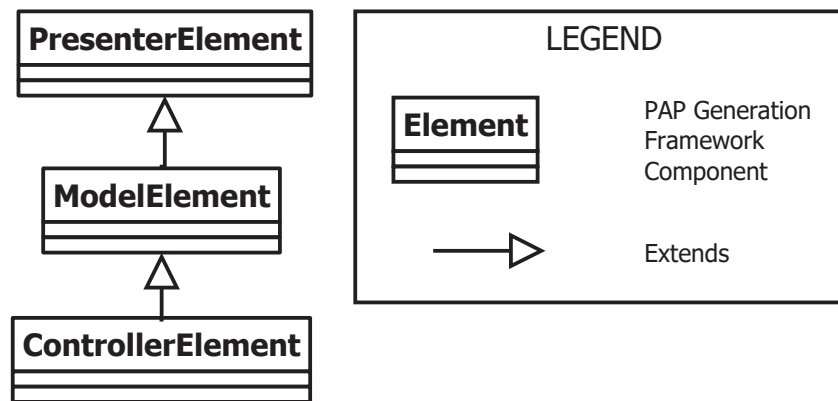


Figure 36: Inheritance Relation between Model, Presenter and Controller Layers

The model element inherits from the presenter element and the controller element inherits from the model element. This allows the controller to access and control data from both other layers. On the one hand, this access is necessary to push information from the model to the presentation layer, for example, when showing a new policy template on the user interface. On the other hand, when the user enters data into the user interface, the respective controller can pull information from the presentation module and store it in the model. This interaction is essential, as the instantiation of the policy template inside the model is used for the transformation from the instantiated SLP into an ILP. In case of data modifications, the presentation elements can trigger respective controller elements to fetch updates and push them into the model.

In addition to the elements of the policy templates, an element representing the whole policy vocabulary is provided on the presentation, model and controller layers. The use of this element is mandatory as it is initialized on PAP startup. Moreover, the element provides access to additional data from the policy vocabulary for the implementation of specification paradigms. An example is the order in which policy templates are shown in the »wizard« specification paradigm or which concrete policies are provided at the different levels of the »security levels« specification paradigm.

The developer can decide on how to use the presentation elements for implementing a specification paradigm while creating a presentation module. As the specification paradigms vary in their interaction process and therefore their user interface, we chose an aggregation relation between the presentation layer and the view layer. The developer can freely assemble view elements in the presentation elements.

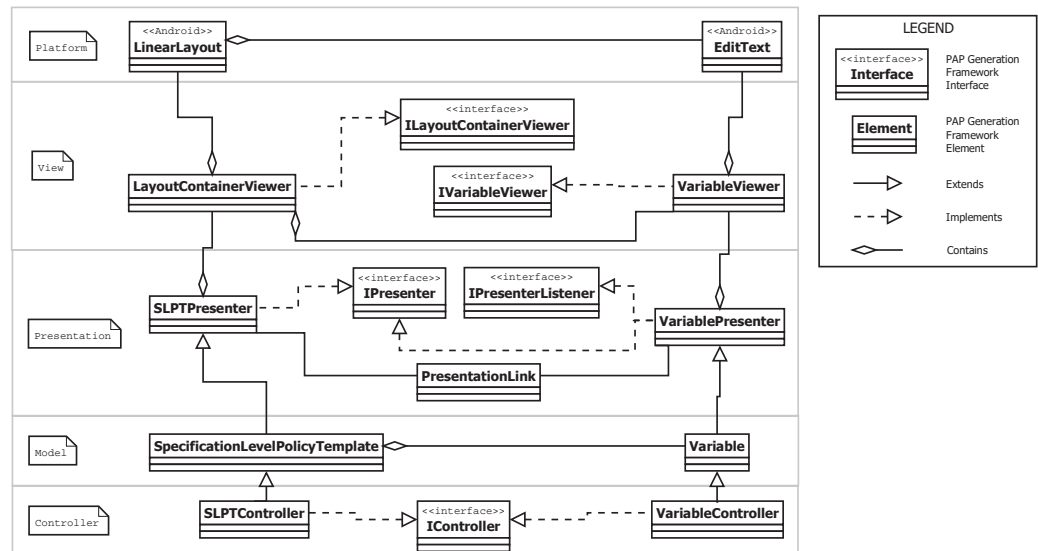


Figure 37: Relation of Elements between Layers

The relation between elements at the different layers is summarized in an example in Figure 37. Details about the elements and interfaces for each specific layer are presented in the following subsections.

## View Layer

The main purpose of the view layer is to provide UI framework independence. This layer offers a basic set of interfaces for view elements that need to be implemented for each module. Each interface includes a specification of necessary functionality. A view element encapsulates a UI framework-specific graphical component and provides it as an independent abstraction to the presentation layer. In the optimal case, the UI framework provides a graphical component that supports all functionality that is required by the interface. If not, missing functions need to be implemented. Thus, developers use the view element to specify the specification paradigm algorithms within the presentation modules. Figure 38 shows an overview of the view elements, their interfaces and their hierarchical relation.

The interfaces contain standardized methods for setting the look and feel (e.g., sizes and colors) as well as the behavior (e.g., action on click or input validation) of the view elements. In the following, the required view elements per module are described. All these elements need to be implemented, because they reflect the elements used in the specification

paradigms identified in the state of the art (see Section 2.5). These view elements reflect the SLPT elements from the policy template model (see Section 4.6).

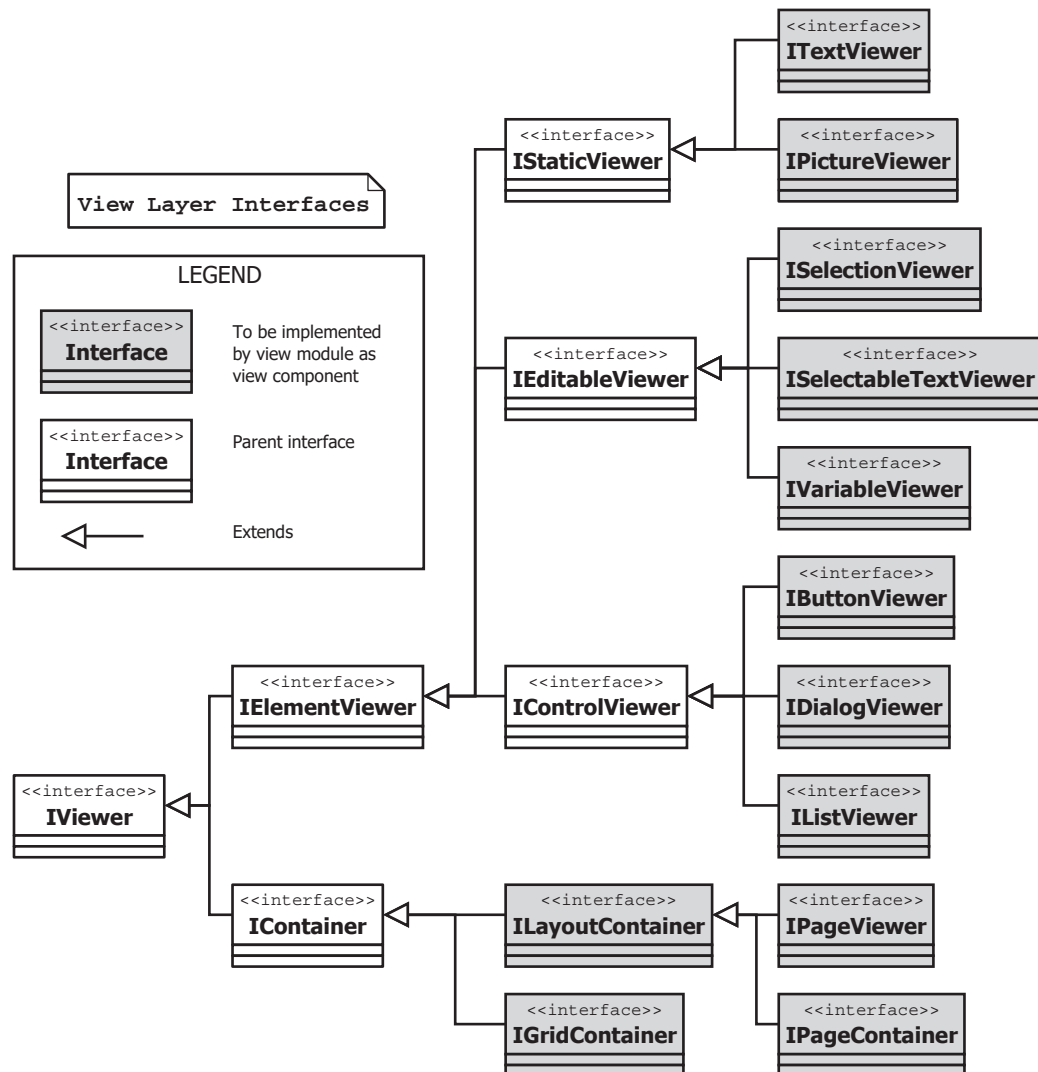


Figure 38: Interfaces for View Elements of the View Layer

The view elements can be subdivided into element viewers and containers. Element viewers are used for displaying static information, for obtaining user input or for providing control and notification features to the user. Containers are used for arranging the layout of other containers and element viewers. The mandatory elements and categories are:

- Static viewers for displaying information on the screen.
  - A **text viewer** (interface `ITextView`) displays a text.
  - A **picture viewer** (interface `IPictureViewer`) displays a picture.
- Editable viewers for input requests from the user.

- A **selection viewer** (interface `ISelectionViewer`) provides the user different options for selection. The content of the selection options is defined by other view elements. The number of options a user may select at minimum and at maximum must be configurable. Typical representations of a selection viewer in UI frameworks are check boxes or radio buttons (e.g. in Swing and Android).
- A **selectable text viewer** (interface `ISelectableTextViewer`) provides the user different text options for selection. The user may select one option. A typical representation for a selectable text viewer in UI frameworks are a spinner in Android, a combobox in JavaFX or a select tag in HTML.
- A **variable viewer** (interface `IVariableViewer`) provides the user the capability to enter information in the form of text, numbers or dates. A typical representation of a variable viewer in UI frameworks is a text field in JavaFX or an input tag in HTML.
- Control viewers for navigation and user notification.
  - A **button viewer** (interface `IButtonViewer`) represents a typical button on the user interface for triggering an action (e.g., saving a policy) or for navigation within the specification paradigm.
  - A **dialog viewer** (interface `IDialogViewer`) shows the user a dialog box with a notification on the user interface.
  - A **list viewer** (interface `IListViewer`) displays the user a list of selectable options on the user interface. This control viewer can, for example, be used to show a list of policy templates for selection.
- Containers for arranging the layout of view elements on the user interface.
  - A **layout container** (interface `ILayoutContainer`) contains and arranges other view elements in one layout direction (on either the horizontal or the vertical axis). The orientation in which the contained elements appear can be specified. The position of all contained elements is set automatically, which means that the UI framework component must support the automatic layout or that the developer must implement an appropriate algorithm.
  - A **grid container** (interface `IGridContainer`) contains and arranges other view elements in a grid layout. The developer can define the number of rows and columns and place one view elements in each cell.
  - A **page viewer** (interface `IPageViewer`) is a special instance of a layout container. The handling of content is similar. However, a page viewer is meant to be used if parts of the user interface

are shown in sequence, for example, in the specification paradigm »wizard«.

- A **page container** (interface `IPageContainer`) contains a list of page viewers and navigation functionality for switching the pages. The developer defines the order of the pages.

In addition to these components, standardized abstractions for colors, pictures, layout orientations, viewer alignments, text styles, viewer dimensions need to be provided for each operation platform.

## Presentation Layer

The main purpose of the presentation layer is to provide the exchangeability of paradigms for policy specification. Therefore, the developer can provide multiple specification paradigms as modules for the PAP generation framework. A presentation module contains an algorithm that defines the generation of the user interface based on a specification paradigm. In the algorithm, the developer describes how the information from a policy template model instance is presented on the user interface. He uses view elements to arrange the visible information and to implement the specification process. The generation algorithms for the four supported specification paradigms of our reference implementation can be found in Section 5.3.2.

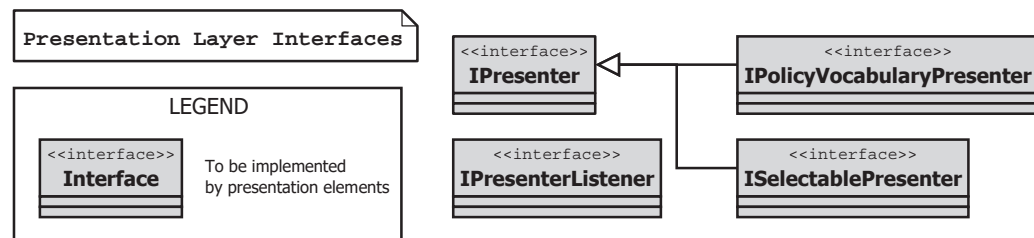


Figure 39: Interfaces for Presentation Elements of the Presentation Layer

As presentation modules are exchangeable and not hard-wired to the controller layer, presentation elements need to implement respective interfaces. The following interfaces exist (see Figure 39):

- Each presentation element inherited by a respective controller element must implement the **IPresenter** interface. This interface covers methods for initializing the presentation element, for retrieving values from the model or pushing user input from view elements back into the model. Additionally, the developer of the presentation module must implement an input checker that validates whether the data entered by the user is valid (e.g., with respect to type or range according to the specification in the template) and complete (e.g., checking that all mandatory fields have been filled out). In case of an invalid input, the user is notified via the PAP listener.

- The **ISelectablePresenter** interface extends the **IPresenter** interface and adds methods for setting and getting the selection state of presentation elements that the user can select or deselect.
- The **IPolicyVocabularyPresenter** interface extends the **IPresenter** interface and adds functionality. There exists exactly one implementation of this interface, the **PolicyVocabularyPresenter**. First, a function for retrieving the reference to the UI framework component that is linked to the parent view container element of the policy specification interface must be provided. This component can be embedded in the surrounding PAP. It contains the complete policy specification interface. Second, in some cases it is necessary to send information from the policy specification interface to the surrounding PAP (e.g., for showing error dialogs or storing instantiated policies). Therefore, a function for registering a listener at the **PolicyVocabularyPresenter** is provided.
- All presentation elements that need to be informed about changes of view elements must implement the **IPresenterListener** interface. This interface provides a list of callback methods for elements that are clicked, that gain or lose focus or whose value was changed. Thus, on each change of the view element, the affected presentation element can retrieve the new data and trigger a controller element to update the model instance. Generic interfaces between the presentation layer and the controller layer ensure the paradigm-independence.

Information about the relations between different presentation elements can be specified via the **IPresentationLink** interface, which contains methods for setting and getting parent-to-child relations.

### ***Model Layer***

We use a policy vocabulary to configure the PAP Generation Framework. The policy vocabulary is an instantiated policy template model. Actually, the **PolicyVocabulary** is an element in the template sub-model and includes SLPTs and corresponding ILPTs for an application domain. The SLPTs are needed for presenting them on the user interface. The ILPTs are used as transformation rules for generating concrete machine-understandable policies from instantiated SLPTs.

Following the model-view-controller approach, access to the model elements is only granted to the respective controller elements.

### ***Policy Language Layer***

The policy language layer contains modules for the supported implementation-level policy languages. As described in Section 4.7, the policy template model is extensible by machine-understandable policy



languages based on XML. A policy language module consists of a model of language elements. ILPTs are assembled from these model elements. Thus, for creating machine-understandable policies, a PAP needs transformation rules in the form of ILPTs and the compatible policy language module.

## Controller Layer

The main purpose of the controller layer is to support the data exchange between the model and presentation layers or more specifically, between the respective elements of those layers. Therefore, for all relevant model elements, corresponding controllers exist. Figure 40 depicts all controller elements and their relation to the controller interfaces. The relation between elements on the controller layer depends on the concrete instantiation of the PAP based on the policy vocabulary being used.

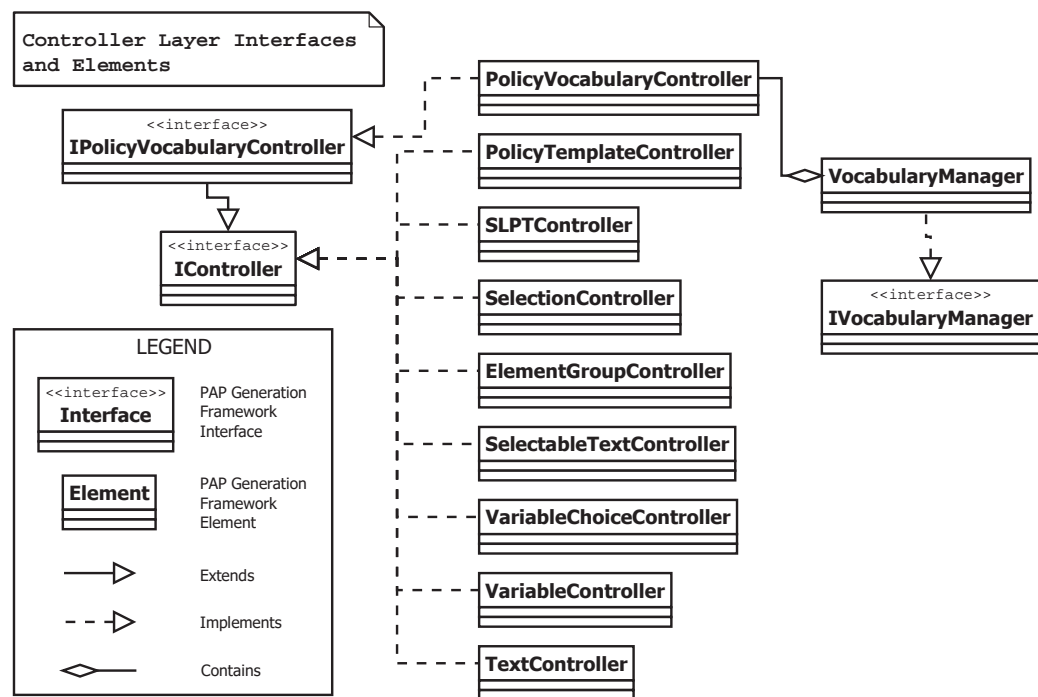


Figure 40: Interfaces and Elements of the Controller Layer

The following interfaces exist:

- The **IController** interface is implemented by all controllers except the policy vocabulary controller. It requires the following functionality. First, the initialization of each policy template is triggered via the policy vocabulary controller and passed through all child elements. Second, a controller can be triggered by a presenter to update data in the model based on user input. Third, the controller triggers the input validation. Last, the respective values for the generation of SLPs and ILPs from the model are provided.

- The **IPolicyVocabularyController** interface extends the IController interface and mainly requests functionality for the generation of SLPs and ILPs. In addition, a PAP listener can be registered to accept callbacks from the surrounding software component.
- The **IVocabularyManager** interface requires functionality for the selection of view and presentation modules and for the initialization of the entire PAP based on a policy vocabulary. Additionally, the interface requires functions for providing references to the PolicyVocabularyController and the corresponding PolicyVocabularyPresenter from the policy specification user interface to the PAP.

The developer must implement exactly one PolicyManager and one PolicyVocabularyController. Those are the main elements of the generated policy specification interface of a PAP.

### **PAP Listener**

The generated policy specification interface is embedded into a PAP. This PAP has to implement the IPAPListener interface in order to accept callbacks from the policy specification user interface. Two functions need to be provided:

- After policy specification, the generated SLPs and ILPs are handed over to the PAP. These policies can then, for example, be stored or deployed.
- On invalid user input, error messages are handed over to the PAP.

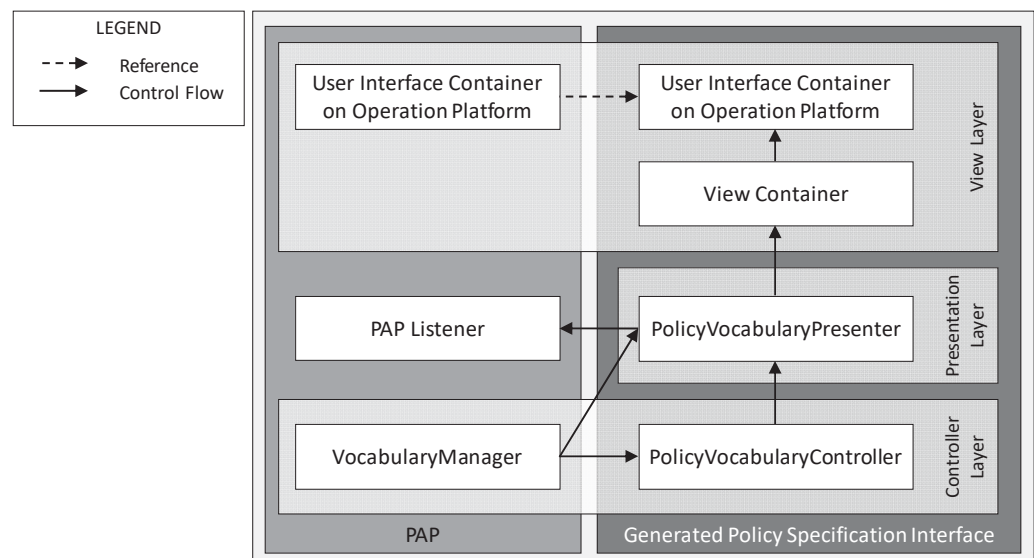


Figure 41: Concept for Embedding a Generated Policy Specification Interface into a PAP

### 5.2.2 Concept for Embedding a Generated Policy Specification Interface into a PAP

A PAP can embed a policy specification user interface that is generated with the PAP generation framework. To adopt a generated user interface, the developer has to perform three tasks:

- The VocabularyManager need to be initialized in the PAP. For the initialization, a policy vocabulary need to be assigned to the VocabularyManager. In addition, the modules for view and presentation must be selected.
- The developer registers a PAP Listener at the VocabularyManager. This listener provides call back functions used by the PolicyVocabularyPresenter.
- The developer needs to embed the parent user interface container from the view layer of the policy specification interface into a UI container of the PAP.

An overview of the embedding concept is shown in Figure 41

## 5.3 Specification Paradigms

### 5.3.1 Selection of Specification Paradigms

In Section 2.5, we identified many different PAPs in the literature and practice that aim to be used by different types of users (e.g., non-experts, administrators) for the specification of security or privacy policies. We derived eight specification paradigms from those tools. It would be beneficial to compare all of those paradigms with respect to usability. However, in order to generate PAPs using those specification paradigms, we would need to implement all of them as presentation modules for our PAP generation framework. As we only have limited resources and the comparison of too many specification paradigms in the evaluation is expedient (e.g., participation in experiments takes too long), we decided to select and implement four representative presentation modules. We identified two characteristics of PAPs in the literature that guided us in assigning the eight specification paradigms to four classes. For each class, we selected one specification paradigm as a representative. The two characteristics are:

- **Expressiveness:** The user needs to make a series of decisions during the specification of a policy with a PAP. We call the number of required decisions the expressiveness of the PAP. On the one hand, a high expressiveness lets users adapt the policies in a more fine-grained manner to their personal security and privacy preferences. On the other

hand, a limited expressiveness lets users focus on the essential decisions and potentially decreases the error potential. Many paradigms with low expressiveness let the user just select from predefined policies.

- **Guidance:** During the specification of policies, the user can be supported in decision making by appropriate additional information, hints, recommendations of the next decision to make, notifications about potential mistakes and many other guidance mechanisms. PAPs can massively influence the process of policy specification by providing a high level of guidance or by leaving the user to his own devices by offering a low level of guidance.

Our decision to differentiate the specification paradigms according to these two characteristics is substantiated in the literature. In 2010, Johnson et al. proposed »new guidelines for usable policy authoring« [25]. They recommend an »appropriate limitation of expressiveness« [25] to optimize PAP usability. Note, however, that the suitability of the expressiveness depends on the user and his skills. We reflect this guideline in our scale »expressiveness«. There, we distinguish between specification paradigms with low expressiveness that allow only the selection of pre-defined policies and paradigms with high expressiveness that let users instantiate policies within given specification boundaries.

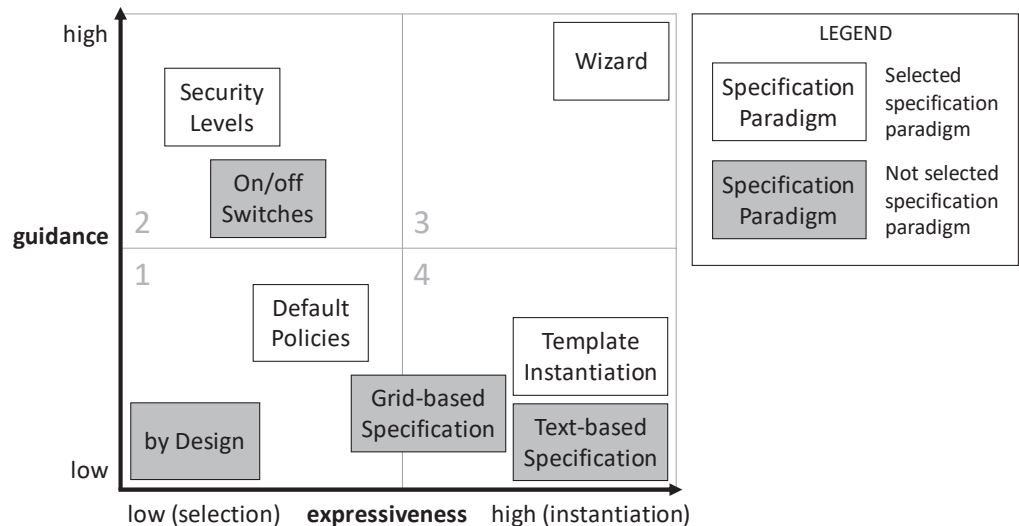


Figure 42: Selection of Specification Paradigms

Johnson et al. also require the provision of access to metadata [25]. That is, they argue that a PAP should provide the users all context information that is necessary to make informed decisions. We partially reflect this guideline in our scale »guidance«. However, we see a need for a broader differentiation. Apart from providing comprehensive information for decision-making, the process of transforming this information into personal policies also needs to be considered.

In addition, Kuo [35] and Boyd [40] consider the user's knowledge as an important requirement for users to specify privacy settings. Thus, the policy specification interface should be usable for the user with his level of knowledge. This motivates our selection of the two characteristics: Missing knowledge can be compensated by limiting the expressiveness and by increasing the guidance.

We assessed the seven specification paradigms that we derived from existing PAP designs along the scales »guidance« and »expressiveness« (see Figure 42). We divided the data area of the diagram into four quadrants and selected one specification paradigm per quadrant. For every specification paradigm selected, we implemented a presentation module for our PAP generation framework:

- **Quadrant 1 – Low guidance, low expressiveness:** The paradigm with the lowest guidance and expressiveness is »security and privacy by design«. The user has no decisions to take, as policies are predefined and immutable. The paradigm »default policies« has very limited expressiveness. The user can only select from a list of predefined policies; this selection is the only choice the user can make. The level of guidance depends on the concrete implementation of this specification paradigm. However, we rate the guidance low compared to other paradigms with low expressiveness.
- **Quadrant 2 – High guidance, low expressiveness:** We rate the paradigms »security levels« and »on/off switches« to have limited expressiveness, as they base on the selection of predefined policies. For the »security levels«, the specification process is very simple: The users just have to select their preferred level. The additional hints and information supporting the users in their decision depend on the concrete implementation. They are supposed to be appropriate for meaningful decision-making.
- **Quadrant 3 – High guidance, high expressiveness:** We assigned the »wizard« as the only specification paradigm to this quadrant. The »wizard« strictly controls the sequence of decisions to be made by the user. Moreover, decisions are split into small steps. Explanations and hints that support the user in the decision-making process are provided at each step. The »wizard« provides a high expressiveness due to many fine-grained decisions in the specification steps. User can generate their individual policy from a variety of decision options.
- **Quadrant 4 – Low guidance, high expressiveness:** The remaining three specification paradigms, »template instantiation«, »grid-based specification« and »text-based specification«, all provide high expressiveness. The »template instantiation« is based on the generation of policies from a policy template with a graphical user interface. The »grid-based specification« allows the user to combine

predefined parts of the policy in a grid view. The »text-based specification« allows the user to assemble words to a policy based on a specific grammar, provided by, for example, a controlled language. Compared to the other paradigms that are based on the selection of predefined policies, the specification process of these paradigms is less restricted and less guided.

In quadrant 1, we chose the paradigm »**default policies**« for implementation in the PAP generation framework, as it provides an actual specification option to the user. Altogether, we derived three specification paradigms that are based on the selection of predefined policies: »default policies«, »on/off switches« and »security levels«. Among these, we picked those two with the highest and lowest guidance for implementation. Thus, we selected the »**security levels**« in the second quadrant. In our opinion, the »**wizard**« is the specification paradigm with the strongest guidance; therefore we selected it as the third specification paradigm for the PAP generation framework. In the fourth quadrant, we voted for the »**template instantiation**«. Johnson et al. positively evaluated the use of policy templates for the process of policy specification for non-experts and suggested to this paradigm [10]. We followed their suggestion and compared this specification paradigm with the others in our evaluation.

In summary, we selected the specification paradigms »default policies«, »security levels«, »wizard« and »template instantiation« for implementation in the PAP generation framework and for evaluation with respect to Hypothesis 1 (usability of specification paradigms; see Section 1.5.1).

### 5.3.2 Specification Paradigm Algorithms

Based on the selection of specification paradigms in the previous section, we developed presentation modules for our PAP generation framework. A presentation module contains a generation algorithm for transforming the information from the policy vocabulary into the graphical user interface representing the specification paradigm.

Below, we describe the generation algorithms for all four selected specification paradigms. For each paradigm, we provide simplified pseudocode explaining the generation algorithm and a mockup showing a simplified user interface resulting from the specification paradigm.

#### ***Default Policies***

The specification paradigm »default policies« provides the user a list of predefined policies per policy template. The user selects a policy template and the PAP shows the respective list of predefined policies. The user

selects the preferred policy and saves it. Figure 43 shows a mockup of a PAP with the specification paradigm »default policies«.

The mockup shows a window titled "Specification Paradigm »Default Policies«". On the left is a sidebar with five categories: "Context-based device security settings", "Access to financial data in different situations", "Mass retrieval of data", "Execution of group analyses", and "Automatic deletion after theft". The main area displays four policy rules, each with a radio button for selection:

- ☐ If a financial advisor is about to access financial data of clients on a business trip and already accessed 10000 data records this week, then allow access and inform mr.blue@bank.de.
- ☒ If a financial advisor is about to access financial data of clients on a business trip and already accessed 100 data records today, then inhibit access and inform mrs.black@bank.de.
- ☐ If a financial advisor is about to access financial data of clients on business trip and already accessed 10 data records this hour, then inhibit access.
- ☐ If a financial advisor is about to access financial data of clients in home office and already accessed 200 data records this hour, then inhibit access and inform mr.blue@bank.de.

A "Save Policy" button is located at the bottom right of the main area.

Figure 43: Mockup of Specification Paradigm »Default Policies«

The PAP generates the list of default policies from the respective information stored in the policy vocabulary (instantiated specification paradigm projection sub-model and instantiated specification-level sub-model). The policy vocabulary contains a definition of default policies per policy template. Each default policy consists of a set of values for each variable element of the policy template (ReferenceableSlptElement; i.e. variable, selectable text or selection). Those values are used to instantiate the respective policy template.

Listing 6 shows the pseudocode for the user interface generation for the specification paradigm »default policies« in a simplified form. We did not include any calls for the styling of the UI components.

Listing 6: Pseudocode of Generation Algorithm for Specification Paradigm »Default Policies«

```

create layout container 1 with horizontal orientation
get list of policy templates from policy template model
create list viewer containing list of policy templates
create layout container 2 with vertical orientation
add list viewer and layout container 2 to layout container 1
on click on policy template in list viewer do
  ..get clicked policy template from policy template model
  ..initialize policy template controller for policy template
  ....create selection viewer with vertical orientation
  ....get default policies for policy template from policy template model
  .....for each default policy do
  .....instantiate policy template with values from default policy

```



```

.....generate SLP of default policy with policy template controller
.....create text viewer
.....set SLP as text in text viewer
.....add text viewer to selection viewer
....add selection viewer to layout container 2
create button for saving policy
add button to layout container 2

```

## Security Levels

The specification paradigm »security levels« provides the user a list of predefined policy sets (each representing a different security level) containing instances of multiple policy templates. The user selects the set of policies that matches his preferred security or privacy level. By choosing a security level, all corresponding policies are selected and can be saved. Figure 44 shows a mockup of a PAP with the specification paradigm »security levels«.

Select the security and privacy level of your preference:

☐ High

- If a financial advisor is about to access financial data of clients on a business trip and already accessed 10 data records this hour, then inhibit access.
- Inhibit group analysis outside the bank.
- Brick device if a theft is reported.

☒ Medium

- If a financial advisor is about to access financial data of clients on a business trip and already accessed 200 data records this today, then inhibit access and inform mr.blue@bank.de.
- Perform a factory reset on the device if a theft is reported.

☐ Low

- If a financial advisor is about to access financial data of clients on a business trip and already accessed 10000 data records this week, then allow access and inform mr.blue@bank.de.
- Delete app data on the device to if a theft is reported.

Save Policies

Figure 44: Mockup of Specification Paradigm »Security Levels«

The PAP generates the security levels from the respective information stored in the policy vocabulary (instantiated specification paradigm projection sub-model and instantiated specification-level sub-model). The policy vocabulary contains a definition of all security levels in the form of a mapping of default policies to security levels. Thus, the experts decides which of the default policies per policy template he wants to assign to a



security level. Listing 7 shows the pseudocode for the user interface generation for the specification paradigm »security levels« in a simplified form. We did not include any calls for the styling of the UI components.

Listing 7: Pseudocode of Generation Algorithm for Specification Paradigm »Security Levels«

```

create layout container with vertical orientation
create text viewer containing instruction for selection of security level
add text viewer to layout container
create selection viewer
get security levels from policy template model
for each security level do
  ..get default policies for security level from policy template model
  ..for each default policy do
    ...instantiate policy template with values from default policy
    ...generate SLP of default policy with policy template controller
    ...create text viewer
    ...set SLP as text in text viewer
  ..add text viewer to selection viewer
add selection viewer to layout container

```

## Wizard

The specification paradigm »wizard« provides the user a series of small specification steps. On instantiating the policy template model, an expert defines the order of these specification steps and their relation to the policy templates. Thus, by carrying out the predefined control flow of the specification steps, the user specifies a set of policies. Figure 45 shows a mockup of a PAP with the specification paradigm »wizard«.

Figure 45: Mockup of Specification Paradigm »Wizard«

The PAP generates the wizard from the respective information stored in the policy vocabulary (instantiated specification paradigm projection sub-

model and instantiated specification-level sub-model). The policy vocabulary contains a definition of the wizard pages and their order. Each page contains an ordered list of descriptive texts and references to the variable elements (ReferenceableSlptElement; i.e. variable, selectable text or selection) of the policy templates stored in the policy vocabulary. For each wizard page, the generation algorithm creates viewer components based on this list. Listing 8 shows the pseudocode for the user interface generation for the specification paradigm »wizard« in a simplified form. We did not include any calls for the styling of the UI components.

Listing 8: Pseudocode of Generation Algorithm for Specification Paradigm »Wizard«

```

create layout container 1 with vertical orientation
create layout container button with horizontal orientation
create button for getting to the previous step
add button for getting to the previous step to layout container button
get list of template references from wizard definition
for each template reference do
  ..resolve template reference and get policy template
  .....from policy template model
  ..initialize policy template controller for policy template
  ..get list of wizard pages from policy template
  ..for each wizard page do
    ....get list of wizard page elements of wizard page
    .....for each wizard page element do
      .....if wizard page element is wizard page description then
        .....create text viewer
        .....set description of wizard page description to text viewer
        .....add text viewer to layout container 1
        .....else if wizard page element is SLPT element
          .....call handleSlptElements(wizard page element, layout container 1)
    ....create button for getting to current step
    ....add button for getting to current step to layout container button
create button for getting to the next step
add button for getting to the next step to layout container button
add layout container button to layout container 1

function handleSlptElements(SLPT elements, container)
begin
  ..for each SLPT element do
    ....if SLPT element is selection then
      .....call handleSelection(SLPT element, container)
    ....else if SLPT element is text then
      .....call handleText(SLPT element, container)
    ....else if SLPT element is variable then
      .....call handleVariable(SLPT element, container)
end

function handleSelection(selection, container)
begin
  ..create selection viewer
  ..if only one element group of selection can be selected then
    ....set selection viewer to radio button mode
  ..else
    ....set selection viewer to check box mode

```

```
..get list of element groups from selection
..for each element group do
...create layout container s with horizontal orientation
...get list of SLPT elements from element group
...call handleSlptElements(SLPT elements, layout container s)
...add layout container s as item to selection viewer
..add selection viewer to container
end

function handleVariable(variable, container)
begin
..create variable viewer
..set value of variable to variable viewer
..set variable type of variable to variable viewer
..add variable viewer to container
end

function handleSelectableText(selectable text, container)
begin
..create selectable text viewer
..add variable choices of selectable text as items to selectable text viewer
..add selectable text viewer to container
end
```

### ***Template Instantiation***

The specification paradigm »template instantiation« provides the user a list of policy templates. The user selects a policy template and the PAP shows the respective template. The user fills all forms according to his security and privacy preferences. Finally, the user saves the policy. Figure 46 shows a mockup of a PAP with the specification paradigm »template instantiation«.

The generation algorithm creates and assembles viewer components based on the policy templates in the policy vocabulary (instantiated specification-level sub-model). In addition, more information from the policy vocabulary is added to the respective viewers, such as descriptive hint texts. Listing 9 shows the pseudocode for the user interface generation for the specification paradigm »template instantiation« in a simplified form. We did not include any calls for the styling of the UI components.

The mockup shows a window titled "Specification Paradigm »Template Instantiation«". On the left is a sidebar with five buttons: "Context-based device security settings", "Access to financial data in different situations", "Mass retrieval of data", "Execution of group analyses", and "Automatic deletion after theft". The main area contains a policy rule configuration for "Access to financial data in different situations". The rule is: "If a financial advisor is about to access financial data of clients". Below this is a selection box with three radio buttons: "inside the bank", "on a business trip", and "in home office" (which is selected). This is followed by the text "and already accessed" with a text input "100", "data records" with a dropdown menu showing "today", and ", then". Below this is another selection box with four radio buttons: "allow access", "allow access and inform" (with a text input "email address"), "inhibit access", and "inhibit access and inform" (with a text input "mrs.black@bank.de" and selected). At the bottom right is a "Save Policy" button.

Figure 46: Mockup of Specification Paradigm »Template Instantiation«

Listing 9: Pseudocode of Generation Algorithm for Specification Paradigm »Template Instantiation«

```

create layout container 1 with horizontal orientation
get list of policy templates from policy template model
create list viewer containing list of policy templates
create layout container 2 with vertical orientation
add list viewer and layout container 2 to layout container 1
on click on policy template in list viewer do
  ..get clicked policy template from policy template model
  ..initialize policy template controller for policy template
  ..get list of SLPT elements from policy template
  ..create layout container 3 with horizontal orientation
  ..call handleSlptElements(list of SLPT elements, layout container 3)
add layout container 3 to layout container 2
create button for saving policy
add button to layout container 2

function handleSlptElements(SLPT elements, container)
begin
  ..for each SLPT element do
    ....if SLPT element is selection then
      .....call handleSelection(SLPT element, container)
    ....else if SLPT element is text then
      .....call handleText(SLPT element, container)
    ....else if SLPT element is variable then
      .....call handleVariable(SLPT element, container)
    ....else if SLPT element is selectable text then
      .....call handleSelectableText(SLPT element, container)
end

```

```
function handleSelection(selection, container)
begin
  ..create selection viewer
  ..if only one element group of selection can be selected then
    ...set selection viewer to radio button mode
  ..else
    ...set selection viewer to check box mode
  ..get list of element groups from selection
  ..for each element group do
    ...create layout container s with horizontal orientation
    ...get list of SLPT elements from element group
    ...call handleSlptElements(SLPT elements, layout container s)
    ...add layout container s as item to selection viewer
  ..add selection viewer to container
end

function handleText(text, container)
begin
  ..create text viewer
  ..set value of text to text viewer
  ..add text viewer to container
end

function handleVariable(variable, container)
begin
  ..create variable viewer
  ..set value of variable to variable viewer
  ..set variable type of variable to variable viewer
  ..add variable viewer to container
end

function handleSelectableText(selectable text, container)
begin
  ..create selectable text viewer
  ..add variable choices of selectable text as items to selectable text viewer
  ..add selectable text viewer to container
end
```

## 5.4 Reference Implementation

We developed a fully functional PAP generation framework that implements the proposed reference architecture. An overview of the modules currently provided is given in Figure 47.

We support four UI frameworks with respective modules: »Swing« and »JavaFX« for Java desktop clients, »Android« for mobile clients and a »Web« user interface. For the presentation layer, we developed modules implementing the specification paradigms »template instantiation«, »wizard«, »default policies« and »security levels« with the generation algorithms described in the previous section. We tested the generation of PAPs in several application domains and provided corresponding policy vocabularies. Examples are:

- Security and privacy demands of bank clients in the context of the case study »SINNODIUM« (see Section 8.2)
- Security demands regarding critical cloud infrastructure solutions in the case study »SECCRIT« (see Section 8.3)
- Security demands of employees and employers with respect to information classification and data protection in a business context in the case study »BeSure« (see Section 9.2)
- Privacy demands of citizens as users of apps for smart rural areas in the context of the case study »Digital Villages« (see Section 9.3)

For further illustration, we show an exemplary PAP instantiations using the policy vocabulary from an IND<sup>2</sup>UCE demonstrator »Construction Site 4.0 (CS4)« in the following. In this example, support the machine-understandable IND<sup>2</sup>UCE policy language for the generation of ILPs in the versions 1.1 and 3.0.46 as well as its productive equivalent MYDATA policy language in version 4.0.

|  |   |  |                                     |                              |
|--|---|--|-------------------------------------|------------------------------|
| <b>View layer:</b><br>Platform Components                      | JavaFx                                      | Web                                    | Swing                               | Android                      |
| <b>Presentation layer:</b><br>Policy Specification Paradigms   | Template Instantiation                      | Wizard                                 | Default Policies                    | Security Levels              |
| <b>Model layer:</b><br>Policy Template Model Instance          | Digital Villages                            | BeSure                                 | SECCRIT                             | CS4                          |
| <b>Policy language layer:</b><br>Transformation Rules          | MYDATA Language (4.0)                       | IND <sup>2</sup> UCE Language (3.0.46) | IND <sup>2</sup> UCE Language (1.1) | (XML-based Policy Languages) |
| <b>Controller layer:</b><br>Interfaces and Basic Functionality | Generic PAP Generation Framework Controller |  |                                     |                              |

Figure 47: Current Modules in the Reference Implementation of the PAP Generation Framework

### View Layer

We developed four view modules. For each module, we had to implement all required interfaces in the form of view elements. We identified the UI framework components that best cover the functionality required by the interfaces and embedded them in the view elements. We added missing functionality that is not provided by the UI components. Table 11 lists examples of UI components embedded into the view elements in the four implemented view modules.

Table 11: Examples for Mapping of View Elements with UI Framework Components

| View Element         | JavaFx (javafx.scene.*)                                  | Web (HTML)  | Swing (javax.swing.*)                      | Android (android.widget.*)                |
|----------------------|--|---|--|---|
| TextViewer           | control.Label  | <span>  | JLabel                                     | TextView                                  |
| SelectionViewer      | layout.VBox with control.RadioButton or control.CheckBox | <fieldset><input /></fieldset> with input types radio or checkbox | ButtonGroup with JRadioButton or JCheckbox | LinearLayout with RadioButton or CheckBox |
| SelectableTextViewer | control.ComboBox   | <select>  | JComboBox                                  | Spinner                                   |
| VariableViewer       | control.TextField  | <input>   | JFormattedTextField                        | EditText                                  |
| ButtonViewer         | control.Button   | <button>  | JButton                                    | Button                                    |
| ListViewer           | control.ListView   | <div>   | JList                                      | ListView                                  |
| LayoutContainer      | layout.FlowPane or layout.VBox                           | <div>   | JPanel                                     | LinearLayout                              |

### **Presentation Layer**

We developed presentation modules for each of the four specification paradigms »template instantiation«, »wizard«, »default policies« and »security levels«. In each module, we implemented the respective specification paradigm algorithms described in Section 5.3.2.

### **Policy Specification Interface Generation at runtime**

The PAP generation framework generates the user interface for the specification of policies. Three aspects affect the generation: the view module, the presentation module and the policy vocabulary.

There are two points in time at which the policy specification interface of a PAP could be generated:

- At **development time**: The developer generates an instance of the PAP with a fixed specification paradigm and with an immutable user interface.
- At **runtime**: The developer creates an instance of the PAP that supports all specification paradigms and that generates the policy specification user interfaces at runtime.

We chose **runtime generation** because it offers higher flexibility for users. The developer creates one instance of the PAP. Thus, it supports one UI framework and all specification paradigms. The user interface of the PAP is generated at runtime based on the selected specification paradigm (presentation module) and the selected policy vocabulary. The developer can decide how the selection of both modules is realized. Both

modules can be selected, for example, via a configuration file or via the user interface. The latter allows a selection by the user itself.

We realized the UI generation at runtime with two concepts and respective technologies:

- To achieve customization to the application domain, we load a policy vocabulary during PAP instantiation. We realized the exchangeability of model modules with the concept **XML binding and unmarshalling**. First, class models representing the policy template model and the policy language for ILPs need to be generated from the respective XML schema files. We use the technology »Java Architecture for XML Binding (JAXB)« [161] with its implementation »Eclipselink MOXy« [162]. At runtime, we bind XML elements from the policy vocabulary to objects in the model module using the same technology. To this end, XML elements from the policy vocabulary are deserialized and respective objects in the model module are instantiated during the initialization of the PAP. Eventually, the model module contains an object tree that represents the XML element structure of the policy vocabulary. After the user has specified an SLP, the PAP can generate a corresponding ILP, provided that the respective transformation rules are defined in the policy vocabulary. These rules are applied to the part of the object tree representing the ILP. The final generation of an ILP is realized as a serialization of this part of the object tree with XML unmarshalling. The output is an XML policy.
- To achieve the flexible selection of presentation modules at runtime, we use **dependency injection**. Dependency injection allows a dynamic binding of objects at runtime. According to our reference architecture, controller, model and presenter elements have an inheritance relation. At development time, it is unclear which concrete presenter the corresponding model element needs to inherit from, as it is not known which presentation modules is used. Presentation modules may even be replaced by others at runtime. Thus, during the initialization of the policy vocabulary, a presenter stub is instantiated and bound to the concrete presentation element after the selection of the presentation module. During the initialization of the selected presentation module, the user interface is created. Therefore, the view elements are also dynamically bound to the presenter elements. The concept of element binding is illustrated in Figure 48. To implement the concept, we use the Google Guice [163] dependency injection framework.



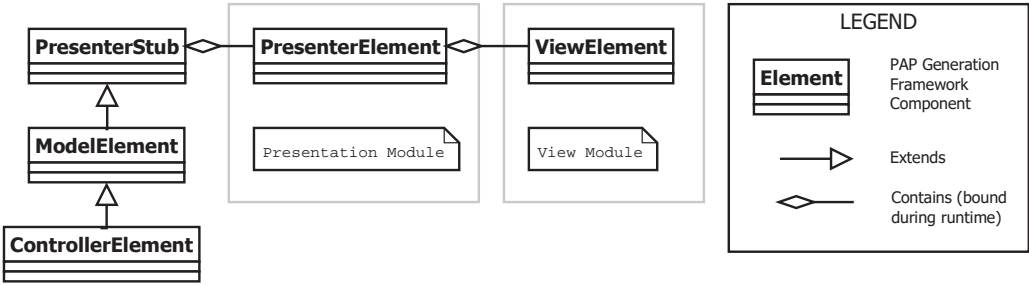


Figure 48: Injection of Presentation Elements at runtime

**Example PAP Framework Instantiations**

In the following, we show some example PAPs that are created with the PAP generation framework based on the UI framework »JavaFX«. Other exemplary instantiations of PAPs created with the PAP generation framework can be found in the descriptions of the four case studies in Section 8.2.3, Section 8.3.3, Section 9.2.3 and Section 9.3.3.

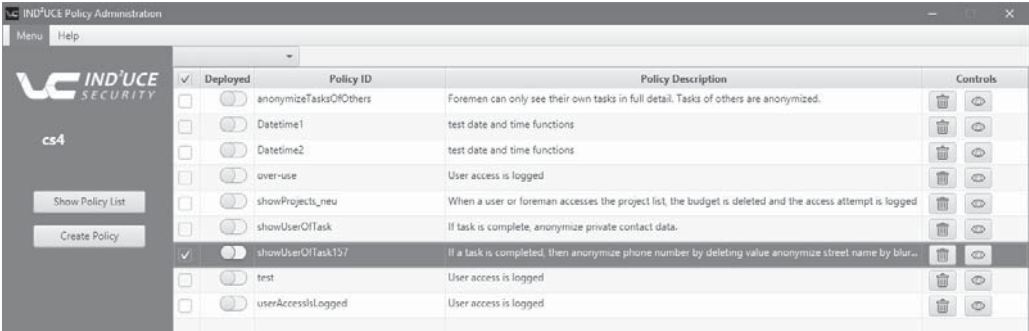


Figure 49: Policy Editor in UI Framework »JavaFX« that Embeds a PAP and Supports Policy Management Functionality

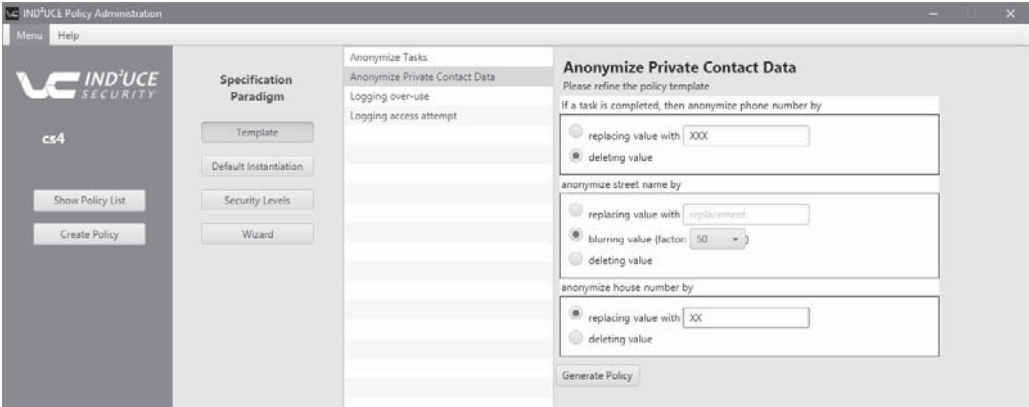


Figure 50: Exemplary PAP Using View Module »JavaFX«, Policy Vocabulary »CS4« and Presentation Module »Template Instantiation«

Figure 49 shows our JavaFX PAP, which imported the policy vocabulary »CS4«. The use of the PAP with the presentation module »template instantiation« is shown in Figure 50. A click on the »Generate Policy« button instructs the PAP to generate an ILP from the instantiated policy

template. The resulting ILP is visible in the editor, as shown in Figure 51. The ILP is based on the policy language »MYDATA Version 4.0«.

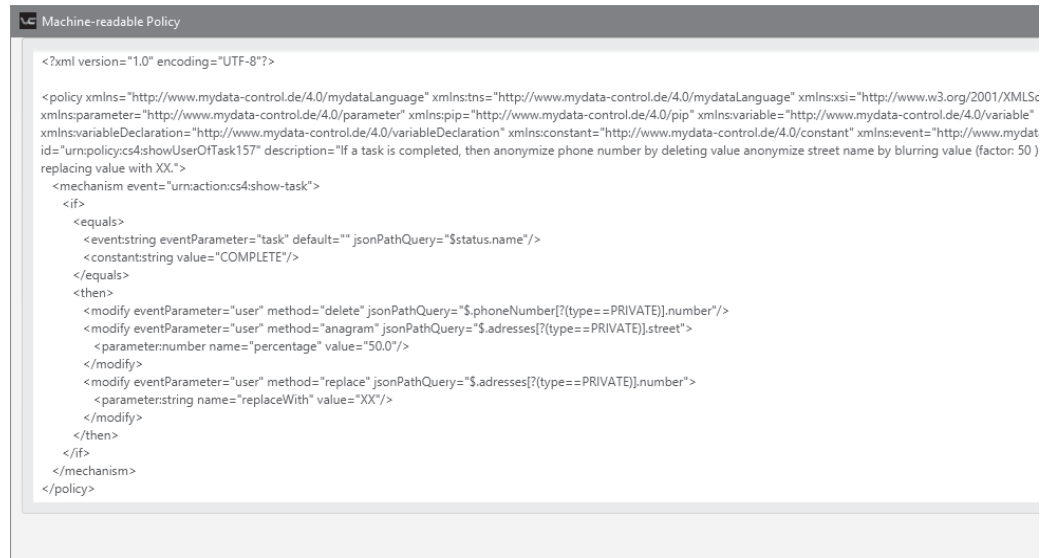


Figure 51: ILP in MYDATA Policy Language Version 4.0 Generated by the PAP in UI Framework »JavaFX«

The JavaFX PAP instance also allows the user to specify policies with different specification paradigms. The use of the »wizard« is illustrated in Figure 52, the »default policies« in Figure 53 and the »security levels« in Figure 54.

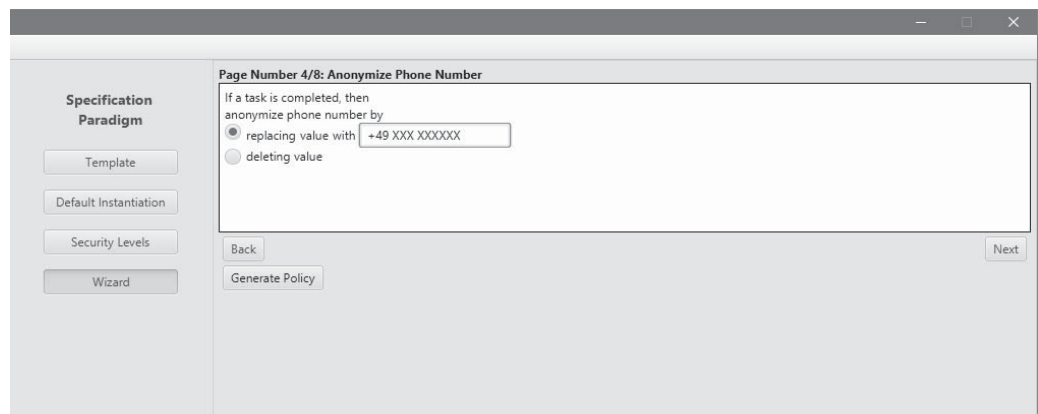


Figure 52: Example PAP using View Module »JavaFX«, Policy Vocabulary »CS4« and Presentation Module »Wizard«

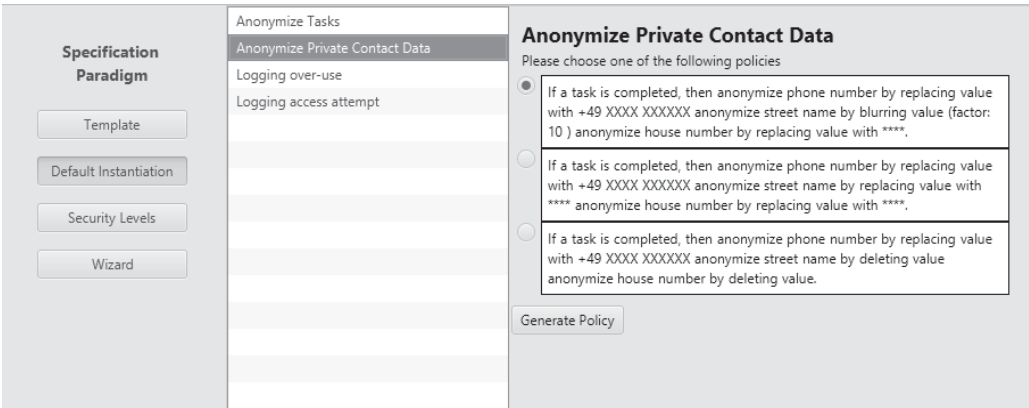


Figure 53: Example PAP using View Module »JavaFX«, Policy Vocabulary »CS4« and Presentation Module »Default Policies«

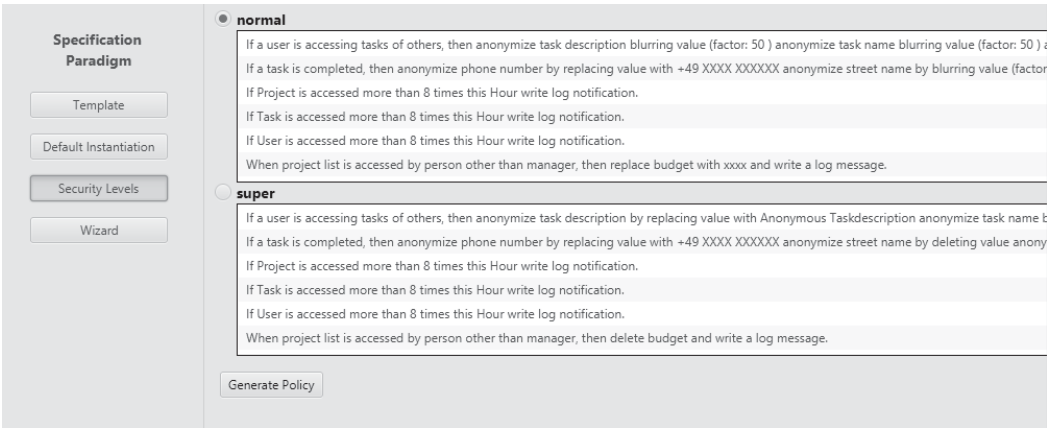


Figure 54: Example PAP using View Module »JavaFX«, Policy Vocabulary »CS4« and Presentation Module »Security Levels«

## 5.5 Summary and Conclusion

In this chapter, we presented our concept and reference implementation of a PAP generation framework. We also explained the selection of specification paradigms for our implemented presentation modules.

To conclude the discussion and to reflect on our achievements, we briefly review the fulfillment of the requirements for the PAP generation framework, which we defined in Section 5.1:

- **Req\_Framework\_UI-Generation:** The policy specification user interfaces in a PAP can be generated by using the PAP generation framework.
- **Req\_Framework\_Modularity:** We built a modular framework with the five layers: controller, presentation, view, model and policy language. Except for the controller layer, the developer can implement multiple modules for each layer and choose the desired modules at runtime.

- **Req\_Framework\_Policy-Templates:** The specification paradigm generation algorithms take all policy templates from a policy vocabulary and represent them on the user interface. To this end, the different algorithms generate the user interfaces for the policy templates or elements from the policy templates, respectively, in different ways, depending on the specification paradigm.
- **Req\_Framework\_Specification-Paradigms:** We created generation algorithms for four specification paradigms and extended the policy template model accordingly (specification paradigm projection sub-model). The user interfaces of the PAP are generated at runtime. Therefore, users can select their preferred specification paradigm at runtime. The PAP can also be configured to use a default specification paradigm or to provide a limited set of specification paradigms for selection. We implemented the four specification paradigms »default policies«, »security levels«, »wizard« and »template instantiation«.
- **Req\_Framework\_Multi-User-Interface-Frameworks:** The PAP generation framework supports multiple UI frameworks. This is achieved with the view layer as an abstraction of the UI frameworks being used. We demonstrated the use of multiple UI frameworks by implementing view modules for »Swing«, »JavaFX«, »Web« and »Android«.
- **Req\_Framework\_Policy-Transformation:** Our PAP is capable of applying the transformation rules provided in policy vocabulary (implementation level template sub-model) to an instantiated policy template using the JAXB marshalling. The outcome is a machine-understandable XML policy. Multiple policy languages can be supported. However, the transformation rules in a policy vocabulary can only be defined for one specific policy language.



## 6 Mapping Users to Specification Paradigms

We ascertained in the problem derivation surveys that users have problems with existing PAPs regarding usability. In this thesis, we propose to solve the usability issues by mapping the specification paradigm to a user that offers him the best usability. Our mapping represents Contribution 1 of this thesis (see Section 1.4).

We enhance policy specification effectiveness by lowering the users' specification mistakes and by increasing the precision of the users' self-evaluation of mistakes made. In addition, we improve the effectiveness by increasing the speed of specification and the satisfaction that the user experience during the specification. In this chapter, we explain our contribution to the mapping of specification paradigms to users.

A recent study by Zhao et al. [11] shows that existing approaches lack understanding of the user group. Thus, we need to consider the resources of users for the task of policy specification. Therefore, we also evaluate the mapping of usability effects of specification paradigms on different user groups represented by personas.

We structured the remainder of this chapter as follows. In Section 6.1, we explain our research approach. Next, we state our assumptions regarding the mapping of specification paradigms to users and their basis in Section 6.2. Section 6.3 describes and justifies our assumptions regarding a mapping of specification paradigms to personas. Section 6.4 summarizes and concludes this chapter.

### 6.1 Research Approach

We applied a two-step approach to our research problem:

First, we identified obstacles that a user may face with different specification paradigms. To this end, we investigated characteristics of users and mapped them on characteristics of the specification paradigms. In order to identify relevant user characteristics, we needed a better understanding of user behavior regarding PAP use. Therefore, we surveyed psychological models describing the user behavior. We built a user intention model aligned to existing behavior models that explains the user behavior observed in the problem derivation surveys. We clarified influences of characteristics of different specification paradigms on usability. Our studies also revealed that PAP users sometimes face

fundamental barriers that hinder them to start the policy specification at all. We identified potential obstacles as discrepancies between available and expected user resources. Based on these findings, we finally derived assumptions about the best matching specification paradigms for users.

Next, we empirically validated our assumptions in the policy specification experiment by measuring the effect of different specification paradigms on the specification process. We measured effectiveness, efficiency and satisfaction for the use of all specification paradigms. We reviewed the results and derived recommendations for specification paradigms. We also evaluated whether the mapping of specification paradigms to different personas differs from the mapping to the entire participant population. For this experiment, we selected a persona model that best matched our needs. The experiment is described in Section 9.4.

We derived the following requirements for our mapping of users to specification paradigms from our Hypothesis 1 (see Section 1.5.1):

- **Req\_Mapping\_User-Characteristics:** We need a better understanding of PAP users in order to map suitable specification paradigms to them, as we require in Hypothesis 1. Therefore, we need to determine the relevant characteristics of users that affect the usability of PAPs with different specification paradigms.
- **Req\_Mapping\_Specification-Paradigms:** We need to identify key characteristics of specification paradigms in order to match them to the identified characteristics of users, as we require in Hypothesis 1.
- **Req\_Mapping\_Personas:** We assume that the mapping of specification paradigms to individual personas representing more homogeneous user subgroups achieves better results with respect to increased usability than a mapping to the entire, heterogeneous user population. Thus, we need to find an appropriate user group or persona model to cluster users into representative user groups. Then, we can propose a mapping for increasing the usability of PAPs based on the characteristics of specification paradigms and user groups.

## 6.2 Mapping Specification Paradigms to Users

We want to improve the usability of PAPs for users by providing the best matching specification paradigm for each type of user. Therefore, we need a better understanding of the user, his characteristics and his behavior. Below, we describe a user intention model, key characteristics of users and our mapping to specification paradigms.

### 6.2.1 User Intention Model

Our problem derivation surveys revealed that users make only moderate efforts to specify their security and privacy policies. In many cases, this contradicts the user's actual need for security and privacy, which is one of the key drivers for performing security and privacy related activities. We consider the need for security and privacy a basic need of humans [142]. We concentrate on those users who are not able to carry out these tasks (i.e., specifying security and privacy policies) appropriately despite their existing needs. Thus, we ignore potential unawareness of security and privacy issues. Lacking need for security and privacy could be compensated by awareness measures, which are out of focus here.

Figure 55 shows our intention model, which is based on established models of user behavior (see Section 2.6.1). Our model abstracts existing problems (e.g., too high complexity of PAP, too much time necessary for policy specification, privacy paradox) to a generic level. The model explains the discrepancy between the user's demand for security and privacy protection (desired result) and the reality of the user ignoring his interaction options (actual behavior).

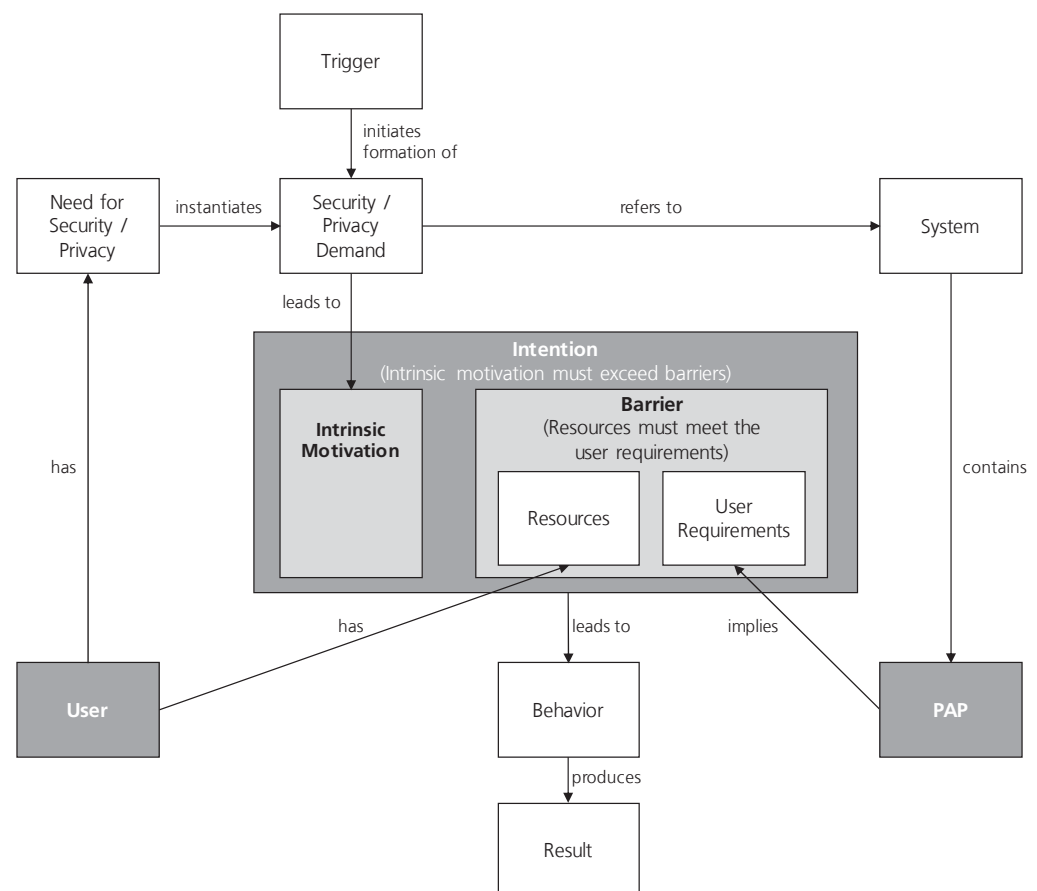


Figure 55: User Intention Model



The utility of PAPs depends on the behavior of the user. If the user does not use or does not want to use PAPs, security and privacy goals cannot be achieved. Thus, we want to achieve a specific user behavior (i.e., usage of PAPs) in order to obtain a result (e.g., specified policies). The actual behavior depends on the user's intention. In an ideal world, the user's intention is a direct consequence of his motivation. For example, as personal-identifiable information in a system directly belongs or relates to the user, he typically has an intrinsic motivation to protect it. Unfortunately, pure motivation is not the only factor influencing the intention. Barriers come into play as a counterpart to motivation. Intention arises when the user's motivation exceeds the barriers he faces. The intention leads to the behavior of specifying policies. We will refine the barriers later and focus on the motivational part first.

The motivation for using PAPs typically stems from situation-dependent security and privacy demands. These concrete demands are based on a general need for security and privacy and arise when the user experiences a certain trigger. A privacy demand could be, for instance, the desire to protect personal data from abuse in a social network or to gather information about the data usage by third parties. In comparison to the need for privacy, the privacy demand does not describe a holistic need, but it refers to a certain system. Examples for a trigger are the use of a new service, a change in the functionality of an existing service or new personal data that is requested by the service.

Barriers influence the user's intention. They emerge from the interrelation of resources available to the user and the requirements on users by the PAP. If the user has sufficient resources, he does not experience barriers. However, if the user's resources do not meet the PAP's requirements, he experiences barriers in using the PAP. As described above, the strength of barriers does not directly determine the intention, but has to be exceeded by the motivation. The instantiation of the user's resources and the requirements on the user and thus the identification of barriers strongly depends on the specific PAP or the specification paradigm being used. In our problem derivation surveys, about 30 percent of the participants responded to the question regarding the reasons for their moderate use of PAPs that these PAPs are too complicated and time-consuming (see Section 1.2). Both reasons represent barriers to specifying policies. However, even facing barriers, the user may overcome them due to his high intrinsic motivation.

We identified multiple categories of requirements, resources and barriers resulting from a discrepancy between user resources and requirements on the user: Domain knowledge, security and privacy knowledge, technical knowledge, available time, cognitive capacity and physical capacity. The identification of those categories was based on expert discussions and our

expertise in this area. In Table 12, we explain the potential discrepancies between user requirements and resources for each category.

So far, we only discussed the intention of a user to specify a policy and its relationship to the behavior of actually doing it. We also need to consider the quality of the specified policies and the specification process. In Research Question 1 of this thesis, we ask for the best specification support with respect to effectiveness, efficiency and user satisfaction, thus, in sum to usability.

However, quality is not directly depending on intention, but the discrepancy between requirements on the user and available user resources can explain usability issues. Barriers caused by missing knowledge or lack of cognitive capacity can lead to bad effectiveness. Barriers caused by discrepancies regarding available time can also cause bad efficiency.

Table 12: Barrier Categories as Discrepancies between User Requirements and User Resources

| Barrier categories           | Description of potential discrepancies between user requirements and user resources  |
|------------------------------|--|
| Domain knowledge             | Required vs. actual knowledge regarding the application domain including the service's use cases for which a policy is to be specified. This knowledge includes information about the personal data that has to be shared with the service. The user needs to understand the domain in order to be capable of making privacy-related decisions.                |
| Security & privacy knowledge | Required vs. actual knowledge of potential and actual use of personal data by the service and potential threats that arise from this use are necessary in order to be capable of making security and privacy-related decisions. This knowledge also includes that users understand the effect of countermeasures for improving their own security and privacy. |
| Technical knowledge          | Required vs. actual knowledge of the functionality of the service and its PAP.   |
| Available time               | Required vs. available time to specify policies in the PAP.  |
| Cognitive capacity           | Amount of security and privacy related information the user needs vs. is capable of processing simultaneously during the specification of policies in a PAP.   |
| Physical capacity            | Required vs actual accessibility of a device that allows the use of the PAP in the respective system.  |

Summarizing, our intention model explains the behavior of people who have a general need for security and privacy, but do not take appropriate actions to enforce it. Thus, the model contributes to the research on the so-called privacy paradox. The concept of barriers explains usability issues that users face when specifying policies with PAPs. In addition, missing motivation of users and low needs for security and privacy may explain

the disinterest in the specification of security and privacy policies, which we determined in our problem derivation surveys (see Section 1.2).

### 6.2.2 Example for Barriers of a PAP

We instantiated the user intention model for the PAP (privacy settings) provided by the social media platform Twitter. This service provides various options that are relevant from the privacy perspective. Most tweets, likes and shares are public by default. In addition, Twitter offers many privacy-relevant features, for example, for connecting the user's contact book (e.g., from Gmail) or for getting SMS notifications on the personal phone number. Although Twitter's primary purpose is the interaction with other users, and thus, the general need for privacy might be comparably low for many Twitter users, profiling, tracking and customized advertisements can be strong motivators for taking privacy protecting measures. Concrete triggers for privacy demands can stem from the Twitter use itself (e.g., visibility of sensitive tweets), reminders by Twitter (e.g., to update your phone number after login) and external triggers (e.g., press articles about Twitter).

Table 13: Potential Barriers for Users of the Twitter PAP

| Barrier categories           | Description of exemplary barriers   |
|------------------------------|---|
| Domain knowledge             | <ul style="list-style-type: none"> <li>The user does not know or does not remember the provided personal information and does therefore not know what to specify.</li> </ul>  |
| Security & privacy knowledge | <ul style="list-style-type: none"> <li>The user does not understand well enough how the personal data can be used by third parties in order to decide on his individual privacy policies.</li> </ul>  |
| Technical knowledge          | <ul style="list-style-type: none"> <li>The user is not aware of technical possibilities for tracking his usage behavior, for example via sensors on smartphones.</li> </ul>   |
| Available time               | <ul style="list-style-type: none"> <li>The Twitter PAP provides many predefined policies. It can be too time-consuming for users to read them all and to make individual decisions.</li> <li>As it is unclear which policy should be checked how often, the user would need to check all policies on every use, which is time consuming.</li> </ul> |
| Cognitive capacity           | <ul style="list-style-type: none"> <li>The PAP overwhelms the user with many options for policy selection and much textual information.</li> </ul>  |
| Physical capacity            | <ul style="list-style-type: none"> <li>Privacy settings can be configured on mobile apps and browsers and are synchronized for all devices, which could be misleading (although explicitly stated).</li> <li>Privacy policies are hidden in the app and are not optimized for navigation on mobile devices.</li> </ul>                              |

We identified potential barriers to using the Twitter PAP. The privacy settings are distributed over 15 categories, which makes it time and effort consuming to maintain them. About 30 percent of the participants in our

problem derivation surveys (see Section 1.2) responded that PAPs are too complicated and time-consuming. Both reasons represent barriers to using a PAP for specifying policies. In Table 13, we show examples for burdens we identified in the Twitter PAP mapped to the barrier categories presented in Table 12.

Of course, these potential barriers are not the result of a comprehensive evaluation, and they lack certain details. However, they should give a first impression of barriers that might exist.

### 6.2.3 Matching Specification Paradigms to Users

Discrepancies between user resources and PAP requirements on users can lead to a barrier that impairs the policy specification or hinders the user to specify policies at all. Nevertheless, even if the user's resources are slightly exceeded to resource requirements of a PAP, the experienced usability can still be very low. We assume that the higher the resources of the user are, the better the usability of a PAP will be experienced.

Table 14: Required user resources of the selected specification paradigms

| Requirements on users        | Default policies | Security levels | Wizard       | Template instantiation |
|------------------------------|------------------|-----------------|--------------|------------------------|
| Domain knowledge             | Medium           | Low             | Medium       | High                   |
| Security & privacy knowledge | Medium           | Low             | Medium       | High                   |
| Technical knowledge          | Medium           | Low             | Medium       | High                   |
| Available time               | Medium           | Low             | High         | High                   |
| Cognitive capacity           | Medium           | Medium          | High         | High                   |
| Physical capacity            | out of focus     | out of focus    | out of focus | out of focus           |

Different specification paradigms require different user resources. Barriers and bad usability experiences can occur if the user's resources are lower than the resource level considered optimal for the specification paradigm under consideration. We base the selection of specification paradigms on two characteristics; »expressiveness« and »guidance« (see Section 5.3.1). We assume that the more expressiveness a PAP provides the more security, privacy, domain and technical knowledge as well as cognitive capacity is needed. Moreover, we assume that strong guidance can lower the required user resources of specification paradigms with respect to security, privacy, domain and technical knowledge, because missing information is provided. We rated the categories of requirements on the user for the four specification paradigms of our PAP generation framework, as shown in Table 14. We did not rate the required user resources of the specification paradigms quantitatively (objectively), but only qualitatively (subjectively). We would need metrics and value

thresholds for objectively measuring the barriers, which currently do not exit. This may be a topic of future research.

In summary, we assume that users need more resources for using the specification paradigm »template instantiation« than for the other paradigms. We also assume that users need fewer resources for the specification paradigm »security level« compared to the other paradigms.

We are aware that the specification paradigm is not the only aspect influencing the requirements of a PAP on the user. The policy vocabulary and the concrete design of the user interface are definitely other dependent variables in this calculation. However, we assume that we can meaningfully compare the specification paradigms according to their requirements on user resources as long as we use the identical policy vocabulary and consult usability experts for the user interface design. The influence of the projection rules within the policy vocabulary on the paradigm requirements needs to be investigated in future work.

6.3 Mapping Specification Paradigms to Personas

In the previous section, we claimed that the requirements of a PAP must match the resources of its user. However, we know that different users have individual levels of resources. Thus, we decided to cluster users into groups in the form of personas with different resources. As a prerequisite for this user classification, we first needed to select an appropriate user model or persona model, respectively.

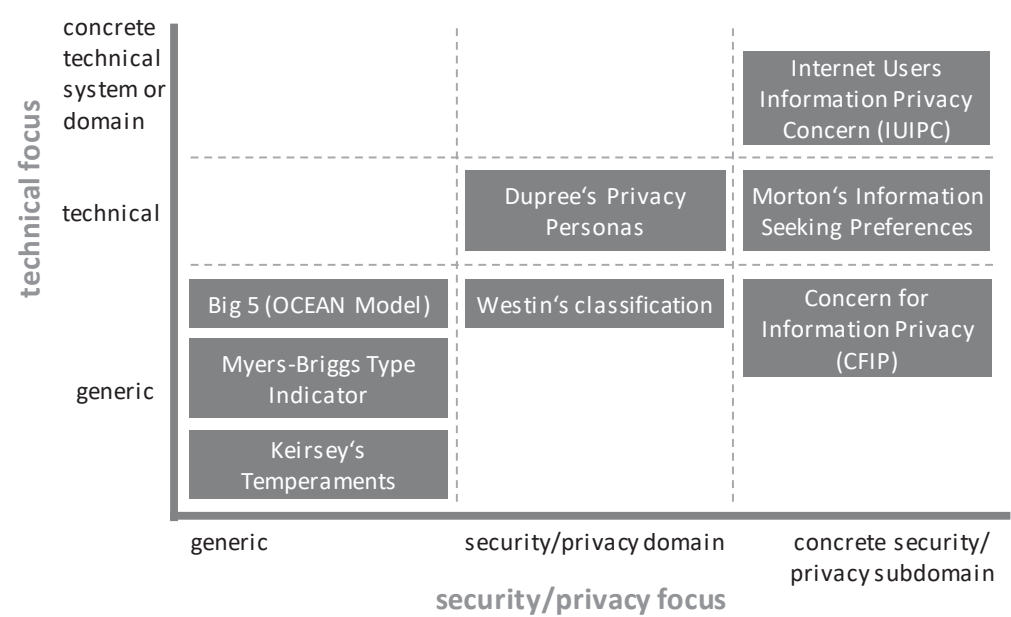


Figure 56: User Type and Persona Models

### 6.3.1 Selection of Persona Model

There are many models for user types and personas, which we surveyed in Section 2.6.2. When searching for the appropriate model for our work, we found that all relevant user group and persona models we reviewed could be characterized by two fundamental properties. They differ in their focus on IT security and privacy as well as in their focus on concrete technical systems. In both cases, there are highly specialized models developed for a specific subdomain or system, but also very generic approaches. We rated the identified models according to those two characteristics, as shown in Figure 56.

The specification of policies addresses the domain of security and privacy. Thus, we need a user group or persona model that reflects these aspects. However, we can specify policies for a multitude of application domains. Hence, we need a model that does not only reflect the users of a specific subdomain of security and privacy such as the »Internet Users Information Privacy Concern Model« [150] which only reflects internet users' concerns about collection and control of personal data in online environments. Apart from the security and privacy aspect, we focus on the technical implications of policy specifications and their enforcement. This constitutes a technical focus of our work.

We identified the persona model by Dupree et al. [14] as a good match. It focuses on the domain of security and privacy without being specialized to a subdomain and it has a technical focus on the use of security and privacy systems without being limited to one concrete system. This matches to our whole approach being applicable to multiple application domains and providing different specification paradigms to users. Thus, we chose this model for our experiment. For brevity, we call it the »Dupree model« in the remainder of this thesis. This model distinguishes users by their motivation (willingness to specify policies) and their knowledge of how to specify appropriate policies. The different personas proposed by the Dupree model are described in Appendix C.

### 6.3.2 Mapping the Specification Paradigms to the Personas of Dupree

After selecting the persona model of Dupree, we need to map our specification paradigms to its five personas. We expect an increased usability of the specification interfaces of a PAP if the user resources of the specification paradigm (compare Section 6.2.3) align to the user characteristics of the personas. However, the user resource categories (domain knowledge, security and privacy knowledge, technical knowledge, available time, cognitive capacity and physical capacity) do not directly match Dupree's categories (knowledge and motivation). We therefore map the categories of user resources to the categories of the Dupree model like follows:

- **Knowledge:** The differentiated types of knowledge of our user resources (domain knowledge, security and privacy knowledge, technical knowledge) and the cognitive capacity are condensed to Dupree's category »knowledge«. Dupree uses this category for defining the user's capabilities of using and understanding security and privacy systems.
- **Motivation:** The motivation of a persona is not directly considered in the user resource categories. Moreover, our user intention model explains that users may overcome barriers if they have a high motivation. However, the willingness to spend a specific amount of time for the specification of policies is an indicator for the user's motivation.

We neglect the user resource category physical capacity, as it does not match to any of the categories of the Dupree model. The proposed fusion of categories is an assumption and has not been evaluated, yet. However, it allows us to map our specification paradigms to the personas of Dupree. This mapping enables us to phrase assumptions that can be used as the baseline for evaluation.

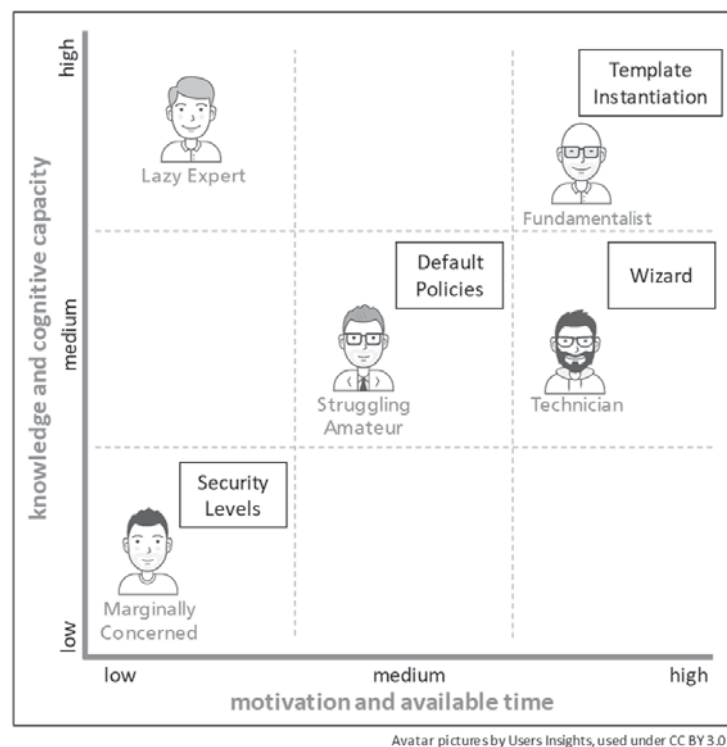


Figure 57: Assumed Matching of our Specification Paradigms to the Personas of Dupree for Best Usability

Thus, we mapped the required user resources of the selected specification paradigms mapping (compare Table 14) to the personas of Dupree (compare Figure 14 in Section 2.6.2). We assume that this mapping leads to best usability (combined results for effectiveness, efficiency and



satisfaction) for personas using the specific specification paradigm. We present our proposed persona to specification paradigm mapping in Figure 57 and evaluate it in an experiment in Section 9.4.

## 6.4 Summary and Conclusion

We created a user intention model to gain a better understanding of the user behavior with respect to policy specification with PAPs. The model explains why users have usability problems with PAPs. We extracted key characteristics of users and specification paradigms to improve the matching of user groups to specification paradigms regarding usability. In addition, we identified a persona model that helps us to confirm our assumption that different user groups, which are distinguished according to their user resources, behave differently and perform differently with respect to PAP usage effectiveness and efficiency.

To summarize our achievements, we review the fulfillment of the requirements defined in Section 6.1:

- **Req\_Mapping\_User-Characteristics:** We derived six categories of user resources. The categories stem from our user intention model. These resources are relevant user characteristics to describe their behavior with respect to policy specification with PAPs.
- **Req\_Mapping\_Specification-Paradigms:** We derived six categories of PAP requirements on users, which correspond to the user resources. With these categories, we can define the requirements that a PAP has on users for the task of effective and efficient policy specification.
- **Req\_Mapping\_Personas:** We identified an appropriate persona model that reflects the derived categories of user resources and requirements provided by the user intention model. In addition, we proposed an assumption on the optimal mapping of personas to specification paradigms in order to increase the usability during the specification of policies in a PAP.

Our discussion also revealed interesting future research topics. There is demand for the quantitative measurement of user resources of specification paradigms. In addition, further exploring the influence of individual policy vocabularies on the required user resources for specification paradigms in PAPs is a promising field of research into user-friendly PAP design.





## 7 Method for Usable PAP Generation

The method for usable PAP generation combines the previous four contributions into a comprehensive approach for generating usable PAPs, as requested in the scientific problem statement. The method for usable PAP generation represents Contribution 5 of this thesis, as defined in Section 1.4.

We structured this chapter as follows. In Section 7.1, we explain the research approach for the creation of the method for usable PAP generation. Section 7.2 presents an overview of the method. The five phases of the method are described in the following sections: the policy template elicitation in Section 7.3, the instantiation of the policy template model in Section 7.4, the instantiation of the PAP generation framework in Section 7.5, the selection of specification paradigms in Section 7.6 and finally the specification of policies with the usable PAP in Section 7.7. Section 7.8 summarizes this chapter.

### 7.1 Research Approach

We iteratively engineered the method for usable PAP generation. In total, we applied (various parts of) the method in four (industrial) case studies—two aiming at the improvement and two aiming at the validation of our method—and one experiment for evaluation. After each of the two case studies for improvement, we enhanced the method and the containing contributions based on our observations and lessons learned from evaluation. After the first case study for validation, we added two more specification paradigms to the policy template model and two corresponding presentation modules to the PAP generation framework. These extensions did not affect the validity of the results of the »BeSure« case study. More details about the research approaches for the four contributions used in this method can be found in the respective chapters. Figure 58 provides an overview of the different versions of the four contributions, which we devised in the different case studies and in the experiment.

We applied the elicitation of policy templates, the instantiation of the policy template and the PAP generation framework in all four case studies and validated our assumptions regarding the most suitable specification paradigms in the experiment. The combination of the case study »Digital Villages« and the policy specification experiment yielded the validation of the complete method for usable PAP generation.

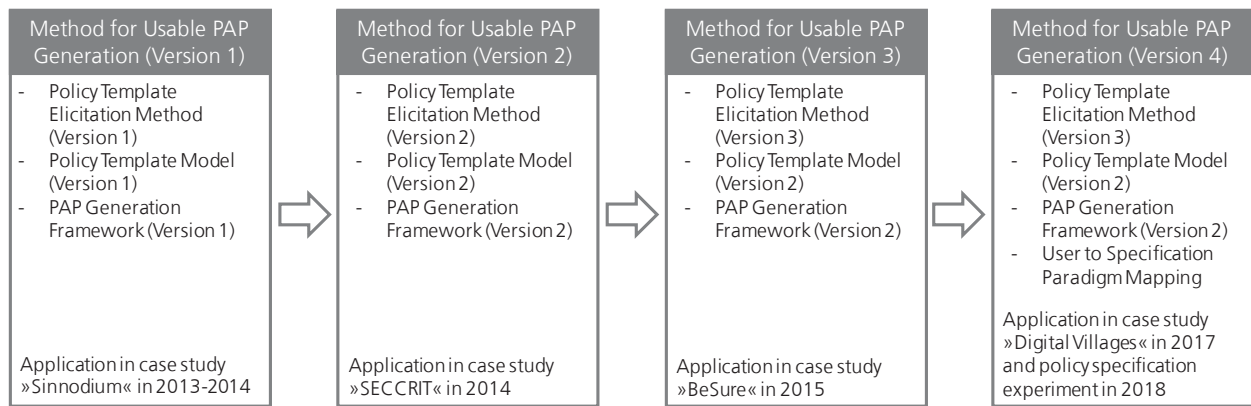


Figure 58: Research Approach for the Method for Usable PAP Generation

## 7.2 Method Overview

The overall goal of our work is the creation of a comprehensive method for generating usable PAPs. Therefore, we elaborated and evaluated four essential contributions that can be combined into this method:

- **Contribution 1 (C1) – User to Specification Paradigm Mapping:** The mapping guides PAP designers to select the appropriate specification paradigms with respect to usability for users based on their resources (e.g., security knowledge or cognitive capacities). We presented the details in Chapter 6.
- **Contribution 2 (C2) – Policy Template Elicitation Method:** The method supports the elicitation of policy templates from an application domain that reflect the security and privacy demands of users. We provided the details in Chapter 3.
- **Contribution 3 (C3) – Policy Template Model:** The model supports the formalization of security and privacy demands as policy templates. In addition, rules for projecting the policy templates on different specification paradigms as well as transformation rules for generating ILPs from instantiated policy templates can be defined. We described the details in Chapter 4.
- **Contribution 4 (C4) – PAP Generation Framework:** The PAP generation framework automates the generation of user interfaces for policy specification with multiple specification paradigms in a PAP. We presented the details in Chapter 5.

Using these contributions, we can implement a PAP that can be tailored to the user and the application domain. Hereby, the selection of the supported UI framework is done at development time. We need to provide a generic PAP for each UI framework. At development time, this generic PAP does not contain user interfaces for policy specification. These are generated at runtime in an automated manner. Thus, the selection of the

policy vocabulary and the specification paradigms is done at runtime of the PAP. The PAP loads the desired policy vocabulary on startup and then generates the user interfaces for policy specification for all supported specification paradigms on runtime. The imported policy vocabulary contains the policy templates provided to the user and the corresponding transformation rules (SLP to ILP) and projection rules (representation of policy templates in specification paradigms). Finally, the user can choose which specification paradigm to use. However, a preselection of specification paradigms that suit the user's capabilities is recommended (compare Contribution 1). We summarize the customization decisions of our PAPs in Figure 59.

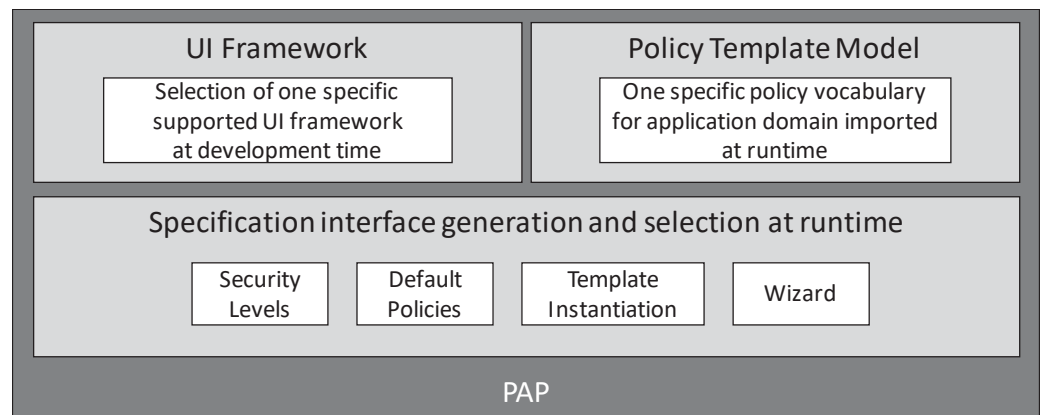


Figure 59:

Customization Decisions for a PAP at Development Time and Runtime

Figure 60 provides an overview of our entire method for usable PAP generation. Our method consists of five process steps, which include the four contributions (marked in Figure 60):

- **Step 1 – Policy Template Elicitation:** The method expert prepares and conducts a policy template elicitation workshop with experts from the application domain. He derives policy templates from the workshop results and validates them together with the domain experts.
- **Step 2 – Instantiation of Policy Template Model:** The method expert uses the elicited information and instantiates the policy template model. Additional information for creating projection and transformation rules in the policy vocabulary must be requested by domain experts. The resulting policy vocabulary reflects the security and privacy demands of users of the application domain.
- **Step 3 – Instantiation of PAP Generation Framework:** The method experts selects a generic PAP supporting the desired the UI framework. Next, he assigns a complete policy vocabulary for the application domain to be loaded by the PAP on startup. The PAP generation framework inside the PAP is capable of generating one user interface for policy specification for each supported specification paradigm.

- **Step 4 – Specification Paradigm Selection:** The method expert can preselect specification paradigms that are suitable for the users of the PAP according to the guidelines we provide in this thesis. This decision may also be delegated to the users themselves by providing paradigm selection at runtime.
- **Step 5 – Specification of Policy with PAP:** A user can specify a security or privacy policy according to his demands with the PAP with the selected specification paradigm.

In the following sections, the method steps are explained in detail.

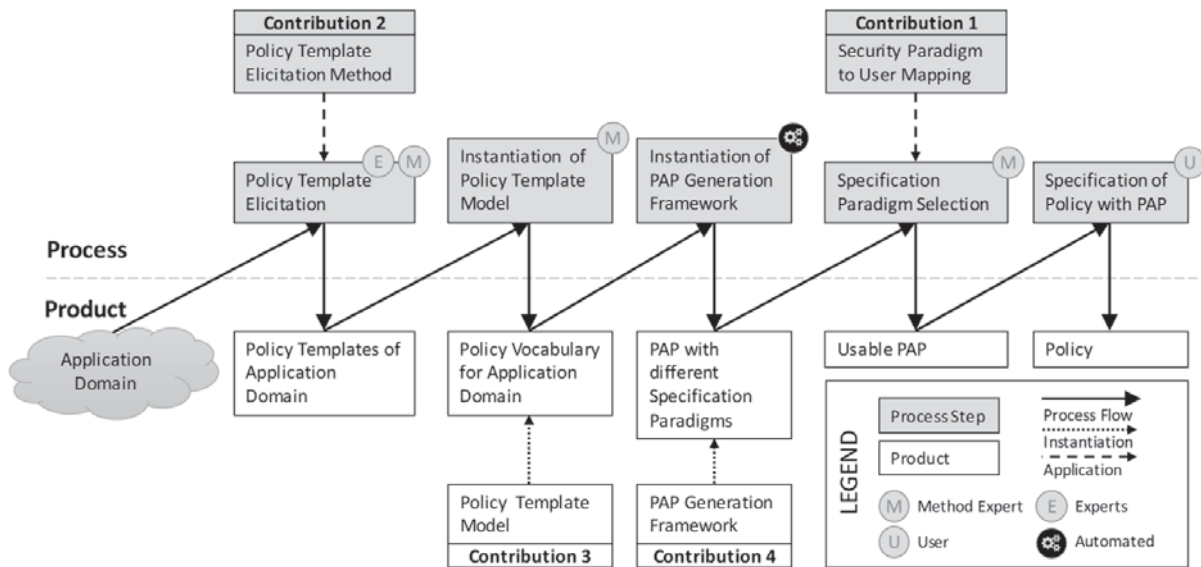


Figure 60: Overview of the Method for Usable PAP Generation

### 7.3 Step 1: Policy Template Elicitation

The purpose of the first method step is the elicitation of policy templates. The step represents the application of Contribution 1 of this thesis, the policy template elicitation method, described in Chapter 3.

#### ***Roles Involved***

- The **method expert** executes the entire policy template elicitation method.
- The **contact person** provides information about the application domain and relevant stakeholders to be involved in the elicitation workshop.
- **Experts** from the application domain participate in the elicitation workshop and validate the results. Experts can be domain experts, technology experts, security experts, legal experts, asset owners and asset users.

### ***Input***

The method expert needs a **contact person** for information retrieval in order to prepare the elicitation workshop. Other information material, such as a **project offer**, can provide valuable additional information for the application of the method.

### ***Output***

After successful application, the policy template elicitation method yields **policy templates** relevant for the application domain. The contact person might reveal additional information, such as the policy languages used for the enforcement of policies in systems of the application domain.

### ***Process Description***

The method expert applies the policy template elicitation method as described in Chapter 3.

## **7.4 Step 2: Instantiation of Policy Template Model**

The purpose of the second method step is the instantiation of the policy template model to create an application domain specific policy vocabulary. We describe the policy template model (Contribution 3 of this thesis) in detail in Chapter 4.

### ***Roles Involved***

- The **method expert** instantiates the policy template model to create an application domain specific policy vocabulary.
- **Experts** from the application domain provide information about projection rules for specification paradigms and transformation rules for generating ILPs.

### ***Input***

The elicited **policy templates** from the previous step are the main input. However, the method expert might need to retrieve additional information about **projection rules** and **transformation rules** from experts of the application domain.

### ***Output***

The output of this method step is an application domain specific **policy vocabulary**.

## ***Process Description***

The creation of the policy vocabulary (instantiation of the policy template model) is a manual step. Currently, we do not provide an editor for creating policy vocabularies, which would support the method expert. This is left to future work.

First, the method expert creates an XML file that represents the policy vocabulary. He uses the grammar, which is provided as an XML schema in Appendix B.1. This XML schema represents the policy template model. The method expert defines the policy templates devised in the previous step (compare the template sub-model in Section 4.5 and the specification-level template sub-model in Section 4.6).

Second, the method expert creates projection rules for the generation of user interfaces with multiple specification paradigms:

- For the specification paradigm »template instantiation«, no projection rules are necessary.
- For the specification paradigm »default policies«, the method expert needs to create default instantiations of the policy templates. For each default policy, he defines the values of all variable parts of a policy template. The PAP instantiates the policy templates according to these default values and provides a list of default policies to the user for selection.
- For the specification paradigm »security levels«, the method expert defines different levels of security or privacy. Each level consists of a set of default policies. Thus, the method expert assigns a set of default policies, which are specified for the specification paradigm »default policies«, to each level.
- For the specification paradigm »wizard«, the method expert first needs to define the order in which the policy templates are processed. Next, the method expert splits each policy template into several wizard pages, in which a small part of the policy template shall be specified by the user. For each page, the method expert combines variable parts of the policy template with descriptive texts. All variable parts need to be referenced in exactly one wizard page to allow the generation of policies.

The method expert must retrieve missing information about the definition of the projection rules from experts of the application domain.

Third, the method expert creates transformation rules for ILP generation. To this end, he creates ILPTs (i.e., templates for machine-understandable policies) in the desired policy language. Currently, our reference

implementation of the PAP generation framework only supports XML-based policy languages. The method expert creates a basic ILPT as a tree of XML nodes. Then, the method expert can extend this ILPT by two types of variable parts:

- **Variable references:** The method expert can assign variable elements or selectable text elements from the SLPT to attributes of the ILPT. Thus, after the user instantiated an SLPT, the values from these elements are inserted as attribute values during the generation of the ILP.
- **ILPT Blocks:** The method expert can insert variable XML blocks into an ILPT. He defines multiple XML nodes that can be added as child elements to an XML node of the ILPT, based on a condition. The condition is assigned to selection elements of the SLPT. Thus, if the user selects an assigned selection element on the specification level, the respective XML block is inserted into the ILP.

An example of an ILPT can be found in the description of the »SECCRIT« case study in Appendix D.1.

Finally, the policy vocabulary must be validated by experts of the application domain with respect to correctness, completeness and understandability of the descriptive texts in the »wizard«.

## 7.5 Step 3: Instantiation of PAP Generation Framework

The purpose of the third method step is the provision of a PAP that supports multiple specification paradigms using the PAP generation framework, described in Chapter 5 as Contribution 4 of this thesis. The user interfaces for policy specification are generated at runtime using the specification paradigm algorithms (see Section 5.3.2), which are applied to the information contained in the policy vocabulary defined in the previous method step.

### *Roles Involved*

- The **method expert** selects a PAP for the desired UI framework. Then, he configures the PAP to load a specific policy vocabulary on startup.
- If necessary, a **software developer** develops a new PAP (view modules and surrounding software component) for an additional UI frameworks or new presentation modules for the PAP generation framework, which implement new specification paradigms (This may imply also changes to the policy template model).



### ***Input***

The main inputs for this method step are the **policy vocabulary** specified in the previous step and a generic **PAP**, built for the desired UI framework, which is capable of creating the policy specification interfaces for different specification paradigms in an automated manner at runtime using the built-in PAP generation framework. Please note that this generic PAP does not yet contain any policy specification interface implemented at development time. All policy specification interfaces are generated at runtime.

### ***Output***

The output of this step is a fully functional **PAP for the application domain with multiple supported specification paradigms**.

### ***Process Description***

The method expert selects a PAP for the desired UI framework (and thus indirectly selects the supported operation platforms). Our reference implementation of the PAP generation framework provides generic PAPs as an Android app, as Java applications (executable on Windows or Linux using the UI frameworks »Swing« and »JavaFx«) and as a web-service. If a different PAP is required, it must be implemented by a software developer using our proposed architecture. This implementation task includes the respective view module.

Next, the method expert configures the PAP to use the policy vocabulary created in the previous step. On start-up, the PAP loads the policy vocabulary and creates an internal instantiation of the policy template model. The PAP generates the user interfaces for the supported specification paradigms at runtime.

## **7.6 Step 4: Specification Paradigm Selection**

The purpose of the fourth method step is the selection of specification paradigms for users. This step is optional as we may delegate the specification paradigm selection to the user. We provide criteria for the selection of specification paradigms for users as Contribution 1 of this thesis, as described in Chapter 6.

### ***Roles Involved***

- The **method expert** selects one or more specification paradigms for the various user types.

***Input***

The **PAP for the application domain with multiple supported specification paradigms** is the input for this method step.

***Output***

The method experts provides a **usable PAP** for a user or user group as output. He achieves improved usability by selecting the most suitable specification paradigms for a specific user group.

***Process Description***

The method expert selects one or more specification paradigms for the users of the PAP. The method expert bases the selection on the recommendations given in Section 6.3.2, whose evaluation is described in Section 9.4.

## **7.7 Step 5: Specification of Policy with PAP**

Finally, the user can specify security or privacy policies with the generated usable PAP that provides the best-suited specification paradigm(s) for achieving an effective, efficient and satisfying policy specification for the user. This generated PAP with an optimal usability experience is the overall output of the method for usable PAP generation.

***Roles Involved***

- The **user** specifies policies with the usable PAP.

***Input***

The user needs access to a **usable PAP** that is tailored to his specific user resources and the application domain.

***Output***

The user specifies a **set of policies** that reflect his security or privacy demands.

***Process Description***

The user starts the usable PAP. If required, he selects a specification paradigm. Then, he instantiates one or more policy templates with the predefined or selected specification paradigm. Finally, he receives a list of SLPs and corresponding ILPs.

## **7.8 Summary and Conclusion**

In this chapter, we presented our entire process for the systematic creation of a usable PAP for users of an application domain. In this method, we combined the four other contributions of this thesis into one comprehensive method.

## 8 Evaluation for Improvements

As the first part of our evaluation, we describe two case studies focusing on our contributions presented in the previous chapters. We gained new insights and discovered improvement potential after each case study.

We structure the remainder of this chapter as follows: In Section 8.1, we explain our research approach. In the subsequent sections, we present the two case studies: the case study »SINNODIUM« in Section 8.2 and the case study »SECCRIT« in Section 8.3. Section 8.4 summarizes our findings.

### 8.1 Research Approach

We chose an explorative and iterative evaluation approach for improving our contributions. We conducted two case studies with the overall goal to gain better insights and to reveal improvement potential for our contributions. In both case studies, we applied our method for usable PAP generation. The two explorative and iterative case studies are:

- The »SINNODIUM« case study was an early application of our preliminary method to demonstrate its general feasibility. The study was conducted between 2013 and 2014 with the industrial partner »vwd« in the application domain of a mobile app for financial advisors that visit clients at home.
- The »SECCRIT« case study was an early application of our preliminary method to test improvements identified in the first case study. The study was conducted in 2014 with the 9 project partners of the European project »SECCRIT« in the application domain of cloud services in critical infrastructure IT.

We describe these case studies and evaluate their results according to our research questions (see Section 1.3) and hypotheses (see Section 1.5) in this thesis. We confirm our results in the evaluation for validation described in the Chapter 9.

### 8.2 Case Study: Software Cluster Project »SINNODIUM«

We performed an initial case study in the context of the research project »SINNODIUM« together with the industrial partner » vwd Vereinigte Wirtschaftsdienste GmbH« (vwd for short) to explore the applicability of our method for usable PAP generation in an actual application domain.

### 8.2.1 Project Summary

The joint project SINNODIUM (Software Innovations for the Digital Enterprise) was funded by the German Ministry of Education and Research under grant number 01IC12S01F. The overall goal of SINNODIUM was the development of prototypical solutions for the next generation of business software, with a focus on the improvement of the qualities interoperability, adaptively, user experience and security. We (as employees of Fraunhofer IESE) worked in cooperation with vwd on the improvement of the security and privacy of financial data in mobile scenarios.

The vwd group develops private banking and asset management software, such as the »vwd portfolio manager«, which is a software solution for the management and controlling of client portfolios. Financial advisors use this software to consult bank clients on their investment strategy. The requirements for this domain are currently changing as the financial advisors increasingly use mobile devices outside the bank. Therefore, vwd developed a prototype for mobile portfolio management, called the »vwd portfolio manager mobile«. With this tool, financial advisors can visit clients at home and prepare meetings on their way to the client. However, this raises concerns about security and privacy, as many new threats occur in different mobile scenarios in comparison to the work conducted solely inside the bank.

In cooperation with vwd, we elicited assets and threats for different use cases of the »vwd portfolio manager mobile« app and derived corresponding policy templates. We instantiated the policy template model and generated a PAP with the specification paradigm »policy templates« for the operation platform »Android«. In this project, we focused on detecting different mobile scenarios, so-called contexts, in which a financial advisor may use the app. Examples are »in the bank«, »at the client's home« and »on a business trip«. We wanted to enforce different security policies based on the current use situation, as different contexts imply different security and privacy demands.

Together with vwd, we developed a demonstrator that shows the enforcement of context-dependent security policies within the »vwd portfolio manager mobile«. The demonstrator employed the usage control enforcement framework IND<sup>2</sup>UCE on a mobile system with the operating system Android. The Android PAP was a core part of this demonstrator.

### 8.2.2 Design and Execution

The main goal of the first case study was to test the applicability of the early versions of our contributions of this thesis. We defined the following

evaluation plan in order to answer our research questions and to confirm our hypotheses.

For the policy template elicitation method, we aimed to find preliminary answers to RQ2 (Elicitation; see Section 1.5.2). More specifically, we tested the feasibility of using policy templates for the specification of policies in a PAP, and we identified mandatory information to be elicited from stakeholders in the application domain. In addition, we examined whether the policy templates generated with our method allow the instantiation of a correct and complete set of policies for the application domain.

In the first meeting with vwd, we identified the application domain to be a mobile app for financial advisors that consult clients in the bank and at the client's home. The financial advisors take the mobile device with access to clients' financial data with them on business trips. Apparently, bank clients and security experts from the bank have many security and privacy demands that need to be enforced on the mobile device. We identified the use cases, assets, threats and countermeasures for this application domain together with domain and technology experts from vwd in unstructured discussions within several consecutive meetings. At least one domain and technology expert and one method expert participated in each of those meetings. After each meeting with vwd, we created and later refined the policy templates. We devised the policy templates without a structured elicitation method. However, we tested a first version of the policy template notation format for creating several policy templates on the specification level (SLPTs).

Moreover, we tested the feasibility of our policy template model for the formalization of security and privacy demands to answer RQ3 (Formalization). To this end, we instantiated a preliminary version of the policy template model including rules for transforming policy instances on the SLP level into ILPs.

Finally, we checked whether the creation of the user interfaces of a PAP can be automated with respect to RQ4 (Automation) and H6 (Feasibility of automation of PAP creation). To assess the feasibility of automated PAP generation, we built a prototype of an Android PAP with the specification paradigm »template instantiation«. The PAP was a core part of a demonstrator of the »SINNODIUM« project.

Between 19.09.2013 and 12.12.2014, we held seven consecutive meetings with vwd to determine and refine the use cases and the policy templates and to assess usability of the PAP prototype.

### 8.2.3 Results

During the elicitation of policy templates, we decided to focus on one essential asset »financial data of client« for the »vwd portfolio manager mobile«. This asset is described in Table 15. We added the prioritization values after the end of the case study, as we initially did not elicit this information.

Table 15: Documented Asset »Financial Data of Client«

|                           |  |
|---------------------------|--|
| Asset ID                  | A1   |
| Asset                     | Financial data of client   |
| Data Owner                | Client   |
| Example Use Case          | The financial advisor accesses financial data of clients on a mobile device during work. Data access can happen at the bank, at the home of a client or on business trips. |
| Policy Authors            | Bank administrator   |
| Prioritization Properties | Monetary value of asset: high (€€€)<br>Sensitivity of asset: highly confidential (high)  |
| Legal Regulations         | Regulations of BaFin   |

We identified the following seven use cases in which the security of the client data is jeopardized and policies must be enforced:

- Unauthorized access to sensitive data when leaving the bank: A financial advisor leaves the bank with the mobile device. Previously, he had viewed or edited client-specific data in the »vwd portfolio manager mobile« app on the device. He forgets to close the app showing sensitive data before leaving the bank. To prevent this threat, the app is automatically closed when the application context changes.
- Third parties want to obtain specific information about the financial status of a client, or they want to falsify data: Outside the bank, the financial advisor may only access the complete client data if the financial advisor is in a client appointment and the client has authenticated himself using a PIN. If the financial advisor is in a client appointment but there is no valid PIN authentication, he can only access anonymous client data. After a successful PIN authentication, the financial advisor has full access to the client data of the visited client.
- Unauthorized disclosure of sensitive internal bank data: The financial advisor is in a meeting with the client in which he holds the tablet in his hand and can see all data, including sensitive information. He places the tablet flat on the table in order to show the client something on the display. In this case, only the

data that the client is allowed to see should be displayed instead of the entire data.

- Unauthorized execution of group evaluations: A financial advisor may only carry out group evaluations with his mobile device inside the bank. Outside the bank, only individual client data records are available for analysis.
- Mass retrieval of data: Financial advisors have full access to client data. If, however, an unusual data retrieval behavior is detected, which indicates a mass retrieval of client data, appropriate reactive measures will be triggered to prevent, for example, the creation of so-called tax CDs that can be sold to the authorities..
- Unauthorized access to sensitive data due to insufficient security settings on the mobile device: A financial advisor's mobile device has inadequate security settings in the field. For example, the set period for the automatic display lock is too long, no screen lock is enabled and sensitive data is potentially visible on the display. Stolen tablets could be accessed by criminals.
- Loss of sensitive data due to loss or theft of a mobile device: Financial advisors' mobile devices contain sensitive client data. The loss or theft of such a tablet therefore represents an immense security risk. Thus, in such a case, the automated deletion of all sensitive data must be ensured.

For each of these seven use cases, we iteratively identified threats. To this end, we created and refined seven respective policy templates. We show one of the elicited threats in Table 16.

Table 16: Documented Threat »Data Theft of Financial Data for Creation of Tax CD«

|                        |  |
|------------------------|--|
| Threat ID              | T5   |
| Related Asset ID       | A1   |
| Related Asset          | Financial data of client   |
| Attackers              | Internal attacker  |
| Threat                 | Data theft of financial data for creation of tax CD <ul style="list-style-type: none"> <li>• probability: likely (medium)</li> <li>• damage: existence-threatening (high)</li> </ul> |
| Existing Documentation | not elicited   |

We present the related policy template in Table 17.



Table 17: Policy Template »Mass Retrieval of Data«

| ID                     | Policy Template Name   | Asset  | Target System                | Policy Author      |
|------------------------|------------------------|--|------------------------------|--------------------|
| S4                     | Mass retrieval of data | Client Data  | vwd portfolio manager mobile | Bank administrator |
| Policy Template Syntax |                        | If the financial advisor wants to access client data and has already accessed <number> data records from different customers within <number> <unit of time> and is [inside the bank   outside the bank], then [forbid access   inform the supervisor via email: <email address>   log the misconduct] +.       |                              |                    |
| Description            |                        | Financial advisors have full access to their clients' data. If, however, an unusual data retrieval behavior is detected which points to the mass retrieval of client data, appropriate reactive measures must be taken. Thus, for example, the creation of so-called tax CDs can be recognized and prohibited. |                              |                    |
| Threat                 |                        | Mass retrieval of data   |                              |                    |
| Security/Privacy Goal  |                        | Confidentiality  |                              |                    |
| Example Instantiation  |                        | If the customer advisor wants to access client data and has already accessed 5 data records from different customers within 30 minutes and is outside the bank, then forbid access and inform the supervisor via email: supervisor@bank.de.  |                              |                    |

We originally elicited all information in German language and derived German policy template. For the documentation in this thesis, we translated the elicited information to English.

The screenshot shows the PAP (Policy Assistant) interface. On the left, there is a sidebar with a list of policy templates. The main area displays the configuration for the 'Mass Retrieval of Data' policy template. The configuration includes a condition: 'Wenn der Kundenberater auf Kundendaten zugreifen will und innerhalb 5 Minuten schon auf 10 Datensätze von unterschiedlichen Kunden zugegriffen hat und sich innerhalb der Bank befindet, dann'. Below this, there are three options: 'verbiete den Zugriff' (checked), 'informiere den Vorgesetzten per E-Mail: E-Mail-Adresse' (checked), and 'logge das Fehlverhalten' (unchecked). At the bottom, there are buttons for 'Policy anzeigen', 'Policy speichern', and 'Policy Details'.

Figure 61: Exemplary PAP Using View Module »Android«, Policy Vocabulary »SINNODIUM« and Presentation Module »Template Instantiation«

Next, we instantiated the policy template model and, thus, created a policy vocabulary with all seven policy templates and respective ILP transformation rules. We imported the policy vocabulary in a prototype of

the Android PAP. Figure 61 shows the generated user interface for the policy template presented in Table 17 (in German language).

The PAP was capable of applying the transformation rules on the instantiated policy template in order to generate machine-understandable representations in the form of ILPs in the IND<sup>2</sup>UCE policy language, as depicted in Figure 62.

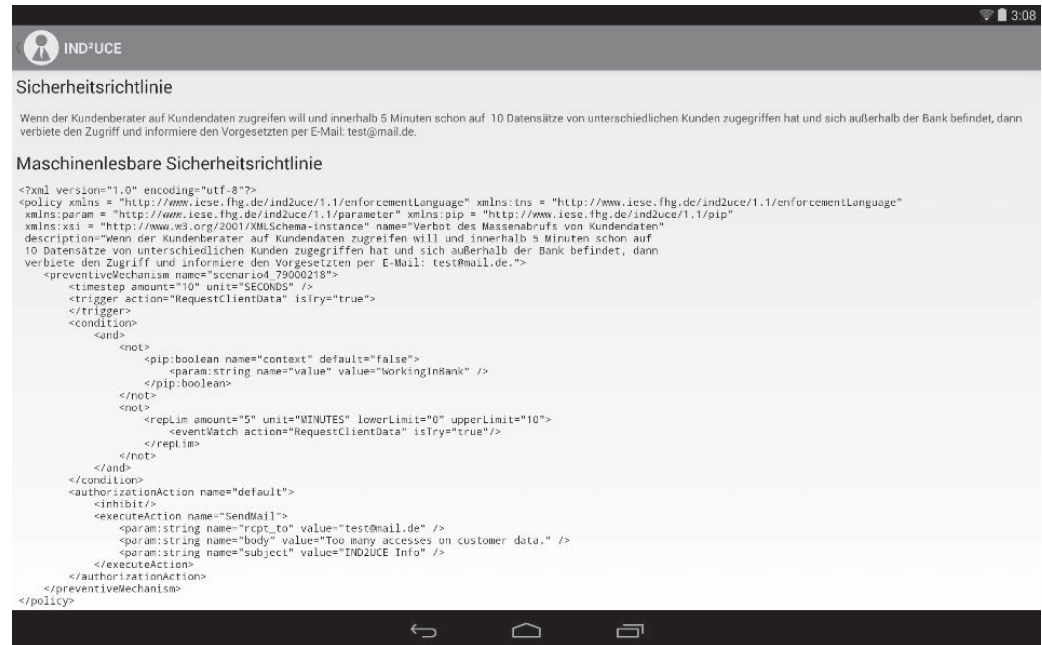


Figure 62: ILP in IND<sup>2</sup>UCE Policy Language Version 1.1 Generated by the Android PAP

## 8.2.4 Observations and Lessons Learned

Regarding RQ2 (Elicitation), we showed that we can create policy templates with the information elicited from the stakeholders of the application domain. This indicates that all relevant information for creating policy templates is elicited with our method. We were also able to demonstrate the use of these policy templates in a PAP for the specification of policies by the targeted user group.

In addition, we observed in this case study that the unstructured elaboration of policy templates is a time-consuming task. We needed seven meetings with our project partners to elicit all necessary information and to define the final versions of the policy templates. We concluded that a more structured approach should allow a faster elicitation of information and a faster derivation of policy templates with fewer stakeholder workshops. In particular, a comprehensive list of relevant assets and threats should be systematically elicited before drafting the policy templates.

We learned that domain and technology experts are valuable information sources regarding security and privacy policies. However, several aspects remained unclear. In many application domains, the relevant security and privacy policies should or even must be affected by legal regulations. Thus, stakeholder with legal expertise should be involved in future elicitations. Moreover, the policies solution providers have in mind may not reflect the actual security and privacy demands of real users. Thus, we need to further investigate how the integration of real users affects the elicitation and its results. We consider both stakeholders in the next case studies.

We confirmed that the provision of policy templates allows users to specify policies that are tailored to their personal privacy and security demands within the limits set by the application domain. A PAP can use these policy templates to provide the instantiation of concrete policies.

Regarding RQ3 (Formalization), we found that all policy templates and transformation rules to generate ILPs from SLPs can be modeled with our proposed policy template model. However, we perceived the model as too complicated and identified potential for improvement and extension. Especially the specification of ILPTs and transformation rules was very error-prone. We considered this first version of the policy template model to be unsuitable for less experienced method experts. A simpler syntax for ILPTs and transformation rules was deemed necessary.

We confirmed the fulfillment of RQ4 (Automation) and H6 (Feasibility of automation of PAP creation) in the context of our case study by demonstrating the generation if the PAP user interface for the specification of policies with the specification paradigm »template instantiation«. This proved the general feasibility to automate the PAP creation. However, further investigations are required in order to evaluate the revealed improvement potential and to show the generation with multiple specification paradigms.

### **8.2.5 Threats to validity**

Our experimental results are subject to several threats to validity. Below, we distinguish between internal, external and conclusion validity:

- Internal validity is the extent to which conclusions about causal relationships can be made based on the research design (e.g., used measures, research setting).
- External validity is the extent to which the results can be generalized (results can be held to be true for other cases, for example, with different participants).

- Conclusion validity is the extent to which conclusions about the relationship among variables are correct and reasonable based on the data.

### ***Internal Validity***

In our case study, vwd was very interested in security and privacy for their own product. Thus, the selections of highly motivated participants and the application domain are threats to internal validity. However, the exemplary assets and threats were not predefined, but jointly elicited with the project partner. Additionally, the stepwise refinement of the policy templates by the method expert might have affected the results.

The project partners did not know our research goals and hypotheses; however, they knew about the project goals that centered on context-aware policy enforcement. Hence, we cannot estimate the influence of guessed hypotheses and expected researcher expectancies.

### ***External Validity***

Many aspects affected the result quality in this case study. The method expert influenced the creation of the policy templates during iterative refinement. To guarantee the general feasibility of our approach, we need to apply it in different application domains with different stakeholders.

### ***Conclusion Validity***

The number of scenarios and associated contextual descriptions that we obtained is limited. Hence, we must confirm threats with regard to low statistical power and consequently low reliability. However, we asked the representatives of the company whether they miss any interesting scenarios or situations, which they denied. Hence, we can be certain in terms of completeness of the elicited policy templates.

## **8.2.6 Summary**

In the »SINNODIUM« case study, we positively evaluated the concept of policy templates for specifying security and privacy policies in a PAP. Together with experts of vwd, we elicited seven policy templates for the instantiation of policies. We built a PAP with which users can specify policies for the »vwd portfolio manager mobile« Android app. In conclusion, we showed the applicability of the method for usable PAP generation.

### 8.3 Case Study: European Project »SECCRIT«

We performed a second case study in the context of the research project »SECCRIT« for further exploring the feasibility of our method for usable PAP generation in a different application domain. In addition, we explored the improvements we made in the second versions of the policy template elicitation method, the policy template model and the PAP generation framework.

In the SECCRIT study, we elicited policy templates together with the industrial partners »Amaris Technologies GmbH (AMARIS)«, »NEC Europe Ltd (NEC)«, »Mirasys Ltd. (MIRASYS)«, »Hellenic Telecommunications Organization S.A. (OTE)«, »Ayuntamiento de Valencia (VLC)« and the research partners »AIT Austrian Institute of Technology GmbH (AIT)«, »ETRA Investigacion Y Desarrollo SA (ETRA)«, »Karlsruher Institut für Technologie (KIT)« and »Lancaster University (ULANC)«.

#### 8.3.1 Project Summary

The goal of the European project SECCRIT (Secure Cloud Computing for Critical Infrastructure IT) was the development of technologies and methodologies to create a secure, trustworthy, and high-assurance cloud-computing environment for critical infrastructure IT. Services for critical infrastructures are used in domains such as transportation systems, financial services or security services. SECCRIT was funded by the European Union within the 7<sup>th</sup> Framework Programme (FP7-SEC-2012-1) under grant number 312758.

In SECCRIT, we aimed to improve the policy specification for security demands in critical cloud infrastructure IT. Therefore, our goal was to provide a usable PAP for specifying security policies. To this end, we first elicited security demands from the industrial partners of the project. Two scenarios were considered: a public video surveillance system in Helsinki, Finland, and a traffic control system in Valencia, Spain. Both systems were supposed to run in the cloud and, thus, faced the security challenges implied by cloud deployment.

In cooperation with the project partners, especially the industrial partners, we elicited the assets, threats and countermeasures for the services of critical infrastructure in the two different scenarios. A partner that acted as a cloud provider and a research partner with legal expertise supported the elicitation task. We derived policy templates, instantiated the policy template model and generated a PAP with the specification paradigms »policy templates« and »default policies« for the operation platforms »Swing« and »Web«. We also built a demonstrator for enforcing the policies regarding secure virtual machine management as well as data

usage control in cloud databases, using the data usage control framework »IND<sup>2</sup>UCE« [110].

### 8.3.2 Design and Execution

Our key concern in the second case study was to test the improvements of our policy template elicitation method (RQ2), the policy template model (RQ3) and the PAP generation framework (RQ4) in relation to the first case study.

In RQ2, we aim to » elicit all relevant information from an application domain«. This means that we need to know what information we need and from where we can obtain it. We confirmed that assets, use cases, threats, countermeasures and example policies are suitable information types for deriving policy templates. In addition, we identified multiple information sources from which we can retrieve assets, use cases, threats, countermeasures and example policies. By applying the policy template elicitation method, we tested how a method expert can elicit security demands from various stakeholders and existing documentation in a more structured way.

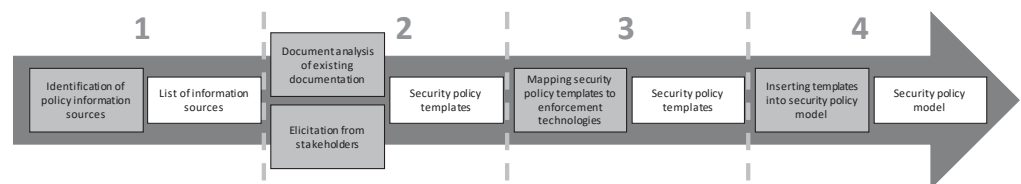


Figure 63: Second Version of the Policy Template Elicitation Method

In this case study, we acted as the method expert and involved other partners from the project as domain, technology, legal and security experts and especially the industrial partners as asset owners and users. We defined the project-related application domain to be: »Cloud systems in critical infrastructures must be protected against multiple threats on different architectural layers of the cloud system«. We used the second version of the policy template elicitation method to elicit all information required to derive policy templates. We first identified information sources by means of discussions with project partners and Internet searches. This second version of the policy template elicitation method, as illustrated in Figure 63, was not yet include an explicit elicitation workshop. Instead, we had unstructured discussions with the project partners and retrieved information by document analyses. Based on these sources, we devised a first set of policy templates and started to create a policy catalog (i.e., a document with all policy templates) for the application domain. We sent this catalog to four project partners (»Amaris«, »ETRA«, »Mirasys« and »OTE«, three of them from industry). These partners were asked to specify their own policy templates based on instructions we attached that



described how to review and extend our policy templates. We consolidated all responses and integrated them into a preliminary policy catalog.

Next, we conducted a review workshop with nine project partners in order to discuss the status of the policy templates catalog. The workshop took place on 03.04.2014 for about two hours and was part of a regular project meeting. Sixteen persons participated including the method expert. Together, the participants had the expertise to represent the different stakeholder roles: domain expert, technology expert, legal expert, security expert, asset owners and asset users. In the workshop, we first presented the concept of policy templates to all project partners. Afterwards, we let the four partners that contributed to the policy template catalog present their policy templates for their own assets, use cases and threats. We discussed the catalog in the group of all project partners. After the workshop, we integrated the discussion results into the policy templates. Then, we asked the remaining project partners to extend the policy catalog. After consolidating and integrating their feedback, we finalized the policy template catalog.

Regarding RQ3, we showed the feasibility and completeness of the improved policy template model. In the project, we decided to select three policy templates for being used in a project demonstrator. We instantiated the policy templates including transformation rules for generating ILPs for those three policy templates. The second version of the policy template model used the simplified transformation rules. Additionally, the new model contained elements for defining default policies for policy templates, which are required by the specification paradigm »default policies«.

Regarding RQ4, we explored the generation of different policy templates in multiple PAPs based on one policy vocabulary. Therefore, we used the PAP generation framework to provide three fully functional PAPs. One was implemented as a Java application using the UI framework »Swing«, the second was an Android app, and the third was based on a preliminary version of the view module »Web«. The Android PAP is a slightly improved version of the PAP used in the »SINNODIUM« case study. We equipped all PAPs with the presentation modules »template instantiation« and »default policies« and imported the policy vocabulary with the three demonstrator policy templates of the »SECCRIT« project.

### **8.3.3 Results**

In the »SECCRIT« case study, we identified the following information sources for eliciting security demands, assets, threats or complete security policies in the application domain:

- Asset users: Users of sensitive assets typically have security demands.
- Domain experts: Domain experts should be considered within the elicitation process to identify application domain-specific assets, threats and countermeasures.
- Technology experts: When eliciting security policy templates, technology experts can explain which security demands can be enforced technically with a policy enforcement framework and which can be enforced only organizationally, for instance, by service level agreements.
- Company regulations: Companies usually have IT security regulations in place that must be met. The documentation of these regulations can be used as an information source. In addition, the responsible IT security officer is a valuable source.
- Risk assessment documents: During risk assessment activities, assets and respective threats are identified. A catalog of typical vulnerabilities and threats for the domain of cloud computing for critical infrastructure IT can, for example, be found in [57].
- Standards and guidelines: Depending on the business operation scope, companies are obliged or encouraged to follow specific standards provided by regulatory authorities or expert groups. These guidelines are a source of security policies the company must enforce. For example, a set of security threats for cloud computing for critical infrastructure is listed in [104].
- Legal aspects: Besides the domain-specific standards, some legal obligations may apply, such as the General Data Protection Regulation (GDPR) [164] and its county-specific implementations.

We analyzed these information sources and elicited additional information from the project partners (as described in the previous section). In total, we identified thirteen assets in the application domain of cloud services for critical infrastructure IT. They are related to different architectural levels: cloud infrastructure level, tenant infrastructure level, service level and user level. We elicited 35 threats for these assets considering the individual threats for those architectural levels. The threats stem from expert discussions in project meetings and existing documentation about related threats and risks (e.g., [57, 104]).

Based on the threats, we defined 40 policy templates that can instantiate policies for mitigating or preventing those threats. Next, we created the policy vocabulary for the three selected policy templates by instantiating the policy template model (see Appendix D.2). We present examples for the elicited assets, threats and countermeasures in Appendix D.1.



We created PAPs that can generate user interfaces for specifying security policies for critical infrastructure cloud solutions using the PAP generation framework. We used the presentation modules »template instantiation« and »default policies« and the view modules »Swing«, »Android« and »Web« for the PAPs. We present screenshots of the generated PAPs in Appendix D.1.

#### **8.3.4 Observations and Lessons Learned**

Our key concern in the second case study was to test the improvements of our policy template elicitation method, of the policy template model and of the PAP generation framework in relation to the first case study.

With respect to RQ2 (Elicitation), we explored how a method expert can elicit security demands from various stakeholders and existing documentation in a more structured way. We applied the improved policy template elicitation method on the application domain of cloud systems in critical infrastructures. In comparison to the first case study, we identified more categories of information sources for eliciting assets, use cases, threats, countermeasures and example policies. We elicited several suitable pieces of information from those sources. As it turned out, the involvement of participants covering multiple stakeholder roles and the analysis of existing documentation about related threats and risks were the most helpful improvements in terms of information quantity. In terms of result quality, the involvement of asset owners and users was particularly beneficial, as they had the intrinsic motivation to seize the opportunity to specify and enforce policies that met their own security demands.

Regarding H2 (Completeness of elicited information), we observed that the experts extended the policy template catalog by 10 policy templates after the workshop. This led to an initial completeness rate of 75 percent.

With respect to H3 (Correctness of elicited information), we observed that the experts found improvement potential in 3 out of 40 policy templates. Thus, 93 percent of the policy templates were correctly elicited according to the feedback we received from six project partners.

In addition, we learned that two unsupportive situations might occur if the method experts provides too much information (e.g., assets, threats or policy templates) to the participants of the elicitation method before eliciting any information from them:

- First, the provision of information can strongly bias the method results. The participants can easily get the wrong impression of completeness if a lot of work has already been done by the method expert, especially if they do not have an intrinsic motivation for eliciting policy templates.

We assume that starting the elicitation »from scratch« with a structured elicitation approach might produce better results. The stakeholders will not be distracted or misled by existing information, but concentrate first on their own major concerns.

- Second, some stakeholders prefer to discuss existing work rather than focus on the elicitation of new information. This influences other participants to join discussion. Of course, discussions about the quality of the information are helpful, but need to be limited to avoid interrupting the elicitation process. The method expert that acts as a moderator in meetings and workshops and needs to remind participants to focus on the elicitation.

Thus, we concluded that we need to test the elicitation »from scratch« with an improved policy template elicitation method in the next case study. Consequently, it was mandatory that the third version of the policy template elicitation method contains an elicitation step for retrieving potential countermeasures from different stakeholder roles. Based on their wide-ranging expertise, they might know »unconventional« countermeasures that fit well in the specific application domain, but are not known to the method expert while creating the policy templates.

Generally, we learned that a more structured and iterative elicitation method with multiple elicitation rounds for assets, threats and countermeasures can improve the current method. The list of assets, use cases, threats and countermeasures might increase rapidly during elicitation. If the time for elicitation is limited, we need to prioritize for which assets we want to identify threats. Only eliciting assets and their properties in the first round facilitates such a prioritization and narrows down the focus of the elicitation. We concluded that we need to explore prioritization scales for assets and threats in order to put a better focus on the relevant ones.

We also observed that a lot of information is orally presented during discussions, but never written down by the participants of the workshop. We need to have a minute taker to capture this information during workshops and other meetings.

Regarding RQ3 (Formalization) and H5 (Completeness of policy template model), we found that policy templates can be adequately formalized with our policy template model. All elements for creating a policy vocabulary (policy template model instance) for the »SECCRIT« demonstrator including ILPTs for the policy language IND<sup>2</sup>UCE in version 1.1 were available.

We observed that our improved policy template model with the simplified transformation rules for ILPs is much easier to instantiate and less error-

prone than the previous version in the first case study. We did not measure the exact time required to instantiate the model or the number of errors made during instantiation to compare the two versions, but the improvement was clearly noticeable.

Finally, with respect to RQ4 (Automation) and H6 (Feasibility of automation of PAP creation), we explored the user interface generation using multiple view and presentation modules based on the same policy vocabulary. We built three PAPs for the UI frameworks »Swing«, »Android« and »Web« by using the respective view modules of the PAP generation framework. In addition, we used the two presentation modules that implement the specification paradigms »template instantiation« and »default policies«. We were able to start the PAPs and import the policy vocabulary for the »SECCRIT« demonstrator. All user interfaces were correctly generated and all policy instances of policy templates could be specified successfully. Thus we regard H6 as confirmed because »PAPs with multiple specification paradigms can be generated from a policy template model instance«.

### **8.3.5 Threats to validity**

Below, we address threats to validity with respect to the policy specification experiment. The threat categories are explained in Section 8.2.5.

#### ***Internal Validity***

The selection of participants for the elicitation was solely based on their willingness to contribute to the policy template catalog and their availability for the validation workshop. We did not select the participants according to their stakeholder roles. However, as we covered all identified roles with participants, we do not see the participant selection as a significant threat to internal validity.

In this case study, the creator of the method for usable PAP generation took the role of as method expert. We decided to do this in order to collect as much experience with the execution of the method as possible to reveal further improvement potential. However, this role assignment poses a threat to internal validity.

We ascertained in this case study that the improved policy template model led to an easier and less error-prone instantiation of the model. We see three threats to internal validity regarding this finding:

- First, we did not objectively measure the ease of use of the model. Our claim solely bases on a subjective estimation by the method expert. However, the efficiency of the policy template model

instantiation is not in the focus of this thesis, and the improvement was apparent.

- Second, the same person performed the model instantiations in both case studies. Thus, we cannot exclude learning effects.
- Third, the policy templates of the two case studies are different. We do not know how the complexity of policy templates influences the instantiation of the policy template model. Further studies are required to gain better insights.

### ***External Validity***

We instantiated the policy template model only for three policy templates. We doubt that those three templates cover all relevant requirements regarding the policy template model. This poses a threat to external validity. Further investigations must be performed to confirm the generalizability of our findings.

We only applied the method for usable PAP generation in one application domain in the context of this case study. Further applications are necessary to generalize the feasibility of our method for different application domains.

### ***Conclusion Validity***

Regarding H6, we conclude that we can automate the PAP creation for multiple specification paradigms. However, we do not know whether this hypothesis applies to all possible specification paradigms. This poses a threat to conclusion validity. We need to explore the automated PAP creation for further specification paradigms.

## **8.3.6 Summary**

In the »SECCRIT« case study, we confirmed that the concept of policy templates is suitable for specifying security and privacy policies in a PAP. We elicited 40 policy templates for the application domain of cloud services for critical infrastructure IT. Ten of those templates were added during the validation phase. In addition, minor errors were found in three policy templates. We successfully demonstrated the instantiation of the policy template model and the generation of user interfaces for policy specification in PAPs with three policy templates. These three policy templates were included in a »SECCRIT« demonstrator. For this demonstrator, we provided three PAPs: one as a Java application with the »Swing« UI framework, the second as an Android app and the last as a web service. We included transformation rules for generating ILPs into the policy vocabulary. Thus, a user of the demonstrator was able to specify a

security policy on the specification level with our PAP and could then try out the effect of the respective enforced ILP.

## 8.4 Summary and Conclusion

Regarding our research questions and hypotheses, the first two case studies yielded the following findings:

- RQ2 (Elicitation): We applied two versions of the policy template elicitation method in the two case studies. These method versions were preliminary, thus the results may not reflect the quality of results that the third version would have produced.
  - H2 (Completeness of elicited information): According to the experts who validated the method results in the case study »SINNODIUM« (seven policy templates), the list of policy templates was complete with respect to the necessary specification options for policies in the application domain. In the »SECCRIT« case study, the experts extended the policy template catalog by ten additional policy templates.
    - Q2.1: Is the policy template elicitation method capable of eliciting 90% of all necessary policy templates for the application domain?
    - M2.1: We elicited 79% of all policy templates from the application domain  $((7+30)/(7+40) = 79\%)$ .
      - ➔  $H_{2_0}$  cannot be rejected as we were not able to elicit 90% of the necessary policy templates for the application domain as we missed some policy templates during the elicitation in the »SECCRIT« case study.
  - H3 (Correctness of elicited information): According to the experts who validated the method results in the case study »SINNODIUM« (seven policy templates), all derived policy templates were correct. In the »SECCRIT« case study, the experts found improvement potential in three out of 40 policy templates.
    - Q3.1: Is the policy template elicitation method capable of eliciting correct policy templates that cover the security and privacy demands from the application domain?
    - M3.1: The policy template elicitation method allowed us to elicit 94% of the policy templates correctly  $((7+37)/(7+40) = 94\%)$ .
      - ➔  $H_{3_0}$  can be rejected.

- H4 (User acceptance of elicitation method): Overall, we received positive feedback on our policy template elicitation method from the participants of the case studies (two participants in »SINNODIUM«, sixteen participants in »SECCRIT«). Still, we got fruitful hints for improving the method, which we considered in the third version of the method.
    - Q4.1: Do users rate a workshop in which the policy template elicitation method is applied as a positive experience?
    - M4.1: 100 percent of the participants (18 out of 18) that we asked gave us positive feedback regarding the participation in a meeting or workshop in which the policy template elicitation method was applied.

→ H4<sub>0</sub> can be rejected.
- RQ3 (Formalization): in the »SINNODIUM« case study, we were able to instantiate a policy vocabulary with all seven derived policy templates. In the »SECCRIT« case study, we selected three policy templates for the demonstrator, which could all be expressed in the policy template model:
  - H5 (Completeness of policy template model)
    - Q5.1: Is the policy template model capable to represent more than 90 percent of the elicited security and privacy demands in the form of policy templates?
    - M5.1: We were able to model 100 percent of the derived policy templates in the policy template model  $((7+3)/(7+3) = 100\%)$ .

→ H5<sub>0</sub> can be rejected.
- RQ4 (Automation): We successfully demonstrated the generation of user interfaces for policy specification in both case studies. This includes two PAPs that use different view modules and fully support the two presentation modules that implement the specification paradigms »template instantiation« and »default policies«. We regard the feasibility for automated PAP creation as approved. However, we still see a need to explore the generation of further specification paradigms.
  - H6 (Feasibility of automation of PAP creation)
    - Q6.1: Is the process of user interface creation for the task of policy specification automatable for multiple specification paradigms and UI frameworks?

- M6.1: The user interface creation for 100 percent (2 of 2) of the tested specification paradigms could be automated.
- M6.2: The user interface creation of PAPs could be automated for 100 percent (3 of 3) of the tested UI frameworks.

➔  $H_{6_0}$  can be rejected.

The »SINNODIUM« and »SECCRIT« case studies successfully demonstrated the application of our method for usable PAP generation (excluding the user to specification paradigm mapping) in two different application domains. In both studies, we elicited policy templates with the policy template elicitation method, instantiated the policy template model to create a policy vocabulary and we used the PAP generation framework to create PAPs for the specification of policies with generated user interfaces. These case studies served to improve the method for usable PAP generation, thus, different versions of our contributions were used in the consecutive case studies.

## 9 Evaluation for Validation

In the second part of our evaluation, we focus on the validation of our contributions. In this chapter, we describe an application of the entire method for usable PAP generation and validate the contributions with respect to the research goals (see Section 1.3) and hypotheses (see Section 1.5) of this thesis.

We structure this chapter as follows: Section 9.1 explains our research approach. In Section 9.2, the »BeSure« case study is presented, followed by the »Digital Villages« case study in Section 9.3. We Section 9.4 describes our policy specification experiment. In Section 9.5, we summarize our validation results.

### 9.1 Research Approach

In the validation of our work, we focus on all of our five contributions. First, we conducted two case studies mainly to test Hypotheses H2 to H5 (see Section 1.5) in order to find valid answers to our research questions RQ2 to RQ4 (see Section 1.3). We applied the method for usable PAP generation in both case studies:

- The »BeSure« case study was the first application of our final method for usable PAP generation (excluding user to specification paradigm mapping). In this case study, we evaluated the usability of a PAP with the specification paradigm »template instantiation«. The study was conducted between 2015 and 2016 with the industrial partner »DATEV« in the application domain of data classification and data-based security policies.
- The »Digital Villages« case study applied the method for usable PAP generation including the mapping of users to specification paradigms. The study was conducted in 2017 with colleagues from »Fraunhofer IESE« in the application domain of digital services in smart rural areas.

Second, we conducted an experiment to test our hypotheses H1.1 to H1.4 in order to answer our research question RQ1. We used the policy vocabulary from the »Digital Villages« case study containing policy templates for generating a realistic PAP with four different specification paradigms. We let participants specify policies according to predefined specification tasks and measured effectiveness, efficiency and user



satisfaction with all specification paradigms in order to confirm hypotheses H1.1-H1.4.

## **9.2 Case Study: Software Campus Project »BeSure«**

To explore the applicability of our method for usable PAP generation in another actual application domain, we carried out a case study in the context of the research project »BeSure« together with the industry company »DATEV«. On this occasion, we assessed the usability of our generated PAP together with »DATEV«.

### **9.2.1 Project Summary**

The goal of the Software Campus project »BeSure« was to gain a better understanding of the specification of security policies from an end-user perspective. In »BeSure«, we developed a holistic methodology that increases the usability of security and privacy PAPs for different stakeholders while providing a reduced complexity and a vocabulary tailored to the application domain for security policy specification. This should enable stakeholders with different levels of knowledge to specify security policies more easily and with fewer mistakes. The results of the project contributed to the method for usable PAP generation. »BeSure« was funded by the German Ministry of Education and Research under grant number 01IS12053.

We performed the evaluation of the project together with the industrial partner DATEV. DATEV must pay special attention to the protection of customer-related data (e.g., financial and tax-related data), as their business model is based on a trustworthy processing of this type of highly sensitive data. In addition, DATEV's business processes have to comply with various regulations and legal obligations. New projects at DATEV require that project-specific security policies are specified, depending to external and internal regulations and the data classification of the project. In the long-term, DATEV wants to provide their project managers with tool-supported policy specification. Thus, the application domain for the case study is the project-based specification of security policies.

We applied the policy template elicitation method in cooperation with DATEV and elicited assets, threats and countermeasures for the application domain. We derived policy templates, instantiated the policy template model and generated a PAP with the specification paradigm »policy templates« for the operation platform »Android«. In a second workshop, we evaluated the usability of the generated PAP.

## 9.2.2 Design and Execution

The main goal of the case study was to verify the general applicability of the method for usable PAP generation. Thus, the individual contributions of this work needed to be examined. Therefore, we split the case study into three parts:

- Policy template elicitation
- Policy template model instantiation and PAP creation
- Usability evaluation of the PAP

### *Policy Template Elicitation*

In the first step of our case study, we applied the policy template elicitation method in the application domain »data classification and data-based security policies« of the industrial partner DATEV. For the policy template elicitation method, we first aimed to find answers to RQ2 (elicitation; see Section 1.5.2) and to prove Hypotheses H2 (completeness of elicited information), H3 (correctness of elicited information) and H4 (user acceptance of elicitation method), described in Section 1.5.2. In the context of this particular case study, we refined RQ2 into the following research questions:

- **RQ2.1 – Feasibility of RE techniques:** Are the applied RE techniques suitable to elicit an assets, threats, and countermeasures for a given application domain?
- **RQ2.2 – Stakeholders:** Which stakeholders or roles need to be involved to elicit required information when applying our policy template elicitation method?
- **RQ2.3 – Derivability of policy templates:** Is the information elicited in the elicitation workshop sufficient and suitable to derive policy templates for the given application domain?

We planned the application of the policy template elicitation method with all five method steps of the policy template elicitation method. Due to the spatial distance of the method expert and the contact person, we decided to initialize the project with phone calls and email communication with the contact person at DATEV in the first method step. Several phone calls and email conversations were required to gather all necessary information for the preparation of the elicitation workshop. We received a data classification guideline from DATEV from which we extracted exemplary assets, threats and countermeasures. We identified the following constraints for the workshop: There are three participants and the workshop is limited to a duration of three hours.

We planned a workshop based on the constraints and available stakeholders. Due to the time constraint of three hours at maximum for the whole workshop and the availability of only three stakeholders as participants, we prepared the workshop as follows: We chose the »brainstorming on cards« method for the asset elicitation (i.e., a group discussion). Data classes were collected as assets and enriched with various information, such as asset owners, monetary value and sensitivity. Additionally, we collected applicable laws, regulations, and typical use cases for each asset on cards. We performed the »ranking method« for the prioritization of assets. We had to limit the scope of the application domain due to the time constraint. However, we did not continue with the top ranked assets, but selected the three most relevant and diverging ones in terms of monetary value and sensitivity in order to evaluate the method for different asset types. Similar to the asset elicitation, we elicited the threats with »brainstorming on cards«, enriched them with various information (e.g., likelihood, potential damage, attackers) and clustered them accordingly. We also chose the »ranking method« for the prioritization of threats and selected the respective top three threats for the elicitation of countermeasures. In order to produce many countermeasures within a short period of time, we applied an adaptation of the »6-3-5 method«.

We conducted the elicitation workshop on April 14, 2015 with three participants, one method expert moderating the workshop and a minute taker. Among the participants, one had the stakeholder role of a security expert, one was a domain expert and one was a legal expert. All participants were asset owners and users. One participant had to leave the workshop directly before the countermeasure elicitation. The elicitation and prioritization of the assets took approximately one hour; we spent about 45 minutes on the threats and applied the »6-3-5 method« as planned for 30 minutes. The results were digitized by the method expert, and he created example policies from the elicited information. In the final step, the method expert derived policy templates. For the derivation, the expert generalized the example policies. To this end, he identified the variable parts of each policy and defined suitable values or value ranges for instantiation. Finally, the elicited policy templates were validated by the DATEV security expert. The post-processing of the workshop (including digitization of elicited information, derivation and validation of templates) required approximately two person days.

### ***PAP Generation***

After the elicitation phase, we instantiated the policy template model with the information collected during the elicitation and generated a PAP. For the policy template model and the PAP generation, we aimed to find answers to Hypotheses H5 (completeness of policy template model) and

H6 (feasibility of automation of PAP creation). After the policy templates had been validated, the method expert instantiated the policy template model and, thus, created a policy vocabulary. When instantiating the policy template model, the method expert checked whether all information could be expressed to confirm the completeness of the model (H5).

Next, the method expert evaluated whether the automated PAP generation (H6) works with the instantiated policy template model and the resulting PAP is fully functional. In order to assess the generated result, he imported the policy vocabulary in the Android PAP that uses the »template instantiation« paradigm. Together with DATEV, we decided not to support the transformation into ILPs in the PAP. Thus, this part of the policy template model was not instantiated and users were only able to specify SLPs in the PAP.

### ***Usability Evaluation of the PAP***

In the final step, we evaluated the usability of the generated PAP together with DATEV. Our goal was to answer RQ1 (see Section 1.5.1). However, we only evaluated the specification paradigm »template instantiation«. Thus, we refined the research question as follows:

- RQ1.1 – Usability of PAP: Is the generated PAP with the specification paradigm »template instantiation« usable for stakeholders of the application domain?

Together with DATEV, we carried out a second workshop to evaluate the usability of the generated PAP. We split the workshop into two phases: exploration and discussion. During the exploration, we asked the participants to apply the Android PAP and to fill out a questionnaire in parallel. The task of the participants during the exploration was to answer the questions with the following mindset: »Imagine that you need to specify security policies for a new project as a project leader in the company«. We did not define concrete specification tasks. The questionnaire contained the following five questions (we used the term »policy editor« instead of PAP):

- The specification of security policies is a challenge in the company (1—low to 5—high)
- Name the three most positive and the three most negative aspects of the policy editor.
- Is there any possible application for such a policy editor? If yes, which?
- What would be the benefits of introducing such a policy editor?

- Which additional features does the policy editor need to provide in order to be acceptable?

In addition, the participants filled out a AttrakDiff word pair sheet [165] to describe the usability of the PAP.

The second workshop phase was a discussion round, in which the feedback of all participants was presented and discussed. We explicitly asked for positive and negative experiences, potential extension points (e.g., other platforms) and scenarios where such a PAP would be beneficial. The questions were similar to the questions already asked in the questionnaire. However, we wanted to elicit information from the discussions between the participants.

We conducted the second workshop for the usability evaluation of the PAP on February 17, 2016. The same three participants from the elicitation workshop joined. They represented the security, domain and end-user perspectives, which we consider the main stakeholder groups for security policy specification at DATEV. The evaluation started with a short introduction of the workshop goals. We explained the functionality of the PAP in a slideshow with screenshots and presented the questionnaire and the AttrakDiff method. Next, the participants tested the PAP for about 25 minutes. They answered the questionnaire and the AttrakDiff sheet in parallel. Finally, we had a discussion for 25 minutes. During the trial phase of the workshop, we did not track the concrete user interactions with the PAP nor did we store the specified security policies.

Table 18: Asset »Communication Data«

|                           |   |
|---------------------------|---|
| Asset ID                  | A8  |
| Asset                     | Communication data (e.g., emails)   |
| Data Owner                | Employees and specialty department  |
| Example Use Case          | Bring your own device   |
| Prioritization Properties | Monetary value of asset: medium (€€)<br>Sensitivity of asset: internal use only |
| Legal Regulations         | German laws HGB, TKG, SigG and GDPR   |

### 9.2.3 Results

We split the results section into three parts: policy template elicitation, PAP generation and PAP evaluation.

Table 19: Threats for Asset »Communication Data«

|                        |   |
|------------------------|---|
| Threat ID              | T4-T6   |
| Related Asset ID       | A8  |
| Related Asset          | Communication Data  |
| Attackers              | Data theft  |
| Top 3 Threats          | <p>T4: Unintentional sending of hidden, sensitive information</p> <ul style="list-style-type: none"> <li>probability: permanently (high)</li> <li>damage: costly (medium)</li> </ul> <p>T5: Falsifying information (e.g. manipulation of draft contracts, obtaining financial advantages, etc.)</p> <ul style="list-style-type: none"> <li>probability: almost impossible (low)</li> <li>damage: costly (medium)</li> </ul> <p>T6: Unintended disclosure to third parties (unencrypted sending or wrong recipient)</p> <ul style="list-style-type: none"> <li>probability: permanently (high)</li> <li>damage: costly (medium)</li> </ul> |
| Other threats          | <ul style="list-style-type: none"> <li>Misdirection / open distributor</li> <li>Phishing</li> <li>Accidental disclosure of highly confidential data internally</li> <li>Generous allocation of mailbox authorizations</li> <li>E-mails with long attachments</li> <li>Use of not permitted communication methods</li> <li>Transmission of data with highest classification to unauthorized persons</li> </ul>   |
| Existing Documentation | not available   |

### ***Policy Template Elicitation***

In total, we identified twelve assets: project data, employee data, supplier data, job data, customer data, communication data, contact information, source code, system logs, information for employees, public data and technical configurations. We selected the assets communication data (see Table 18), job data (see Table 42 in Appendix E.1) and public data (see Table 44 in Appendix E.1) for the threat elicitation.

Table 20: Countermeasures for Threat »T4: Unintentional Sending of Hidden, Sensitive Information«

|   |
|---|
| Countermeasures for threat:<br>T4: Unintentional sending of hidden, sensitive information |
| Reminder before sending email   |
| Provide deletion function for removing sensitive information from email                   |
| Regular awareness raising through warning messages  |
| Automatic removal of sensitive data (data loss prevention)                                |

For these three assets, we elicited 27 threats in total (see Table 19 as well as Table 43 and Table 45 in Appendix E.1). Ultimately, we identified 39 countermeasures for the elicited threats. We present examples in Table 20 as well as Table 46 and Table 47 in Appendix E.1. From these threats and countermeasures, we finally extracted fourteen policy templates with the elicited information (see Table 21 and Table 48 in Appendix E.1).

Table 21: Policy Template »Secure Email Sending«

| ID                     | Policy Template Name | Asset  | Target System           | Security/Privacy Goal      |
|------------------------|----------------------|--|-------------------------|----------------------------|
| 1                      | Secure email sending | Communication Data   | Email client and server | Confidentiality, integrity |
| Policy Template Syntax |                      | If [any employee   <employee>   <employee role>] sends an email [with attachments   containing sensitive information]*, then [inform the user   enforce encryption of email   enforce digital signature of email   delay the delivery of the email for <amount> <time unit> in order to enable revocation   remove sensitive information [automatically   after user confirmation] ]+. |                         |                            |
| Description            |                      | Employees often communicate via email with internal as well as external recipients. This communication must be protected because email content as well as attachments can contain sensitive information. This template allows the control of email sending.  |                         |                            |
| Threat                 |                      | Information leakage or manipulation of sensitive information   |                         |                            |
| Example Instantiation  |                      | If service employees send an email containing sensitive information, then inform the user, enforce encryption of email, and delay delivery of the email for 5 minutes in order to enable revocation.   |                         |                            |

### **PAP Generation**

We used the output of the policy template elicitation method to instantiate the policy template model. We were able to model all policy templates. We used the resulting instance of the policy template model, the policy vocabulary, for the automated PAP creation. We used the PAP generation framework with the view module »Android« and the presentation module »template instantiation« to create an Android PAP. Figure 64 shows a screenshot of the PAP, which was originally provided in German language.

### **PAP Evaluation**

Regarding our questionnaire, the participants reported that specifying security policies at DATEV is considered a rather challenging task. The average rating was 3.7 out of 5 points, where larger values denote bigger challenges. Thus, better guidance (e.g., by a usable PAP tailored to stakeholders of the application domain) could be beneficial for DATEV. The participants named the simple handling, the clarity and the structured,



unified specification process as benefits of the PAP. Regarding the specification paradigm »template instantiation«, the restricted variety of the templates, the unified and domain-specific diction of the policies and the structuring of the policies were positively mentioned.

The participants also found improvement potential. The set of 14 templates was perceived as confusing, although the PAP provides search and filter mechanisms for the policy template handling. Furthermore, the participants experienced the policy templates linguistically speaking as not yet »human«. Rephrasing the templates or providing a specification paradigm with more guidance could improve usability. An example would be a wizard with detailed explanations of individual customization options in the policy templates.

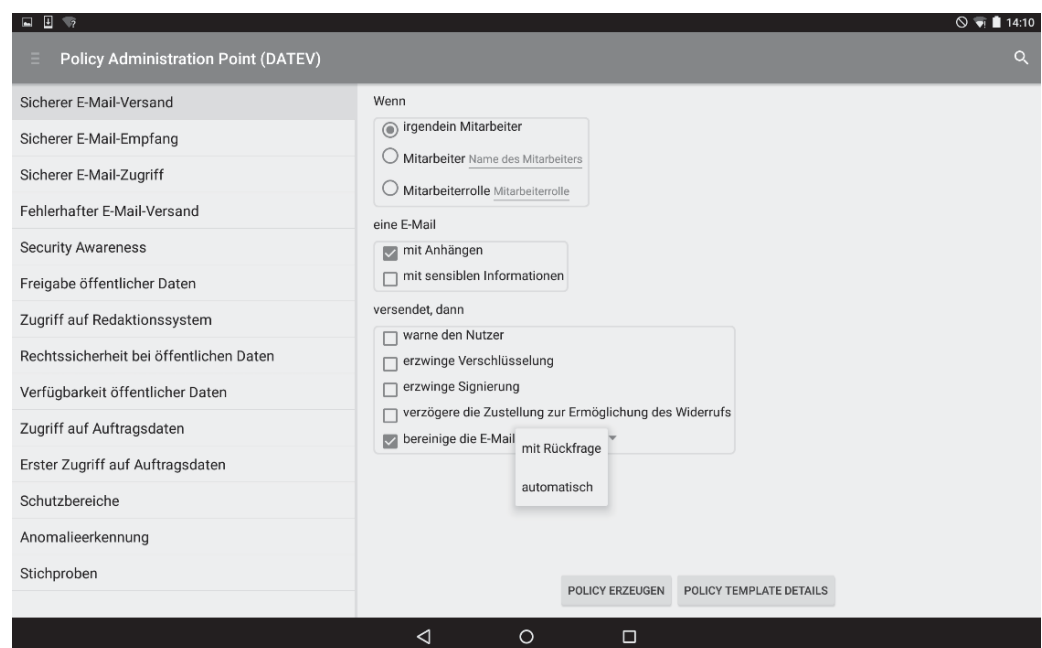


Figure 64: Example PAP Using View Module »Android«, Policy Vocabulary »BeSure« and Presentation Module »Template Instantiation«

The participants stated that their company could benefit from using such a PAP because it would foster a standardized and centralized procedure of specifying security policies. Currently, policies are specified in a more unstructured way using checklists and documentation of security policies in text files. Moreover, a PAP could empower data owners, especially non-experts, to specify security policies that reflect their personal protection needs.

The participants named the adaption of the PAP generation framework to existing systems as the most valuable extension point. That is, policies should be automatically transformed into ILPs that can be enforced in the target systems. Actually, this feature is supported by the PAP generation framework for XML-based policies. However, we excluded this aspect in



the first discussions with the contact person prior to the policy elicitation workshop. Another concern raised by the participants was that a clear process for the specification and maintenance of security policy templates at the company would be necessary in practice. However, we do not address such a process in this work.

The AttrakDiff method revealed that the user interface of the PAP was rated as »fairly practice-oriented« regarding the pragmatic dimension; the PAP was rated between neutral and task-oriented. This means that users can achieve their tasks with the PAP, but there is room for improvement. Users seemed to be stimulated by the PAP from the hedonic point of view, but only on an average level. Thus, there is potential for improvement for the hedonic quality as well. Regarding the hedonic quality »identity«, the PAP's mean value was located slightly above average level. Thus, our PAP met ordinary standards, but a higher value would bind the user more strongly to the PAP. With respect to the hedonic quality »stimulation«, the mean value was rated slightly above the average level. Thus, our PAP met ordinary standards, but improvements would motivate users more strongly. The attractiveness was rated moderate. The AttrakDiff test revealed that the generated PAP was accepted as a user-friendly and attractive tool, but there are still improvement potential regarding usability and attractiveness.

#### **9.2.4 Observations and Lessons Learned**

The elicitation part of our case study was based on the following evaluation plan in order to answer our research questions. It was verified whether the outcome of our elicitation matches the known assets, threats and countermeasures in the application scenario (RQ2.1). As we did not have a baseline, we relied on a subjective evaluation by DATEV experts during the validation of the policy templates. We checked whether we involved enough different stakeholders to elicit all information required by subsequent process steps (RQ2.2). In addition, we checked whether security policy templates could be actually derived from the information elicited and processed in the first four method steps (RQ2.3). Furthermore, domain experts were asked to validate the completeness (H2) and correctness (H3) of the derived policy templates. Obviously, they needed to consider that we only elicited threats for a limited set of assets and collected countermeasures only for the top ranked threats. Regarding completeness, we additionally checked whether all example policies could be instantiated with the policy templates. Finally, we asked for feedback regarding the policy template elicitation method and the elicitation workshop (H4).

Regarding RQ2.1, we found that the selected elicitation techniques led to a set of assets, threats and countermeasures that well reflect the application domain. The results were quite homogeneous and included

technical and organizational countermeasures. Some of them have already been implemented at DATEV and some are desired future extensions. The experts approved the elicited information in the validation. Thus, we rate the selected RE techniques as suitable in our setting.

With respect to RQ2.2, we assess the selection of stakeholders for our elicitation as positive. The participants covered all relevant perspectives on security policies for the application domain: technical and organizational IT security perspective, legal perspective and end user perspective. The stakeholders actively engaged in the workshop and revealed a lot of information during the elicitation. Thus, the selection of stakeholders was successful in our setting.

Regarding RQ2.3, we can confirm that the method provides sufficient information for the derivation of policy templates. We were able to derive 14 policy templates from the elicited information. Variable parts were easily identifiable in the example policies, and the information elicited during the workshop was sufficient for the method experts to define the concrete variables. The derivation of security policy templates worked efficiently.

The DATEV experts confirmed that the derived policy templates were all correct. That is, the templates correctly reflect the information elicited in the workshop, yielding a correctness of 100 percent (H3). Obviously, the policy template cannot completely cover the application domain, as we only partially elicited threats and countermeasures due to the time constraints of the workshop. We asked the experts to consider this fact in the evaluation. The experts neither reported any missing templates nor did they identify any desired policies that cannot be instantiated with the derived policy templates. Thus, we achieved a completeness of 100 percent for this application domain (H2). We asked the participants via email to validate our results. We cannot rate how diligently they performed the validation and thus, how reliable the query results described below are.

The participants perceived the policy template elicitation method as an applicable process for eliciting and deriving domain-specific policy templates for the application domain »data classification and data-based security policies« at DATEV (H4). They also appreciated the structure of the method and the requirements engineering techniques selected for the workshop. In fact, we did not face any kind of resistance during the workshop and received only positive feedback regarding the method after the workshop.

After completing the template derivation, we instantiated the policy template model with the output from the policy template elicitation method. We were able to express nine of fourteen policy templates

completely in the model. The other five policy templates contained a construct that was not yet supported by the policy template model: In selections, each element could only be selected once (concept of radio buttons and check boxes); however, the five policy templates would require multiple instances of selection elements. In the specific case, a variable inside a selection element let the user specify employees for which the policy should be enforced. Thus, as only one instance of a selection element could be created, the user would need to specify an individual policy for each user. This is neither usable for the specification nor for the management of policies. Thus, this finding indicated that the policy template model was not yet complete (H5).

We imported the policy vocabulary into the Android PAP app with the integrated PAP generation framework. The user interfaces for the specification of all policy templates were generated during the runtime of the PAP in an automated manner. Thus, we demonstrated that H6, which claims the feasibility of automation in the PAP creation process, holds in the context of this case study.

Finally, we evaluated the usability of the generated PAP with experts from DATEV. They overall liked the Android PAP app with the specification paradigm »template instantiation«, which positively answered RQ1.1. However, they gave valuable feedback regarding improvement potential, for example, regarding the grammar of policy templates and more guidance during the specification.

In summary, the case study yielded the following evidence supporting our research questions and hypotheses:

- **RQ1 (Usability of Specification Paradigms):** We evaluated the usability of an Android PAP with the specification paradigm »template instantiation«. The feedback of the participants and the results of an AttrakDiff test were positive, but revealed improvement potential.
- **H2 (Completeness of elicited information):** According to the experts who validated the method results, the derived policy templates cover the security demands of the application domain. However, the experts noted that not all assets and threats were investigated during the elicitation due to time constraints.
- **H3 (Correctness of elicited information):** According to the experts who validated the method results, all derived policy templates were correct.
- **H4 (User acceptance of elicitation method):** Overall, we got positive feedback on our policy template elicitation method.
- **H5 (Completeness of policy template model):** We were able to instantiate a policy vocabulary with all derived policy templates.

However, the version of the policy template model used in the case study lacked a construct of multiple instances of selection elements. Thus, in the context of this case study, we were not able to approve the completeness of the policy template model.

- **H6 (Feasibility of automation of PAP creation):** We demonstrated the feasibility of an automated PAP creation with a PAP realized as an Android app. At runtime, the PAP was capable of generating all user interfaces for the specification of policy templates.

Besides the contributions to our research questions and hypotheses, we made the following observations.

During the elicitation workshop, we found that a better alignment of the elicitation to use cases or business processes could improve effectiveness and efficiency of the elicitation. If assets are used in multiple use cases, the threats and countermeasures vary greatly. The elicitation of threats based on use cases could ease their elicitation, as the participants only have to explore one usage scenario at a time. This could also improve the understandability of the policy templates, as they are focusing on one use case.

The combination of the stakeholder roles in our elicitation workshop led to good results and fruitful discussions. Domain, security and legal expertise was combined during the elicitation. We will further aim to have a mixture of these stakeholder roles in elicitation workshops.

Participants in the PAP usability evaluation rated the policy templates as not yet »human« from a linguistic perspective. This artificial appearance is most probably caused by the concept of policy templates. It is very challenging or on parts even impossible to define a syntax that allows the instantiation of policies with natural English or German grammar (on the specification level). It needs to be investigated whether better grammar in instantiated policy templates can be obtained by rephrasing the policy templates. Another improvement idea is to subdivide policy templates into less complex ones. We assume that it is easier to provide a more natural grammar a simplified policy template syntax. However, simpler policy templates require the user to instantiate more policy templates. This tradeoff requires further investigation.

According to their feedback, the participants required more guidance. They stated that they would appreciate some kind of wizard with detailed explanations of the variation points of the policy templates. This feedback led us to develop a new specification paradigm that supports a wizard approach for the specification of policies.

Together with the contact person, we agreed at the beginning of the case study that we would not create transformation rules for ILPs. The main reason was that our PAP generation framework currently only supports XML-based policy languages. At the time of this decision, the contact person informed us that DATEV did not have policy enforcement technology in place with a XML-based policy language in the area of the application domain. However, during the usability evaluation of the PAP, the participants stated that a connection of the PAP to target systems would be desirable. They named the Windows Group Policies as an exemplary target system. To support this connection, we would need to support non-XML-based policy languages, either directly with in the PAP generation framework or with an intermediate XML-based language that can be transformed into the policy language of the target system after the specification in the PAP. However, adding support for other ILP languages was not within the focus of this thesis work.

From a practical point of view, a well-defined process for the maintenance of security policy templates at the company would be mandatory. However, we do not address this aspect in this work.

### **9.2.5 Threats to validity**

In this section, we address threats to validity with respect to the policy specification experiment. The threat categories are explained in Section 8.2.5.

#### ***Internal Validity***

We have several dependent variables in the case study. Regarding the study design, the constellation and selection of participants is an issue. All participants were highly experienced and stemmed from complementary departments. They were extremely motivated, as they had an intrinsic interest in the results. One participant knew about our goals and hypotheses, which might have influenced his behavior during the case study. We do not know how these factors influenced the results and whether less motivated or less experienced participants would have caused worse results.

The selection and application of different RE techniques is another influencing factor. We tested »brainstorming on cards« and the »6-3-5 method« in the elicitation workshop. Both worked fine in our case study. However, we do not know whether other techniques are more suitable and produce even better results. We also do not know whether these techniques perform equally well in other application domains, and how strongly the stakeholders influence the success of the application of RE techniques.

Finally, the creator of the method for usable PAP generation executed the method on his own, including the moderation of both workshops. We decided to do this in order to collect as much experience with the application of the method. However, this poses a threat to internal validity. Future case studies and experiments should be performed by independent persons not related to this thesis work.

### ***External Validity***

Many aspects affect the quality of the results of our approach. We do not know how much the quality of the output of our policy template elicitation method depends on the expertise of the participants regarding the application domain, security and legal aspects. Moreover, we only evaluated the usability of the PAP with a small, potentially biased group of persons. We need to apply the method more often in different application domains with different stakeholders to confirm the generalizability of our method's feasibility, the user experience during its application and the completeness and correctness of its results.

### ***Conclusion Validity***

The derived security policy templates were validated by the participants of the case study. More meaningful validation results may have been achieved if the validation had been applied by different, independent security experts from the same application domain.

## **9.2.6 Summary**

In the »BeSure« case study, we positively evaluated the policy template elicitation method, the policy template model and the PAP generation framework together with DATEV. We applied the policy template elicitation method with stakeholders from DATEV. In a half-day workshop, we elicited twelve assets. For three of these assets, we identified 27 threats. For the nine major threats, we elicited 36 countermeasures. From this information, we derived fourteen policy templates. Overall, we obtained valuable results and received positive feedback from the participants. Next, we instantiated the policy template model. We identified one construct that we were unable to model, but was required by five policy templates. This deficiency rendered the policy specification for users more complicated, as potentially more policies need to be specified to compensate for the weakness of the model. Still, all policies could be specified. Thus, we assigned this problem a low severity. We used the resulting policy vocabulary to generate an Android PAP. We evaluated the usability of this PAP in a second workshop with experts from DATEV.



## 9.3 Case Study: »Digital Villages«

Together with our colleagues from Fraunhofer IESE, we performed another case study in the context of the research project »Digital Villages«. The goal of this study was to confirm the feasibility of our method for usable PAP generation in a real application domain. The generated PAP was also used for the policy specification experiment described in Section 9.4. Our work was not funded by the project.

### 9.3.1 Project Summary

The goal of the »Digital Villages« is to provide novel digital solutions to better connect rural regions, to strengthen the community and open up new opportunities for local businesses. The new services supporting these objectives run on a service platform. To illustrate the approach, consider the following examples: With the »DorfFunk« service as the communication center of the regions, citizens can offer their help, exchange goods and services, submit applications or simply chat with each other in a casual way. The »BestellBar« is a completely new kind of online marketplace. It combines the advantages of online shopping with those of local shopping. Retailers in the region present their products, which citizens can order online. The »LieferBar« is the bring-along service for the community. Here, citizens can see which parcels from the local online shop »BestellBar« are still waiting for delivery and can then take them along to their neighbors.

In all these services, citizens provide personal data. We want to enable the citizens to control the use of their data. Therefore, citizens shall be involved in the policy specification process so that they can express their own security and privacy demands. To this end, a user-friendly PAP is required.

We applied the policy template elicitation method in cooperation with colleagues from Fraunhofer IESE, who are developing the »Digital Villages« platform and are supporting citizens in using the platform services. We elicited assets, use cases, threats and countermeasures for the application domain in a workshop. We derived policy templates and validated them with the workshop participants. We instantiated the policy template model to create a policy vocabulary, and based on this vocabulary, we generated a PAP with the specification paradigms »policy templates«, »default policies«, »security levels« and »wizard« for the operation platform »Web«.

The elicitation of policy templates is part of the integration of the data usage control enforcement framework »IND<sup>2</sup>UCE« into the »Digital Villages« platform. However, as we had not integrated IND<sup>2</sup>UCE into the

platform at the time of the case study, we did not yet define ILP transformation rules for the IND<sup>2</sup>UCE policy language.

### 9.3.2 Design and Execution

Our main goal was to find evidence regarding Hypotheses H4 (User acceptance of elicitation method), H5 (Completeness of policy template model) and H6 (Feasibility of automation of PAP creation). As a by-product of the validation study, we wanted to elicit real policy templates to be used in the policy specification experiment, described in Section 9.4.

At the beginning of the study, we applied the policy template elicitation method in the application domain »Digital services in smart rural areas«. To this end, we first met with the contact person, the project leader of the »Digital Villages« project, to clarify the constraints for the policy template elicitation, but no specific constraints were imposed. We got access to information material about the »Digital Villages« platform including use case and architecture documentation. We were able to derive lists of relevant assets, user roles and use cases for the application domain from the provided documentation. Having this material available, we decided to extend and confirm those lists in the beginning of the workshop in a group discussion rather than eliciting assets, user roles and use cases from scratch.

We prepared an introductory slide show to explain the elicitation process and to present the initial lists of assets, user roles and use cases to the participants. In addition, we prepared the material for the chosen RE techniques and a catalog of exemplary threats and countermeasures in the event that the participants need some assistance in identifying the required information. Furthermore, we pinned exemplary results of each method step on a pin board so that participants were able to gain a better understanding of our expectations. As we had a positive experience with a half-day elicitation workshop in the »BeSure« case study, we planned the »Digital Villages« elicitation workshop to last four hours. We invited developers of the »Digital Villages« platform and project members that directly interact with citizens in the villages in which the »Digital Villages« services are offered.

Before the workshop, our understanding of the mapping of assets, users and use cases of the application domain (which user roles are actually using which assets within which use case) was incomplete. Therefore, we prepared a large matrix at a pin board with use cases and assets at the two axes. Participants were asked to insert the user roles in the cells that use the respective assets within the respective use cases.

We decided to elicit threats per use case, because we assumed it would better fit the mindset of the participants. The well documented use cases



represent the services of the »Digital Villages« platform and were well known to all participants. Thus, we assumed that the participants could easier think their way into the use cases for the threat elicitation than into assets. Due to the limited number of use cases, we decided to assess all of them and thus skipped their prioritization.

We elicited the threats for the use cases with the »3-6-5 method«. We chose this method to rapidly identify as many threats as possible. We asked the participants to collect threats for one specific use case on each of the 3-6-5 sheets. After the threat elicitation, we let each participant rate the likelihood and severity of each threat per asset. We selected the most relevant threats based on their severity and likelihood rating.

Last, we applied the »brainstorming on cards« method to elicit countermeasures. We chose this method because it stimulates discussion, as we wanted to find consensus on applicable countermeasures across the entire group.

The workshop took place on July 7, 2017 with five participants, one method expert moderating the workshop, a minute taker for documenting the results and an assistant for organizing the input from the participants (e.g., to pin moderation cards on the pin boards or to fill information into the slides). The author of this thesis acted as the method expert.

Five developers of the »Digital Villages« platform participated in the workshop. To some degree, they also interact directly with citizens of the villages where the platform is rolled out. Thus, the group of participants represented the stakeholder roles domain expert, technology expert, asset owner and asset user. We did not have access to legal or security experts for this elicitation.

In the first elicitation round of the workshop, the participants refined the lists of relevant assets, use cases and users of the application domain in a group discussion based on our initial lists. The assistant changed the lists on the fly. We ended after all participants approved the updated lists. Next, we mapped assets, use cases and users on each other with the prepared matrix. After that, we let the participants collect threats per use case with the »3-6-5 method« and prioritize the identified threats. Finally, we identified countermeasures and potential drawbacks of those countermeasures for selected threats with the »brainstorming on cards method«. At the end of the workshop, we asked all participants for feedback about their experiences to confirm the user acceptance of our policy template elicitation method (H4). Afterwards, we digitized and archived all workshop results with a photo protocol of all pin boards, and we scanned the 3-6-5 sheets. From the elicited information, we formulated exemplary policies that represent security and privacy

demands of users of the application domain. We used these example policies to derive policy templates.

To obtain evidence for the completeness of the policy template model (H5), we instantiated it with derived policy templates. We added projection rules for the different specification paradigms. We defined default policies for the specification paradigm »default policies«, the order of specification and smaller specification steps for the »wizard« paradigm and the different levels including the assigned default policies for the specification paradigm »security levels«. Due to the missing enforcement technology for security and privacy policies in the »Digital Villages« platform at the time of the case study, we abstained from specifying any ILP transformation rules.

To demonstrate the automation in the PAP creation process (H6), we generated user interfaces for policy specification with all four specification paradigm algorithms of the PAP generation framework and provided those in a PAP for the operation platform »Web«.

### 9.3.3 Results

We refined seven relevant use cases for the application domain »digital services in smart rural areas«, and we identified eight relevant assets that are used in these use cases. Ultimately, we agreed on nine user roles that use the assets in the use cases. The elicited use cases, assets and user roles are shown in Table 22.

Table 22: Lists of Elicited Use Cases, Assets and User Roles

| Use Cases                         | Assets  | User roles                              |
|-----------------------------------|---|---|
| Create account and authentication | Person data: Ordering person / Consumer of Exchange | Ordering person / Exchange Consumer (1) |
| Ordering via BestellBar           | Person data: Deliverer / Provider of Exchange       | Deliverer / Exchange provider (2)       |
| Delivering via LieferBar          | Merchant  | Merchant (3)                            |
| Exchanging via DorfFunk           | Order Data  | Platform operator (4)                   |
| Scientific analysis               | Delivery Data                                       | Care taker (5)                          |
| Debugging                         | Chat Data   | Scientist (6)                           |
| Administration                    | Trade Data  | Ministry (7)                            |
| Create account and authentication | Achievements  | External provider (8)                   |
|                                   | Log Data  | Third party operator (9)                |

In order to better understand the mapping of assets, user roles and use cases, we elicited this information in a matrix, as shown in Table 23. The user roles are represented by the numeric value assigned in Table 22.

Table 23: Mapping of Use Cases (X-Axis), Assets (Y-Axis) and User Roles (Numbers in Cells)

|  | Create account and authentication | Ordering via BestellBar | Delivering via LieferBar | Exchanging via DorfFunk | Scientific analysis | Debugging | Administration |
|--|-----------------------------------|-------------------------|--------------------------|-------------------------|---------------------|-----------|----------------|
| Person data: Ordering person / Consumer of Trade | 4,9                               | 4, 5, 8, 9              | 4, 5, 8, 9               | 1, 2, 4                 | 3-9                 | 4         | 4, 5           |
| Person data: Deliverer / Provider of Trade       | 4,9                               | 4                       | 4, 5, 8, 9               | 1, 2, 4                 | 3-9                 | 4         | 4, 5           |
| Merchant   | 4,9                               | 4, 5, 8, 9              | 4, 5, 8, 9               |                         | 3-9                 | 4         | 4, 5           |
| Order Data                                       |                                   | 4, 5, 8, 9              | 4, 5, 8, 9               |                         | 3-9                 | 4         | 4, 5           |
| Delivery Data                                    |                                   | 4, 5, 8, 9              | 4, 5, 8, 9               |                         | 3-9                 |           | 4, 5           |
| Chat Data  |                                   | 1, 4                    | 4                        | 1, 2, 4                 | 3-9                 |           |                |
| Trade Data                                       |                                   |                         |                          | 1, 2, 4, 6, 7           | 3-9                 |           |                |
| Achievements                                     | 4                                 | 4                       | 4                        | 1, 2, 4, 6, 7           | 3-9                 |           |                |
| Log Data   | 4                                 | 4                       | 4                        | 4                       |                     |           | 4              |

Using the »3-6-5 method«, we elicited 68 threats for the assets used in the use cases. Based on their prioritization, we selected 27 of them for the countermeasure elicitation. Table 24 shows an exemplary sheet from the threat elicitation for the use case »Exchanging«.

For the selected 27 threats, we elicited 53 potential countermeasures from the participants of the workshop. As an example, Table 25 shows the countermeasures that we retrieved for the use case »exchanging«.

Table 24: 3-6-5 Sheet for Threat Elicitation of Use Case »Exchanging« (Dmg: Damage; Pb: Probability)

| Use Case   |                 | Exchanging  |                 |   |                 |
|--|-----------------|---|-----------------|---|-----------------|
| Threats  |                 |   |                 |   |                 |
| Conclusions about life situations of a person through collection of all current and past offers / requests | Dmg: M<br>Pb: L | Identifying, when person is outside the home to provide help to others → Burglary | Dmg: L<br>Pb: L | Collect home addresses of persons → creating profiles of districts / villages                                       | Dmg: L<br>Pb: L |
| Teaser → Attack on the exchange consumer   | Dmg: H<br>Pb: L | Fake exchanges for collecting DigiTaler (currency in Digital Villages)            | Dmg: M<br>Pb: H | Using the DorfFunk for advertisement (e.g., advertisement in picture uploads)                                       | Dmg: N<br>Pb: N |
| Fraud with defective, fake or similar products   | Dmg: M<br>Pb: L | Defamation through bad recessions   | Dmg: M<br>Pb: L | -   |                 |
| Blackmailing with a bad rating, which everyone then sees   | Dmg: M<br>Pb: L | Use pictures for collecting details about residence, for example, for breaking in | Dmg: M<br>Pb: M | Giving information about storage location of exchange goods (e.g. in allotment garden). Can be used for burglaries. | Dmg: H<br>Pb: M |
| Fake account for scamming items  | Dmg: H<br>Pb: H | Health insurance companies / authorities check behavior or identify property      | Dmg: M<br>Pb: L | Employers trace what employees do in free time  | Dmg: M<br>Pb: L |
| Commercial use   | Dmg: M<br>Pb: M | Fake accounts for stealing  | Dmg: H<br>Pb: H | Lent items are never returned   | Dmg: H<br>Pb: L |

Table 25: Identified Countermeasures for Use Case »Exchanging«

| Use Case |  | Exchanging  |                                |
|----------|--|---|--------------------------------|
| ID       | Threat   | Countermeasure  | Side Effect                    |
| T1       | Commercial use   | Report function for fraud   |                                |
| T2       | Creation of fake account for stealing goods or data or for collecting data | Report function for fraud   |                                |
|          |  | Identification of users with ID cards on account creation   | Effort hinders potential users |
| T3       | Use of exchange data for burglary  | Trust through <ul style="list-style-type: none"> <li>Personal data</li> <li>Verified persons (post-Ident)</li> <li>Picture-based confirmation of persons by others</li> </ul> |                                |
|          |  | Configurable trust level for seeing offer → Only friends see offer  | Effort hinders potential users |
|          |  | Show address coarse-grained   |                                |
|          |  | Do not propose date and time of exchange  |                                |
| T4       | Fake exchange for collecting DigiTaler                                     | Limit number of exchanges per time frame  |                                |
|          |  | Upper limit of DigiTaler per exchange   |                                |

After the policy elicitation workshop, we documented all elicited information. Using this information, we created 44 example policies that represent security and privacy demands of the application domain. In our role as security experts, we extended this list of example policies by sixteen additional example policies, which can in our opinion support citizens in protecting their security and privacy. For illustration, Table 26 lists all example policies for the use case »exchanging«.

Table 26: Derived Example Policies for Use Case »Exchanging«

| Example Policy   | Type  |
|--|---|
| If citizens access exchange data, the provider's name will not be displayed and address only be displayed roughly before the exchange is accepted (e.g. only postal code or house number range: main street [1-50]). | IND <sup>2</sup> UCE                        |
| If citizens access exchange data, the date/time is not or only roughly displayed before the exchange is accepted.  | IND <sup>2</sup> UCE                        |
| Limit the number of exchanges per user and time period to prevent fake exchanges.  | IND <sup>2</sup> UCE                        |
| Own exchanges can only be seen by friends.   | IND <sup>2</sup> UCE                        |
| The number of DigiTaler per exchange is limited.   | Security Requirement / IND <sup>2</sup> UCE |
| Users can report fraud if the TauschBar is used commercially or data is collected for other purposes.  | Security Requirement                        |
| In order to create accounts, a valid identity card must be presented or postIdent must be performed.   | Security Requirement                        |

In total, we derived fourteen policy templates from the example policies. Table 27 shows the exemplary policy template »DorfFunk: Help requests and Offers«, which we also used in the policy specification experiment. We created the policy template only for help requests and offers (i.e. services that citizens can offer and request in their community), but not for all types of exchange data (e.g., trading of goods). Of course, another variable could be added to the policy template for setting the type of the exchange data. However, we did not want to make the policy templates too complicated. The trust levels were not part of the actual elicitation, but added by the method expert in response to a suggestion during the workshop that was supported by the participants.

The elicitation workshop and its results were originally documented in German. In this work, we presented English translations of the policy templates for the policy specification experiment.

One major goal of this case study was to elicit real policy templates for the policy specification experiment. Thus, we needed to create a policy vocabulary for the PAP used in the experiment. Therefore, we selected six policy templates that cover all three services. We instantiated the policy template model with these six policy templates. Additionally, we defined

projection rules so that we could generate all four specification paradigms supported by the PAP generation framework.

Table 27: Exemplary Policy Template »DorfFunk: Help Requests and Offers«

| ID                     | Policy Template Name               | Asset   | Target System | Security/Privacy Goal       |
|------------------------|------------------------------------|---|---------------|-----------------------------|
| 1                      | DorfFunk: Help requests and Offers | Personal Data   | DorfFunk      | Confidentiality / necessity |
| Policy Template Syntax |                                    | My help requests and offers can be viewed [by every citizen only by my friends only by citizens with at least the trust level [gold silver bronze]]. Before accepting the help request or offer, they are allowed to look at [my complete name not my name], [my complete address only street and city of my address only zip code and city of my address not my address] and [not the preferred appointment   only the date of the preferred appointment   only date and daytime of the preferred appointment   the date and exact time of the preferred appointment]. |               |                             |
| Description            |                                    | The user of the DorfFunk service can defined the visibility of own personal data when offering help requests.   |               |                             |
| Threat                 |                                    | <ul style="list-style-type: none"> <li>• T3: Use of exchange data for burglary</li> <li>• T2: Creation of fake account for stealing goods or data or for collecting data</li> </ul>   |               |                             |
| Example Instantiation  |                                    | My help requests and offers can be viewed by every citizen. Before accepting the help request or offer, they are allowed to look at not my name, only zip code and city of my address and only the date of the preferred appointment.   |               |                             |

We generated a PAP instance using the view module »Web« for the specification of privacy demands in the context of the project »Digital Villages«. The PAP provides the four presentation modules »template instantiation«, »default policies«, »security levels« and »wizard«. We present screenshots of the PAP in Appendix F.1.

### 9.3.4 Observations and Lessons Learned

We did not perform a proper validation of the workshop results and the derived policy templates, thus we do not have evidence to approve H2 (completeness of elicited information) and H3 (correctness of elicited information) for this case study.

Regarding H4 (user acceptance of elicitation method), we received positive feedback from all participants. They commended the elicitation with respect to use cases. In particular, they confirmed that it was good to emphasize the use cases in order to identify threats. The participants also stressed that they would have liked to have worked longer on the elicitation of countermeasures. They also liked that the workshop was not restricted to eliciting policies that are technically enforced, but that also

organizational policies were identified. However, we also received suggestions for improving our method. The process of mapping assets, use cases and user roles with the matrix needs to be better explained to the participants. The users also proposed to rate the damage and probability in teams. We agree that this discussion would be more meaningful if performed in teams or with the entire group. However, to save time we decided to let those values be set by individual participants. The participants suggested to have a longer break in the middle of the workshop, for example a lunch break. We opted against a lunch break, because in similar workshops, we experienced a decrease in the productivity of participants after lunch breaks.

With respect to H5 (Completeness of policy template model), we ascertained that we were able to instantiate the policy template model and the respective projection rules for the four specification paradigms for all six selected policy templates. Thus, the results of this case study indicate that the policy template model is complete.

Regarding H6 (Feasibility of automation of PAP creation), we demonstrated the automated creation of the user interfaces for policy specification within a web-based PAP, which implement the following specification paradigms to the user: »template instantiation«, »default policies«, »security levels« and »wizard«.

### **9.3.5 Threats to validity**

In this section, we address threats to validity with respect to the policy specification experiment. The threat categories are explained in Section 8.2.5.

#### ***Internal Validity***

In this case study, we acted as the method experts while conducting the policy template elicitation method. This poses a threat to internal validity. However, we strictly adhered to the documented process instructions we created, and we did not try to influence the results in any way.

We selected the participants for the workshop, which might have biased the results of the case study. In addition, all participants were researchers at Fraunhofer IESE, having a similar mindset and using similar terminology. We cannot exclude an influence of this homogeneity of the participant group on the results of the case study.

#### ***External Validity***

We elicited information from only five participants, all from the same organization, and only in a single workshop session. To confirm the



generalizability of the policy template elicitation method, we would need to apply this method with multiple participants groups for the same application domain.

The policy template model was only instantiated with six policy templates. We do not know whether the selected templates cover all potential constructs that need to be expressed in an instance of the policy template model. Thus, the small number of used policy templates poses a threat to external validity.

### **Conclusion Validity**

We did not have access to legal or security experts during the elicitation. This may have affected the workshop results. However, with our own expertise in the areas of security, privacy and applicable privacy protection laws (e.g., GDPR), we are convinced that we sufficiently took these aspects into account during the post processing of the workshop results (i.e., during the creation of example policies and the derivation of policy templates). Therefore, we rate this threat to conclusion validity as low.

We did not validate the final results of the policy template elicitation method with the workshop participants or domain experts. Thus, we did not evaluate the completeness and correctness of the elicited policy templates. Thus, based on this case study, we cannot conclude that the output of our elicitation method is correct or complete.

### **9.3.6 Summary**

In the »Digital Villages« case study, we positively evaluated the user acceptance of the policy template elicitation method, the completeness of the policy template model and the feasibility of automating the PAP creation process. During the half-day elicitation workshop, we were able to elicit nine assets, which are used in eight use cases by nine user roles. We elicited 68 threats based on these use cases. We selected the 25 threats with the highest damage potential and probability for further assessment, for which we identified 53 countermeasures. From the elicited information, we created 60 example policies that reflect security and privacy demands from the application domain. Using these example policies, we derived fourteen policy templates. Overall, we obtained valuable results, and we received positive feedback from the participants regarding the elicitation method. We instantiated the policy template model for six selected policy templates, which we then used in the policy specification experiment described in the next section. Moreover, we generated a web-based PAP that supports four different specification paradigms and that lets users specify policies with the six selected policy templates. The successful realization of a versatile PAP demonstrates the feasibility of the automated PAP generation.



## 9.4 Policy Specification Experiment

Users can use different specification paradigms for the specification of security and privacy policies. Each paradigm requires the user to make a certain number of decisions during the specification of his requirements and guides the user differently through the specification process. We assume that the appropriate selection of the specification paradigm for a user can have a positive effect on the usability of the PAP. Thus, we analyzed the PAP created in the »Digital Villages« case study with all four specification paradigms regarding effectiveness (objective and perceived correctness of specified policies), efficiency (necessary time span for specification) and user satisfaction (how much users like the paradigm).

With respect to the objective effectiveness of the PAP, we want to minimize the number of mistakes made by users during the specification of policies. In our experiment, we define a mistake as a deviation of the user input from the sample solution. The objective correctness of specified policies can be improved if fewer mistakes are made. The perceived correctness by a user can be improved if the user is better aware of his mistakes. Ideally, a PAP should enable the user to accurately self-estimate the objective correctness of his specified policies.

Another objective of the experiment was to find out which paradigms are suitable for certain users in terms of satisfaction and efficiency. We define satisfaction as the indicator of how much the users like to use the specification paradigm. We define efficiency as the time needed to specify policies with the given specification paradigm.

The experiment's main objective was to confirm our Hypotheses 1.1 to 1.4 about the effect of appropriate specification paradigm selection on the usability of a PAP for users (compare Section 1.5.1). The hypotheses are:

- H1.1: Objective effectiveness of PAP: On average, users make at least 30% fewer mistakes with a PAP when comparing the best matching specification paradigm to the worst matching specification paradigm.
- H1.2: Perceived effectiveness of PAP: On average, the users' self-evaluation regarding policy correctness (perceived correctness) when specifying policies with a PAP has at least a 30% higher accuracy when comparing the best matching specification paradigm to the worst matching specification paradigm.
- H1.3: Efficiency of PAP: On average, users are specifying policies at least 30% faster when specifying policies with a PAP comparing the best matching specification paradigm to the worst matching specification paradigm.

- H1.4: Satisfaction with PAP: On average, the user satisfaction during a policy specification with a PAP when using the best matching specification paradigm is significantly higher than with the worst matching specification paradigm.

We assume that the positive effect on usability holds for the entire group of users. To clarify this, we clustered users into five groups according to the persona model of Dupree [14], which differentiates users in terms of their security and privacy knowledge as well as their motivation to interact with PAPs. We refined RQ1 into the following research questions, which we want to answer in this experiment as well:

- RQ1.1: Do particular types of persons (represented by a persona) differ in terms of objective correctness when specifying policies with different specification paradigms?
- RQ1.2: Do particular types of persons (represented by a persona) differ in terms of correctly estimated perceived correctness (confidence regarding objective correctness) when specifying policies with different specification paradigms?
- RQ1.3: Do particular types of persons (represented by a persona) differ in terms of efficiency when specifying policies with different specification paradigms?
- RQ1.4: Do particular types of persons (represented by a persona) differ in terms of satisfaction when specifying policies with different specification paradigms?

In the following, we describe the design and execution of the experiment in Section 9.4.1 including details about the user tasks, the procedures and instruments, the technical setup and the invitation of participants. In Section 9.4.2, we describe our data analysis and the results. We discuss the results in Section 9.4.3 and the threats to validity in Section 9.4.4. We summarize our findings in Section 9.4.5.

### 9.4.1 Design and Execution

#### *Scenario and Tasks*

We aimed for evaluating our contributions in a realistic scenario. Therefore, we used the PAP that was generated with the method for usable PAP generation in the »Digital Villages« case study, which is described in Section 9.2.

In this scenario, village citizens use digital services such as a digital village bulletin board (called DorfFunk), an online marketplace with local merchants (called BestellBar) and a delivery service where citizens deliver

goods from local merchants to other citizens (called LieferBar). The participants of our experiment were asked to imagine that they use these novel digital services and that this could potentially affect their privacy, as their personal data is used in those services. The participants had the task to adjust the privacy policies of these services to given privacy demands. The presetting of the privacy requirements was necessary so that all participants could use the PAP in a comparable way. This enabled us to compare the observed mistakes made by the participants, their speed of specification and their satisfaction with the different specification paradigms of the PAP. The privacy demands were described as part of the six tasks formulated from the ego perspective:

- »When I place an order in the BestellBar app, I do not under any circumstances want to receive advertising from other providers that refers to the ordered product. They may not use my data.«
- »I do not like that all the citizens in my village know where I order goods. Therefore, only people who are considered to be as trustworthy as possible and my friends should be able to view my delivery requests in the LieferBar app.«
- »Before someone accepts my order in the LieferBar app, this person may know my name, but not exactly where I live. The name of my village with postal code would be ok. The potential deliverer may also know the dimensions of the package. However, further information on the address and the parcel should only be provided to the person after acceptance of the delivery request.«
- »If I am not at home, the delivery may be deposited at my house. If a person has accepted my delivery order, he/she is only allowed to find out via the App as close as possible to my front door where the storage location is.«
- »I want everyone in the DorfFunk to see my help requests and offers, but I don't want them to know that they are from me. Therefore, the other users should not be able to see my name or my exact address. My place of residence with zip code and the concrete day on which I need help should be sufficient. Further details, such as the exact address and the proposed time of day, can be seen after accepting my offer.«
- »I think it is important that scientists contribute to society through research. Therefore, I am willing to provide them my data for these purposes as long as they do not use my name.«

The requirements did not match one-to-one the wording in the policy templates and therefore the formulation in the user interfaces of the different specification paradigms in the PAP. The reason is that a one-to-one match would cause the participants to compare the buzzwords of the

task descriptions with the texts in the PAPs, but not their semantic content.

The scenario description and the tasks were provided on a digital handout, which is shown in Appendix G.2. Participants were advised to print out the handout. The scenario description was supplemented by a short video that introduces the novel, digital services for citizens of a village.

Policy specification interfaces representing the four specification paradigms »template instantiation«, »default policies«, »security levels« and »wizard« were provided in the PAP (all these specification paradigms are presented in Section 5.3.1). The participants had to complete the same six tasks for each of the four specification paradigms.

The paradigms »template instantiation« and »wizard« let the participant instantiate concrete policies from the templates. The paradigms »default policies« and »security levels« provide a limited list of already instantiated policies to choose from. In the paradigm »security levels«, the user can choose from three different sets of policies. All tasks in the experiment can be solved with all four specification paradigms. The sample solution can be found in Appendix G.4.

During the experiment design, we had to decide whether we should provide a perfect match with the tasks for the paradigm »security levels«. This means that one of the levels solves all tasks of the scenario. Such a perfect match is unlikely in real life. However, the lack of a perfect solution could confuse the participants in the experiment cause them to abort, which would probably severely affect the outcome of the experiment. Thus, we decided to offer a perfect match, because we did not want to spoil the experiment.

In this experiment, we determined the suitability of different specification paradigms for different user types. To conduct an experiment in which participants can actively specify policies, each specification paradigm must be implemented. However, this mixes the findings on the concepts of the specification paradigms and those on the corresponding implementations. To minimize the effects of the implementation, we asked usability experts to analyze the implementations of the specification paradigms in the PAP in order to make them as unobtrusive as possible. After the experiment, we also searched the free text comments by participants of the experiment for hints regarding problems with the operation of the PAP, but did not find any.

### ***Procedures and Instruments***

We designed our experiment as a publically available online experiment, which implies that it was uncontrolled to some extent. To avoid misuse,

we tried to control the experiment as closely as possible. Participants were only able to start the experiment with an individual, unique eight-digit participant ID. This unique ID was printed on the handout sent to each participant prior to participation. Each participant ID could be used only once to start the experiment. It was possible to interrupt the experiment and to continue with the participant id at the current step of the experiment. However, it was not possible to repeat already executed steps.

When entering the website provided in the handout, the participant first had to select their preferred language. The handout and the experiment were provided in German and English. Next, the participant started the experiment by entering their participant ID. With the start, participants agreed to an informed consent. The minimum age for participation was set at 18 years, as we would have needed the consent of the parents of minors in Germany for the analysis and exploitation of the recorded data. Thereafter, the experiment started. In the course of the experiment, each participant had to execute 18 steps (Screenshots showing the user interface for each step are presented in Appendix G.3):

1. The participant had to answer demographic questions about age, gender educational level.
2. The participant had to answer questions regarding the relationship to Fraunhofer IESE and to the research topic »IND<sup>2</sup>UCE«.

These answers were used to determine whether the participants' characteristics and capabilities have an impact on the results of the experiment.

3. The participant had to carry out a self-assessment regarding his knowledge of IT security measures and his use of IT security measures and web services.
4. The participant had to answer questions about his motivation to use PAPs and about reasons that hinder users to use PAPs more frequently.

The information acquired in the latter two steps was used to confirm the correct mapping of a persona to the participant.

5. The participant was asked to select one of the five personas by Dupree [14] that fits best to the own character and behavior. All personas were described based on nine to twelve original character traits formulated from the ego-perspective. The participant had to choose between the personas »marginally concerned«, »amateur«, »technician«, »lazy expert« and »fundamentalist«. The persona descriptions were presented to each participant in an individual, random order. A detailed description of the personas can be found in Appendix C.

6. The participant was asked to watch a video<sup>2</sup> describing the scenario and to read the task descriptions on the second page of the handout.
7. The participant received instructions on how to specify privacy policies in the following steps. More specifically, he was informed that he has to solve all six tasks with each of the four specification paradigms and that he has to rate each one directly after using it.

In the following steps, the participant was asked to specify privacy policies according to the tasks assigned to him on the handout. Therefore, we provided a PAP with four different policy specification interfaces, from which each one implements one of the four specification paradigms »template instantiation«, »default policies«, »security levels« and »wizard«. After each PAP use, the participant was asked whether he thinks that he solved all tasks correctly (perceived correctness). Next, he was asked how he liked this specification paradigm for configuring the given privacy policies. Finally, the participant had to rate whether he would like to use this specification paradigm in real life and whether the provided expressiveness is appropriate. Free text comments were welcome. The order in which the specification paradigms were presented to the participants was randomly determined to minimize and to statistically cancel out learning effects.

8. The participant had to solve all six tasks with the first PAP interface using the first specification paradigm.
9. The participant had to rate the first PAP interface.
10. The participant had to solve all six tasks with the second PAP using the second specification paradigm.
11. The participant had to rate the second PAP interface.
12. The participant had to solve all six tasks with the third PAP interface using the third specification paradigm.
13. The participant had to rate the third PAP interface.
14. The participant had to solve all six tasks with the fourth PAP interface using the fourth specification paradigm.
15. The participant had to rate the fourth PAP interface.

To analyze the effectiveness of the different PAP interfaces, we captured all specified policies in order to compare them to the sample solution.

---

<sup>2</sup> The video describing the scenario is available online: <https://www.youtube.com/watch?v=uNOP4R-SsxY>

16. The participant had to rank the four specification types according to his preference of using them in real life. We also requested free text comments about the reasons of the ranks.
17. The participant were asked to rate how well he can identify with the scenario and the chosen persona.
18. On the final screen, we thanked the participants for their participation and showed them their achieved objective correctness for each of the four specification paradigms.

### ***Technical Setup***

We implemented the experiment as a web application based on the Spring framework [166]. We used the Spring modules Boot to run the experiment as a web service and MVC (Model-View-Controller) for request and data handling. We used Bootstrap [167] as the front-end framework and Thymeleaf [168] as the server-side HTML template engine to create the user interfaces for each step of the experiment. However, the PAP interfaces for the four specification paradigms were generated with the PAP generation framework. We extended the code of the PAP to capture the user input and embedded it into the web application. We ran the Spring Boot application on a webserver during the execution of the experiment.

For each step of the experiment, we measured the elapsed time between the loading of the content of the current step and the confirmation of the participant to move to the next step. We also stored all entered data.

### ***Participant Invitation and Execution***

We acquired the participants by means of a non-binding invitation via e-mail on February 7, 2018. We invited persons in the circle of friends and acquaintances of the author as well as in the author's institution, Fraunhofer IESE. The participants were asked to forward this non-binding invitation to other persons. This initial email contained the information that the experiment is provided in English and German. On each reply of an interested person, we sent a specific invitation email (see Appendix G.1) with an attached handout (see Appendix G.2) in the preferred language. The handout contained all necessary instructions to start the experiment, including an individual participant id and the scenario description. We sent approximately 120 personal invitation emails. Because we strongly respect the anonymity of our participants, we deleted all these invitation emails from our outboxes directly after sending. Therefore, we do not know who of the invited persons actually participated in the experiment, and cannot establish any relation between participant IDs and natural persons.



We informed the participants in the first non-binding invitation email that the online experiment was accessible for 14 days. We closed the experiment and collected the results on February 22, 2018. Participants were also informed about the expected duration of the experiment of about 30-40 minutes, but we did not define a time limit for completion.

## 9.4.2 Data Analysis and Results

### *Data Analysis*

We executed all statistical analyses with SPSS 19 [169] and Microsoft Excel 2016 [170].

First, we checked the plausibility of the self-selection of personas by analyzing whether the self-reported security knowledge and motivation match the persona classification by Dupree [14]. Moreover, we analyzed how well participants identify with their selected persona.

To answer RQ1.1, we analyzed the number of mistakes the participants made. The different specification paradigms provided different expressiveness and thus required a different number of decisions being made by the participant: one decision for the specification paradigm »security levels«, six decisions for the »default policies«, 18 decisions for the »template instantiation« and 18 decisions for the »wizard«. As a consequence, the pure number of mistakes is not directly comparable. Therefore, we compared the ratio of incorrect decisions to all decisions. We performed Kruskal-Wallis tests ( $\alpha = 0.05$ ) to investigate whether the selection of the paradigm has an influence on the objective correctness for the entire participant group (compare Q1.1.1 in Section 1.5.1) and for each persona (compare Q1.1.2) as well as whether the persona has an influence on the objective correctness (compare Q1.1.3). For calculating the effect sizes, we used Cohen's  $d$  value ( $d_c$ : small effect:  $|d_c|=0.2$ ; middle effect  $|d_c|=0.5$ ; large effect  $|d_c|=0.8$ ). The persona »fundamentalist« was excluded from the persona analysis due to the small number of participants that selected this persona.

To answer RQ1.2, we analyzed the self-evaluation with respect to the objective correctness. To measure the perceived correctness, we asked the participants after the use of each specification paradigm whether they think that they solved all tasks correctly using this paradigm (zero mistakes). Finally, we compared perceived correctness with the objective correctness (zero mistakes) to obtain the self-evaluation. We performed Fisher's exact tests ( $\alpha = 0.05$ ) to determine whether the selection of the paradigm has an influence on the correctness of the self-evaluation for the entire participant group (compare Q1.2.1 in Section 1.5.1) and for each persona (compare Q1.2.3) as well as whether the persona has an influence on the correctness of the self-evaluation (compare Q1.2.3). To



calculate the effect sizes, we used Cramer's  $\phi$  value ( $\phi_c$ : for  $df=3$ ; small effect:  $|\phi_c|=0.06$ ; middle effect  $|\phi_c|=0.17$ ; large effect  $|\phi_c|=0.29$ ).

To answer RQ1.3, we measured the elapsed times to perform the policy specification steps in our experiment with the four PAP interfaces implementing the four specification paradigms under investigation. We calculated averages of the measured times per paradigm and per persona. Finally, we compared the average values. We performed Kruskal-Wallis tests ( $\alpha = 0.05$ ) to determine whether the selection of the paradigm has an influence on the time needed for policy specification for the entire participant group (compare Q1.3.1 in Section 1.5.1) and for each persona (compare Q1.3.3) as well as whether the persona has an influence on the time needed for policy specification (compare Q1.3.3). To calculate the effect sizes, we used Cohen's  $d$  value ( $d_c$ : small effect:  $|d_c|=0.2$ ; middle effect  $|d_c|=0.5$ ; large effect  $|d_c|=0.8$ ). The persona »fundamentalist« was excluded from the persona analysis due to the small number of participants that selected this persona. We excluded two samples from the analysis as their elapsed time was extremely high, which indicates a longer break during the experiment.

To answer RQ1.4, we asked the participants after each specification with a specification paradigm to rate the satisfaction with the PAP on a scale from 1 (»I really dislike this specification paradigm«) to 5 (»I really like this specification paradigm«). After completion of all four specification rounds, we asked participants to rank the four specification paradigms according to their personal preference. We calculated mean and median values per paradigm and per persona. We are aware that the calculation of mean values on Likert scales is controversially discussed in the literature. We therefore only label the extreme values of the scale and assume it to represent an interval scale. In addition, we calculated the percentage of participants per persona that ranked the specification paradigm on a specific rank. Finally, we compared these values. We also performed Kruskal-Wallis tests ( $\alpha = 0.05$ ) to determine whether the selection of the paradigm has an influence on the satisfaction (compare Q1.4.1 in Section 1.5.1) and whether the persona has an influence on the satisfaction (compare Q1.4.2). To calculating the effect sizes, we used Cohen's  $d$  value ( $d_c$ : small effect:  $|d_c|=0.2$ ; middle effect  $|d_c|=0.5$ ; large effect  $|d_c|=0.8$ ). The persona »fundamentalist« was excluded from the persona analysis due to the small number of participants that selected this persona.

### ***Participant Description***

Of the approximately 120 invited participants, 63 started the experiment. 61 participants completely finished the experiment with complete and valid data sets. We checked by hand whether the results are plausible to avoid any kind of misuse. In particular, we checked whether the elapsed

time per step was plausible and whether all participants tried to solve all six tasks with all four specification paradigms. In addition, we read all free text comments in order to find hints for problems during the experiment. After careful analysis, we found no reason to exclude a participant's result. However, we ascertained that two participants had a longer break within an experiment step. We excluded those samples from the analysis of efficiency, because this assessment depends on correct execution times.

Table 28: Personas Chosen by Participants of the Experiment

| Persona              | Number | Ratio |
|----------------------|--------|-------|
| Marginally Concerned | 12     | 20%   |
| Amateur              | 21     | 34%   |
| Lazy Expert          | 11     | 18 %  |
| Technician           | 14     | 23%   |
| Fundamentalist       | 3      | 5%    |

Of the 61 participants, 43 percent are female. The participants' age ranges from 18 to 82 ( $M=40.54$ ;  $SD=14.37$ ). The majority of the participants (33 out of 61) hold a university degree as highest educational level, nine participants hold a doctoral degree, seven have an entrance qualification for higher education and eleven a secondary school leaving certificate as highest level of education. About half of the participants (54%) were related to the authors' institution, 20 of them being scientific and eight non-scientific employees and five being students working with the authors' institution. 28 participants (46%) had no relation to the authors' institution.

Table 28 shows the distribution of the personas selected by the participants. Most participants chose the amateur (34%). The fundamentalist was only selected by three participants. The other personas were chosen between 11 and 14 times (18% to 23%).

To verify the plausibility of the persona self-selection, we asked the participants to rate their IT security knowledge and their motivation to use IT security measures. The participants' security knowledge fits well to their chosen personas, except for the lazy experts (see Figure 65). Based on Dupree's categorization (see Figure 14 on page 56), we expected the lazy experts to have higher self-estimated knowledge. The participants' security motivation fits to the model of Dupree as well (see Figure 66). Moreover, we asked the participants how well the chosen persona matches them on a scale from 1 (»Not very well, but it matched best out of the five options«) to 5 (»I can identify myself very well with the persona«). On average, the participants responded with a score of 3.75. Not a single person reported the value 1.

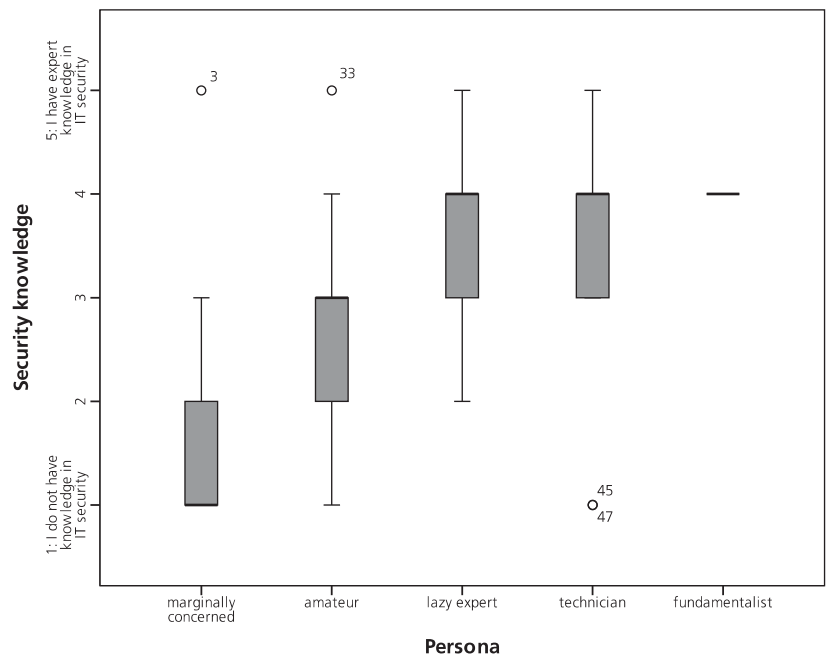


Figure 65: Security Knowledge to Persona Mapping

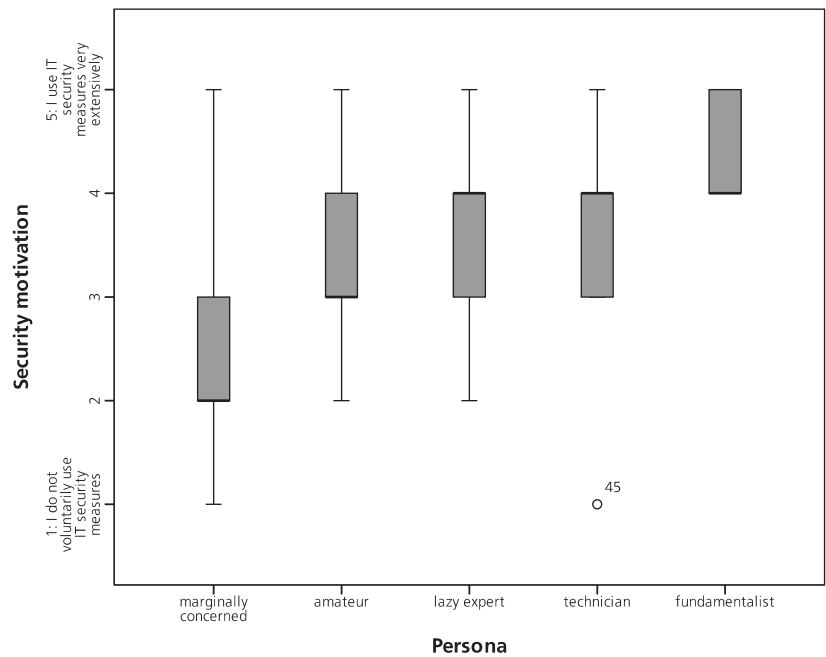


Figure 66: Security Motivation to Persona Mapping

The participants' age ranged from 18 to 82 ( $M=40.54$ ;  $SD=14.37$ ). Figure 67 shows the range and the median of the participants' age sorted by personas. Since the boxplots overlap, it is unlikely that the difference in age across the personas is significant. Thus, we conclude that the participants' age has no significant influence on the chosen personas. Nevertheless, there are differences in the range sizes of the age values. The figure shows that the personas marginally concerned and amateur have the biggest ranges in age including the oldest participants.

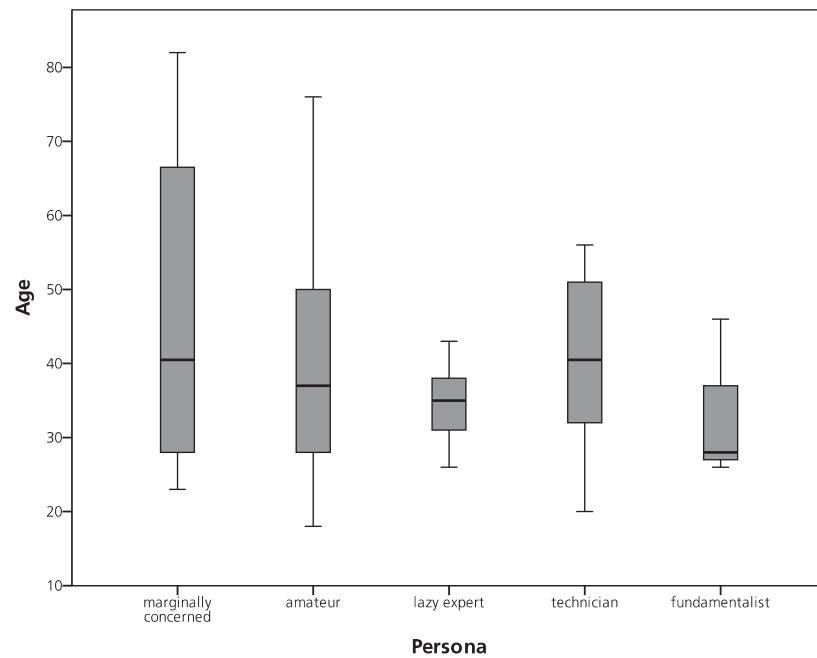


Figure 67: Boxplot Diagram of the Participants' Age.

### Objective Correctness

Three different aspects were taken into account in the analysis of objective correctness: First, we compared the cumulative number of mistakes of the participants that occurred per specification paradigm. Secondly, we determined how many perfect results with zero mistakes the participants achieved with each specification paradigm. Finally, the actual number of mistakes in relation to the possible number of mistakes per paradigm were analyzed. In the latter two cases, the influence of the personas was also determined.

Table 29: Mistakes per Paradigm

| Paradigm               | Necessary decisions (expressiveness) | Total mistakes made | Ratio of wrong decisions |
|------------------------|--------------------------------------|---------------------|--------------------------|
| Default Policies       | 6                                    | 79                  | 23.1%                    |
| Security Levels        | 1                                    | 4                   | 7.0%                     |
| Template Instantiation | 18                                   | 224                 | 21.8%                    |
| Wizard                 | 18                                   | 228                 | 22.2%                    |

Table 29 shows the cumulative number of mistakes made in total per specification paradigm. Not surprisingly, participants made the fewest mistakes with the »security levels«. Seven percent of participants chose the wrong security level. In the other three paradigms, which provided more decision options, about one in five decisions was taken incorrectly. Thus, for the entire population of the experiment there is no significant difference in objective correctness, except for the paradigm »security levels«.

Table 30: Participants with 100 Percent Objective Correctness

| Persona              | Number of participants with all paradigms correct | n per persona | % of participants per persona |
|----------------------|---|---------------|-------------------------------|
| Marginally Concerned | 1   | 12            | 8.3%                          |
| Amateur              | 4   | 21            | 19.1%                         |
| Lazy Expert          | 1   | 11            | 9.1%                          |
| Technician           | 4   | 14            | 36.4%                         |
| Fundamentalist       | 0   | 3             | 0%                            |
| All participants     | 10  | 61            | 16.4 %                        |

Only 10 out of 61 participants made no mistakes (see Table 30) and achieved 100 percent objective correctness in all paradigms. Of these ten participants, four participants chose the persona technicians (36.36% of all technicians) and four the persona amateur (19.1% of all amateurs). None of the three fundamentalists achieved 100 percent objective correctness. About one third of all participants made mistakes in three out of four paradigms. Four participants made mistakes in all paradigms. It is interesting to mention that the personas with high motivation performed better, especially the technicians.

Table 31: Participants per Personas Making Zero Mistakes per Paradigm

| Persona              |               | Default Policies | Security Levels | Template Instantiation | Wizard |
|----------------------|---------------|------------------|-----------------|------------------------|--------|
| Marginally Concerned | # no mistakes | 2                | 9               | 1                      | 1      |
|                      | % no mistakes | 16.7%            | 75%             | 8.3%                   | 8.3%   |
| Amateur              | # no mistakes | 14               | 20              | 7                      | 7      |
|                      | % no mistakes | 66.7%            | 95.2%           | 33.3%                  | 33.3%  |
| Lazy Expert          | # no mistakes | 7                | 11              | 2                      | 1      |
|                      | % no mistakes | 63.6%            | 100%            | 18.2%                  | 9.1%   |
| Technician           | # no mistakes | 8                | 14              | 6                      | 6      |
|                      | % no mistakes | 57.1%            | 100%            | 42.9%                  | 42.9%  |
| Fundamentalist       | # no mistakes | 3                | 3               | 1                      | 0      |
|                      | % no mistakes | 100%             | 100%            | 33.3%                  | 0%     |
| All participants     | # no mistakes | 36               | 57              | 17                     | 15     |
|                      | % no mistakes | 59.0%            | 93.4%           | 27.9%                  | 24.6%  |

Table 31 shows the number of participants per persona that made zero mistakes in the specific paradigms. The »security levels« paradigm has an acceptable success rate between 75 and 100 percent for all personas (93.4 percent on average). The »default policy« paradigm was correctly used by

the fundamentalists, but only 16.9 percent of the marginally concerned made zero mistakes (59.0 percent on average). For the paradigms »template instantiation« and »wizard«, the highest success rate was 42.9 percent for the technicians. On average, only 27.9 percent of the participants achieved perfect objective correctness with the specification paradigm »template instantiation« and only 24.6 percent with the »wizard«. It can be mentioned that fewer expressiveness of the specification paradigms leads to a higher objective correctness.

Table 32: Ratio of Mistakes Made by Personas per Paradigm to All Decisions

| Mistakes made in relation to necessary decisions (expressiveness) |                    | Default Policies | Security Levels | Template Instantiation | Wizard |
|---|--------------------|------------------|-----------------|------------------------|--------|
| Necessary Decisions   |                    | 6                | 1               | 18                     | 18     |
| Marginally Concerned  | Normalized Average | 0.56             | 0.25            | 0.49                   | 0.50   |
|   | Std. Deviation     | 0.36             | 0.45            | 0.29                   | 0.29   |
| Amateur   | Normalized Average | 0.12             | 0.05            | 0.12                   | 0.12   |
|   | Std. Deviation     | 0.22             | 0.22            | 0.16                   | 0.14   |
| Lazy Expert   | Normalized Average | 0.15             | 0.00            | 0.16                   | 0.21   |
|   | Std. Deviation     | 0.26             | 0.00            | 0.16                   | 0.21   |
| Technician  | Normalized Average | 0.17             | 0.00            | 0.15                   | 0.11   |
|   | Std. Deviation     | 0.27             | 0.00            | 0.25                   | 0.16   |
| Fundamentalist  | Normalized Average | 0.00             | 0.00            | 0.06                   | 0.13   |
|   | Std. Deviation     | 0.00             | 0.00            | 0.06                   | 0.08   |
| All participants  | Normalized Average | 0.22             | 0.07            | 0.20                   | 0.21   |
|   | Std. Deviation     | 0.31             | 0.25            | 0.25                   | 0.24   |

Figure 68 and Table 32 present the normalized ratio of the number of mistakes made by personas per paradigm to all decisions (i.e. deviations from the sample solution). Since the paradigms require a different number of decisions (mistake potential), we show the ratio of wrong decisions to all decisions per specification paradigm. In addition, we show standard deviation values of the absolute number of mistakes in Table 32.

There are only a few outliers (see Figure 68), for instance the participants with the IDs 19 and 45. Both were outlier in more than one paradigm. The outliers could be caused by a poor choice of the persona. However, no participant reported not to identify with the chosen persona.

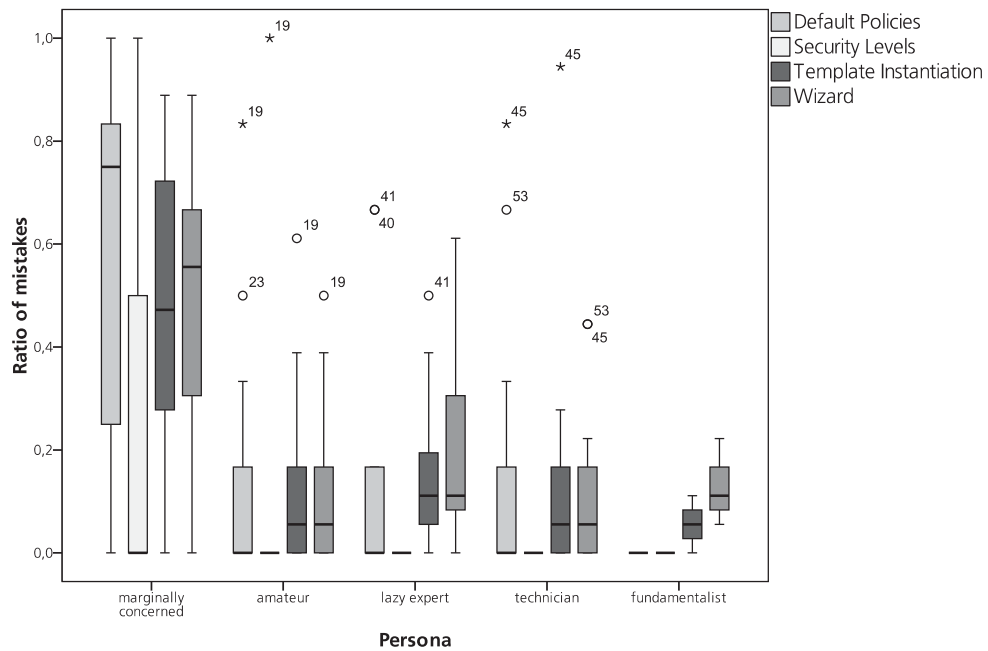


Figure 68: Ratio of Mistakes Made by Personas per Paradigm to All Decisions

In Hypothesis H1.1, we assume that the best matching specification paradigm leads to 30 percent fewer mistakes than the worst matching specification paradigm (see Section 1.5.1).

Regarding Q1.1.1, the entire group of participants made the most mistakes with the specification paradigm »default policies« (22% of all decisions) and the fewest mistakes with the »security levels« (7%). If we compare the results, we find that on average, all participants made 68 percent fewer mistakes with the paradigm »security levels« than with »default policies«. We found a significant influence with a large effect of the used specification paradigm on the made mistakes (Kruskal-Wallis test with Cohen's  $d$ :  $\alpha=0.05$ ,  $H=48.94$ ,  $p<0.01$ ,  $d_c=0.97$ ). This means that at least two paradigms significantly differ in terms of mistakes made. When we compare the paradigms pairwise, we find that users perform significantly better with the best paradigm »security levels« compared to »default policies« ( $z=4.24$ ,  $p<0.01$ ), »template instantiation« ( $z=5.93$ ,  $p<0.01$ ) and »wizard« ( $z=-6.18$ ,  $p<0.01$ ).

Regarding Q1.1.2, we see that all personas reduced their number of mistakes by more than 30 percent when comparing the best to the worst matching specification paradigm (marginally concerned by 55%, amateurs by 58%, the others by 100%). However, the influence of the selected specification paradigm on the objective correctness is not for all personas significant (Kruskal-Wallis test with Cohen's  $d$ :  $\alpha=0.05$ ). It is significant for the »amateurs« ( $H=16.15$ ,  $p<0.01$ ,  $d_c=0.89$ ), for the »lazy experts« ( $H=16.63$ ,  $p<0.01$ ,  $d_c=1.44$ ) and for the »technicians« ( $H=11.15$ ,  $p=0.01$ ,  $d_c=0.86$ ), but not for the »marginally concerned« ( $H=4.98$ ,

$p=0.17$ ,  $d_c=0.43$ ). Due to the small sample size, the test could not provide meaningful significant results for the »fundamentalists«.

Regarding Q1.1.3, we found a significant influence with a large effect of the persona selection on the mistakes made (Kruskal-Wallis test with Cohen's  $d$ :  $\alpha=0.05$ ,  $H=35.23$ ,  $p<0.01$ ,  $d_c=0.81$ ). We explain this effect of the persona with the significant difference regarding objective correctness of the marginally compared to the other personas, as they perform significantly worse. We see the influence of the persona selection in each paradigm: »default policies« ( $H=13.88$ ,  $p<0.01$ ), »template instantiation« ( $H=14.10$ ,  $p<0.01$ ), and »wizard« ( $H=17.04$ ,  $p<0.01$ ), and also for the »security levels« ( $H=7.99$ ,  $p<0.05$ ), but not that strong.

Thus, we can reject the Null Hypothesis  $H1.1_0$ . We present detailed diagrams of the statistical tests in Appendix G.5.

### **Perceived Correctness**

We measured the perceived correctness of the specification tasks per specification paradigm. Therefore, we asked the participants after each specification paradigm they used whether they think that they solved all tasks correctly. As shown in Table 33, the overall perceived correctness is very high. All participants were most skeptical about the »security levels« (78.7%) and most confident about the »wizard« (91.8%).

Table 33: Perceived Correctness per Specification Paradigm

|                      | Default Policies | Security Levels | Template Instantiation | Wizard |
|----------------------|------------------|-----------------|------------------------|--------|
| Marginally Concerned | 91.7%            | 83.3%           | 83.3%                  | 100.0% |
| Amateur              | 85.7%            | 76.2%           | 90.5%                  | 90.5%  |
| Lazy Expert          | 81.8%            | 72.7%           | 90.9%                  | 81.8%  |
| Technician           | 92.9%            | 85.7%           | 85.7%                  | 92.9%  |
| Fundamentalist       | 100.0%           | 66.7%           | 100.0%                 | 100.0% |
| All Participants     | 88.5%            | 78.7%           | 88.5%                  | 91.8%  |

The results of our experiment reveal that the persona selection does not significantly influence the perceived correctness in any paradigm (template instantiation:  $p=0.96$ ; default policies:  $p=0.87$ ; security levels:  $p=0.85$ ; wizard:  $p=0.62$ ). This means that there is no difference in how optimistic or pessimistic the participants of the different personas are regarding the specification paradigms.



### ***Self-evaluation regarding Objective Correctness***

In our experiment, we aimed at identifying which paradigm suits best for a correct self-evaluation (perceived correctness) regarding the objective correctness. Participants achieved a correct positive self-evaluation if they made zero mistakes with a specification paradigm and were confident about the perfect solution. Participants achieved a correct negative self-evaluation if they made at least one mistake and were confident that they made mistakes. Table 34 shows the positive self-evaluations (P) and the negative self-evaluations (N) as well as the ratio of correct self-evaluations to the number of participants per persona.

Overall, the self-evaluation was best with the »security levels« (78.7%) and worst with the »wizard« (29.5%). We ascertained that more decisions during specification led to worse self-evaluation. Overall, 42 participants thought that they used all paradigms correctly, however, only eight of them actually made no mistakes in all paradigms. Only four persons had a too pessimistic self-evaluation; that is, they achieved perfect results, but thought they made mistakes.

The marginally concerned achieved the worst self-evaluation, which can be explained with the significantly worse objective correctness they have. Only 25 percent of the participants with this persona correctly self-evaluated themselves with the specification paradigms »default policies« and »template instantiation«, and only one participant correctly estimated their mistakes using the »wizard« (8.3%). The technicians performed best with the specification paradigms »template instantiation« (57.1%) and »wizard« (50.0%).

Table 34: Accuracy of Perceived Correctness (Correct Positive (P) and Negative (N) Self-Evaluations)

|                      | Default Policies |      | Security Levels |      | Template Instantiation |      | Wizard |      |
|----------------------|------------------|------|-----------------|------|------------------------|------|--------|------|
|                      | P/N              | %    | P/N             | %    | P/N                    | %    | P/N    | %    |
| Marginally Concerned | 2/1              | 25.0 | 8/1             | 75.0 | 1/2                    | 25.0 | 1/0    | 8.3  |
| Amateur              | 12/1             | 61.9 | 16/1            | 81.0 | 6/1                    | 33.3 | 6/1    | 33.3 |
| Lazy Expert          | 7/2              | 81.8 | 8/0             | 72.7 | 2/1                    | 27.3 | 1/2    | 27.3 |
| Technician           | 8/1              | 64.3 | 12/0            | 85.7 | 6/2                    | 57.1 | 6/1    | 50.0 |
| Fundamentalist       | 3/0              | 100  | 2/0             | 66.7 | 1/0                    | 33.3 | 0/0    | 0.0  |
| All Participants     | 32/5             | 60.7 | 46/2            | 78.7 | 16/6                   | 36.1 | 14/4   | 29.5 |

Regarding Hypothesis H1.2, we assume that the best matching specification paradigm leads to 30 percent higher accuracy regarding the self-evaluation of objective correctness than the worst matching specification paradigm (see Section 1.5.1).

Regarding Q1.2.1, the entire participant group achieved the best self-evaluation with the »security levels« paradigm (78.7%) and the worst with the »wizard« paradigm (29.5%). If we compare the results, we find that on average, the accuracy of self-estimation for all participants is 167 percent higher with the »security levels« than with the »wizard« paradigm. We found a significant influence with a large effect of the used specification paradigm on the correct self-evaluation (Fisher's exact test and Cramer's  $\phi$ :  $\alpha=0.05$ ,  $T=38.69$ ,  $p<0.01$ ,  $\phi_c=0.39$ ).

Regarding Q1.2.2, we see that all personas increased their number of mistakes by more than 30 percent when comparing the best to the worst matching specification paradigm (marginally concerned by 804%, amateurs by 143%, lazy experts by 200%, technicians by 71%; the percentage increase for fundamentalists is infinite). However, the influence of the selected specification paradigm on the objective correctness is not for all personas significant (Fisher's exact test and Cramer's  $\phi$ :  $\alpha=0.05$ ). It is significant for the »marginally concerned« ( $T=12.49$ ,  $p=0.01$ ,  $\phi_c=0.53$ ), the »amateurs« ( $T=13.78$ ,  $p<0.01$ ,  $\phi_c=0.41$ ), for the »lazy experts« ( $T=10.86$ ,  $p=0.01$ ,  $\phi_c=0.51$ ), but not for the »technicians« ( $T=4.44$ ,  $p=0.26$ ,  $\phi_c=0.28$ ) or the »fundamentalists« ( $T=6.00$ ,  $p=0.24$ ,  $\phi_c=0.75$ ).

Regarding Q1.2.3, we found that the selection of the persona also has an influence on the correct self-evaluation ( $T=10.08$ ,  $p=0.04$ ,  $\phi_c=0.20$ ), but not a very strong one and with only a medium effect size.

Thus, we can reject Null Hypothesis H1.2<sub>0</sub>. We present detailed diagrams of the statistical tests in Appendix G.5.

## **Efficiency**

We measured the time it took each participant to complete the specification of all six tasks with each of the four specification paradigms. We excluded the data sets of two participants from the analysis, as each had an extreme outlier in one paradigm. This can only be explained by a longer pause during the experiment. The other time data are reasonable regarding the minimum time to fulfill a task properly. Thus, the total number of participants for the analysis of efficiency is 59.

In regard to all participants, the paradigm »security levels« proved to be the most efficient ( $M=1.8$  minutes) method for specifying privacy settings (see Table 35). There are smaller differences in the average time of the other paradigms, ranging between 3.1 and 3.8 minutes. The second most efficient paradigm is the »template instantiation«; the participants needed the longest time for the »wizard«.

Table 35: Mean Time in Minutes of Specification with Different Specification Paradigms

| Mean times in minutes | Default | Security Levels | Template Instantiation | Wizard | All paradigms |
|-----------------------|---------|-----------------|------------------------|--------|---------------|
| Marginally Concerned  | 4.3     | 2.6             | 3.4                    | 4.0    | 14.3          |
| Amateur               | 3.4     | 1.6             | 3.0                    | 3.8    | 11.8          |
| Lazy Expert           | 2.7     | 1.1             | 2.7                    | 3.7    | 10.3          |
| Technician            | 3.5     | 1.8             | 3.5                    | 3.5    | 12.3          |
| Fundamentalist        | 3.5     | 1.4             | 3.5                    | 4.5    | 12.9          |
| All Participants      | 3.5     | 1.8             | 3.1                    | 3.8    | 12.2          |

Table 35 and Figure 69 show the time needed to complete all six tasks with a specification paradigm per persona. The lazy experts required less time to solve all tasks in all four paradigms than the other personas (on average 2 minutes less than the remaining participants). On average, the marginally concerned needed about 2.5 minutes longer than the remaining population. The other three personas needed between 11.8 and 12.9 minutes on average for all paradigms. Lazy experts, fundamentalists and amateurs needed longest for the »wizard« and performed fastest with the »security levels«. For technicians and fundamentalists, the time needed for the »template instantiation« and for the »default policies« is almost equal. The technicians are also clearly fastest when using the paradigm »security levels«; however, they took an equal amount of time in all other paradigms.

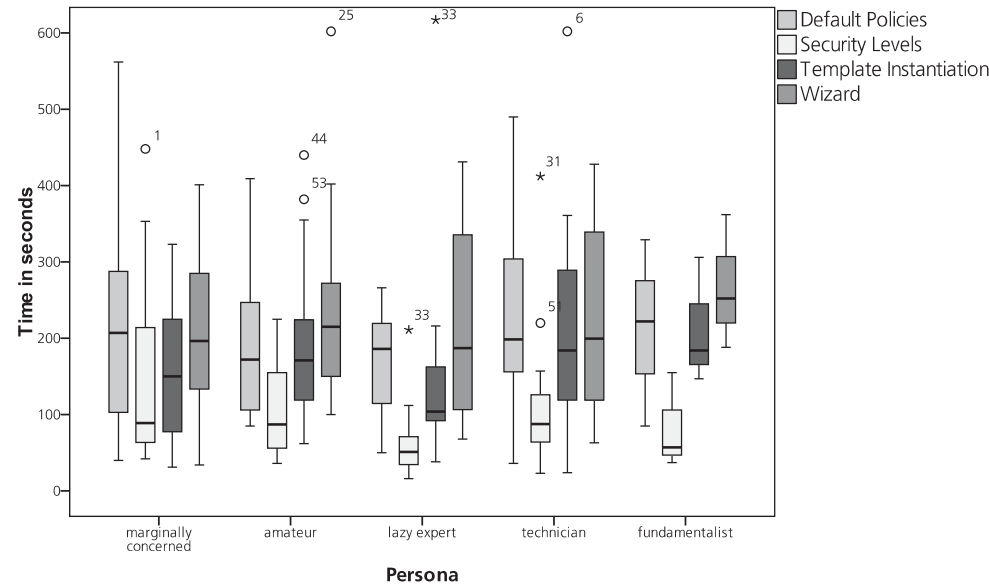


Figure 69: Time Needed in Seconds to Complete all Six Tasks with a Specification Paradigm per Persona

According to Hypothesis H1.3, we assume that the best matching specification paradigm lets users to specify policies 30 percent faster than the worst matching specification paradigm (see Section 1.5.1).

Regarding Q1.3.1, the entire participant group specified fastest with the specification paradigm »security levels« (1.8 minutes on average) and slowest with the »wizard« (3.8 minutes on average). If we compare the results, we find that all participants increased efficiency by 53 percent with the »security levels« compared to the »wizard«. We found a significant influence with a large effect of the used specification paradigm on the time needed for policy specification, that is, the users' efficiency (Kruskal-Wallis test with Cohen's  $d$ :  $\alpha=0.05$ ,  $H=46.89$ ,  $p<0.01$ ,  $d_c=0.95$ ). This means that at least two paradigms significantly differ with respect to the time needed. When we compare the paradigms pairwise, we find that users perform significantly faster with the best paradigm »security levels« compared to »default policies« ( $z=5.11$ ,  $p<0.01$ ), »template instantiation« ( $z=4.23$ ,  $p<0.01$ ) and »wizard« ( $z=-6.17$ ,  $p<0.01$ ).

Regarding Q1.3.2, we found that for all personas, the selection of the best matching specification paradigm lead to a decrease of the time needed by more than 30 percent compared to the most inefficient choice (marginally concerned by 40%, amateurs by 58%, lazy experts by 70%, technicians by 49% and fundamentalists by 69%). However, the influence of the selected specification paradigm on the efficiency is not for all personas significant (Kruskal-Wallis test with Cohen's  $d$ :  $\alpha=0.05$ ). It is significant for the »amateurs« ( $H=23.64$ ,  $p<0.01$ ,  $d_c=1.19$ ), for the »lazy experts« ( $H=13.09$ ,  $p<0.01$ ,  $d_c=1.16$ ) and for the »technicians« ( $H=9.85$ ,  $p=0.02$ ,  $d_c=0.79$ ), but not for the »marginally concerned« ( $H=2.57$ ,  $p=0.46$ ,  $d_c=0.20$ ). Due to the small sample size, the test could not be meaningfully applied to the »fundamentalists«.

Regarding Q1.3.3, we did not find a significant effect of the persona selection on the time needed with the Kruskal-Wallis test ( $\alpha=0.05$ ,  $H=3.90$ ,  $p=0.27$ ,  $d_c=0.13$ ). Thus, the distribution of time needed is similar across all personas.

In summary, we can reject Null Hypothesis H1.3<sub>0</sub>. We present detailed diagrams of the statistical tests in Appendix G.5.

## ***Satisfaction***

We asked the participants directly after they had used a specification paradigm to indicate how much they like it. They used a five-point scale ranging from 1 (»I really dislike this specification paradigm« to 5 (»I really like this specification paradigm«). After all four specification rounds had been completed, we asked the participants to rank the four specification paradigms according to their personal preference.

Overall, participants liked the »template instantiation« paradigm most (see Figure 70 and Table 36). The participant rated the »wizard« slightly worse. The »default policies« were placed third. The »security level

paradigm« was considered least satisfying. The participants also ranked the paradigms according to their preference. In the ranking, the »security level« paradigm was most often ranked last, regardless of the chosen persona.

Table 36: Satisfaction with Specification Paradigms for Personas (SD: Standard Deviation)

|                             | Mean | SD  | Median | Rank 1 | Rank 2 | Rank 3 | Rank 4 |
|-----------------------------|------|-----|--------|--------|--------|--------|--------|
| <b>Marginally Concerned</b> |      |     |        |        |        |        |        |
| Template Inst.              | 3.9  | 0.9 | 4      | 17%    | 58%    | 25%    | 0%     |
| Default Policies            | 3.3  | 1.4 | 3.5    | 17%    | 8%     | 42%    | 33%    |
| Security Levels             | 3    | 1.2 | 3      | 17%    | 25%    | 8%     | 50%    |
| Wizard                      | 4    | 1.2 | 4      | 50%    | 8%     | 25%    | 17%    |
| <b>Amateur</b>              |      |     |        |        |        |        |        |
| Template Inst.              | 3.8  | 0.9 | 4      | 43%    | 48%    | 10%    | 0%     |
| Default Policies            | 3.3  | 1.2 | 4      | 24%    | 10%    | 48%    | 19%    |
| Security Levels             | 2.1  | 1.2 | 2      | 0%     | 14%    | 19%    | 67%    |
| Wizard                      | 3.8  | 0.7 | 4      | 33%    | 29%    | 24%    | 14%    |
| <b>Lazy Expert</b>          |      |     |        |        |        |        |        |
| Template Inst.              | 4    | 1.1 | 4      | 45%    | 45%    | 0%     | 9%     |
| Default Policies            | 3    | 1.1 | 3      | 0%     | 9%     | 64%    | 27%    |
| Security Levels             | 1.9  | 0.8 | 2      | 9%     | 9%     | 27%    | 55%    |
| Wizard                      | 3.8  | 1.3 | 4      | 45%    | 36%    | 9%     | 9%     |
| <b>Technician</b>           |      |     |        |        |        |        |        |
| Template Inst.              | 4.1  | 1.1 | 4      | 43%    | 7%     | 29%    | 21%    |
| Default Policies            | 3.4  | 1.3 | 4      | 14%    | 21%    | 43%    | 21%    |
| Security Levels             | 3.2  | 1.5 | 3      | 29%    | 14%    | 7%     | 50%    |
| Wizard                      | 3.8  | 1.2 | 4      | 14%    | 57%    | 21%    | 7%     |
| <b>Fundamentalist</b>       |      |     |        |        |        |        |        |
| Template Inst.              | 4.3  | 1.2 | 5      | 67%    | 0%     | 33%    | 0%     |
| Default Policies            | 4.3  | 0.6 | 4      | 0%     | 33%    | 33%    | 33%    |
| Security Levels             | 3.3  | 2.1 | 4      | 0%     | 0%     | 33%    | 67%    |
| Wizard                      | 4.3  | 0.6 | 4      | 33%    | 67%    | 0%     | 0%     |
| <b>All Participants</b>     |      |     |        |        |        |        |        |
| Template Inst.              | 4    | 1   | 4      | 39%    | 38%    | 16%    | 7%     |
| Default Policies            | 3.3  | 1.2 | 4      | 15%    | 13%    | 48%    | 25%    |
| Security Levels             | 2.6  | 1.4 | 2      | 11%    | 15%    | 16%    | 57%    |
| Wizard                      | 3.9  | 1   | 4      | 34%    | 34%    | 20%    | 11%    |

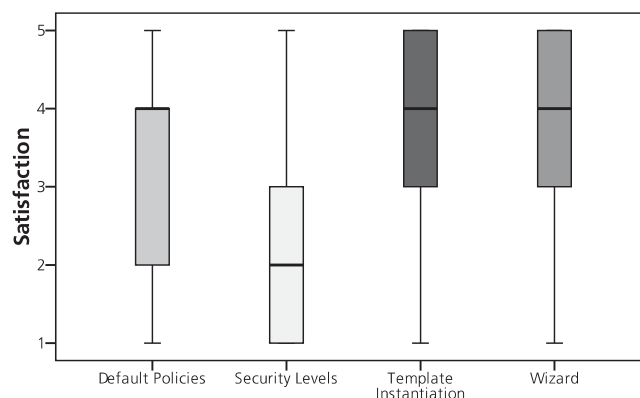


Figure 70: Participant's Satisfaction with Specification Paradigms

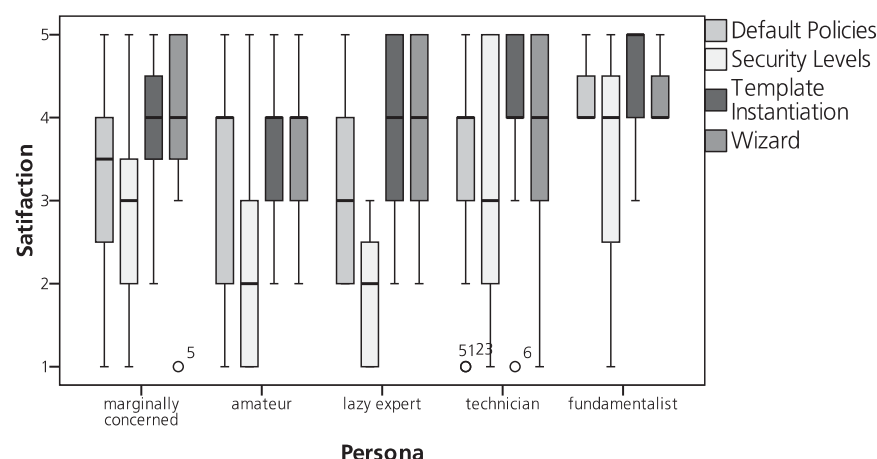


Figure 71: Participant's Satisfaction with Specification Paradigms per Persona

Table 36 and Figure 71 show the satisfaction results broken down by personas. They underline that the »security levels« paradigm is least satisfying within all persona groups and that the »default policies« paradigm is in third place for all personas except the fundamentalists. The marginally concerned preferred the paradigms »wizard« and »template instantiation« most. In the final ranking, 50 percent voted the »wizard« in first place and a majority voted the »template instantiation« in second place. Immediately after the individual specifications with each paradigm, the amateurs almost equally liked the paradigms »template instantiation« and »wizard«. However, in the final ranking, they clearly voted the paradigm »template instantiation« in first place, followed by »wizard«. 67 percent ranked »security levels« in last place. The lazy experts voted similar to the amateurs. Overall, the »template instantiation« paradigm was liked most, followed by the »wizard« paradigm. The technicians liked the »template instantiation« paradigm the most, followed by the »wizard« paradigm. However, the other two paradigms also received quite high ratings. This rather even distribution of satisfaction over the paradigms is also evident in the rankings, where each paradigm was ranked first by at least 14 percent of the technicians. The »template instantiation« paradigm was ranked top by 43 percent of the participants. The fundamentalists answered that the specification paradigms »template

instantiation«, »default policies« and »wizard« are equally satisfying. In the final ranking, two participants voted for »template instantiation« in the first place; the »wizard« was ranked second. However, due to the small sample size, the results for the fundamentalists are not very meaningful.

Regarding Hypothesis H1.4, we assume that the satisfaction during a policy specification for users when using the best matching specification paradigm is 30 percent better than with the worst matching specification paradigm (see Section 1.5.1).

Regarding Q1.4.1, the entire participant group liked the paradigm »template instantiation« most (rating of 4 out of 5 and 39% of participants ranked it in first place, 7% in the last place). The participants liked the paradigm »security levels« least (rating of 2.6 out of 5 and 11% of participants ranked it in first place, 57% in the last place). If we compare the mean and median values of the paradigms, we achieve a higher satisfaction with the paradigm »template instantiation« than with the »security levels« (mean: 1.4; median: 2). In addition, the rankings indicate a significantly better satisfaction with the »template instantiation« than with the »security levels«. We found a significant influence with a large effect of the used specification paradigm on the users' satisfaction based on the rating with the 5-point scale (Kruskal-Wallis test with Cohen's  $d$ :  $\alpha=0.05$ ,  $H=42.62$ ,  $p<0.01$ ,  $d_c=0.89$ ). This means that at least two paradigms significantly differ with respect to satisfaction. When we compare the paradigms pairwise, we find that users like the specification paradigm »template instantiation« significantly more than the »default policies« ( $z=2.83$ ,  $p=0.03$ ) and the »security levels« ( $z=5.84$ ,  $p<0.01$ ). In addition, participants like the specification paradigm »default policies« ( $z=3.02$ ,  $p=0.02$ ) and »wizard« ( $z=-5.33$ ,  $p<0.01$ ) significantly more than the »security levels«.

Regarding Q1.4.2, we show that for all personas, the selection of the best matching specification paradigm leads to an increase in satisfaction compared to the least satisfying choice (marginally concerned: mean by 1, median by 1; amateurs: mean by 1.7, median by 2; lazy experts: mean by 2.1, median by 2; technicians: mean by 0.9, median by 1; and fundamentalists: mean by 1, median by 1). However, the influence of the selected specification paradigm on satisfaction is not for all personas significant (Kruskal-Wallis test with Cohen's  $d$ ,  $\alpha=0.05$ ). It is significant for the »amateurs« ( $H=24.23$ ,  $p<0.01$ ,  $d_c=1.20$ ), for the »lazy experts« ( $H=16.50$ ,  $p<0.01$ ,  $d_c=1.43$ ), but not for the »marginally concerned« ( $H=6.41$ ,  $p=0.09$ ,  $d_c=0.58$ ) or the »technicians« ( $H=4.06$ ,  $p=0.26$ ,  $d_c=0.29$ ). Due to the small sample size, the test did not yield meaningful results for the »fundamentalists«.



Regarding Q1.4.3, we did not find a significant influence of the persona selection on the satisfaction (Kruskal-Wallis test with Cohen's  $d$ ,  $\alpha=0.05$ ,  $H=5.87$ ,  $p=0.12$ ,  $d_c=0.23$ ).

In summary, we can reject Null Hypothesis H1.4<sub>0</sub>. We present detailed diagrams of the statistical tests in Appendix G.5.

### ***Comparison of Effectiveness, Efficiency and Satisfaction***

In the previous sections, we analyzed the results with respect to effectiveness, efficiency and satisfaction separately. We show that the selection of the appropriate specification paradigm has an effect on the qualities effectiveness, efficiency and satisfaction. However, we need to analyze the relation between the three qualities based on the selection of a specification paradigm.

In our results, effectiveness and efficiency of specification paradigms are aligned. Satisfaction behaves contrary. People do not like the »security levels« but perform most efficiently and most effectively with this paradigm. Vice versa, people like the specification paradigms »wizard« and »template instantiation«, but they are less effective and less efficient using them.

### ***Summary of Results regarding Personas***

Using the »security levels« paradigm, the **marginally concerned** made the fewest mistakes and achieved the best self-evaluation compared to other paradigms (Average Mistakes (AM): 25%, see Table 32; Correct Self-Evaluation (CSE): 75%, see Table 34)). In all other paradigms, this group of people made more mistakes. This persona performed fastest with the »security levels« (2.6 minutes on average, see Table 35) and slowest with the »default policies« (4.3 minutes). The marginally concerned liked the »wizard« most (Average Rating (AR): 4 out of 5; 50% ranked in first place (FP), see Table 36) and the »security levels« least (AR: 3, 17% FP).

The **amateurs** also achieved the best results with the »security levels« (AM: 5%; CSE: 81%). For the other paradigms, the AM values are equal at 12 percent. Regarding the self-evaluation, participants assessed themselves rather well with the »default policies« (CSE 61.9%). Amateurs did rather few mistakes with the paradigms »template instantiation« and »wizard«, but the self-assessment is worse than with other paradigms. This persona performed fastest with the »security levels« (1.6 minutes) and slowest with the »wizard« (3.8 minutes). The amateurs liked the »template instantiation« (AR: 3.8, 43% FP) most, directly followed by the »wizard« (AR: 3.8, 33% FP). The »security levels« are least satisfying (AR: 2.1, 0% FP).



The **lazy experts** are described by Dupree as people with a high level of knowledge and low motivation in terms of security and privacy. It is interesting to note that they performed worse than amateurs and technicians in many direct value comparisons. The values for the »default policies« (AM: 15%; CSE: 81.8%) and »security levels« (AM: 0%; CSE: 72.7%) are best. They performed fastest with the »security levels« (1.1 minutes) and slowest with the »wizard« (3.7 minutes). The lazy experts liked the »template instantiation« (AR: 4, 45% FP) most, directly followed by the »wizard« (AR: 3.8, 33% FP). The »security levels« are least satisfying (AR: 1.9, 9% FP).

Like all other personas, **technicians** achieved best results in the paradigms »security levels« (AM: 0%; CSE: 86%) and »default policies« (AM: 17%; CSE: 64%). However, the technicians achieved best values regarding the self-evaluation and rather low numbers of mistakes for the paradigms »template instantiation« (AM: 15%; CSE: 57%) and »wizard« (AM: 11%; CSE: 50%). The technicians performed fastest with the »security levels« (1.8 minutes) and equally fast with all other specification paradigms (3.5 minutes). They liked the »template instantiation« (AR: 4.1, 43% FP) most and the »security levels« least (AR: 3.2, 29% FP). However, surprisingly many technicians voted the »security levels« in first place. In contrast, half of the technicians voted them in the last place.

Since only three participants have chosen the persona **fundamentalist**, no conclusions can be made about this persona. Still, our preliminary results confirm the persona scheme of Dupree [14]. On average, the fundamentalists made the fewest mistakes (AM values between 0% and 12%). They achieved perfect scores for the paradigms »default policies« and »security levels«, except one participant being too pessimistic about his success in the »security levels« paradigm. Also, the fundamentalists made very few mistakes in the paradigms »template instantiation« (AM: 6%) and »wizard« (AM: 13%). However, they overestimated the objective correctness in those two paradigms (CSE: 33% and 0%). Fundamentalists performed fastest with the »security levels« (1.4 minutes) and slowest with the »wizard« (4.5 minutes). They liked the »template instantiation« most (AR: 4.3; 67% FP). However, the paradigms »default policies« and »wizard« received the same average rating. The »security levels« were least satisfying (AR: 3.3, 0% FP).

The experiment results partially confirm the mapping of personas to specification paradigms for increasing usability that we assumed in Section 6.3.2. The marginally concerned performed best (objective correctness, self-evaluation regarding objective correctness and efficiency) with the »security levels«. However, they do not like this paradigm. Also all the other personas achieved best objective correctness with the »security levels«. This seems reasonable, as this paradigm requires less user resources than the others do. The amateurs performed best with respect

to the correct self-estimation with the »default policies«. However, they had similar results regarding objective correctness and efficiency with the three specification paradigms »default policies«, »template instantiation« and »wizard«. The lazy experts do not have a specification paradigm directly mapped to them. It is interesting that they performed worse than amateurs and technicians in many direct value comparisons. This indicates that the motivation has a significant influence on the results. The technicians also reached best results with »security levels« and »default policies«. However, they achieved the best results of all personas with respect to objective correctness and correctly perceived correctness with the specification paradigms »template instantiation« and »wizard«. Due to the small number of fundamentalists, we cannot draw conclusions about our assumption for the best mapping to increase usability.

In summary, the experiment shows that a mapping of users to specification paradigms can increase the objective and perceived correctness, which users can achieve when specifying policies with a PAP. In most cases, users perform better with specification paradigms that require fewer resources. However, if more user resources are required, the personas with the appropriate level of user resources performed better than these personas not having this level. We did not identify effects of the mapping on efficiency and satisfaction. We need to investigate the user to specification paradigm mapping for increasing usability further in future work.

### 9.4.3 Discussion

In the experiment, we investigated how the selected persona and the specification paradigm affect objective correctness (RQ1.1), perceived correctness (RQ1.2), efficiency (RQ1.3) and satisfaction (RQ1.4) with our research questions.

With respect to RQ1.1, we observed that all personas made the fewest mistakes with the specification paradigm »security levels«. The number of mistakes differed only marginally between the other paradigms. However, the persona marginally concerned differs significantly from the others with respect to objective correctness, as participants belonging to this group made more mistakes. The cumulated mistakes are higher than expected by the author. This raises the question about the difficulty of the tasks to be solved. Ten of 61 participants achieved the perfect objective correctness (zero mistakes in total). Thus, it was possible to solve all tasks without making mistakes. None of the participants complained that he did not understand the tasks or the scenario in free text comments at the end of the experiment. Thus, we see a relevant differentiation of the marginally concerned in relation to the other personas with respect to objective correctness.

Regarding RQ1.2, we found that the perceived correctness is related to the number of decisions of a paradigm. In our experiment, more freedom led to worse perceived correctness. However, there is no significant difference in how personas perform regarding perceived correctness in these paradigms. Regarding the self-evaluation, we ascertained that marginally concerned performed worse than the other personas. In summary, we did not expect that only few participants (8 out of 61) would correctly perceive their objective correctness. Most participants overestimated themselves; only four underestimated their correctness. In practice, overestimation could frustrate a PAP user, as the system is not reacting as expected. This could reduce trust in the PAP and its provider. Those participants who underestimated their achieved correctness may appreciate the correct specification and the effect by the system, but they may also be frustrated, because they have the feeling of not having control over the system. We see a relevant differentiation of the marginally concerned in relation to the other personas with respect to self-evaluation of objective correctness.

With respect to RQ1.3, we found that the different personas differ in how fast they were able to specify policies. Lazy experts were faster than all other personas and marginally concerned were slower than all other personas. In summary, we do not see a relevant difference in the efficiency of the different personas.

Regarding RQ1.4, we ascertained that the persona has no significant influence on the satisfaction with different specification paradigms. All participants voted similarly. This is interesting, as we had assumed that less skilled or less motivated participants would prefer less expressive specification paradigms. However, our experiment revealed the opposite result. In their free text comments, participants pointed out that they like to have options for specification. Thus, we do not see a relevant difference in satisfaction of the different personas with the specification paradigms.

Our experiment relies on the personas developed by Dupree (see Appendix C). We decided to select these personas since they were developed based on empirical data, and the personas mainly differ in the user's motivation and security knowledge, which matches the barriers identified with the user intention model (see Section 6.2.1). In addition, Dupree's personas explain other character traits that reveal more valuable information, such as the preference for convenience over security. Moreover, they contain concrete security behaviors, such as the use of strong passwords. We assume that such concrete information eases the self-classification compared to a scale with short statements, which are prone to a subjective interpretation (i.e., expert knowledge might be interpreted differently).

Our two questions in the experiment about security knowledge and motivation were intended to control whether the persona selection is

reasonable. However, we do not consider these to questions as sufficient to replace the personas. In practice, it would be preferable to have a small selection questionnaire for the user to persona mapping. To the best of our knowledge, though, such questionnaires do not exist.

In the study by Dupree [14], the number of fundamentalists was the smallest by far. We experienced the same in our experiment. More fundamentalists are needed to draw conclusions about an appropriate specification paradigm. The other personas were represented by 11, 12, 14, and 21 participants, respectively. These numbers still seem small, but we chose statistical tests for small sample sizes to produce meaningful results. Nevertheless, the experiment needs to be repeated with more participants in other scenarios to improve the generalizability of our results.

In our experiment, many participants were academics or related to an academic work environment (69% academics, 54% employees of the author's institution, 93% German-speaking participants). Obviously, the group of participants does not reflect the overall population (e.g., there are only 15% academics in Germany). We cannot rule out that this had an influence on the results and a negative impact on their generalizability. It seems unlikely to us that the level of education has a direct impact, but indirect effects seem reasonable. The level of education is related to certain jobs and interests, which also affects knowledge about IT security. In future experiments, questions that are more precise have to be asked to assess the relation of education to effectiveness more deeply. Questions could be, for example: »Is your job related to IT security or privacy?« or »Do you spend time in your spare time to learn more about privacy?«

We showed that the selection of the specification paradigm has contrary effects on effectiveness and efficiency than on satisfaction. Participants performed ineffectively and made many mistakes with specification paradigms they like. In contrast, the participants do not like the specification paradigm with which they performed most effectively and efficiently. This poses a dilemma for a PAP vendor that needs to select the appropriate specification paradigm for the privacy specification interfaces of his product. High effectiveness and efficiency are usually desired by the users. However, low satisfaction with a PAP can prevent users from specifying policies at all. On the other hand, a satisfactory PAP that specifies incorrect policies can undermine trust in the vendor.

Apart from efficiency, effectiveness and user satisfaction, other requirements may also need to be met, such as legal obligations or the need for the vendor to collect data based on his business model. Therefore, we cannot make general recommendations for the selection of specification paradigms based on our current results. Vendors must carefully weigh the pros and cons before selecting a specification

paradigm based on the personas that best represent their user community.

#### **9.4.4 Threats to validity**

In this section, we address threats to validity with respect to the policy specification experiment. The threat categories are explained in Section 8.2.5.

##### ***Internal Validity***

We did not control the participants during or after the experiment, which is a threat to internal validity.

We adequately instructed participants with a text handout, a scenario video and instructions in various steps during the experiment, as we would have done in a controlled setting. We did not find any hint for an inadequate introduction (e.g., in the feedback at the end of the experiment). Thus, we assess this threat as low.

We cannot exclude the possibility that the participants talked about the experiment with other participants before their participation, nor that the participants could not find the necessary information or concentration to solve the tasks adequately. Distraction might increase the number of mistakes. We excluded two participants when analyzing the efficiency due to obvious large breaks.

A participant who could not identify with the scenario or the provided privacy demands in the tasks well may have lower motivation to take effort in correctly using the specification paradigms in the experiment. This may negatively affect the objective correctness and is a threat to internal validity.

The participants used a PAP (tool) for the specification of privacy policies, which uses implementations of the specification paradigms (concepts). This mixes findings at concept and tool level. To minimize this threat to internal validity, usability experts supported us to make the policy specification interfaces according to the four specification paradigms in the PAP as unobtrusive as possible.

##### ***External Validity***

The experiment tried to simulate the use of privacy demands in real life. In reality, participants would have their own individual demands. However, we had to preset the privacy demands in the form of six tasks in order to measure the correctness as the discrepancy between the participants' results and the sample solution. Thus, we cannot be sure

whether the same correctness values would be achieved in the real world with personal privacy demands. This poses a threat to external validity.

The paradigm »security levels« in combination with the given tasks does most likely not reflect the reality, since the preset tasks matched perfectly to one of the levels. This is rarely the case in real life and therefore jeopardizes external validity. However, we decided to propose a perfect solution, because the lack of a perfect match may have influenced the measured correctness and irritated the participants, which would have been a threat to internal validity.

Furthermore, the experiment was conducted in a scenario that represents a single use case for privacy demands (mono-operation bias). Further experiments that confirm our results in different scenarios with different participants would improve the generality of the results and therefore the external validity.

In addition, a large number of participants were academics or related to Fraunhofer IESE. This does not truly reflect the overall population. We assess this as a threat to external validity. Moreover, we cannot guarantee that the participants sufficiently reflect the population regarding security knowledge and motivation.

The number of participants, especially per persona, is quite small. This limits the generalizability. Further studies with more participants could mitigate this threat to external validity.

### **Conclusion Validity**

The selection of the specification paradigms is based on our observations of the paradigms most commonly used in practice. We cannot rule out the possibility that there are other paradigms leading to better results in a comparable experiment. This implies a threat to conclusion validity with respect to our recommendations of most suitable specification paradigms.

The small number of participants per persona is also a threat to conclusion validity, as any recommendation for personas has a low statistical power.

## **9.4.5 Summary and Conclusion**

Regarding our hypotheses, we showed the following:

- **H1.1 – Objective effectiveness of PAP:** We approved that the best matching specification paradigm leads to 30 percent fewer mistakes than the worst matching specification paradigm. The whole of the participants made most mistakes with the specification paradigm »default policies« (23% of all decisions) and fewest mistakes with the



»security levels« (7%). If we compare the results, we find that all participants made on average 68 percent fewer mistakes with the paradigm »security levels« than with »default policies«. We showed that the selection of the paradigm has a significant influence with a large effect on the conducted mistakes ( $\alpha=0.05$ ,  $H=48.94$ ,  $p<0.01$ ,  $d_c=0.97$ ) as well as the selection of the persona ( $\alpha=0.05$ ,  $H=35.23$ ,  $p<0.01$ ,  $d_c=0.81$ ). We confirm that the correct mapping of users to specification paradigms can increase objective correctness. Thus, we can reject the null hypothesis.

- **H1.2 – Perceived effectiveness of PAP:** We approved that the best matching specification paradigm leads to 30 percent higher accuracy regarding the self-evaluation of objective correctness than the worst matching specification paradigm. The whole of the participants performed best with respect to the self-evaluation with the paradigm »security levels« (78.7%) and worst with the »wizard« (29.5%). If we compare the results, we find that the accuracy of self-estimation for all participants is on average 167 percent higher with the paradigm »security levels« than with the »wizard«. We showed that the selection of the paradigm has a significant influence with a large effect on the correct self-evaluation with respect to conducted mistakes ( $\alpha=0.05$ ,  $T=38.7$ ,  $p<0.01$ ,  $\phi_c=0.39$ ). In addition, the selection of the persona ( $\alpha=0.05$ ,  $T=10.08$ ,  $p=0.04$ ,  $\phi_c=0.20$ ) has a significant influence with a medium effect. We confirm that the correct mapping of users to specification paradigms can increase the correctness of perceived correctness. Thus, we can reject the null hypothesis.
- **H1.3 – Efficiency of PAP:** We approved that the users are 30 percent faster (efficiency) with the best matching specification paradigm than with the worst matching specification paradigm. The whole of the participants specified fastest with the specification paradigm »security levels« (1.8 minutes on average) and slowest with the »wizard« (3.8 minutes on average). If we compare the results, we find that all participants increased efficiency by 111 percent with the »security levels« than with the »wizard«. We showed that the selection of the paradigm has a significant influence with a large effect on the time needed for policy specification with respect to conducted mistakes ( $\alpha=0.05$ ,  $H=46.89$ ,  $p<0.01$ ,  $d_c=0.95$ ), but not for the selection of the persona ( $\alpha=0.05$ ,  $H=3.90$ ,  $p=0.27$ ,  $d_c=0.13$ ). Nevertheless, we can reject the null hypothesis.
- **H1.4 – Satisfaction with PAP:** We approved that the satisfaction during a policy specification for users when using the best matching specification paradigm is significantly better than with the worst matching specification paradigm. The whole of the participants liked the paradigm »template instantiation« most (rating of 4 out of 5) and 39 percent of participants ranked it on first place, 7 percent on the last place. The participants liked the paradigm »security levels« least (rating

of 2.6 out of 5) and 11 percent of participants ranked it on first place, 57 percent on the last place. If we compare the mean and median values of the paradigms, we achieve a higher satisfaction with the paradigm »template instantiation« than with the »security levels« (mean: 1.4; median: 2). In addition, the rankings indicate a significantly better satisfaction with the »template instantiation« than with the »security levels«. We showed that the selection of the paradigm has a significant influence with a large effect on the satisfaction ( $\alpha=0.05$ ,  $H=42.62$ ,  $p<0.01$ ,  $d_c=0.89$ ). However, the selection of the personas has no significant influence on the satisfaction ( $\alpha=0.05$ ,  $H=5.87$ ,  $p=0.12$ ,  $d_c=0.23$ ). In summary, we can reject the null hypothesis.

In summary, the selection of the specification paradigm has a significant effect on the effectiveness, efficiency and satisfaction of the PAP (H1.1 – H1.4). However, these effects do not significantly differ between the personas of Dupree, except the marginally concerned, which performed worse with respect to effectiveness and efficiency (RQ1.1 – RQ1.4). Thus, we recommend to only use two user groups in future experiments. The marginally concerned as one user group and all other participants as the other group.

## 9.5 Summary and Conclusions

Reviewing our hypotheses and research questions, we draw the following conclusions from our evaluation experiment:

### ***RQ1 (Usability of Specification Paradigms)***

We evaluated the usability of the web-based PAP in the policy specification experiment. We confirmed that the selection of the specification paradigm has an effect on the usability. We considered the qualities effectiveness, efficiency and satisfaction. However, we ascertained that they behave contrary to each other: High effectiveness and efficiency was aligned to low satisfaction and otherwise.

- **H1.1 – Objective effectiveness of PAP:** We measured the mistakes made by the participants in the policy specification experiment with each specification paradigm. We compared the results and derived the best matching specification paradigm with respect to objective effectiveness to be the »security levels«.
  - Q1.1.1: Can the optimal mapping of specification paradigms of PAPs to users reduce the number of specification mistakes at least by 30%?
  - M1.1.1: The participants made the most mistakes with the specification paradigm »default policies« (23% of all decisions)



and the fewest mistakes with the »security levels« (7%). If we compare the results, we find that all participants made on average 68 percent fewer mistakes with the paradigm »security levels« than with »default policies«. We found a significant influence with a large effect of the used specification paradigm on the conducted mistakes ( $\alpha=0.05$ ,  $H=48.94$ ,  $p<0.01$ ,  $d_c=0.97$ ). In detail, the paradigm »security levels« leads to a significantly worse objective effectiveness compared to »default policies« ( $z=4.24$ ,  $p<0.01$ ), »template instantiation« ( $z=5.93$ ,  $p<0.01$ ) and »wizard« ( $z=-6.18$ ,  $p<0.01$ ).

- Q1.1.2: Is the optimal mapping of specification paradigms reducing the number of specification mistakes for each persona by at least 30%?
- M1.1.2: For 100 percent of the personas (five out of five), the selection of the best matching specification paradigm decreased the number of mistakes by more than 30 percent (marginally concerned by 55%, amateurs by 58%, all other personas by 100%). However, the influence of the selected specification paradigm on the objective correctness is only significant for the »amateurs« ( $H=16.15$ ,  $p<0.01$ ,  $d_c=0.89$ ), for the »lazy experts« ( $H=16.63$ ,  $p<0.01$ ,  $d_c=1.44$ ) and for the »technicians« ( $H=11.15$ ,  $p=0.01$ ,  $d_c=0.86$ ), but not for the »marginally concerned« ( $H=4.98$ ,  $p=0.17$ ,  $d_c=0.43$ ).
- Q1.1.3: Does the persona selection influence the objective effectiveness when using the different specification paradigms?
- M1.1.3: we found a significant influence with a large effect of the persona selection on the mistakes made ( $\alpha=0.05$ ,  $H=35.23$ ,  $p<0.01$ ,  $d_c=0.81$ ). We explain this effect with the significant difference with respect to objective correctness of the marginally concerned compared to the other personas. The marginally concerned performed significantly worse. We see the influence of the persona selection in each paradigm: »default policies« ( $H=13.88$ ,  $p<0.01$ ), »template instantiation« ( $H=14.10$ ,  $p<0.01$ ), and »wizard« ( $H=17.04$ ,  $p<0.01$ ), and also for the »security levels« ( $H=7.99$ ,  $p<0.05$ ), but not that strong.

➔ In summary, we can reject the Null Hypothesis H1.1<sub>0</sub>.

- **H1.2 – Perceived effectiveness of PAP:** We asked participants in the policy specification experiment to self-evaluate the correctness of the specified policies after each specification with a different specification paradigm. We compared the self-evaluation with the actual correctness and determined the best matching specification paradigm regarding perceived effectiveness to be the »security levels«.

- Q1.2.1: Can the optimal mapping of specification paradigms of PAPs to users increase the accuracy of estimations regarding objectively correct specified policies by at least 30%?
- M1.2.1: The entire participant group performed best with respect to the self-evaluation with the paradigm »security levels« (78.7%) and worst with the »wizard« (29.5%). If we compare the results, we find that the accuracy of self-estimation for all participants is 167% higher on average with the paradigm »security levels« than with the »wizard«. We found a significant influence with a large effect of the used specification paradigm on the correct self-evaluation ( $\alpha=0.05$ ,  $T=38.69$ ,  $p<0.01$ ,  $\phi_c=0.39$ ).
- Q1.2.2: Does the optimal mapping of specification paradigms increase the accuracy of estimations regarding objectively correct specified policies for each persona by at least 30%?
- M1.2.2: For 100 percent of the personas (five out of five), the selection of the best matching specification paradigm increased the accuracy of estimations regarding objectively correct specified policies by more than 30 percent (marginally concerned by 838%, amateurs by 143%, lazy experts by 200%, technicians by 71 %. The increase for the fundamentalist is infinite as 0% of the fundamentalists made a correct estimation with the paradigm »wizard«, but 100% estimated correctly with the paradigm »default policies«). However, the influence of the selected specification paradigm on the correct self-evaluation is not for all personas significant ( $\alpha=0.05$ ). It is significant for the »marginally concerned« ( $T=12.49$ ,  $p=0.01$ ,  $\phi_c=0.53$ ), the »amateurs« ( $T=13.78$ ,  $p<0.01$ ,  $\phi_c=0.41$ ), for the »lazy experts« ( $T=10.86$ ,  $p=0.01$ ,  $\phi_c=0.51$ ), but not for the »technicians« ( $T=4.44$ ,  $p=0.26$ ,  $\phi_c=0.28$ ) or the »fundamentalists« ( $T=6.00$ ,  $p=0.24$ ,  $\phi_c=0.75$ ).
- Q1.2.3: Does the persona selection influence the perceived correctness?
- M1.2.3: We found that the selection of the persona has an influence on the correct self-evaluation ( $T=10.08$ ,  $\phi_c=0.20$ ), but only with a medium effect size. This means that there is little difference in how optimistic or pessimistic the participants of the different personas are when using the specification paradigms.

➔ In summary, we can reject the Null Hypothesis H1.2<sub>o</sub>.

- **H1.3 – Efficiency of PAP:** We measured the time needed for solving six tasks with four specification paradigms. We compared the results

and identified that users perform fastest with the specification paradigm »security levels«.

- Q1.3.1: Can the optimal mapping of specification paradigms of PAPs to users decrease the time needed to specify policies by at least 30%?
- M1.3.1: Participants specified fastest with the specification paradigm »security levels« (1.8 minutes on average) and slowest with the »wizard« (3.8 minutes on average). The selection of an appropriate specification paradigm can decrease the time needed to specify policies by 53%. We found a significant influence with a large effect of the used specification paradigm on the time needed for policy specification, that is the users' efficiency ( $\alpha=0.05$ ,  $H=46.89$ ,  $p<0.01$ ,  $d_c=0.95$ ). This can be explained by the significantly better efficiency of users with the best paradigm »security levels« compared to »default policies« ( $z=5.11$ ,  $p<0.01$ ), »template instantiation« ( $z=4.23$ ,  $p<0.01$ ) and »wizard« ( $z=-6.17$ ,  $p<0.01$ ).
- Q1.3.2: Is the optimal mapping of specification paradigms for decreasing the time needed to specify policies valid for all personas?
- M1.3.2: For 100 percent of the personas (five out of five), the selection of the best matching specification paradigm decreased the needed time by more than 30 percent (marginally concerned by 40%, amateurs by 58%, lazy experts by 70%, technicians by 49% and fundamentalists by 69%). However, the influence of the selected specification paradigm on the efficiency is not for all personas significant ( $\alpha=0.05$ ). It is significant for the »amateurs« ( $H=23.64$ ,  $p<0.01$ ,  $d_c=1.19$ ), for the »lazy experts« ( $H=13.09$ ,  $p<0.01$ ,  $d_c=1.16$ ) and for the »technicians« ( $H=9.85$ ,  $p=0.02$ ,  $d_c=0.79$ ), but not for the »marginally concerned« ( $H=2.57$ ,  $p=0.46$ ,  $d_c=0.20$ ).
- Q1.3.3: Does the persona selection influence the time needed to specify policies?
- M1.3.3: We did not find a significant effect of the persona selection on the time needed ( $\alpha=0.05$ ,  $H=3.90$ ,  $p=0.27$ ,  $d_c=0.13$ ). Thus, the distribution of time needed is similar for the different personas.

➔ In summary, we can reject the Null Hypothesis H1.3<sub>0</sub>.

- **H1.4 – Satisfaction with PAP:** After each specification with a specification paradigm, we asked the participants how much they liked the paradigm. At the end of the policy specification experiment, we asked the participants to rank all four specification paradigms. We

compared the results and determined the best matching specification paradigm regarding satisfaction to be the »template instantiation«.

- Q1.4.1: Can the optimal mapping of specification paradigms of PAPs to users significantly increase the satisfaction experienced by users during the policy specification?
- M1.4.1a: The entire participant group liked the paradigm »template instantiation« most (rating of 4 out of 5). The participants liked the paradigm »security levels« least (rating of 2.6 out of 5). If we compare the mean and median values of the paradigms, we achieve a higher satisfaction with the paradigm »template instantiation« than with the »security levels« (mean: 1.4; median: 2). We found a significant influence with a large effect of the used specification paradigm on the users' satisfaction ( $\alpha=0.05$ ,  $H=42.62$ ,  $p<0.01$ ,  $d_c=0.89$ ). When we compare the paradigms pairwise, we find that users like the specification paradigm »template instantiation« significantly more than the »default policies« ( $z=2.83$ ,  $p=0.03$ ) and the »security levels« ( $z=5.84$ ,  $p<0.01$ ). In addition, participants like the specification paradigm »default policies« ( $z=3.02$ ,  $p=0.02$ ) and »wizard« ( $z=-5.33$ ,  $p<0.01$ ) significantly more than the »security levels«.
- M1.4.1b: 39% of the participants ranked the specification paradigm »template instantiation« in first place, 7% in the last place. Only 11% of participants ranked the specification paradigm »security levels« in the first place, 57% in the last place. The rankings indicate a significantly better satisfaction with the »template instantiation« than with the »security levels«.
- Q1.4.2: Is the optimal mapping of specification paradigms for increasing the satisfaction experienced by users during the policy specification valid for all personas?
- M1.4.2: For 40 percent of the personas (two out of five), the selection of the best matching specification paradigm increased satisfaction significantly ( $\alpha=0.05$ ). It is significant for the »amateurs« ( $H=24.23$ ,  $p<0.01$ ,  $d_c=1.20$ ), for the »lazy experts« ( $H=16.50$ ,  $p<0.01$ ,  $d_c=1.43$ ), but not for the »marginally concerned« ( $H=6.41$ ,  $p=0.09$ ,  $d_c=0.58$ ) or the »technicians« ( $H=4.06$ ,  $p=0.26$ ,  $d_c=0.29$ ).
- Q1.4.2: Does the persona selection influence the satisfaction with specification paradigms?
- M1.4.2: We did not find a significant influence of the persona selection on the users' satisfaction ( $\alpha=0.05$ ,  $H=5.87$ ,  $p=0.12$ ,  $d_c=0.23$ ).

→ In summary, we can reject the Null Hypothesis H1.4<sub>0</sub>.

## **RQ2 (Elicitation)**

We applied the policy template elicitation method in the two case studies. However, in the »Digital Villages« case study we only focused on the user acceptance of the method with respect to RQ2.

- **H2 (Completeness of elicited information):** In the »BeSure« case study, we only validated the completeness of elicited information. According to the experts that validated the method results (14 policy templates), the list of policy templates was complete.

- Q2.1: Is the policy template elicitation method capable of eliciting 90 percent of all necessary policy templates for the application domain?
- M2.1: We elicited 100 percent of all policy templates from the application domain (14/14 = 100%).

→ In summary, we can reject the Null Hypothesis H2<sub>0</sub>.

- **H3 (Correctness of elicited information):** In the »BeSure« case study, we did only validate the correctness of elicited information. According to the experts that validated the method results (14 policy templates), all derived policy templates were correct.

- Q3.1: Is the policy template elicitation method capable of eliciting policy templates that cover more than 90% of the security and privacy demands from the application domain?
- M3.1: The policy template elicitation method allowed us to elicit 100% of the policy templates correctly (14/14 = 100%).

→ In summary, we can reject the Null Hypothesis H3<sub>0</sub>.

- **H4 (User acceptance of elicitation method):** In both case studies, we received positive feedback with respect to our policy template elicitation method (3 participants in »BeSure«, 5 participants in »Digital Villages«).

- Q4.1: Do users rate a workshop in which the policy template elicitation method is applied as a positive experience?
- M4.1: 100% of the participants (8 out of 8) that we asked gave us positive feedback regarding the policy template elicitation method (participation in the elicitation workshop).

→ In summary, we can reject the Null Hypothesis H4<sub>0</sub>.

### **RQ3 (Formalization)**

In the case study »Digital Villages«, we were able to instantiate a policy vocabulary with six derived policy templates. However, we identified a remaining challenge for our model in the »BeSure« case study. Five out of 14 policy template could not be completely expressed in our policy template model, as one required construct is not supported by the model. Thus, we cannot confirm the completeness of the policy template model. However, even without this construct, all policies can be specified by users, but with less comfort.

- H5 (Completeness of policy template model)
  - Q5.1: Is the policy template model capable to represent more than 90 percent of the elicited security and privacy demands in the form of policy templates?
  - M5.1: We were able to model 65% of the derived policy templates in the policy template model  $((6+9)/(6+14) = 65\%)$ .
    - ➔ In summary, we cannot reject the Null Hypothesis  $H5_0$ . The policy template model is still incomplete and needs to be completed in future work.

### **RQ4 (Automation)**

We successfully demonstrated the automated PAP creation in two case studies. This includes the use of two different UI frameworks and four different specification paradigms (»template instantiation«, »default policies«, »security levels« and »wizard«). We confirmed the feasibility of automated PAP creation. Thus, the developer does not need to implement user interfaces for each application domain as this task is automated by generating from the policy vocabulary.

- H6 (Feasibility of automation of PAP creation)
  - Q6.1: Is the process of user interface creation for the task of policy specification automatable for multiple specification paradigms and UI frameworks?
  - M6.1: The user interface creation for 100% (4 of 4) of the tested specification paradigms could be automated.
  - M6.2: The user interface creation of PAPs could be automated for 100% (2 of 2) of the tested UI frameworks.
    - ➔ In summary, we can reject the Null Hypothesis  $H6_0$ .

## **Conclusion**

We conclude that we successfully showed the application of the method for usable PAP generation in two different application domains in the case studies within the projects »BeSure« and »Digital Villages«. In each case study, we successfully applied our contributions of this thesis. We elicited correct and complete policy templates with the policy template elicitation method, we instantiated the complete policy template model to create a policy vocabulary and we used the PAP generation framework for the automated creation of PAPs that provide up to four different specification paradigms.

In addition, we confirmed that the mapping of users to specification paradigms might increase the effectiveness of a PAP. We did not find a strong relation between our proposed mapping and the qualities efficiency and satisfaction. We need to investigate the user to specification paradigm mapping further in future work.



## 10 Summary and Future Work

More and more data is exchanged between users and organizations, such as personal data users are sending to online services. This data is collected, stored, analyzed, reused and partially resold by companies. Users become increasingly afraid of data misuse, and their need for a better protection of their security and privacy is increasing. They want to gain more self-determination in the form of controlling and self-expressing their security and privacy demands for personal data they share with online services. Therefore, they need PAPs to specify security and privacy policies. Many online services provide a PAP. However, studies reveal that many users do not use these tools or have usability issues when doing so. Unfortunately, service providers are somewhat reluctant to improve usability by better tailoring their PAPs to the users, because this required substantial development effort as this is currently a manual process.

We identified the limited usability of existing PAPs and the huge development effort to improve PAP usability as the two key problems. To solve these problems, we devised the method for usable PAP generation as the overall contribution of this thesis. The four main contributions, which we discuss and evaluate within this thesis, are: the policy template elicitation method, the policy template model, the PAP generation framework and the user to specification paradigms mapping.

Overall, we demonstrated the feasibility of the method for usable PAP generation in four case studies. More specifically, we showed that the policy template elicitation method provides correct and complete policy templates and that the method is accepted by the participants of elicitation workshops. We successfully modelled most elicited security and privacy demands as policy templates with our proposed policy template model, even though one construct was missing in the model, which indicates the incompleteness of the model. We showed that a PAP could generate policy specification interfaces implementing multiple supported specification paradigms at runtime using the PAP generation framework. Thus, the developer does not need to implement user interfaces for each application domain as this task is automated by generating them from the policy vocabulary. We provided respective PAPs for all four application domains of the case studies.

In an experiment, we achieved usability improvements by selecting the most appropriate specification paradigm. Our empirical results reveal that users perform differently with respect to effectiveness, efficiency and satisfaction when using different specification paradigms. These three



qualities are individually significantly increased on average (effectiveness and efficiency by more than 30 percent) when selecting the appropriate specification paradigm. However, the results regarding these qualities are contrary as high effectiveness and high efficiency do not imply high satisfaction and vice versa. We showed that these results are valid for a heterogeneous user group as a whole. The clustering of users into personas according to their knowledge and motivation provided similar results.

We conclude the thesis in this chapter by summarizing our methodological and technological contributions in Section 10.1, our empirical contributions in Section 10.2 and our validation results in Section 10.3. Finally, we discuss open issues and future work in Section 10.4.

## 10.1 Methodological and Technological Contributions

We summarize the five methodological and technological contributions of this thesis in the following list:

- **Contribution 1 (C1) – User to Specification Paradigm Mapping:** We provide guidance for selecting the appropriate specification paradigms for users in terms of effectiveness, efficiency and satisfaction.
- **Contribution 2 (C2) – Policy Template Elicitation Method:** We elaborated a method for eliciting policy templates from an application domain. The method consists of five steps: First, information about the application domain is retrieved from a contact person. Next, based on this information, an elicitation workshop is prepared and conducted. In the workshop, the method expert elicits assets, use cases, threats and countermeasures from participating stakeholders of the application domain. Finally, the method expert derives policy templates from the elicited information and validates them together with stakeholders from the application domain. A policy template abstracts security or privacy demands of stakeholders into a variable and instantiable construct. The user employs a PAP to instantiate a policy template into a concrete policy.
- **Contribution 3 (C3) – Policy Template Model:** We created a model that supports the formalization of security and privacy demands as policy templates. The formalization of these demands is a necessary requirement for adding automation to the PAP creation process. The model contains sub-models for describing the domain, security aspects, policy templates on the specification level, transformation rules for generating machine-understandable policies on the implementation level and projection rules for representing policy templates in multiple specification paradigms. We call the instantiation

of the policy template model a policy vocabulary, which can be used to represent security and privacy demands of an application domain.

- **Contribution 4 (C4) – PAP Generation Framework (Concept and Implementation):** We designed the PAP generation framework for the automation of the PAP creation process. This framework can be embedded into a PAP. It enables the generation of user interfaces for the specification of policies at runtime based on a policy vocabulary. The framework is modular and supports the use of multiple UI frameworks, specification paradigms and policy languages. Different types of user interfaces can be generated according to the selected specification paradigm. Our reference implementation of the PAP generation framework is capable of generating a fully functional PAP from a policy vocabulary, and it supports multiple specification paradigms. We provide generation algorithms for four different specification paradigms: »template instantiation«, »default policies«, »security levels« and »wizard«.
- **Contribution 5 (C5) – Method for Usable PAP Generation:** We combined the four aforementioned contributions into a comprehensive method. The method can be used for generating usable PAPs, as requested in the scientific problem statement.

## 10.2 Empirical Contributions

Our empirical contributions comprise three problem derivation surveys, four case studies (three of them with industrial partners) and one experiment. We conducted problem derivation surveys to substantiate our practical problems:

- The **»SECCRIT« survey** was conducted with 15 company representatives. They were asked whether to involve users in the process of policy specification and whether this imposes security risks. This survey revealed that companies want to provide PAPs to users; however, some fear to jeopardize security when letting users specify policies.
- The **»MPK« survey** was conducted with 1,391 visitors of a museum exhibition. We asked them how often they check their security and privacy settings in online services. If they do it only rarely, we asked for their reasons. Of all respondents, about 40 percent stated that they check their security and privacy settings too infrequently because they face usability issues.
- In the **survey in the context of the policy specification experiment**, we tried to confirm the results of the »MPK« survey with 61 participants. Here, more than 60 percent of the respondents update security and privacy settings too infrequently due to usability issues.

They stated that PAPs are too time-consuming and too complicated. In addition, they said that they do not feel competent enough to use the PAPs or that they just forget to do it.

We conducted the first two case studies for improving our contributions:

- In the **»SINNODIUM« case study**, we positively evaluated the concept of policy templates for specifying security and privacy policies in a PAP. We elicited policy templates together with experts of vwd and built an Android PAP with which users can specify policies for the »vwd portfolio manager mobile« Android app.
- In the **»SECCRIT« case study**, we confirmed that the concept of policy templates is suitable for specifying security and privacy policies in a PAP. We elicited policy templates for the application domain of cloud services for critical infrastructure IT together with eight partners from industry and research. We demonstrated the instantiation of the policy template model and the generation of user interfaces for policy specification in PAPs with three selected policy templates in a project demonstrator. To this end, we provided two PAPs (a Java application with the »Swing« UI framework and an Android app) that were capable of letting users specify policies with the specification paradigms »template instantiation« and »default policies«. Thus, we confirmed the feasibility of the policy template elicitation method, the policy template model and the PAP generation framework.

In the second part of our evaluation, we focused on the validation of our contributions:

- In the **»BeSure« case study**, we positively evaluated the policy template elicitation method, the policy template model and the PAP generation framework together with the industry partner DATEV. We applied the policy template elicitation method with stakeholders from DATEV and elicited 14 policy templates in total. Participants enjoyed the workshop. Next, we instantiated the policy template model. During this process, we identified one construct that currently cannot be modeled. The extension of the model to fix this issue is part of future work. Finally, we created an Android PAP with the PAP generation framework. We evaluated the usability of this PAP with the specification paradigm »template instantiation« in a second workshop with experts from DATEV. We got positive feedback and valuable improvement suggestions, such as the provision of a specification wizard. We added this idea in the form of the specification paradigm »wizard« in the final version of the PAP generation framework.
- In the **»Digital Villages« case study**, we positively evaluated the user acceptance of the policy template elicitation method, the completeness of the policy template model and the feasibility of

automation in the PAP creation process. We were able to elicit 14 policy templates. We created a policy vocabulary containing a subset of six selected policy templates. This policy vocabulary was used in the final policy specification experiment. Last, we generated a web-based PAP that supports all four specification paradigms of the PAP generation framework.

- Finally, we conducted a **policy specification experiment** in which we assessed the usability improvements in terms of effectiveness, efficiency and user satisfaction. We let users specify policies with the four specification paradigms of our PAP generation framework and compared the results. We demonstrated usability improvements when selecting the appropriate specification paradigm. Our empirical results reveal that users perform differently with respect to effectiveness, efficiency and satisfaction when using different specification paradigms. These three qualities are individually significantly increased on average (effectiveness and efficiency by more than 30 percent) when selecting the most appropriate specification paradigm compared to the least suitable one. However, the results regarding these qualities are contrary, as high effectiveness and high efficiency do not imply high satisfaction and vice versa. In addition, we clustered users into different persona groups and investigated whether different personas perform significantly different from the participant group as a whole, which we partially confirmed. We showed that users that are unskilled and unmotivated behave differently from all other users. In addition, we partially confirmed our user to specification paradigm mapping based on the user resources that different specification paradigms require from the user.

Overall, we showed the feasibility of the method for usable PAP generation in the four case studies and in the evaluation experiment. More specifically, we demonstrated that the policy template elicitation method provides correct and complete policy templates and that the method is accepted by the participants of elicitation workshops. We successfully modelled security and privacy demands as policy templates with our proposed policy template model, which indicates the completeness of the model, except one missing construct, which we intend to add in the future. Using the PAP generation framework, we generated PAPs with multiple supported specification paradigms in the four different application domains of the case studies.

### 10.3 Validation Results

In our four case studies and the experiment, we answered the questions of our hypotheses from our GQM approach, which we introduced in Section 1.5.

- **H1.1 (Objective effectiveness of PAP):** We approved in one experiment that the best matching specification paradigm leads to 30 percent fewer mistakes than the worst matching specification paradigm. The whole of the participants made most mistakes with the specification paradigm »default policies« (23% of all decisions) and fewest mistakes with the »security levels« (7%). If we compare the results, we find that all participants made on average 68 percent fewer mistakes with the paradigm »security levels« than with »default policies«. We showed that the selection of the paradigm has a significant influence with a large effect on the conducted mistakes ( $\alpha=0.05$ ,  $H=48.94$ ,  $p<0.01$ ,  $d_c=0.97$ ) as well as the selection of the persona ( $\alpha=0.05$ ,  $H=35.23$ ,  $p<0.01$ ,  $d_c=0.81$ ).
- **H1.2 (Perceived effectiveness of PAP):** We approved in one experiment that the best matching specification paradigm leads to 30 percent higher accuracy regarding the self-evaluation of objective correctness than the worst matching specification paradigm. The whole of the participants performed best with respect to the self-evaluation with the paradigm »security levels« (78.7%) and worst with the »wizard« (29.5%). If we compare the results, we find that the accuracy of self-estimation for all participants is on average 167 percent higher with the paradigm »security levels« than with the »wizard«. We showed that the selection of the paradigm has a significant influence with a large effect on the correct self-evaluation with respect to conducted mistakes ( $\alpha=0.05$ ,  $T=38.7$ ,  $p<0.01$ ,  $\phi_c=0.39$ ). In addition, the selection of the persona ( $\alpha=0.05$ ,  $T=10.08$ ,  $p=0.04$ ,  $\phi_c=0.20$ ) has a significant influence with a medium effect.
- **H1.3 (Efficiency of PAP):** We approved in one experiment that the users is 30 percent faster (efficiency) with the best matching specification paradigm than with the worst matching specification paradigm. The whole of the participants specified fastest with the specification paradigm »security levels« (1.8 minutes on average) and slowest with the »wizard« (3.8 minutes on average). If we compare the results, we find that all participants increased efficiency by 111 percent with the »security levels« than with the »wizard«. We showed that the selection of the paradigm has a significant influence with a large effect on the time needed for policy specification with respect to conducted mistakes ( $\alpha=0.05$ ,  $H=46.89$ ,  $p<0.01$ ,  $d_c=0.95$ ), but not for the selection of the persona ( $\alpha=0.05$ ,  $H=3.90$ ,  $p=0.27$ ,  $d_c=0.13$ ).
- **H1.4 (Satisfaction with PAP):** We approved in one experiment that the satisfaction during a policy specification for users when using the best matching specification paradigm is significantly better than with the worst matching specification paradigm. The whole of the participants liked the paradigm »template instantiation« most (rating of 4 out of 5) and 39 percent of participants ranked it on first place,

7 percent on the last place. The participants liked the paradigm »security levels« least (rating of 2.6 out of 5) and 11 percent of participants ranked it on first place, 57 percent on the last place. If we compare the mean and median values of the paradigms, we achieve a higher satisfaction with the paradigm »template instantiation« than with the »security levels« (mean: 1.4; median: 2). In addition, the rankings are indicate a significantly better satisfaction with the »template instantiation« than with the »security levels«. We showed that the selection of the paradigm has a significant influence with a large effect on the satisfaction ( $\alpha=0.05$ ,  $H=42.62$ ,  $p<0.01$ ,  $d_c=0.89$ ). However, the selection of the personas has no significant influence on the satisfaction ( $\alpha=0.05$ ,  $H=5.87$ ,  $p=0.12$ ,  $d_c=0.23$ ).

- **H2 (Correctness of elicited information):** According to the experts that validated the method results in the case studies »SINNODIUM« (7 policy templates) and »BeSure« (14 policy templates), the list of policy templates was complete. In the »SECCRIT« case study, the experts extended the initial 30 policy templates by 10 additional ones during validation. In total, we elicited 84 percent (51 out of 61) of all relevant policy templates from the application domain.
- **H3 (Correctness of elicited information):** According to the experts that validated the method results in the case studies »SINNODIUM« (7 policy templates) and »BeSure« (14 policy templates), all derived policy templates were correct. In the »SECCRIT« case study, the experts found improvement potential in 3 out of 40 policy templates. Thus, the policy template elicitation method allowed us to elicit 95 percent (58 out of 61) of the policy templates correctly.
- **H4 (User acceptance of elicitation method):** Overall, we received positive feedback on our policy template elicitation method in the case studies (2 participants in »SINNODIUM«, 16 participants in »SECCRIT«, 3 participants in »BeSure« and 5 participants in »Digital Villages«). 100 percent (26 out of 26) of the participants perceived the workshop participation as a positive experience. Still, we obtained valuable improvement suggestions for the method from the participants, which we will consider in future work.
- **H5 (Completeness of policy template model):** We were able to instantiate a policy vocabulary with all six derived policy templates in the case study »SINNODIUM«. In the »SECCRIT« case study, we selected three policy templates for the demonstrator, which could all be expressed in the policy template model. In the »Digital Villages« case study, we were able to model all six selected policy templates. However, we identified a remaining challenge for our model in the »BeSure« case study. Five out of 14 policy template could not be completely expressed, as one required construct is currently not supported by the model. Thus, we cannot approve the policy template model to be complete in the context of our case studies. However,



even without this construct, all policies could be instantiated from the policy templates by users, but with less comfort. In total, we were able to formalize 83 percent (25 out of 30) of the derived policy templates in the policy template model.

- **H6 (Feasibility of automation of PAP creation):** We successfully demonstrated the generation of user interfaces for policy specification in all four case studies. This includes PAPs that use four different view modules («Swing», «JavaFx», «Android», «Web») and that support the four presentation modules which implement the specification paradigms «template instantiation», «default policies», «security levels» and «wizard». Our results confirm the feasibility for automated PAP creation. We showed that the generation of user interfaces for specifying policies works for 100 percent (4 out of 4) of the tested specification paradigms and for 100 percent (4 out of 4) of the tested UI frameworks.

Overall, we gained valuable insights into the processes of PAP creation and policy specification with multiple specification paradigms. We identified open issues and topics for future research, which we present in the next section.

## 10.4 Open Issues and Future Work

In this section, we address open issues and future work with respect to our contributions.

### *Policy Template Elicitation Method*

Regarding our policy template elicitation method, we identified the following future research topics:

- **RE techniques:** We tested a limited number of established RE techniques for the elicitation of assets, use cases, threats and countermeasures in our case studies. However, further techniques exist that may provide better results. This needs to be investigated.
- **Validation of Policy Templates:** Our validation is currently an unstructured process performed by stakeholders of the application domain. A more structured approach is desirable.
- **Policy Template Complexity:** The complexity and expressiveness of a policy template depends on the judgement of the method expert. He derives the policy templates based on the example policies. However, we do not provide specific rules for this derivation process. As a consequence, the resulting policy templates can be simple or complex. This potentially influences the number of policies that a user must

instantiate from the templates. Thus, we face a tradeoff between template complexity and number of policies to be instantiated. This tradeoff needs to be researched.

- **Policy maintenance:** A well-defined process for the maintenance of policy templates was requested in one of our case studies by the participating company. We agree on the usefulness of such a process. However, we did not address this maintenance aspect in this work.

### ***Policy Template Model***

The policy template model described in this thesis reflects the status of our research in this field. Due to the immense size of the model and limited time for research and evaluation, not all aspects of the model could be sufficiently scientifically investigated in the context of this dissertation. Thus, some limitations still apply:

- **Multiple instances of selection elements:** Specification-level policy templates may contain multiple paths, which we call selection elements. During the instantiation of an SLPT, at most one selection element may be used. This constraint caused the incompleteness of the model that we discovered in the case study »BeSure«. There are several reasons why we currently do not support the cloning of selection elements during the instantiation of an SLPT. First, we did not yet investigate the effects of selection element cloning on the generation of ILPs. If element group cloning will be allowed, the method expert may have to consider special rules for the specification of ILPTs. Second, the implementation of the PAP framework currently does not support the cloning of the graphical representations of selection elements in the user interface. We would need to provide interface functions to the user to define the number of clones. Third, the specification paradigm projection model currently only supports references to one instance of a selection element. The whole concept of the projection rules needs to be revised in order to support selection element clones. However, selection element cloning is a valuable extension to our approach, as it would enrich the expressiveness of SPLTs. This needs to be further investigated.
- **Boolean logic for element group references in generation rules:** When designing ILPTs, the method expert can define transformation rules for generating ILPs from instantiated SLPTs. One design element are references of ILPT blocks to selection elements of an SLPT. Such a reference means that if a selection element is selected when instantiating the SLPT, the ILPT block is also selected and integrated into the ILP at a defined XML node. Currently we can only model the selection of a single selection element as the condition. In the future, it would be desirable to integrate Boolean logic so that more complex conditions for the ILPT block selection can be specified.



- **Support of more specification paradigms:** The concrete behavior of a SLPT when presented on the user interface with a specific specification paradigm is specified in projection rules. The method expert defines these projection rules with the specification paradigm projection sub-model. As each specification paradigm requires a unique type of projection rules, the specification paradigm projection sub-model must be extended to support additional specification paradigms. Currently, only the four specification paradigms »template instantiation«, »default policies«, »security levels« and »wizard« are supported. The support of more specification paradigms is a worthwhile extension of the model.

### ***PAP Generation Framework***

We presented the concept and our reference implementation of the PAP generation framework in this thesis. Our case studies revealed several potential extensions:

- **Specification paradigm switching:** Currently, a user or the provider of a PAP can select one specification paradigm for policy specification. However, it would be desirable to switch specification paradigms during the specification of policies on the fly. We would need to investigate under which conditions such a switch is possible (e.g., according to different expressiveness of specification paradigms) and implement this functionality into the PAP generation framework. This needs to be further researched.
- **More information for the user:** We could use the information provided by the domain and security and privacy sub-models of the policy template model in order to better support the user with information about risks and threats on the user interface, as requested by Johnson et al. [25] (»Communicate Risk and Threats«). This remains an open issue.

### ***User to Specification Paradigm Mapping***

In this thesis, we described the matching of users and personas to specification paradigms. The following open questions remain to be addressed:

- **Improved mapping:** We showed in the experiment that our assumed mapping of users to specification paradigms is partially increasing the usability. We see positive effects on the effectiveness, but no significant effects on the efficiency and satisfaction. Further research might reveal a method for creating better mappings.
- **User characteristics:** An interesting question is whether characteristics of users have an influence on the specification of

policies other than the ones we identified with our user intention model and their abstraction to »knowledge« and »motivation« with the persona model of Dupree.

- **Objective measurement of barriers:** Metrics and value thresholds to measure the barriers of our user intention model objectively are urgently needed, but do not exist. This may be a direction of future research.

### ***Method for Usable PAP Generation***

We combined the different contributions of our work in the method for usable PAP generation. We use the policy template model to create policy vocabularies within our method. However, the method lacks structured processes for the creation of projection rules for multiple specification paradigms and for the creation of transformation rules for ILP generation. These steps are currently manual and expert-based. Further research into more structured approaches is required to provide better process guidance.

Presently, we also do not provide a usable editor for creating policy vocabularies. Currently, a policy vocabulary must be written in an ordinary text editor. Such a tool could facilitate the method expert's task significantly.

### ***Experimental Validation***

We conducted the policy specification experiment with 61 participants. Splitting them up into five personas resulted in rather small sample sizes per persona. To confirm the results of our policy specification experiment, we need to perform non-exact replications of our experiment, including a larger sample of participants from all personas and additional scenarios. We need to find out whether optimizations in the implementations of the paradigms can positively influence their usability. Therefore, we also need to explore the use of additional paradigms and discuss the current look and feel as well as the interaction process of the paradigms used.



# References

- [1] DOMO, "Data Never Sleeps 6.0," [Online] Available: [https://web-assets.domo.com/blog/wp-content/uploads/2018/06/18\\_domo\\_data-never-sleeps-6verticals.pdf](https://web-assets.domo.com/blog/wp-content/uploads/2018/06/18_domo_data-never-sleeps-6verticals.pdf).
- [2] European Commission, Special Eurobarometer 431 - Data Protection. [Online] Available: [http://ec.europa.eu/commfrontoffice/publicopinion/archives/ebs/ebs\\_431\\_en.pdf](http://ec.europa.eu/commfrontoffice/publicopinion/archives/ebs/ebs_431_en.pdf).
- [3] Symantec, State of Privacy Report 2015. [Online] Available: <https://www.symantec.com/content/en/us/about/presskits/b-state-of-privacy-report-2015.pdf>. Accessed on: Dec. 20 2018.
- [4] L. F. Cranor and N. Buchler, "Better together: Usability and security go hand in hand," *IEEE Security & Privacy*, vol. 12, no. 6, pp. 89–93, 2014.
- [5] OASIS, eXtensible Access Control Markup Language (XACML) Version 3.0. [Online] Available: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>. Accessed on: Jan. 16 2019.
- [6] K. Strater and H. R. Lipford, "Strategies and struggles with privacy in an online social networking community," in *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1*, 2008, pp. 111–119.
- [7] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Analyzing Facebook privacy settings: User expectations vs. reality," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011, pp. 61–70.
- [8] B. Stepien, A. Felty, and S. Matwin, "A non-technical user-oriented display notation for XACML conditions," in *International Conference on E-Technologies*, 2009, pp. 53–64.
- [9] K. Fu et al., Safety, Security, and Privacy Threats Posed by Accelerating Trends in the Internet of Things. [Online] Available: <https://cra.org/cra/wp-content/uploads/sites/2/2017/02/Safety-Security-and-Privacy-Threats-in-IoT.pdf>. Accessed on: Dec. 21 2018.
- [10] M. Johnson, J. Karat, C.-M. Karat, and K. Grueneberg, "Usable Policy Template Authoring for Iterative Policy Refinement," in *IEEE International Symposium on Policies for Distributed Systems and Networks: POLICY 2010* : 21-23 July 2010, Fairfax, Virginia, USA, Fairfax, VA, USA, 2010, pp. 18–21.
- [11] J. Zhao, R. Binns, M. van Kleek, and N. Shadbolt, "Privacy Languages: Are we there yet to enable user controls?," in *Proceedings of the 25th international conference companion on world wide web*, 2016, pp. 799–806.
- [12] M. Rudolph, R. Schwarz, C. Jung, A. Mauthe, and N. u. H. Shirazi, "SECCRIT Deliverable 3.2: Policy Specification Methodology," 2014.
- [13] ISO 9241-11:2018 Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts, ISO 9241-11, 2018.
- [14] J. L. Dupree, R. Devries, D. M. Berry, and E. Lank, "Privacy personas: clustering users via attitudes and behaviors toward security practices," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 5228–5239.
- [15] K. B. Sheehan, "Toward a typology of Internet users and online privacy concerns," *The Information Society*, vol. 18, no. 1, pp. 21–32, 2002.
- [16] F. Abbattista et al., "Improving the usability of an e-commerce web site through personalization," *Recommendation and Personalization in eCommerce*, vol. 2, pp. 20–29, 2002.
- [17] M. Belk, P. Germanakos, C. Fidas, and G. Samaras, "A personalization method based on human factors for improving usability of user authentication tasks," in *International Conference on User Modeling, Adaptation, and Personalization*, 2014, pp. 13–24.
- [18] H. R. Lipford, A. Besmer, and J. Watson, "Understanding Privacy Settings in Facebook with an Audience View," *UPSEC*, vol. 8, pp. 1–8, 2008.
- [19] B. Lampson, "Privacy and security: Usable security: How to get it," *Commun. ACM*, vol. 52, no. 11, p. 25, 2009.
- [20] W. Li et al., "Service-oriented smart home applications: composition, code generation, deployment, and execution," *Service oriented computing and applications*, vol. 6, no. 1, pp. 65–79, 2012.
- [21] G. Fraser and A. Gargantini, "An evaluation of specification based test generation techniques using model checkers," in *Testing: Academic and Industrial Conference-Practice and Research Techniques*, 2009. TAIC PART'09, 2009, pp. 72–81.

- [22] P. Marwedel, "Code generation for embedded processors: An introduction," in *Code Generation for Embedded Processors*: Springer, 2002, pp. 14–31.
- [23] A. van Lamsweerde, S. Brohez, R. Landtsheer, and D. Janssens, "From System Goals to Intruder Anti-Goals: Attack Generation and Resolution for Security Requirements Engineering," *Proceedings of the RE03 workshop on requirements for high assurance systems*, pp. 49–56, 2003.
- [24] A. Whitten, "Making Security Usable," PhD thesis, School of Computer Science, Carnegie Mellon University, 2004.
- [25] M. Johnson, J. Karat, C.-M. Karat, and K. Grueneberg, "Optimizing a policy authoring framework for security and privacy policies," in *Proceedings of the Sixth Symposium on Usable Privacy and Security*, Redmond, Washington, 2010, p. 1.
- [26] L. Fang and K. LeFevre, "Privacy Wizards for Social Networking Sites," in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 351–360.
- [27] A. Whitten and J. D. Tygar, Why Johnny can't encrypt: a usability evaluation of PGP 5.0: USENIX Association.
- [28] S. Cheng, L. Broderick, J. Hyland, and C. Koranda, "Why Johnny still can't encrypt: evaluating the usability of email encryption software," in *Symposium On Usable Privacy and Security 2006*.
- [29] S. Ruoti, J. Andersen, D. Zappala, and K. Seamons, Why Johnny Still, Still Can't Encrypt: Evaluating the Usability of a Modern PGP Client.
- [30] L. F. Cranor and S. Garfinkel, *Security and usability: Designing secure systems that people can use* / edited by Lorrie Faith Cranor & Simson Garfinkel. Beijing: Farnham : O'Reilly, 2005.
- [31] M. E. Zurko, "User-Centered Security: Stepping Up to the Grand Challenge," in *21st Annual Computer Security Applications Conference: Proceedings : 5-9 December, 2005, Tucson, Arizona, Tucson, AZ, USA, 2005*, pp. 187–202.
- [32] C. Kuo, V. Goh, A. Tang, A. Perrig, and J. Walker, *Empowering Ordinary Consumers to Securely Configure their Mobile Devices and Wireless Networks: CyLab*.
- [33] R. W. Reeder, C.-M. Karat, J. Karat, and C. Brodie, "Usability Challenges in Security and Privacy Policy-Authoring Interfaces," in 2007, pp. 141–155.
- [34] K. Vaniea, C.-M. Karat, J. B. Gross, J. Karat, and C. Brodie, Evaluating assistance of natural language policy authoring: ACM. Available: [http://dl.acm.org/ft\\_gateway.cfm?id=1408674&type=pdf](http://dl.acm.org/ft_gateway.cfm?id=1408674&type=pdf).
- [35] C. Kuo, *Reduction of End User Errors in the Design of Scalable, Secure Communication*: Carnegie Mellon University, 2008.
- [36] C. Morisset and D. Sanchez, "VisABAC: A Tool for Visualising ABAC Policies," in *ICISSP, 2018*, pp. 117–126.
- [37] M. Narouei, H. Takabi, and R. Nielsen, "Automatic Extraction of Access Control Policies from Natural Language Documents," *IEEE Transactions on Dependable and Secure Computing*, 2018.
- [38] A. Gerl and F. Prey, "LPL Personal Privacy Policy User Interface: Design and Evaluation," *Mensch und Computer 2018-Workshopband*, 2018.
- [39] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, 1996, pp. 336–343.
- [40] D. Boyd and E. Hargittai, "Facebook privacy settings: Who cares?," *First Monday*, vol. 15, no. 8, <https://journals.uic.edu/ojs/index.php/fm/article/view/3086>, 2010.
- [41] M. Madejski, M. Johnson, and S. M. Bellovin, "A study of privacy settings errors in an online social network," in *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, Lugano, Switzerland, 2012, pp. 340–345.
- [42] J. Dörr, *Elicitation of a Complete Set of Non-Functional Requirements*. Stuttgart: Fraunhofer Verlag, 2011.
- [43] I. F. Alexander, "Misuse cases: use cases with hostile intent," *Software, IEEE*, vol. 20, no. 1, pp. 58–66, 2003.
- [44] J. Doerr, D. Kerkow, T. Koenig, T. Olsson, and T. Suzuki, "Non-functional requirements in industry-three case studies adopting an experience-based NFR method," in *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, 2005, pp. 373–382.
- [45] T. Olzak, "A Practical Approach to Threat Modeling," Mar. 2006. [Online] Available: [http://adventuresinsecurity.com/blog/wp-content/uploads/2006/03/A\\_Practical\\_Approach\\_to\\_Threat\\_Modeling.pdf](http://adventuresinsecurity.com/blog/wp-content/uploads/2006/03/A_Practical_Approach_to_Threat_Modeling.pdf). Accessed on: Dec. 23 2014.
- [46] C. B. Haley, J. D. Moffett, R. Laney, and B. Nuseibeh, "A framework for security requirements engineering," in *Proceedings of the 2006 international workshop on Software engineering for secure systems*, 2006, pp. 35–42.

- 
- [47] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, no. 1, 2011.
  - [48] N. R. Mead and E. D. Hough, "Security Quality Requirements Engineering (SQUARE) Methodology,"
  - [49] K. K. Fletcher and X. Liu, "Security Requirements Analysis, Specification, Prioritization and Policy Development in Cyber-Physical Systems," in *5th International Conference on Secure Software Integration and Reliability Improvement Companion (SSIRI-C)*, 2011, Jeju Island, 2011, pp. 106–113.
  - [50] T. Phan, J. Han, I. Mueller, M. Kapuruge, and S. Versteeg, "SOABSE: An approach to realizing business-oriented security requirements with Web Service security policies," in *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, 2009: 14 - 15 Jan. [i.e. December] 2009, Taipei, Taiwan ; proceedings, Taipei, Taiwan, 2009, pp. 1–10.
  - [51] H. Hibshi, T. D. Breaux, and S. B. Broomell, "Assessment of risk perception in security requirements composition," in *Requirements Engineering Conference (RE)*, 2015 IEEE 23rd International, 2015, pp. 146–155.
  - [52] J. Karat, C.-M. Karat, C. Brodie, and J. Feng, "Privacy in information technology: Designing to enable privacy policy management in organizations," *International Journal of Human-Computer Studies*, vol. 63, no. 1–2, pp. 153–174, <http://www.sciencedirect.com/science/article/pii/S1071581905000649>, 2005.
  - [53] D. Callele and K. Wnuk, "More than requirements: Applying requirements engineering techniques to the challenge of setting corporate intellectual policy, an experience report," in *2011 Fourth International Workshop on Requirements Engineering and Law (RELAW)*, Trento, Italy, pp. 35–42.
  - [54] Z. Sainan and H. Yu, "Research and application of XACML-based fine-grained security policy for distributed system," in *Proceedings / 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*: 20 - 22 Dec. 2013, Shenyang, China, Shengyang, China, 2013, pp. 1848–1851.
  - [55] E. A. Oladimeji, S. Supakkul, and L. Chung, "Representing Security Goals, Policies, and Objects," in *5th IEEEACIS International Conference on Computer and Information Science: (ICIS 2006) in conjunction with 1st IEEEACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (COMSAR 2006) proceedings 10-12 July 2006, Honolulu, Hawaii, Honolulu, HI, USA, 2006*, pp. 160–167.
  - [56] A. Behnia, R. Abd Rashid, and J. Ahsenali Chaudhry, "A Survey of Information Security Risk Analysis Methods," *Smart Computing Review*, vol. 2, no. 1, 2012.
  - [57] J. Busby, L. Langer, M. Schöller, N. Shirazi, and P. Smith, "SECCRIT Deliverable 3.1 - Methodology for Risk Assessment and Management," 2013. [Online] Available: <https://secrit.eu/upload/D3-1-Methodology-for-Risk-Assessment-and-Management.pdf>. Accessed on: Sep. 12 2014.
  - [58] R. A. Caralli, J. F. Stevens, L. R. Young, and W. R. Wilson, *Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process: TECHNICAL REPORT CMU/SEI-2007-TR-012*. [Online] Available: [https://resources.sei.cmu.edu/asset\\_files/TechnicalReport/2007\\_005\\_001\\_14885.pdf](https://resources.sei.cmu.edu/asset_files/TechnicalReport/2007_005_001_14885.pdf). Accessed on: Jan. 14 2019.
  - [59] F. den Braber et al., "The CORAS Methodology: Model-based Risk Assessment Using UML and UP: UML and the Unified Process," in L. Favre, Ed., Hershey, PA, USA: IGI Global, 2003, pp. 332–357.
  - [60] M. S. Lund, B. Solhaug, and K. Stølen, *Model-driven risk analysis: the CORAS approach*: Springer Science & Business Media, 2010.
  - [61] D. Elliott Bell and Leonard J. LaPadula, "Secure Computer Systems: Mathematical Foundations," MITRE Technical Report 2547, Volume I, 1973.
  - [62] Kenneth J. Biba, "Integrity Considerations for Secure Computer Systems," Bedford, Massachusetts, MITRE Technical Report 3153, Revision 1, 1977.
  - [63] C. E. Landwehr, C. L. Heitmeyer, and J. McLean, "A security model for military message systems," *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 3, pp. 198–222, 1984.
  - [64] B. W. Lampson, "Protection," *ACM SIGOPS Operating Systems Review*, vol. 8, no. 1, pp. 18–24, 1974.
  - [65] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (RBAC): Features and motivations," in *Proceedings of 11th annual computer security application conference*, 1995, pp. 241–248.
  - [66] J. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 1, pp. 4–23, 2005.
  - [67] M. Leitner, S. Rinderle-Ma, and J. Mangler, "Responsibility-driven Design and Development of Process-aware Security Policies," in *International Workshop on Secure Software Engineering (SecSE) on Sixth International Conference on Availability, Reliability and Security (ARES)*, Vienna, Austria, 2011, pp. 334–341.
  - [68] C. Choi, J. Choi, and P. Kim, "Ontology-based access control model for security policy reasoning in cloud computing," *J Supercomput*, vol. 67, no. 3, pp. 711–722, 2014.



- [69] S. Haguouche and Z. Jarir, "Toward a generic access control model," in *Proceedings of 2015 IEEE World Conference on Complex Systems*, Marrakech, Morocco, 2015, pp. 1–6.
- [70] M. Ed-Daibouni, A. Lebbat, S. Tallal, and H. Medromi, "A formal specification approach of Privacy-aware Attribute Based Access Control (Pa-ABAC) model for cloud computing," in *2016 Third International Conference on Systems of Collaboration (SysCo)*, Casablanca, Morocco, 2016, pp. 1–5.
- [71] J. Caramujo et al., "RSL-IL4Privacy: a domain-specific language for the rigorous specification of privacy policies," *Requirements Engineering*, pp. 1–26, 2018.
- [72] J. Park and R. Sandhu, "The UCONABC Usage Control Model," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 128–174, 2004.
- [73] J. Jürjens, "UMLsec: Extending UML for secure systems development," in *International Conference on The Unified Modeling Language*, 2002, pp. 412–425.
- [74] D. Basin, M. Clavel, and M. Egea, "A decade of model-driven security," in *Proceedings of the 16th ACM symposium on Access control models and technologies*, 2011, pp. 1–10.
- [75] R. Neisse and J. Doerr, "Model-based specification and refinement of usage control policies," in *2013 Eleventh Annual Conference on Privacy, Security and Trust (PST)*, Tarragona, Spain, pp. 169–176.
- [76] P. Kumari, "Model-Based Policy Derivation for Usage Control," PhD Thesis, Fakultät für Informatik, Technische Universität München, Munich, 2015.
- [77] M. Rudolph, S. Polst, and J. Doerr, "Enabling Users to Specify Correct Privacy Requirements," in *REFSQ 2019*, Essen, Germany, 2019.
- [78] Manuel Rudolph, "User-friendly and Tailored Policy Administration Points," *1st International Conference on Information Systems Security and Privacy (ICISSP), Doctorial Symposium*, 2015.
- [79] M. Rudolph and D. Feth, "Usable Security Policy Specification," *Mensch und Computer 2016-Workshopband*, 2016.
- [80] M. Rudolph, D. Feth, J. Doerr, and J. Spilker, "Requirements Elicitation and Derivation of Security Policy Templates—An Industrial Case Study," in *24th International Requirements Engineering Conference (RE)*, Beijing, China, 2016, pp. 283–292.
- [81] M. Rudolph, D. Feth, and S. Polst, "Why Users Ignore Privacy Policies: A Survey and Intention Model for Explaining User Privacy Behavior," in *19th International Conference on Human-Computer Interaction (HCI)*, Las Vegas, USA, 2018.
- [82] M. Rudolph, C. Moucha, and D. Feth, "A Framework for Generating User-and Domain-Tailored Security Policy Editors," in *3rd Evolving Security & Privacy Requirements Engineering Workshop (ESPRE)*, Beijing, China, 2016, pp. 56–61.
- [83] M. Rudolph and S. Polst, "Satisfying and Efficient Privacy Settings," *Mensch und Computer*, 2018.
- [84] M. Rudolph, R. Schwarz, and C. Jung, "Security policy specification templates for critical infrastructure services in the cloud," in *Workshop Cloud Applications and Security (CAS)*, London, United Kingdom, 2014, pp. 61–66.
- [85] A. Osborn, *Applied Imagination*. New York, NY, USA: Charles Scribner's Sons, 1979.
- [86] D. Zowghi and C. Coulin, "Requirements Elicitation: A Survey of Techniques, Approaches, and Tools," in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Eds.: Springer Berlin Heidelberg, 2005, pp. 19–46.
- [87] R. Agarwal and M. Tanniru, "Knowledge acquisition using structured interviewing: an empirical investigation," *Journal of Management Information Systems*, vol. 7, no. 1, pp. 123–140, 1990.
- [88] W. H. Foddy, *Constructing questions for interviews and questionnaires: theory and practice in social research*: Cambridge University Press, 1994.
- [89] J. Richardson, T. C. Ormerod, and A. Shepherd, "The role of task analysis in capturing requirements for interface design," *Interacting with Computers*, vol. 9, no. 4, pp. 367–384, 1998.
- [90] K. Pohl, *Requirements Engineering – Grundlagen, Prinzipien, Techniken*, 2nd ed.: dpunkt.verlag, 2008.
- [91] K. Pohl and C. Rupp, *Requirements Engineering Fundamentals: A study guide for the certified professional for requirements engineering exam-foundation level / IREB compliant*: Rocky Nook, Inc, 2011.
- [92] C. Rupp, *Requirements Engineering und Management: Professionelle, iterative Anforderungsanalyse für die Praxis*, 5th ed.: Hanser Verlag, 2009.
- [93] Z. Zhang, "Effective Requirements Development - A Comparison of Requirements Elicitation techniques," *Software Quality Management XV: Software Quality in the Knowledge Society*, pp. 225–240, 2007.
- [94] I. F. Alexander, "A Taxonomy of Stakeholders: Human Roles in System Development," *International Journal of Technology and Human Interaction*, vol. 1, no. 1, pp. 23–59, 2005.
- [95] S. Adam, J. Doerr, M. Eisenbarth, and A. Gross, "Using Task-oriented Requirements Engineering in Different Domains: Experiences with Application in Research and Industry," in *Requirements Engineering Conference*, 2009. RE '09. 17th IEEE International, 2009, pp. 267–272.

- 
- [96] R. K. Mitchell, B. R. Agle, and D. J. Wood, "Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts," (English), *The Academy of Management Review*, vol. 22, no. 4, p 853-886, <http://www.jstor.org/stable/259247>, 1997.
  - [97] B. G. Cameron, T. Seher, and E. F. Crawley, "Goals for space exploration based on stakeholder value network considerations," *Acta Astronautica*, vol. 68, no. 11-12, pp. 2088-2097, 2011.
  - [98] U.S. National Institute of Standards and Technology, NIST Special Publication 800-30, Revision 1: Guide for Conducting Risk Assessments. [Online] Available: <http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800-30-r1.pdf>.
  - [99] U.S. National Institute of Standards and Technology, NIST Special Publication 800-37, Revision 1: Guide for Applying the Risk Management Framework to Federal Information Systems. [Online] Available: <http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf>.
  - [100] U.S. National Institute of Standards and Technology, NIST Special Publication 800-53, Revision 4: Security and Privacy Controls for Federal Information Systems and Organizations. Accessed on: Sep. 11 2014.
  - [101] Bundesamt für Sicherheit in der Informationstechnik, IT-Grundschutz-Kompendium: 1. Edition 2018. [Online] Available: [https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKompendium/itgrundschutzKompendium\\_node.html](https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKompendium/itgrundschutzKompendium_node.html). Accessed on: Jan. 14 2019.
  - [102] Common Criteria for Information Technology Security Evaluation (Version 3.1, Revision 5). [Online] Available: <http://www.commoncriteriaportal.org/cc/>. Accessed on: Jan. 14 2019.
  - [103] ISO/IEC 27001 Information security management systems - Requirements, ISO 27001, 2015.
  - [104] ETSI Industry Specification Group (ISG), Information Security Indicators (ISI); Indicators (INC); Group Specification, Part 1: A full set of operational indicators for organizations to use to benchmark their security posture. [Online] Available: [https://www.etsi.org/deliver/etsi\\_gs/ISI/001\\_099/00101/01.01.01\\_60/gs\\_isi00101v010101p.pdf](https://www.etsi.org/deliver/etsi_gs/ISI/001_099/00101/01.01.01_60/gs_isi00101v010101p.pdf). Accessed on: Jan. 05 2019.
  - [105] S. Barker, "The next 700 access control models or a unifying meta-model?," in *SACMAT'09: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies*, Stresa, Italy, 2009, p. 187.
  - [106] J. Park and R. Sandhu, "Towards usage control models: beyond traditional access control," in *Proceedings of Seventh ACM Symposium on Access Control Models and Technologies: SACMAT 2002* : June 3-4, 2002, Naval Postgraduate School, Monterey, California, USA / sponsored by ACM SIGSAC, Monterey, California, USA, 2002, p. 57.
  - [107] X. Zhang, F. Parisi-Presicce, R. Sandhu, and J. Park, "Formal model and policy specification of usage control," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 4, pp. 351-387, 2005.
  - [108] D. Basin, M. Clavel, M. Egea, and M. Schläpfer, "Automatic generation of smart, security-aware GUI models," in *International Symposium on Engineering Secure Software and Systems*, 2010, pp. 201-217.
  - [109] J. L. de Coi and D. Olmedilla, "A Review of Trust Management, Security and Privacy Policy Languages," in *SECURITY*, 2008, pp. 483-490.
  - [110] Fraunhofer IESE, IND<sup>2</sup>UCE. [Online] Available: [www.ind2uce.de](http://www.ind2uce.de). Accessed on: Jul. 04 2018.
  - [111] Fraunhofer IESE, MYDATA Policy Language Documentation. [Online] Available: <https://developer.mydata-control.de/language/>. Accessed on: Jan. 14 2019.
  - [112] M. Hilty, A. Pretschner, D. Basin, C. Schaefer, and T. Walter, "A Policy Language for Distributed Usage Control," in *Lecture Notes in Computer Science*, Computer Security – ESORICS 2007, J. Biskup and J. López, Eds.: Springer Berlin Heidelberg, 2007, pp. 531-546.
  - [113] Information Systems Security Research Group, University of Kent, Permis. [Online] Available: <http://sec.cs.kent.ac.uk/permis/>. Accessed on: Dec. 16 2018.
  - [114] A. Uszok et al., "KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement," in *Policies for Distributed Systems and Networks*, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on, 2003, pp. 93-96.
  - [115] A. Uszok and J. Bradshaw, KAoS Tutorial. [Online] Available: <http://ontology.ihmc.us/KAoS/KAoS%20Tutorial.pdf>.
  - [116] A. Uszok et al., "New Developments in Ontology-Based Policy Management: Increasing the Practicality and Comprehensiveness of KAoS," in *IEEE Workshop on Policies for Distributed Systems and Networks*, 2008, Policy 2008: 2 - 4 June 2008, Palisades, New York, USA ; proceedings, Palisades, NY, USA, 2008, pp. 145-152.
  - [117] C. A. Brodie, C.-M. Karat, and J. Karat, "An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench," in *Proceedings of the second symposium on Usable privacy and security*, Pittsburgh, Pennsylvania, 2006, p. 8.



- [118] University of Hamburg, Hades Java Applet Permission Editor. [Online] Available: <https://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/java-policy-editor.html>. Accessed on: Dec. 16 2018.
- [119] P. Inglesant, M. A. Sasse, D. Chadwick, and L. L. Shi, "Expressions of expertness: the virtuous circle of natural language for access control policy specification," in *Proceedings of the 4th symposium on Usable privacy and security*, 2008, pp. 77–88.
- [120] F. Autrel, F. Cuppens, N. Cuppens-Bouahia, and C. Coma, "MotOrBAC 2: a security policy tool," in *3rd Conference on Security in Network Architectures and Information Systems (SAR-SSI 2008)*, Loctudy, France, 2008, pp. 273–288.
- [121] Telecom Bretagne, MotOrBAC: An OrBAC Security Policy Editor. [Online] Available: <http://motorbac.sourceforge.net/>. Accessed on: Dec. 16 2018.
- [122] University of Murcia, Umu-XACML-Editor. [Online] Available: <http://umu-xacmleditor.sourceforge.net/>. Accessed on: Dec. 16 2018.
- [123] C. Vollat, "Graphical User Interface Development for Usable Policy Administration Points (PAPs)," Bachelor Thesis, TU Kaiserslautern, Kaiserslautern, Germany, 2012.
- [124] K. Verlaenen, B. de Win, and W. Joosen, "Towards simplified specification of policies in different domains," in *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, Munich, Germany, pp. 20–29.
- [125] R. W. Reeder et al., "Expandable grids for visualizing and authoring computer security policies," in *The 26th annual CHI conference on Human Factors in Computing Systems: CHI 2008 / editors: Margaret Burnett ... [et al.]*, Florence, Italy, 2008, p. 1473.
- [126] R. Conti, I. Matteucci, P. Mori, and M. Petrocchi, "An expertise-driven authoring tool of privacy policies for e-Health," *Computer-Based Medical Systems*, Tech. Rep. IIT-CNR TR-02-2014, 2014.
- [127] M. E. Villarreal, S. R. Villarreal, C. M. Westphall, and J. Werner, "Privacy Token: A Mechanism for User's Privacy Specification in Identity Management Systems for the Cloud," *ICN 2017*, p. 64, 2017.
- [128] Facebook Inc., Facebook. [Online] Available: <https://www.facebook.com/>. Accessed on: Dec. 16 2018.
- [129] Google LLC, Google. [Online] Available: <https://www.google.de>. Accessed on: Dec. 16 2018.
- [130] Twitter Inc., Twitter. [Online] Available: <https://twitter.com/>. Accessed on: Dec. 16 2018.
- [131] statcounter GlobalStats, Desktop Browser Market Share Worldwide. [Online] Available: <http://gs.statcounter.com/browser-market-share/desktop/worldwide#monthly-201811-201811-bar>. Accessed on: Dec. 21 2018.
- [132] Google LLC, Google Chrome. [Online] Available: <https://www.google.de/chrome/>. Accessed on: Dec. 16 2018.
- [133] Mozilla Corporation, Mozilla Firefox. [Online] Available: [https://www.mozilla.org/en-US/firefox/new/?utm\\_medium=referral&utm\\_source=firefox-com](https://www.mozilla.org/en-US/firefox/new/?utm_medium=referral&utm_source=firefox-com). Accessed on: Dec. 16 2018.
- [134] Microsoft Corporation, Microsoft Edge. [Online] Available: <https://www.microsoft.com/de-de/windows/microsoft-edge>. Accessed on: Dec. 16 2018.
- [135] Microsoft Corporation, Internet Explorer. [Online] Available: <https://support.microsoft.com/de-de/hub/4230784/internet-explorer-help>. Accessed on: Dec. 16 2018.
- [136] Microsoft Corporation, Windows 10. [Online] Available: <https://www.microsoft.com/de-de/windows>. Accessed on: Dec. 16 2018.
- [137] IBM Corporation, IBM P3P Policy Editor. [Online] Available: <https://www.w3.org/P3P/imp/IBM/>. Accessed on: Dec. 16 2018.
- [138] IBM Corporation, Policy Design Tool. [Online] Available: <https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUid=e84c047d-957c-47d9-a128-699506cdc96e>. Accessed on: Dec. 16 2018.
- [139] WSO2, WSO2 Identity Server- Policy Editor. [Online] Available: <https://docs.wso2.com/display/IS410/Editing+an+XACML+Policy>. Accessed on: Dec. 16 2018.
- [140] I. Ajzen, "The theory of planned behavior," *Organizational behavior and human decision processes*, vol. 50, no. 2, pp. 179–211, 1991.
- [141] B. J. Fogg, "A behavior model for persuasive design," in *Proceedings of the 4th international Conference on Persuasive Technology*, 2009, p. 40.
- [142] A. H. Maslow, "A theory of human motivation," *Psychological review*, vol. 50, no. 4, p. 370, 1943.
- [143] S. Kokolakis, "Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon," *Computers & Security*, vol. 64, pp. 122–134, 2017.
- [144] J. M. Digman, "Personality Structure: Emergence of the Five-Factor Model," *Annual Review of Psychology*, vol. 41, no. 1, pp. 417–440, 1990.
- [145] D. Keirse, *Please understand me 2*: Prometheus Nemesis Book Company, 1998.

- 
- [146] I. B. Myers, M. H. McCaulley, and R. Most, *Manual: A guide to the development and use of the Myers-Briggs Type Indicator*: Consulting Psychologists Press Palo Alto, CA, 1985.
  - [147] P. Kumaraguru and L. Cranor, *Privacy indexes: a survey of Westin's studies*. [Online] Available: <http://repository.cmu.edu/isr/856>.
  - [148] J. M. Urban and C. J. Hoofnagle, "The Privacy Pragmatic as Privacy Vulnerable," in *Workshop on Privacy Personas and Segmentation*, Menlo Park, CA, 2014.
  - [149] H. J. Smith, S. J. Milberg, and S. J. Burke, "Information privacy: measuring individuals' concerns about organizational practices," *MIS Quarterly*, pp. 167–196, 1996.
  - [150] N. K. Malhotra, S. S. Kim, and J. Agarwal, "Internet users' information privacy concerns (IUIPC): The construct, the scale, and a causal model," *Information systems research*, vol. 15, no. 4, pp. 336–355, 2004.
  - [151] A. Morton and M. A. Sasse, "Desperately seeking assurances: Segmenting users by their information-seeking preferences," in *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*, 2014, pp. 102–111.
  - [152] D. Mellado, C. Blanco, L. E. Sánchez, and E. Fernández-Medina, "A systematic review of security requirements engineering," *Computer Standards & Interfaces*, vol. 32, no. 4, pp. 153–165, 2010.
  - [153] B. Karabacak and I. Sogukpinar, "ISRAM: Information Security Risk Analysis Method," *Computers & Security*, vol. 24, no. 2, pp. 147–159, 2005.
  - [154] M. Rudolph and R. Schwarz, "A Critical Survey of Security Indicator Approaches," in *Availability, Reliability and Security (ARES), 2012 Seventh International Conference on*, 2012, pp. 291–300.
  - [155] ISO/IEC 27000 Information security management systems - Overview and vocabulary, ISO 27000, 2018.
  - [156] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, and D. Le Métayer, "Privacy and Data Protection by Design – from policy to engineering," ENISA, 2014. [Online] Available: [https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design/at\\_download/fullReport](https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design/at_download/fullReport). Accessed on: Jun. 20 2018.
  - [157] P. Kumari and A. Pretschner, "Deriving implementation-level policies for usage control enforcement," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*, 2012, pp. 83–94.
  - [158] R. Neisse, A. Pretschner, and V. Di Giacomo, "A Trustworthy Usage Control Enforcement Framework," in *International Workshop on Secure Software Engineering (SecSE) on Sixth International Conference on Availability, Reliability and Security (ARES), Vienna, Austria, 2011*, pp. 230–235.
  - [159] P. Kumari and A. Pretschner, "Model-Based Usage Control Policy Derivation," in *Lecture Notes in Computer Science, Engineering Secure Software and Systems*, J. Jürjens, B. Livshits, and R. Scandariato, Eds.: Springer Berlin Heidelberg, 2013, pp. 58–74.
  - [160] T. Reenskaug and J. O. Coplien, *The DCI Architecture: A New Vision of Object-Oriented Programming*. [Online] Available: <https://klevas.mif.vu.lt/~donatas/Vadovavimas/Temos/DCI/2009%20The%20DCI%20Architecture%20-%20A%20New%20Vision%20of%20OOP.pdf>.
  - [161] JAXB project. [Online] Available: <https://github.com/javaee/jaxb-v2>.
  - [162] EclipseLink MOXy. [Online] Available: <https://wiki.eclipse.org/EclipseLink/Examples/MOXy>.
  - [163] Google Guice. [Online] Available: <https://github.com/google/guice>. Accessed on: Dec. 13 2018.
  - [164] Official Journal of the European Union, REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). [Online] Available: <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679>.
  - [165] M. Hassenzahl, M. Burmester, and F. Koller, "AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität," in *Mensch & Computer 2003*: Springer, 2003, pp. 187–196.
  - [166] Pivotal Software, Inc., *Spring Framework*. [Online] Available: <https://spring.io/>. Accessed on: Dec. 22 2018.
  - [167] Bootstrap Core Team, *Bootstrap*. [Online] Available: <https://getbootstrap.com/>. Accessed on: Dec. 22 2018.
  - [168] D. Fernández, *Thymeleaf*. [Online] Available: <http://www.thymeleaf.org/>. Accessed on: Dec. 22 2018.
  - [169] IBM Corporation, *IBM SPSS Statistics 19*. [Online] Available: <https://www.ibm.com/analytics/spss-statistics-software>. Accessed on: Dec. 22 2018.
  - [170] Microsoft Corporation, *Microsoft Excel 2016*. [Online] Available: <https://www.office.com/>. Accessed on: Dec. 22 2018.



# Appendix A      Security Policy Template Elicitation

## A.1      Elicitation Techniques

In the following section, several elicitation techniques are described in detail that could be applied within the policy template elicitation method. The main goal of all elicitation techniques is in supporting ascertaining knowledge and requirements of the stakeholders involved in a particular project. How and when a certain technique can be applied depends on individual constraints and characteristics of a given project. The most important influencing factors when choosing an elicitation technique are as follows:

- Distinction between conscious, unconscious and subconscious requirements that are to be elicited
- Time, budget constraints and availability of stakeholders
- Experience of requirements engineer
- Chances and risks of the project

Thus, the first important step when choosing a suitable elicitation technique is to identify risk factors of a project. Mostly, these result from:

**Human influences:** during elicitation, good communication is essential. To assure high-quality communication between the requirements engineer and stakeholders, it is important to investigate the following influencing factors:

- the type of requirement and the desired level of detail
- the experience of the requirements engineer and the interviewees with different elicitation techniques
- social, group-dynamic, and cognitive capabilities of the stakeholders
- whether the elicited knowledge is explicit (consciously known) by each individual stakeholder or it is implicit or unconscious

**Organizational influences:** these risk factors comprise, among others, factors like:

- distinction between fixed price contracts and service contracts

- whether system to be built is a new development or an extension of a legacy system
- spatial and temporal availability of the stakeholders

**Operational influences:** it is necessary to consider the content of the requirements, i.e., if the system is very complex, it is advisable to employ a structuring approach during elicitation

**Desired level of detail:** abstract requirements can be elicited rather well using creativity techniques, survey techniques or observational techniques can aid in eliciting requirements of a medium level of detail. Finely detailed requirements can be elicited well by making use of document-centric techniques.

It is advisable to combine different techniques because this minimizes many project risks. Weaknesses and pitfalls of a particular technique can be balanced out with another technique whose strong points lie where the first technique may have deficits. In the following, several techniques are described in more detail.

## Survey Techniques

Survey techniques aim at eliciting as precise and unbiased statements as possible from stakeholders regarding their requirements. These techniques are usually driven by the requirements engineer because he/she asks questions. This, however, might result in the fact that stakeholder concerns are forgotten, superseded, or disregarded.

During an **interview** [91, 92], the requirements engineer asks predetermined questions to one or more stakeholders and documents the answers. Questions that arise during the conversation can be discussed immediately. Moreover, the requirements engineer may uncover subconscious requirements through clever questions. An experienced interviewer individually controls the course of the conversation, completely commits himself/herself to each stakeholder, inquires about specific aspects, and thus ensures the completeness of the answers.

Interview questions have to be formulated neutrally without suggesting any answers. Interviews can be structured with the help of a questionnaire. Prior to the interview, the stakeholders should be informed about the topic / scope and duration of the interview. It should also be clarified (the very latest at the beginning of the interview) whether interview results should be taken confidential. The answers given by the stakeholder should be documented and sent to the interviewee within 48 hours after the interview for the purpose of validation (e.g. to assure correctness).

*Advantages of interviews:* the requirements engineer can individually adapt the conversation and respond to the stakeholder. That is, the requirements engineer can directly react and ask further questions in case of incomplete answers or if further questions arise.

*Disadvantages of interviews:* interviews are time consuming and the selection of suitable stakeholder representatives is critical to the success of the elicitation. Effectiveness of conducting an interview strongly depends on the experience of the requirements engineer. The formulation of the interview questions can have a tremendous effect on the answers given by the stakeholder as also facial expressions or intonation of the interviewer can influence the stakeholder.

**Questionnaires** [91, 92] make use of open and/or closed questions (e.g., multiple-choice questions). If there are a large number of participants that must be surveyed, an online questionnaire is a valuable option. Questionnaires can elicit a magnitude of information in a short amount of time and at low costs. As long as answers are predetermined, even stakeholders that are not able to express their knowledge explicitly can deliver an assessment. A disadvantage of questionnaires is that can be only employed to gather requirements the requirements engineer already knows or conjectures. Creating a proper questionnaire is often tricky and time-consuming and requires thorough knowledge of the domain in question and the psychological guidelines for creating questionnaires. In addition, questionnaires do not provide immediate feedback between the surveyor and the surveyed so it becomes apparent that questions were forgotten or badly formulated only once the questionnaires have been evaluated.

*Advantages of questionnaires:* this technique allows elicitation of requirements from a large number and locally distributed stakeholders with low budget and time effort as questionnaires can be distributed electronically and afterwards (tool-based) analyzed.

*Disadvantages of questionnaires:* questionnaires are not useful to elicit implicit knowledge. Some types of requirements (such as non-functional requirements) are difficult to elicit with a questionnaire, as they are hardly quantifiable. As all questions are fixed in written form, it is tedious to ask further questions that arise during analysis. Furthermore, the formulation of the questions can influence the answers given by the stakeholders.

## Creativity Techniques

Creativity techniques serve the purpose of developing innovative requirements, delineating an initial vision of the system, and eliciting excitement factors.

During **brainstorming** [91, 92], ideas are collected within a certain period, usually in groups of 5 to 10 people. The ideas are documented by a moderator without discussing, judging, or commenting on them at first. Participants use ideas of other participants to develop new or original ideas or to modify existing ideas. After that, collected ideas are subjected to a thorough analysis. This technique is especially effective when a large number of people of different stakeholder groups are involved. Among the advantages of this techniques is that a large number of ideas can be collected in a short amount of time and multiple people can expand on these ideas collaboratively. Brainstorming is usually less effective when the dynamics of the group are muddled or when participants with very varied levels of dominance are involved. For such situations, other creativity techniques may be better suited, such as the *6-3-5 method*.

*Advantages of brainstorming:* Many ideas can be collected within short amount of time. New / innovative solutions can be developed that no one has thought of before.

*Disadvantages of brainstorming:* not effective in case of difficult group dynamics or of participants have different levels of dominance. If participants are locally distributed, it takes effort to organize a brainstorming session.

The **6-3-5 method** [91, 92] is a written variant of the brainstorming method where six participants individually develop three ideas and write these ideas down on cards. After 3-5 minutes, the cards are handed off to the next participant. This participant reads the written ideas and - inspired by those ideas - the participant adds three new ideas and hands off the card to the next participant etc. This handoff is repeated until every participant has received each card once (altogether fivefold handoff).

*Advantages of 6-3-5 method:* can be used if group dynamic is difficult as written form avoids possible conflicts during discussion. Could also be used in case those stakeholders are locally distributed (via email).

*Disadvantages of 6-3-5 method:* compared to brainstorming, the written form of generating ideas might not be that effective, as the collaboration between the participants is less active. The process might also negatively influence the creativity due to the limited time of generating and writing ideas.



**Brainstorming paradox** [91, 92] is a modification of regular brainstorming in that events that must not occur are collected. Afterwards, the group develops measures to prevent the events collected earlier from happening. Through this process, participants often realize which actions may entail negative results. With this method, risks can be identified early on and countermeasures can be developed. Advantages and disadvantages of this technique are identical to those of classic brainstorming.

*Advantages of brainstorming paradox:* participants analyze problem from opposing viewpoint and consciously reflect on issues that might lead to negative results. This method is very effective to identify risks and – similar to Brainstorming – supports the identification of a large number of ideas within a short timeframe.

*Disadvantages of brainstorming paradox* are the same as those of Brainstorming (see above).

## Document-centric Techniques

Document-centric Techniques reuse solutions and experiences made with existing systems. When a legacy system is replaced, this technique ensures that the entire functionality of the legacy system can be identified. Document-centric techniques should be combined with other elicitation techniques so that the validity of the elicited requirements can be determined and new requirements for the new system can be identified.

**System archaeology** [91] is a technique that extracts information required to build a new system from the documentation or implementation (code) of a legacy system or a competitor's system. This technique is often applied when explicit knowledge about the system logic has been lost partially or entirely. This method leads to a large amount of very detailed requirements and is very laborious.

**Perspective-based reading** [91] is applied when documents need to be read with a particular perspective in mind, e.g. the perspective of the implementer or tester. Aspects that are contained in the document but do not pertain to the current perspective are ignored.

## Support Techniques

Support techniques serve as an addition to the elicitation techniques and try to balance out the weaknesses and pitfalls of the chosen technique.

In **mind mapping** [91], a graphical representation of the refined relationships and interdependencies between terms is created. Is often



used as a supporting technique for brainstorming or brainstorming paradox.

Complex processes that involve a large number of stakeholders require cooperative elicitation of requirements. During a joint meeting (**workshop / focus group** [91, 92]), stakeholders with required knowledge and expertise meet to elaborate and discuss goals or details of a certain functionality of the system collaboratively. For example, previously elicited requirements in individual interview sessions can be consolidated, discussed, validated, prioritized, etc., or open issues can be clarified. Each workshop should follow a predefined agenda and rules that should be observed and followed by the moderator.

*Advantages of workshops / focus groups:* direct communication promotes common understanding and willingness to compromise to finally achieve validated results within the team.

*Disadvantages of workshops / focus groups:* negative group dynamics can negatively influence the effectiveness of this technique. In case of limited availability and locally distributed stakeholders, the organization of workshops / focus groups is very difficult and almost impossible to realize.

With the **CRC (Class-Responsibility Collaboration)** [91] technique, context aspects and their respective attributes and properties are denoted on index cards. Requirements are then formulated using these cards.

Further details and references regarding elicitation techniques can be found in [90–92]. The following Table 37 summarizes the suitability of the introduced elicitation techniques based on different influencing factors.

Table 37: Selection of Elicitation Techniques

| Legend<br>»-« : not recommended<br>»0«: no influence (technique can be used)<br>»+«: recommended<br>»++« strongly recommended | Survey Techniques | Interview | Questionnaires | Creativity Techniques | Brainstorming | Brainstorming paradox | 6-3-5 Method | Document-centric Techniques | System archaeology | Perspective-based reading | Supporting Techniques | Workshop / Focus Group |
|---|-------------------|-----------|----------------|-----------------------|---------------|-----------------------|--------------|-----------------------------|--------------------|---------------------------|-----------------------|------------------------|
| Human Influences  |                   |           |                |                       |               |                       |              |                             |                    |                           |                       |                        |
| Stakeholders have varied levels of dominance  |                   | 0         | 0              |                       | -             | -                     | +            |                             | 0                  | 0                         |                       | -                      |
| Stakeholders are not capable of explicitly expressing their knowledge   |                   | -         | +              |                       | -             | -                     | -            |                             | ++                 | ++                        |                       | -                      |
| Stakeholders are not committed to invest time and effort for elicitation  |                   | +         | 0              |                       | -             | -                     | -            |                             | +                  | -                         |                       | -                      |
| Stakeholders have less communicative skills   |                   | -         | 0              |                       | -             | -                     | -            |                             | +                  | +                         |                       | -                      |
| Difficult group dynamics  |                   | 0         | 0              |                       | -             | -                     | +            |                             | 0                  | 0                         |                       | -                      |
| Low skills of requirements engineer in technique  |                   | -         | +              |                       | -             | -                     | +            |                             | ++                 | ++                        |                       | -                      |
| Organizational Influences   |                   |           |                |                       |               |                       |              |                             |                    |                           |                       |                        |
| Elicitation involves a large number of stakeholders   |                   | 0         | ++             |                       | +             | +                     | -            |                             | 0                  | 0                         |                       | +                      |
| Stakeholders are only spatially or temporally available   |                   | ++        | +              |                       | +             | +                     | 0            |                             | ++                 | -                         |                       | +                      |
| Stakeholders are distributed over several locations   |                   | +         | +              |                       | -             | -                     | 0            |                             | +                  | 0                         |                       | -                      |
| Fixed and low budget available  |                   | +         | -              |                       | ++            | ++                    | ++           |                             | -                  | +                         |                       | ++                     |
| Domain / Content related Influences   |                   |           |                |                       |               |                       |              |                             |                    |                           |                       |                        |
| Elicitation of fine-grained requirements  |                   | +         | -              |                       | -             | -                     | -            |                             | +                  | +                         |                       | 0                      |
| Elicitation of high-level requirements  |                   | ++        | +              |                       | ++            | ++                    | ++           |                             | -                  | -                         |                       | 0                      |
| Complex system  |                   | +         | -              |                       | 0             | 0                     | 0            |                             | +                  | +                         |                       | 0                      |
| No domain expertise of requirements engineer  |                   | -         | -              |                       | 0             | 0                     | 0            |                             | +                  | +                         |                       | -                      |

## A.2 Documentation Techniques

In the following section, documentation techniques are described that could be applied within the policy template elicitation method.

**Documentation of goals** [90, 91]: Goals are very well suited to refine the vision of the system. Refining a goal is known as goal decomposition. Goals can be documented using natural language, e.g., by using **goal description templates** [90] (see Table 38) or using goal models. A widely known and very common goal modeling technique is the use of **AND/OR trees** [91] that can be used to document hierarchical decompositions (see Figure 72 and Figure 73).

Table 38: Goal Description Template

| Goal Description Template |  |
|---------------------------|--|
| Goal ID                   | Unique identifier for the goal   |
| Name of goal              | Unique name of the goal  |
| Description of goal       | Detailed description of the goal   |
| Rationale for goal        | Description of the goal's rationale  |
| Super-Goal(s)             | Name and ID of related super-goals   |
| Sub-Goal(s)               | Name and ID of related sub-goals   |
| Supported stakeholders    | Stakeholders can benefit from the fulfillment of the goal  |
| Further relations         | Further relations to other artifacts / requirements (e.g., conflicts, relations to use cases that address this goal, etc.) |
| Priority                  | Priority of the goal   |
| Criticality               | Criticality of the fulfillment of goal (e.g. for project success)  |
| Source                    | Stakeholder, Document or system where the goal has been identified   |
| Author                    | Name(s) of authors that have documented the goal   |
| Version                   | Current version of goal description  |
| Change History            | Change history of goal description   |

According to [90]: A precise and understandable formulation of goals improves the benefit of using goals in requirements engineering. The following **goal description rules** can support the goal formulation:

- Rule 1: Formulate goals on a short and precise manner
- Rule 2: Formulate goals using active voice (avoid passive voice)
- Rule 3: Formulate goals so that they are verifiable
- Rule 4: In case that a goal can't be formulated in a verifiable manner, the goal should be refined into verifiable goals
- Rule 5: The benefit of the goal should be precisely included in the goal description
- Rule 6: The rationale of a goal should be included in the goal description
- Rule 7: Avoid to include solution ideas in the goal description

Using AND/OR trees, two types of decomposition relationships can be distinguished: OR decomposition and AND decomposition. In case of AND decomposition, every sub-goal must be fulfilled so that the super-goal (the root) is fulfilled. In contrast, in OR decompositions, it suffices if at least one sub-goal is fulfilled so that the super-goal is met. Figure 72 and Figure 73 illustrate how these two types of decomposition can be visualized:

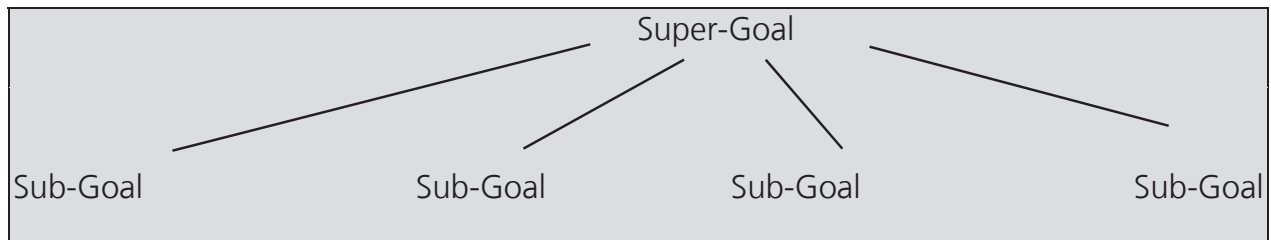


Figure 72: Goal Tree - OR Decomposition

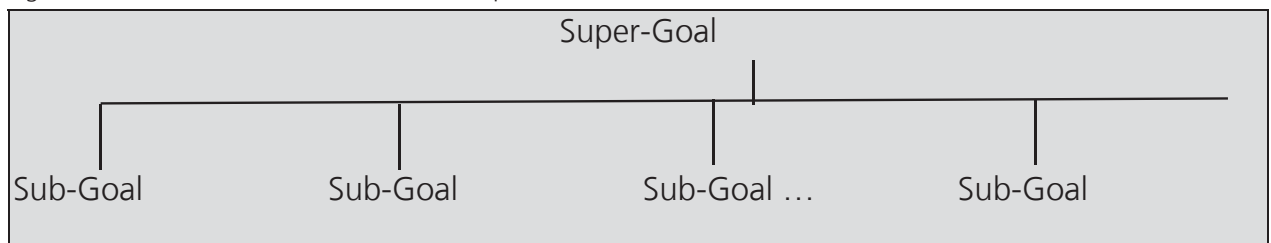


Figure 73: Goal Tree - AND Decomposition

**The documentation of stakeholders:** According to [90], the simplest form to document information about relevant stakeholder is the usage of a **structured stakeholder description template** as illustrated in Table 39.

Table 39: Stakeholder Description Template

| Stakeholder Description Template |   |
|----------------------------------|---|
| Stakeholder ID                   | Unique identifier for the stakeholder   |
| Role                             | Description of role / function that the stakeholder has within the project  |
| Name                             | Name of contact person that is representative of this stakeholder group   |
| Contact                          | Further contact data (email, phone, address, etc.)  |
| Availability                     | Information about availability of stakeholder (e.g., daily via email / phone from 9-15 o'clock, 30% involvement in project, etc.) |
| Knowledge / experience           | Description of knowledge / experience that the stakeholder can bring into the project   |
| Interests and goals              | Description of interests / goals that are important for the stakeholder within the project  |
| Priority                         | Priority of stakeholder (e.g., decision maker, delivers information about certain topic, etc.)                                    |

Further details and references regarding documentation techniques can be found in [90–92].

### A.3 Validation Techniques

In the following section, several validation techniques are described in detail that could be applied within the policy template elicitation method.

During **commenting** [91], the author hand his or her requirements over to another person (co-worker, stakeholder). The goal is to receive the other person's opinion with regard to the quality of a requirement. The other person reviews the requirement with the goal to identify issues that impair requirements quality (e.g., ambiguity or errors) with respect to predetermined quality criteria. The identified flaws are marked in the requirements document and briefly explained.

**Inspections** [91] are done to systematically check artifacts for errors by applying a strict process. An inspection is typically separated into various phases:

- **Planning:** among other things, the goal of the inspection, the work results that are to be inspected, and the roles and participants are determined during this phase
- **Overview:** the author explains the requirements to be inspected to all team members so that there is a common understanding about the requirements among all inspectors
- **Error detection:** the inspectors search through the requirements for errors. Error detection can be performed individually by each inspector or collaboratively in the team. During the course of error detection, any errors that are found are purposively documented.
- **Error collection and consolidation:** all identified errors are collected, consolidated, and documented. During consolidation, errors that have been identified multiple times or errors that are not really errors are identified. Along with consolidation, the identified errors and correcting measures are documented in an error list.

For an inspection to be performed, the following roles must be staffed:

- **Organizer** that plans and supervises the inspection process.
- **Moderator** that leads the session. It is advisable to select a neutral moderator because the moderator could potentially balance out opposing opinions of authors and inspectors.
- **Author** that explains the requirements to the inspectors in the overview phase and later on corrects the identified errors.

- **Reader** that introduces the requirements to be inspected successively and guides the inspectors through them. The role of the reader should be assigned to a neutral stakeholder (often it is the moderator).
- **Inspectors** that are responsible for finding errors and communicating their findings.
- **Minutes-taker** that takes minutes of the results of the inspection.

**Perspective-based reading** [91] is a technique for requirements validation in which requirements are checked by adopting different perspectives. Typically, perspective-based reading is applied in conjunction with other review techniques (e.g., inspections).

Focusing on particular perspectives when reading a document verifiably leads to improved results during requirements validation. Typical perspectives for validation include:

- User / customer perspective
- Software architect perspective
- Tester perspective

Furthermore, three quality aspects also describe three possible perspectives for requirements validation:

- **Content perspective:** the auditor verifies the content of requirements and focuses on the quality of the content of the documented requirements
- **Documentation perspective:** the auditor ensures that all documentation guidelines for requirements and requirements documents have been met
- **Agreement perspective:** the auditor checks if all stakeholders agree on a requirement, i.e., if the requirements are agreed upon and conflicts have been resolved.

In addition, further perspective that emerge from the individual context of the development project can be created as need be.

During perspective-based validation, each auditor is assigned a perspective from which he/she reads and validates the requirement. For each perspective defined, detailed instructions for performing the validation should be laid down because the auditor might not be familiar with all relevant details of his/her assigned perspective. It is advisable to associate questions with each validation instruction that must be answered by the content of the requirements or by the auditor after he/she has read the requirement, respectively. In addition, validation instructions can be

amended with a checklist that summarizes the most important context aspects that ought to be addressed by a requirement with regard to the appropriate perspective.

During the course of a follow-up to a perspective-based reading session, the results of the chosen perspective are analyzed and consolidated.

Further details and references regarding validation techniques can be found in [90–92].

## A.4 Prioritization Techniques

In the following section, several validation techniques are described in detail that could be applied within the policy template elicitation method.

For prioritization, multiple techniques exist that mainly differ with regard to the time and effort needed but also with regard to the suitability of the different prioritization criteria and project properties. Two well-established techniques for requirements prioritization are:

- **Ranking** [91] in which a number of selected stakeholders arrange the requirements to be prioritized with respect to a specific criterion and
- **Top-ten technique** [91] in which the  $n$  most important requirements for a defined criterion are selected. For these requirements, a ranking order is determined afterward. This ranking order represents the importance of the selected requirements with regard to the defined criterion.

Another prioritization technique that is often used in practice is the **single-criterion classification** [91]. This technique is based on the classification of requirements with respect to the importance of the realization of the requirements for the system's success by assigning each requirement to one of the following priority classes:

- A **mandatory** requirement is a requirement that must be implemented at all costs or else the success of the system is threatened.
- An **optional** requirement is a requirement that does not necessarily need to be implemented. Neglecting a few requirements of this class does not threaten the success of the system
- **Nice-to-have** requirements are requirements that do not influence the system's success if they are not implemented.

In practice, differentiating between »optional« and »nice-to-have« requirements can be very difficult. Therefore, requirements classification demands classification criteria that are as objectively verifiable as possible.

Further details and references regarding prioritization techniques can be found in [90–92].

## **A.5 Generic Attacker Roles, Threats and Countermeasures**

This section lists generic exemplary attacker roles, threats and countermeasures.

### **Attacker roles:**

- Script kiddie: hacker that conducts hacking to proof own skills
- Internal attacker: attacker from inside the organization
- Accidental attacker: Internal attacker that causes harm by accident due to misoperation
- Thief: Person stealing information for the goal to sell them
- Rival: Competitor or organization with similar business
- Activist: Person that wants to enforce any social, political, economic, or environmental reform
- Avenger: Person hating the organization for any reason and conducting revenge
- Terrorist: Person intentionally indiscriminating violence as a means to create terror among masses of people
- Vandal: Person destroying stuff for fun
- Jealous partner: Person that wants to retrieve personal information of the partner due to jealousy

### **Threats:**

- Unauthorized access to data
  - Theft of specific data (e.g., documents containing sensitive information)
  - Mass retrieval of data (high number of accesses to data category)
  - Denial of service (frequency of access to data/data memory)
- Unauthorized modification of data
  - data corruption
  - obfuscation of facts
- Unauthorized deletion of data



- data destruction
  - repudiation
- Unauthorized copying of data
  - Reproduction of data
- Unauthorized data flow
  - Entering/leaving the corporate network/security level in the corporate network/certain computer
  - Copying to external removable media device
  - Copy to an externally exposed location
  - Upload in the cloud/to social media network

**Countermeasures:**

- Data accesses
  - Prohibit data access
    - Prevent reading of the data
    - Prevent writing or modifying of the data
    - Prohibition of data access for time period
  - Regulate data accesses
    - <n> accesses to same data
    - <n> accesses the same data category
    - <n> accesses in time span to same data
    - <n> accesses in time span to data category
    - <n> accesses to any data in time period
  - Delay data flow/access by time period
  - Allow/deny context-based access
    - Access only from home/working place/...
    - Access only from certain computer
    - Access only at certain times or dates
  - Set access conditions
    - 4-eyes principle
    - Approval of a specific role/data owner before access
    - 1-factor authentication (knowledge)

- 2-factor authentication (knowledge and ownership)
- Modify the data
  - Anonymization of data
  - Pseudonymization of data
  - Aggregation of data
  - Delete data after access
  - Create a copy before modification (version management)
- Classical data protection
  - Encryption of the data
  - Digital signing of the data
  - Building checksums of the data
  - Performing regular data backups
  - Enforcing high availability of data
- Additional actions (Information regarding data access)
  - Information by e-mail
  - Information by text
  - Logging of accesses (accessing entity, time, duration, data, context)
  - Data flow tracking

**Enforce countermeasures only in specific contexts:**

- In time period (time/date/after other action)
- Triggered by previous action
- Consider current location of the data (corporate network, security level in the corporate network, at the customer, on the Internet)
- Consider current flow of data (entering/leaving the app/ DD platform)
- Consider current location of the user (at home, at work, on business trip, ...)



# Appendix B      PAP Generation Framework

## B.1      XML Schema for Policy Vocabularies

```
<?xml version="1.0" encoding="utf-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.iese.fraunhofer.de/ind2uce/3.0.53/policyVocabulary"
targetNamespace="http://www.iese.fraunhofer.de/ind2uce/3.0.53/policyVocabulary"
xmlns:ind2uce="http://www.iese.fraunhofer.de/ind2uce/3.2.46/ind2uceLanguage"
xmlns:llxsdInd2uce="http://www.iese.fraunhofer.de/ind2uce/3.0.53/llInd2uce"
elementFormDefault="qualified">

  <import namespace="http://www.iese.fraunhofer.de/ind2uce/3.0.53/llInd2uce"
schemaLocation="llxsd.ind2uce/llxsd_ind2uce_llbt.xsd" />
  <import namespace="http://www.iese.fraunhofer.de/ind2uce/3.2.46/ind2uceLanguage"
schemaLocation="llxsd.ind2uce/ind2uceLanguage.xsd" />

  <complexType name="IND2UCEPolicyType">
    <sequence>
      <element name="policy" type="ind2uce:PolicyType" minOccurs="1" maxOccurs="1" />
    </sequence>
  </complexType>

  <complexType name="llptBlockType">
    <group ref="llxsdInd2uce:llptBlockInd2uce" />
    <attribute name="id" type="ID" use="required" />
    <attribute name="use" type="string" use="required" />
  </complexType>

  <complexType name="llptGroupType">
    <sequence>
      <element name="ilptBlock" type="tns:llptBlockType" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
    <attribute name="id" type="ID" use="required" />
  </complexType>

  <complexType name="llptPartsType">
    <sequence>
      <element name="ilptGroup" type="tns:llptGroupType" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
  </complexType>

  <complexType name="llptType">
    <choice>
      <element name="ind2ucePolicy" type="tns:IND2UCEPolicyType" />
    </choice>
  </complexType>

  <complexType name="SpecificationLevelPolicyDescriptionType">
    <attribute name="value" type="string" />
  </complexType>

  <complexType name="ElementGroupType">
```

```
<sequence>
  <group ref="tns:SpltElements" minOccurs="1" maxOccurs="unbounded" />
</sequence>
<attribute name="id" type="ID" use="required" />
<attribute name="description" type="string" use="optional" />
<attribute name="longDescription" type="string" use="optional" />
<attribute name="value" type="string" use="optional" />
<attribute name="selected" type="boolean" use="optional" default="false" />
<attribute name="cloneParent" type="string" use="optional" />
<attribute name="page" type="string" use="optional" />
</complexType>

<complexType name="VariableChoiceType">
  <attribute name="id" type="ID" use="required" />
  <attribute name="description" type="string" use="required" />
  <attribute name="value" type="string" use="required" />
</complexType>

<simpleType name="ConjunctionTypes">
  <restriction base="string">
    <enumeration value="and" />
    <enumeration value="or" />
  </restriction>
</simpleType>

<simpleType name="VariableTypesType">
  <restriction base="string">
    <enumeration value="string" />
    <enumeration value="integer" />
  </restriction>
</simpleType>

<complexType name="SelectionType">
  <sequence>
    <element name="elementGroup" type="tns:ElementGroupType" minOccurs="1"
maxOccurs="unbounded" />
  </sequence>
  <attribute name="id" type="ID" use="required" />
  <attribute name="conjunction" type="tns:ConjunctionTypes" use="optional" default="and" />
  <attribute name="minSelectedElements" type="long" use="required" />
  <attribute name="maxSelectedElements" type="long" use="required" />
</complexType>

<complexType name="SelectableTextType">
  <sequence>
    <element name="variableChoice" type="tns:VariableChoiceType" minOccurs="1"
maxOccurs="unbounded" />
  </sequence>
  <attribute name="id" type="ID" use="required" />
  <attribute name="description" type="string" use="optional" />
  <attribute name="longDescription" type="string" use="optional" />
  <attribute name="value" type="string" use="optional" />
  <attribute name="type" type="tns:VariableTypesType" use="required" />
</complexType>

<complexType name="VariableType">
  <attribute name="id" type="ID" use="required" />
  <attribute name="description" type="string" use="optional" />
  <attribute name="longDescription" type="string" use="optional" />
```

```

    <attribute name="value" type="string" use="optional" />
    <attribute name="type" type="tns:VariableTypesType" use="required" />
    <attribute name="numberMinValue" type="long" use="optional" />
    <attribute name="numberMaxValue" type="long" use="optional" />
</complexType>

<complexType name="TextType">
    <attribute name="id" type="ID" use="optional" />
    <attribute name="description" type="string" use="optional" />
    <attribute name="longDescription" type="string" use="optional" />
    <attribute name="value" type="string" use="required" />
</complexType>

<group name="SpltElements">
    <choice>
        <element name="text" type="tns:TextType" />
        <element name="variable" type="tns:VariableType" />
        <element name="selectableText" type="tns:SelectableTextType" />
        <element name="selection" type="tns:SelectionType" />
        <element name="elementGroup" type="tns:ElementGroupType" />
    </choice>
</group>

<complexType name="SpltType">
    <group ref="tns:SpltElements" minOccurs="1" maxOccurs="unbounded" />
</complexType>

<complexType name="SpltReferenceType">
    <attribute name="ref" type="string" />
</complexType>

<complexType name="DefaultValueType">
    <attribute name="value" type="string" />
    <attribute name="ref" type="string" />
</complexType>

<complexType name="DefaultPolicyType">
    <sequence>
        <element name="defaultValue" type="tns:DefaultValueType" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
    <attribute name="id" type="ID" use="required" />
    <attribute name="description" type="string" use="required" />
</complexType>

<complexType name="DefaultPolicyListType">
    <sequence>
        <element name="DefaultPolicy" type="tns:DefaultPolicyType" maxOccurs="unbounded" />
    </sequence>
</complexType>

<complexType name="FilterListType">
    <sequence>
        <element name="filter" type="tns:FilterType" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
</complexType>

<complexType name="FilterType">
    <attribute name="typeId" type="long" use="required" />

```

```
<attribute name="filterId" type="long" use="required" />
</complexType>

<complexType name="SlptDescriptionType">
  <attribute name="description" type="string" use="required" />
</complexType>

<group name="PageElements">
  <choice>
    <element name="slptReference" type="tns:SlptReferenceType" />
    <element name="slptDescription" type="tns:SlptDescriptionType" />
  </choice>
</group>

<complexType name="PageType">
  <group ref="tns:PageElements" minOccurs="1" maxOccurs="unbounded" />
  <attribute name="id" type="ID" use="required" />
  <attribute name="title" type="string" use="required" />
</complexType>

<complexType name="WizardPageDetailsType">
  <sequence>
    <element name="page" type="tns:PageType" minOccurs="1" maxOccurs="unbounded" />
  </sequence>
</complexType>

<complexType name="PolicyTemplateType">
  <all>
    <element name="description" minOccurs="1" maxOccurs="1">
      <complexType>
        <attribute name="value" type="string" use="required" />
      </complexType>
    </element>
    <element name="exemplary_instantiation" minOccurs="1" maxOccurs="1">
      <complexType>
        <attribute name="value" type="string" use="required" />
      </complexType>
    </element>
    <element name="asset" minOccurs="1" maxOccurs="1">
      <complexType>
        <attribute name="value" type="string" use="required" />
      </complexType>
    </element>
    <element name="threat" minOccurs="1" maxOccurs="1">
      <complexType>
        <attribute name="value" type="string" use="required" />
      </complexType>
    </element>
    <element name="filters" type="tns:FilterListType" minOccurs="1" maxOccurs="1" />
    <element name="wizardPageDetails" type="tns:WizardPageDetailsType" minOccurs="1"
maxOccurs="1" />
    <element name="DefaultPolycys" type="tns:DefaultPolicyListType" minOccurs="0" />
    <element name="slpt" type="tns:SlptType" minOccurs="1" maxOccurs="1" />
    <element name="ilptParts" type="tns:IlptPartsType" minOccurs="0" maxOccurs="1" />
    <element name="ilpt" type="tns:IlptType" minOccurs="1" maxOccurs="1" />
  </all>
  <attribute name="id" type="ID" use="required" />
  <attribute name="name" type="string" use="required" />
</complexType>
```

```

<complexType name="FilterTypeType">
  <sequence>
    <element name="filterValue" minOccurs="1" maxOccurs="unbounded">
      <complexType>
        <attribute name="filterId" type="long" use="required" />
        <attribute name="name" type="string" use="required" />
      </complexType>
    </element>
  </sequence>
  <attribute name="typeId" type="long" use="required" />
  <attribute name="name" type="string" use="required" />
</complexType>

<complexType name="filterDefinitionType">
  <sequence>
    <element name="filterType" type="tns:FilterTypeType" minOccurs="1" maxOccurs="unbounded" />
  </sequence>
</complexType>

<simpleType name="LanguageTypes">
  <restriction base="string">
    <enumeration value="english" />
    <enumeration value="german" />
  </restriction>
</simpleType>

<complexType name="DefaultPolicyReferenceType">
  <attribute name="defaultPolicyReference" type="string" use="required" />
  <attribute name="templateReference" type="string" use="required" />
</complexType>

<complexType name="SecurityLevelType">
  <sequence>
    <element name="defaultPolicyReference" type="tns:DefaultPolicyReferenceType" minOccurs="0"
maxOccurs="unbounded" />
  </sequence>
  <attribute name="id" type="ID" use="required" />
  <attribute name="name" type="string" use="required" />
</complexType>

<complexType name="SecurityLevelDefinitionType">
  <sequence>
    <element name="securityLevel" type="tns:SecurityLevelType" minOccurs="1" maxOccurs="unbounded"
/>
  </sequence>
</complexType>

<complexType name="TemplateReferenceType">
  <attribute name="reference" type="string" use="required" />
  <attribute name="description" type="string" use="required" />
</complexType>

<complexType name="WizardDefinitionType">
  <sequence>
    <element name="templateReference" type="tns:TemplateReferenceType" minOccurs="1"
maxOccurs="unbounded" />
  </sequence>
</complexType>

```



```
<element name="policyModelInstance">
  <complexType>
    <sequence>
      <element name="filterDefinition" type="tns:filterDefinitionType" minOccurs="1" maxOccurs="1" />
      <element name="securityLevelDefinition" type="tns:SecurityLevelDefinitionType" minOccurs="1"
maxOccurs="1" />
      <element name="wizardDefinition" type="tns:WizardDefinitionType" minOccurs="1" maxOccurs="1"
/>
      <element name="policyTemplate" type="tns:PolicyTemplateType" minOccurs="1"
maxOccurs="unbounded" />
    </sequence>
    <attribute name="language" type="tns:LanguageTypes" use="required" />
  </complexType>
</element>

</schema>
```

## Appendix C The Personas of the Dupree Model

In this section, the personas of the Dupree model are described and mapped to the user resources of our user intention model. Dupree created the five personas based on character traits she discovered from participants during her studies. We used those traits for describing the personas. Therefore, we formulated those character traits in the ego-perspective. In the following, we state our assumptions how the personas will perform with respect to effectiveness and efficiency when using PAPs in the policy specification experiment. We do not make any assumptions regarding the satisfaction with PAPs as we think that more influence factors than knowledge and motivation apply.

### Fundamentalist

The »fundamentalist« has a high intrinsic motivation and many resources (see Figure 74); he has extensive knowledge about security and privacy technologies and measures. Thus, for a fundamentalist the barriers might be lower than for other personas, while the motivation is high. This results in a strong intention to use PAPs. Therefore, we assume good results in our evaluation with respect to effectiveness and efficiency. In addition, it is unlikely that a fundamentalist is prone to the privacy paradox.



Figure 74: Character Traits for Persona »Fundamentalist«

### Amateur

The »amateur« has medium motivation to specify policies, however this depends on the situation (see Figure 75). His motivation to protect his bank account is higher than protecting his wireless network. The degree of motivation is probably influenced by his awareness of security and privacy security issues. He has only medium knowledge, which is why he

cannot judge the quality of an advice. To sum it up, we expect medium usability issues for the amateur. In addition, the amateur is prone to the privacy paradox regarding some technologies or data but not regarding all of them.



-  I try to **learn** stuff about security and privacy and already had some success.
-  Sometimes I feel good that there are others having less **knowledge** in security and privacy than I do.
-  When I get a new hint how to improve my privacy or security, I cannot always judge whether the **advice** is good or bad. However, I am willing to take action to protect myself if I get sound advice.
-  I also spent a lot of time to **increase my security and privacy**, after I was attacked this one time.
-  I use some **software** tools to protect my computers and devices, usually an anti-virus and on some devices a firewall or an ad blocker.
-  I have one strong **password** for the important web sites and some more simple passwords for the others. However, I have to write all these passwords down, because I cannot remember them all.
-  For protecting my **bank account**, I take extra care.
-  Of course, I trust my own **wireless network** at home, but I never change its configuration.
-  When I **share information online**, for example on Facebook, I am setting myself at least some limits about what I am posting.
-  Generally, I am concerned about my computer being infected by viruses and I fear an intrusion that **harms or steals my data** (e.g., identity theft).

Figure 75: Character Traits for Persona »Amateur«

## Marginally Concerned

The »marginally concerned« has low motivation to protect his security and privacy (see Figure 76). His knowledge is low, thus we assume that his resources do not meet the user requirements of many PAPs. Consequently, his intention is low and the probability that he performs the desired behavior (privacy actions) is low. He is not affected by the privacy paradox, since the paradox implies that a person has the motivation to protect his privacy. However, we assume the marginally concerned to have low efficiency and effectiveness when using PAPs. Thus, he has significant usability issues.



-  My **knowledge** of security and privacy is based on information told to me by friends and relatives or on TV shows such as CSI. However, most of the time, I do not understand all these technical terms.
-  I know that there are **security and privacy threats**, but I do not worry so much about them. What could go wrong?
-  The only security **software** I use is an antivirus scanner.
-  I only have a few simple **passwords**, but I use one of them most often. I only change passwords if a website forces me to do so, e.g., if a more complicated password is required. Therefore, I have to write all these complex passwords down, as I cannot remember them.
-  I like the option to **reset passwords** on websites, when I have forgotten a password.
-  I use **public wireless networks** without further protection measures.
-  If a website is popular, I **trust** it. In addition, I trust a website more, if it claims to be secure, e.g., by providing a security label or promising not to sell any information about me.
-  I do not care that some company might **monitor** my online activities.
-  I am an honest person: I have **nothing to hide**.

Figure 76: Character Traits for Persona »Marginally Concerned«

## Lazy Expert

The name »lazy expert« describes the motivation and knowledge of this persona well. He has low motivation but expert knowledge (see Figure 77). When only considering the knowledge, his barriers could be low. However, according to his motivation, he will hardly start the specification of any policy unless there is an acute trigger, such as an attack on this data. In such a case his motivation seems to increase for a short time resulting in actions. The most of the time the barriers seem to outrange the motivation a bit. This could be explained by resources the lazy expert is lacking, such as available time and cognitive resources. Like the marginally concerned, the lazy expert is not affected by the privacy paradox. We expect that he can perform well with respect to effectiveness and efficiency, if the trigger is acute enough.



Figure 77: Character Traits for Persona »Lazy Expert«

## Technician

The »technician« has high motivation and medium knowledge (see Figure 78). With his medium knowledge, he cannot meet high user requirements of PAPs. Thus, he faces some barriers. However, his motivation is high. That is why he manages to overcome several barriers. The privacy paradox applies to some regard to him. The technician would like to take sound security and privacy actions, however, he faces some problems when doing so. He instead only performs easy privacy actions. We assume that the technician performs better with respect to effectiveness and efficiency than the marginally concerned and amateurs, but worse than fundamentalists.



- I am highly motivated to take care of my security and privacy, and I think I have a solid **knowledge** of these topics.
- I regularly read online news and blogs to keep up with the newest trends. I first try to **understand** things before using them.
- I have limited **trust** in privacy settings on sites like Facebook.
- I am only a passive user of social networking, because my **privacy** is more important than being social online.
- All my **passwords** look somehow similar, but they are still unique for most services.
- Generally, I am concerned about my computer being infected by viruses and I fear an intrusion that may **harm or steal my data** (e.g., identity theft). However, sometimes I forget that worry.
- I pay special attention to protect my **bank account**.
- If I had the choice between security and convenience, I would take **security**.
- While surfing on new websites, I **trust** my first impressions about the look and feel. I know whether I can trust a site when I see it. In addition, a higher popularity of a website increases my confidence in its trustworthiness.
- I am willing to change my security and privacy behavior on sound **advice**.

Figure 78: Character Traits for Persona »Technician«

## Appendix D Case Study: »SECCRIT«

### D.1 Excerpt of »SECCRIT« Study Results

One asset, for example, is a »critical service« that is operated on the tenant infrastructure level (see Table 40). One exemplary policy templates about »Critical VM Migration« is presented in Table 41.

Table 40: Documented Asset »Critical Service«

|                           |  |
|---------------------------|--|
| Asset ID                  | A1   |
| Asset                     | Critical service (tenant infrastructure level)   |
| Data Owner                | Service owner  |
| Example Use Case          | A service owner (tenant) is running a critical service in the cloud, for example, for running the software for video surveillance on a public place. |
| Policy Authors            | Service owner  |
| Prioritization Properties | (not elicited)   |
| Legal Regulations         | (not elicited)   |

Listing 10: Example Specification Level Policy Template for Policy Template

```
<slpt>
..<text value="If a critical virtual machine is moved to a host already
running a critical VM, then" />
..<selection id="t1_countermeasure_selection" conjunction="and"
.....minSelectedElements="1" maxSelectedElements="4">
..<element id="t1_countermeasure_move_vm" description="move VM to free host"
.....longDescription="The virtual machine that was migrated to the
.....unsuitable host will be removed to a host not yet running a
.....critical VM.">
...<text value="move virtual machine to a host not yet running a
.....critical VM" />
...</element>
..<element id="t1_countermeasure_notification" description="email
.....notification" longDescription="An email notification is sent to the
.....defined recipient.">
...<text value="notify" />
...<variable id="t1_notification_email" type="string" description="email
.....address" longDescription="Enter the email address to which the
.....notification is sent." />
...<text value="via email" />
...</element>
..<element id="t1_countermeasure_log" description="logging"
.....longDescription="writes a log entry">
...<text value="write a log entry" />
...</element>
..<element id="t1_countermeasure_ui" description="UI notification"
.....longDescription="shows a notification on the user interface">
```

```

...<text value="show notifications on the user interface" />
..</element>
.</selection>
</slpt>

```

Table 41: Policy Template »Critical VM Migration«

| ID                     | Policy Template Name  | Asset  | Target System | Security/Privacy Goal         |
|------------------------|-----------------------|--|---------------|-------------------------------|
| T19                    | Critical VM Migration | Critical service   | Cloud system  | Confidentiality, availability |
| Policy Template Syntax |                       | If a critical virtual machine is moved to a host already running a critical VM, then [move VM to a host not yet running a critical VM   notify <email> via email   write a log entry   show notifications on the user interface]*.   |               |                               |
| Description            |                       | A tenant has a VM running a critical infrastructure IT service on a virtual datacenter. The service VM is not allowed to run on a host with another critical infrastructure IT service. The colocation of two critical services can endanger their confidentiality and availability. The colocation increases the attack surface and the hack of one service threatens both services. The templates provides policies for preventing this situation and for notification if such a situation occurs. |               |                               |
| Threats                |                       | <ul style="list-style-type: none"> <li>• Unintended access from one service to another critical infrastructure IT cloud service</li> <li>• Single point of failure for services intended to run independently</li> </ul>   |               |                               |
| Example Instantiation  |                       | If a critical virtual machine is moved to a host already running a critical VM, then move VM to a host not yet running a critical VM and notify manuel.rudolph@iese.fraunhofer.de via email.   |               |                               |

Listing 10 presents the specification level representation of the policy template presented in Table 41 in XML. The corresponding implementation level policy template is listed in Listing 11. The complete policy vocabulary is printed in Appendix D.1.

Listing 11: Example Implementation Level Policy Template for Policy Template

```

<ilpt>
.<ind2ucePolicy>
..<policy name="Critical_VM_Migration">
...<ind2uce:detectiveMechanism name="Migratel"
.....ilptGroupReference="tl_countermeasure_blocks">
....<ind2uce:description>...</ind2uce:description>
....<ind2uce:timestep amount="30" unit="SECONDS" />
....<ind2uce:trigger action="urn:event:ind2uce:vmware:VmMigratedEvent"
.....isTry="false" />
....<ind2uce:condition>
.....<pip:boolean name="urn:ind2uce:vmware:criticalService" default="false">
.....<param:string name="method" value="criticalServiceOnHost" />
.....<param:event name="host" value="host.morValue" />
.....<param:event name="ignoreVM" value="vm.morValue" />
.....</pip:boolean>

```



```

....</ind2uce:condition>
...</ind2uce:detectiveMechanism>
..</policy>
</ind2ucePolicy>
</ilpt>

<ilptParts>
..<ilptGroup id="tl_countermeasure_blocks">
...<ilptBlock id="tl_countermeasure_blocks_move_vm"
.....use="tl_countermeasure_move_vm">
...<llxsdInd2uce:executeAction name="urn:action:ind2uce:vmware:MigrateVM">
...<param:string name="priority" value="highPriority" />
...<param:event name="vm.morType" value="vm.morType" />
...<param:event name="vm.morValue" value="vm.morValue" />
...<param:string name="host.morType" value="HostSystem" />
...<pip:string name="urn:ind2uce:vmware:getFreeHost"
.....paramName="host.morValue" default="host-38439">
.....<param:string name="method" value="getFreeHost" />
.....</pip:string>
...</llxsdInd2uce:executeAction>
...</ilptBlock>
...<ilptBlock id="tl_countermeasure_blocks_notification"
.....use="tl_countermeasure_notification">
...<llxsdInd2uce:executeAction name="urn:action:ind2uce:vmware:sendMail">
...<param:string name="msgPlain"
.....value="Dear Customer, \n\nwe detected a policy violation that
.....critical services were migrated to the same physical host!
.....\nCompensating actions have been performed. \n\nBest Regards,
.....\nIND2UCE" />
...<param:string name="msgHTML"
...<param:boolean name="ind2uceLogo" value="true" />
...<param:string name="subject" value="Policy Violation" />
...<param:string name="recipient" value="$ref:tl_notification_email" />
...</llxsdInd2uce:executeAction>
...</ilptBlock>
...<ilptBlock id="tl_countermeasure_blocks_log" use="tl_countermeasure_log">
...<llxsdInd2uce:executeAction name="urn:action:ind2uce:vmware:log">
...<param:string name="msg" value="Two cricitcal services have been
.....migrated to the same physical host. Compensating actions are
.....running." />
...</llxsdInd2uce:executeAction>
...</ilptBlock>
...<ilptBlock id="tl_countermeasure_blocks_ui" use="tl_countermeasure_ui">
...<llxsdInd2uce:executeAction name="urn:action:ind2uce:http_get">
...<param:string name="paramName" value="msg" />
...<param:string name="paramValue" value="Two cricitcal services have been
.....migrated to the same physical host. Compensating actions are
.....running." />
...<param:string name="urlPrefix" value="http://212.9.140.33:8081" />
...</llxsdInd2uce:executeAction>
...</ilptBlock>
</ilptGroup>
</ilptParts>

```

We generated multiple PAPs with the PAP generation framework. The use of the specification paradigm »template instantiations« using the view module »Swing« is demonstrated in Figure 79. We integrated



transformation rules for generating ILPs. A click on the »Generate Machine-understandable Policy« button instructs the PAP to generate an ILP out of the instantiated policy template. The resulting ILP is based on the policy language »IND<sup>2</sup>UCE Version 1.1« (see Figure 80). As an alternative, the users is able to specify policies using the specification paradigm »default policies« as it can be seen in Figure 81.

The screenshot shows a window titled "PAP Machine-readable Policy". On the left, there is a list of policy templates: "Critical VM Migration", "VM Network Connection", "Virtual Machine CPU Load", and "Virtual Machine CPU Load 2". The "Critical VM Migration" template is selected. The main area displays the following text: "If a critical virtual machine is moved to a host already running a critical VM, then". Below this, there are four checkboxes with corresponding actions in text boxes:
 

- ☒ move virtual machine to a host not yet running a critical VM
- ☒ notify  via email
- ☐ write a log entry
- ☒ show notifications on the user interface

 At the bottom, there is a "Generate Policy" button.

Figure 79: Example PAP Using View Module »Swing«, Policy Vocabulary »SECCRIT« and Presentation Module »Template Instantiations«

The screenshot shows the same "PAP Machine-readable Policy" window, but now displaying the generated ILP in the main area. At the top, there are "deploy" and "clear" buttons. The ILP code is as follows:
 

```

    <pip:string name="urn:ind2uce:vmware:getFreeHost" paramName="host.morValue"
    default="host-38439">
      <param:string name="method" value="getFreeHost"/>
    </pip:string>
    </executeAction>
    <executeAction name="urn:action:ind2uce:vmware:log">
      <param:string name="msg" value="Two cricitcal services have been migrated to the same
    physical host. Compensating actions are running."/>
    </executeAction>
    <executeAction name="urn:action:ind2uce:http_get">
      <param:string name="paramName" value="msg"/>
      <param:string name="paramValue" value="Two cricitcal services have been migrated to the
    same physical host. Compensating actions are running."/>
      <param:string name="urlPrefix" value="http://212.9.140.33:8081"/>
    </executeAction>
    </detectiveMechanism>
    </policy>
    
```

Figure 80: ILP in IND<sup>2</sup>UCE Policy Language Version 1.1 Generated by PAP in UI Framework »Swing«

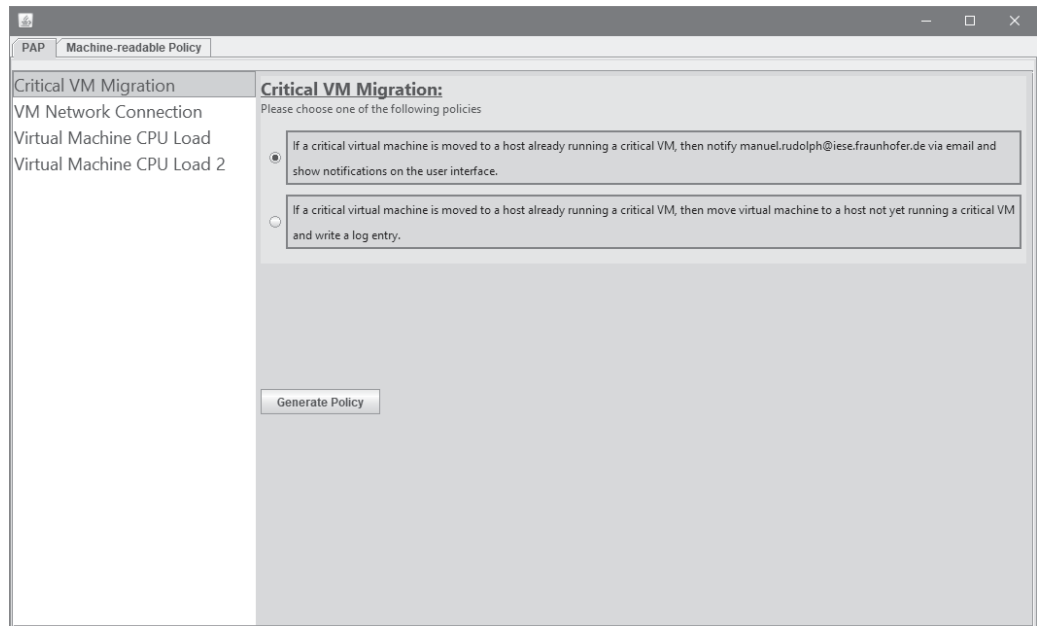


Figure 81: Example PAP Using View Module »Swing«, Policy Vocabulary »SECCRIT« and Presentation Module »Default Policies«

We used the policy vocabulary from the study and the two presentation modules for creating an Android PAP. The »template instantiations« can be seen in Figure 82 and the »default policies« in Figure 84. The Android PAP is able to generate ILPs as depicted in Figure 83.



Figure 82: Example PAP Using View Module »Android«, Policy Vocabulary »SECCRIT« and Presentation Module »Template Instantiations«

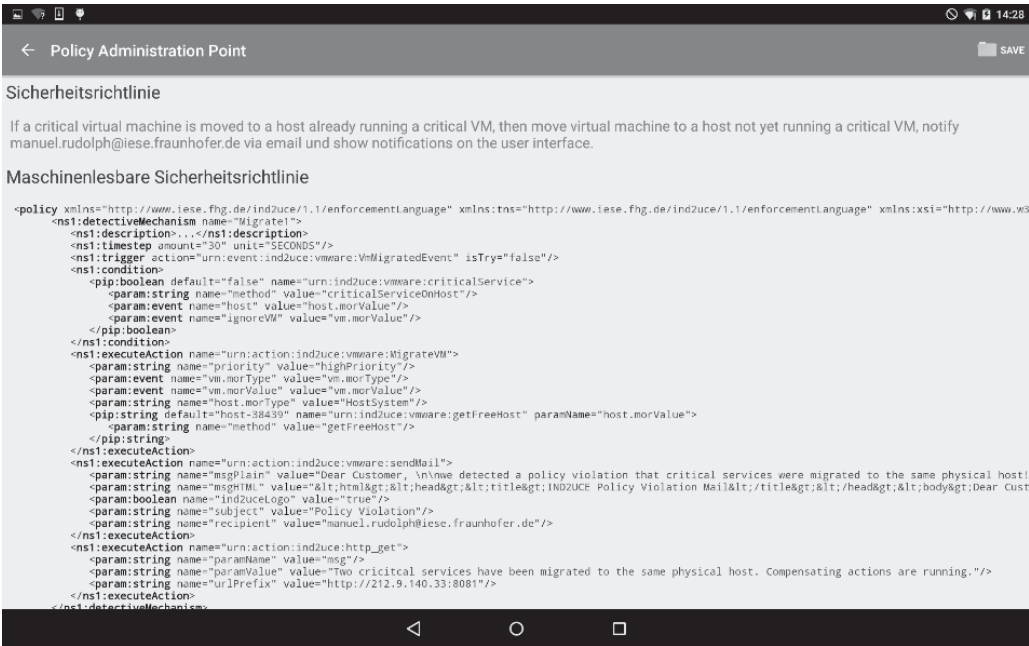


Figure 83: ILP in IND²UCE Policy Language Version 1.1 Generated by PAP in UI Framework »Android«

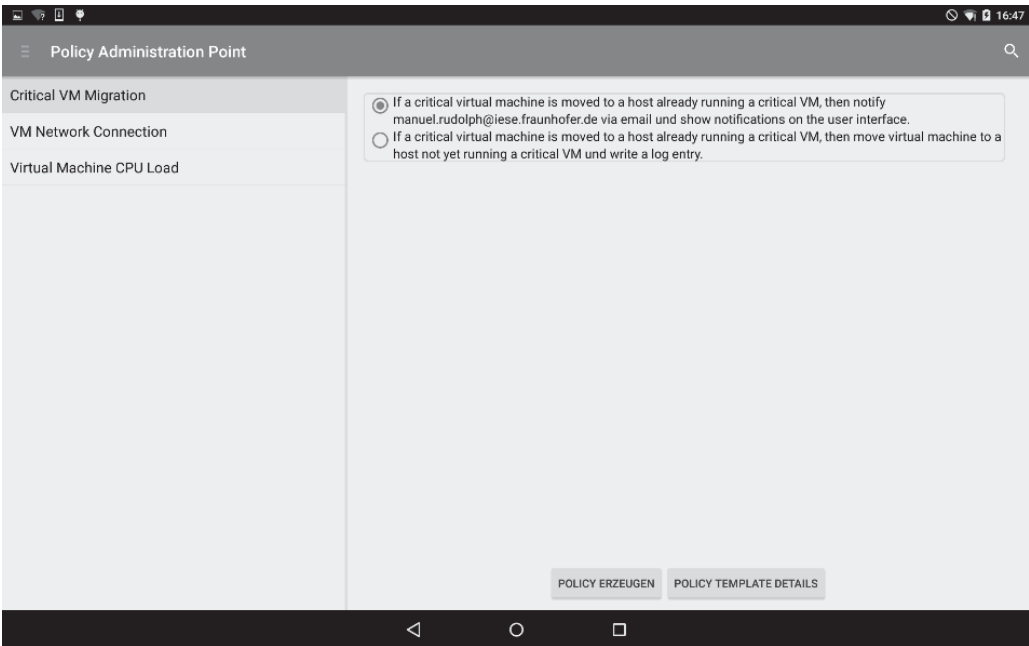


Figure 84: Example PAP Using View Module »Android«, Policy Vocabulary »SECCRIT« and Presentation Module »Default Policies«

In addition, we provided a PAP with the same policy vocabulary and the identical presentation modules for creating a web-based PAP (see Figure 85).

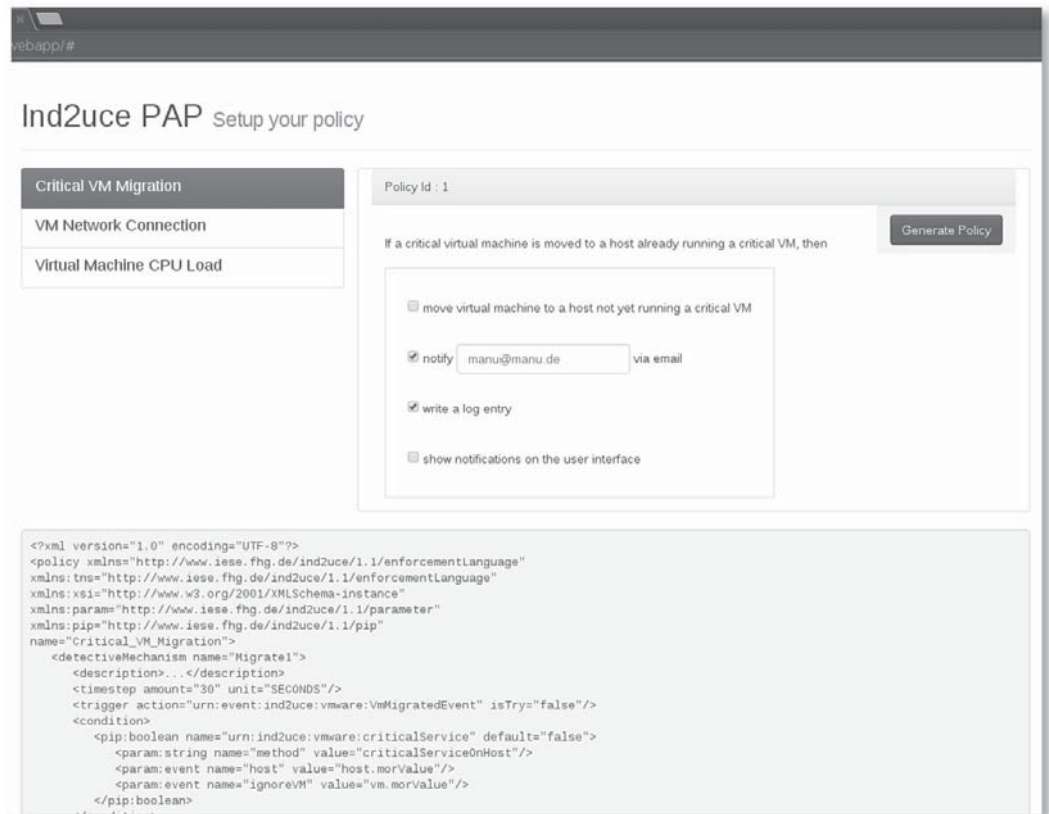


Figure 85: Example PAP Using a Preliminary Version of the View Module »Web«, the Policy Vocabulary »SECCRIT« and the Presentation Module »Default Policies«

## D.2 Example of Policy Template in Policy Vocabulary

```

<policyTemplate id="1" name="Critical VM Migration">
  <description
    value="Tenant A has a VM running a critical infrastructure IT service on a virtual datacenter. The service
    is not allowed to leave a given geolocation or jurisdiction. In case of stored person-related data, different laws
    may apply on this data depending on the geolocation or jurisdiction." />
  <exemplary_instantiation
    value="If VM1 running a critical service of tenant A is about to be moved to a jurisdiction outside the
    EU, then inhibit the movement and notify the tenant infrastructure provider." />
  <asset value="Critical Service" />
  <threat value="Unintended movement of critical infrastructure IT service to another geolocation or
  jurisdiction in which other laws may apply." />
  <filters>
    <filter typeld="1" filterId="1" />
    <filter typeld="2" filterId="1" />
    <filter typeld="3" filterId="1" />
    <filter typeld="4" filterId="1" />
  </filters>
  <defaultInstantiations>
    <defaultInstantiation description="default 1" id="t1_default1">
      <defaultValue ref="t1_countermeasure_move_vm" value="false" />
      <defaultValue ref="t1_countermeasure_notification" value="true" />
      <defaultValue ref="t1_countermeasure_log" value="false" />
      <defaultValue ref="t1_countermeasure_ui" value="true" />
      <defaultValue ref="t1_notification_email" value="manuel.rudolph@iese.fraunhofer.de" />
    </defaultInstantiation>
  </defaultInstantiations>

```

```

</defaultInstantiation>
<defaultInstantiation description="default 1" id="t1_default2">
  <defaultValue ref="t1_countermeasure_move_vm" value="true" />
  <defaultValue ref="t1_countermeasure_notification" value="false" />
  <defaultValue ref="t1_countermeasure_log" value="true" />
  <defaultValue ref="t1_countermeasure_ui" value="false" />
</defaultInstantiation>
</defaultInstantiations>

<slpt>
  <text value="If a critical virtual machine is moved to a host already running a critical VM, then" />
  <selection id="t1_countermeasure_selection" conjunction="and" minSelectedElements="1 "
maxSelectedElements="4">
    <elementGroup id="t1_countermeasure_move_vm" description="move VM to free host"
      longDescription="The virtual machine that was migrated to the unsuitable host will be removed to
a host not yet running a critical VM.">
      <text value="move virtual machine to a host not yet running a critical VM" />
    </elementGroup>
    <elementGroup id="t1_countermeasure_notification" description="email notification"
longDescription="An email notification is sent to the defined recipient.">
      <text value="notify" />
      <variable id="t1_notification_email" type="string" description="email address"
longDescription="Enter the email address to which the notification is sent." />
      <text value="via email" />
    </elementGroup>
    <elementGroup id="t1_countermeasure_log" description="logging" longDescription="writes a log
entry">
      <text value="write a log entry" />
    </elementGroup>
    <elementGroup id="t1_countermeasure_ui" description="UI notification" longDescription="shows a
notification on the user interface">
      <text value="show notifications on the user interface" />
    </elementGroup>
  </selection>
</slpt>

<ilptParts>
  <ilptGroup id="t1_countermeasure_blocks">
    <ilptBlock id="t1_countermeasure_blocks_move_vm" use="t1_countermeasure_move_vm">
      <llxsdInd2uce:executeAction name="urn:action:ind2uce:vmware:MigrateVM">
        <param:string name="priority" value="highPriority" />
        <param:event name="vm.morType" value="vm.morType" />
        <param:event name="vm.morValue" value="vm.morValue" />
        <param:string name="host.morType" value="HostSystem" />
        <pip:string name="urn:ind2uce:vmware:getFreeHost" paramName="host.morValue"
default="host-38439">
          <param:string name="method" value="getFreeHost" />
        </pip:string>
      </llxsdInd2uce:executeAction>
    </ilptBlock>
    <ilptBlock id="t1_countermeasure_blocks_notification" use="t1_countermeasure_notification">
      <llxsdInd2uce:executeAction name="urn:action:ind2uce:vmware:sendMail">
        <param:string name="msgPlain"
value="Dear Customer, \n\nwe detected a policy violation that critical services were migrated
to the same physical host! \nCompensating actions have been performed. \n\nBest Regards, \nIND2UCE" />
        <param:string name="msgHTML"
value="&#x3C;/html&#x3E;&#x3C;/head&#x3E;&#x3C;/title&#x3E;IND2UCE Policy Violation
Mail&#x3C;/title&#x3E;&#x3C;/head&#x3E;&#x3C;/body&#x3E;Dear Customer,
&#x3C;br/&#x3E;&#x3C;br/&#x3E;we detected a policy violation that two &#x3C;b&#x3E;critical

```

```

services&#x3C;/b&#x3E; were migrated to the &#x3C;b&#x3E;same physical host&#x3C;/b&#x3E;!
&#x3C;br/&#x3E;Compensating actions have been performed.&#x3C;br/&#x3E;&#x3C;br/&#x3E;Best Regards,
&#x3C;br/&#x3E;IND2UCE&#x3C;br/&#x3E;&#x3C;br/&#x3E;&#x3C;br/&#x3E;Powered by
ind2uce&#x3C;br/&#x3E;&#x3C;img width=&#x22;200&#x22; height=&#x22;50&#x22;
id=&#x22;Picture_x0020_1&#x22;
src=&#x22;cid:image001.jpg@01D04148.7350F2C0&#x22;&#x3E;&#x3C;/body&#x3E;&#x3C;/html&#x3E;" />
    <param:boolean name="ind2uceLogo" value="true" />
    <param:string name="subject" value="Policy Violation" />
    <param:string name="recipient" value="$ref:t1_notification_email" />
  </llxsdInd2uce:executeAction>
</ilptBlock>
<ilptBlock id="t1_countermeasure_blocks_log" use="t1_countermeasure_log">
  <llxsdInd2uce:executeAction name="urn:action:ind2uce:vmware:log">
    <param:string name="msg" value="Two cricitcal services have been migrated to the same
physical host. Compensating actions are running." />
  </llxsdInd2uce:executeAction>
</ilptBlock>
<ilptBlock id="t1_countermeasure_blocks_ui" use="t1_countermeasure_ui">
  <llxsdInd2uce:executeAction name="urn:action:ind2uce:http_get">
    <param:string name="paramName" value="msg" />
    <param:string name="paramValue" value="Two cricitcal services have been migrated to the
same physical host. Compensating actions are running." />
    <param:string name="urlPrefix" value="http://212.9.140.33:8081" />
  </llxsdInd2uce:executeAction>
</ilptBlock>
</ilptGroup>
</ilptParts>

<ilpt>
  <ind2ucePolicy>
    <policy name="Critical_VM_Migration">
      <ind2uce:detectiveMechanism name="Migrate1 "
ilptGroupReference="t1_countermeasure_blocks">
        <ind2uce:description>...</ind2uce:description>
        <ind2uce:timestep amount="30" unit="SECONDS" />
        <ind2uce:trigger action="urn:event:ind2uce:vmware:VmMigratedEvent" isTry="false" />
        <ind2uce:condition>
          <pip:boolean name="urn:ind2uce:vmware:criticalService" default="false">
            <param:string name="method" value="criticalServiceOnHost" />
            <param:event name="host" value="host.morValue" />
            <param:event name="ignoreVM" value="vm.morValue" />
          </pip:boolean>
        </ind2uce:condition>
      </ind2uce:detectiveMechanism>
    </policy>
  </ind2ucePolicy>
</ilpt>
</policyTemplate>

```



## Appendix E Case Study: »BeSure«

### E.1 Excerpt of »BeSure« Study Results

Table 42: Asset »Job Data«

|                           |  |
|---------------------------|--|
| Asset ID                  | A1   |
| Asset                     | Job data (e.g., professional data and client data)   |
| Data Owner                | Clients  |
| Example Use Case          | <ul style="list-style-type: none"> <li>• Printing and dispatch of payroll invoices in the production area</li> <li>• Clarification of error situations by service employees</li> </ul> |
| Prioritization Properties | Monetary value of asset: high (€€€)<br>Sensitivity of asset: highly confidential   |
| Legal Regulations         | Professional law of tax consultants, StGB §203 (violation of private secrets) and BaFin (confidentiality regulations)  |

Table 43: Threats for Asset »Job Data«

|                        |   |
|------------------------|---|
| Threat ID              | T1-T3   |
| Related Asset ID       | A1  |
| Related Asset          | Job Data  |
| Attackers              | Data theft  |
| Top 3 Threats          | T1: Data theft <ul style="list-style-type: none"> <li>• probability: likely (medium)</li> <li>• damage: costly (medium)</li> </ul> T2: Insufficient deletion <ul style="list-style-type: none"> <li>• probability: likely (medium)</li> <li>• damage: costly (medium)</li> </ul> T3: Manipulation of payment flows <ul style="list-style-type: none"> <li>• probability: almost impossible (low)</li> <li>• damage: existence-threatening (high)</li> </ul> |
| Other threats          | <ul style="list-style-type: none"> <li>• Use of not permitted communication methods</li> <li>• External attackers gain access to job data for blackmailing</li> <li>• Software bugs</li> <li>• Misdirection / misdelivery</li> <li>• External technician copies data (e.g. remote support)</li> <li>• Industrial espionage to obtain internal information from clients (e.g. stock market speculation)</li> </ul>   |
| Existing Documentation | not available   |



Table 44: Asset »Public Data«

|                           |   |
|---------------------------|---|
| Asset ID                  | A9  |
| Asset                     | Public data (e.g., marketing material, website, product descriptions) |
| Data Owner                | Marketing and press departments                                       |
| Example Use Case          | The editor creates new advertising materials                          |
| Prioritization Properties | Monetary value of asset: low (€)<br>Sensitivity of asset: public      |
| Legal Regulations         | German law TMG and GDPR   |

Table 45: Threats for Asset »Public Data«

|                        |   |
|------------------------|---|
| Threat ID              | T7-T9   |
| Related Asset ID       | A9  |
| Related Asset          | Public Data   |
| Attackers              | Script kiddie, Accidental attacker, Rival   |
| Top 3 Threats          | <p>T7: Non-compliance with legal regulations</p> <ul style="list-style-type: none"> <li>probability: likely (medium)</li> <li>damage: costly (medium)</li> </ul> <p>T8: Falsification of information</p> <ul style="list-style-type: none"> <li>probability: likely (medium)</li> <li>damage: costly (medium)</li> </ul> <p>T9: Distributed Denial of Service</p> <ul style="list-style-type: none"> <li>probability: likely to permanently (medium-high)</li> <li>damage: costly (medium)</li> </ul> |
| Other threats          | <ul style="list-style-type: none"> <li>Unintentional publication of internal information</li> <li>Blackmailing through DDoS</li> <li>Reputation gain through in hacker community through information falsification</li> <li>Missing/inadequate data protection declaration</li> <li>Release of internal data</li> </ul>   |
| Existing Documentation | <ul style="list-style-type: none"> <li>Risk management at company level <ul style="list-style-type: none"> <li>Information security risks</li> <li>HighLevel</li> </ul> </li> <li>Threat Modelling in individual projects <ul style="list-style-type: none"> <li>Without a fixed schema</li> <li>Rapid Risk Analysis</li> </ul> </li> </ul>   |

Table 46: Countermeasures for Threat »T5: Intentional Tampering«

| Countermeasures for threat:<br>T5: Intentional tampering    |
|---|
| Sign emails by default                                      |
| Documented, analyzed process                                |
| Warning message on unsigned emails (inbox and outbox)       |
| Protective mechanisms at email client (no authorized usage) |

Table 47: Countermeasures for Threat »T6: Unencrypted Sending of Confidential Emails«

| Countermeasures for threat:<br>T6: Unintended disclosure to third parties (unencrypted sending or wrong recipient) |
|--|
| Encrypt emails by default  |
| Delayed sending of emails (possibility to revoke emails)   |
| Recurring sensitization of employees (intranet, training, ...)   |
| Prevention of sending with data loss prevention mechanism  |

Table 48: Policy Template »Secure Email Receiving«

| ID                     | Policy Template Name   | Asset  | Target System           | Security/Privacy Goal      |
|------------------------|------------------------|--|-------------------------|----------------------------|
| 2                      | Secure email receiving | Communication Data   | Email client and server | Confidentiality, integrity |
| Policy Template Syntax |                        | If [any employee   <employee>   <employee role>] receives an email, which [is not encrypted   is not digitally signed   contains attachments   contains sensitive information   was not scanned for viruses   was sent by an unknown sender]*, then warn the user. |                         |                            |
| Description            |                        | Employees often communicate via email with internal as well as external recipients. This communication must be protected because email content as well as attachments can contain sensitive information. This template facilitates the control of email receipt.   |                         |                            |
| Threat                 |                        | Information leakage or manipulation of sensitive information   |                         |                            |
| Example Instantiation  |                        | If service employees receive an email, which is not digitally signed, contains attachments, and was not scanned for viruses, then warn the user.   |                         |                            |



## Appendix F Case Study: »Digital Villages«

### F.1 Excerpt of »Digital Villages« Study Results

A screenshot of a policy specification with the specification paradigm »template instantiation« is presented in Figure 86. The specification with the »default policies« is depicted in Figure 87. The use of the specification paradigm »wizard« is demonstrated in Figure 88. Figure 89 shows the specification paradigm »security levels« based on the same policy vocabulary.

The screenshot shows a web interface for policy specification. On the left is a sidebar with a list of items: 'BestellBar: Forwarding of order data', 'LieferBar: Acceptance of a delivery', 'LieferBar: Information prior to acceptance of the delivery request', 'LieferBar: Displaying the storage location for packages', 'DorfFunk: Help requests and offers' (which is highlighted), and 'Scientific Evaluation'. The main content area is titled 'My help requests and offers can be viewed'. It contains a form with three radio buttons: 'by every citizen', 'by my friends', and 'by citizen with at least the trust level' (which is selected). To the right of the selected radio button is a dropdown menu with the text 'Please select' and a list of options: 'gold', 'silver', and 'bronze'. Below this, there is a text input field with the placeholder 'my complete name' and a dropdown menu with the text 'only street' and a list of options: 'my complete name', 'my street', and 'my address'. To the right of this is another dropdown menu with the text 'only the date of the preferred appointment'.

Figure 86: Example PAP Using View Module »Web«, Policy Vocabulary »Digital Villages« and Presentation Module »Template Instantiation«

The screenshot shows a web interface for policy specification. On the left is a sidebar with a list of items: 'BestellBar: Forwarding of order data', 'LieferBar: Acceptance of a delivery', 'LieferBar: Information prior to acceptance of the delivery request' (which is highlighted), 'LieferBar: Displaying the storage location for packages', 'DorfFunk: Help requests and offers', and 'Scientific evaluation'. The main content area contains a list of four radio buttons, each followed by a text description of a policy option. The second radio button is selected. The descriptions are: 'Before accepting the delivery request, the deliverer does not obtain my name and does only obtain the following parts of my address: city. Furthermore, he will not be informed about any delivery details.', 'Before accepting the delivery request, the deliverer does obtain my name and does only obtain the following parts of my address: zip code and city. Furthermore, he will only be informed about the parcel size.', 'Before accepting the delivery request, the deliverer does obtain my name and does only obtain the following parts of my address: street and city. Furthermore, he will only be informed about the parcel size and my desired delivery time.', and 'Before accepting the delivery request, the deliverer does obtain my name and does obtain my full address. Furthermore, he will be informed about all delivery details.'

Figure 87: Example PAP Using View Module »Web«, Policy Vocabulary »Digital Villages« and Presentation Module »Default Policies«

### Scientific Evaluation

In order to optimise the quality and benefits of digitisation in rural areas, all data will be scientifically evaluated. You can decide whether and how your data may be used.

My data will be ...

☐

excluded from scientific evaluations

☒

permitted for scientific evaluation

☐ if my name has been made anonymous

☒ if all my personal data have been made anonymous

Previous Step

1

2

3

4

5

6

7

8

Next Step

Figure 88: Example PAP Using View Module »Web«, Policy Vocabulary »Digital Villages« and Presentation Module »Wizard«

### Privacy Level Blue

☐

When a merchant forwards my order data to an advertisement company, I want to be informed and my personal data needs to be anonymized.

All citizens may see and accept my delivery requests.

Before accepting the delivery request, the deliverer does obtain my name and does only obtain the following parts of my address: street and city. Furthermore, he will only be informed about the parcel size and my desired delivery time.

After acceptance of the delivery order, the supplier shall be notified of the secret storage location 100 meters from the place of delivery.

My help requests and offers can be viewed by my friends and by citizens with at least the trust level gold. Before accepting the help request or offer, they are allowed to look at not my name, only street and city of my address and not the preferred appointment.

My data will be excluded from scientific evaluations.

### Privacy Level Purple

☐

When a merchant forwards my order data to an advertisement company, I want to be informed.

Only my friends and deliverers with a trust level of at least silver may see and accept my delivery requests.

Before accepting the delivery request, the deliverer does not obtain my name and does only obtain the following parts of my address: city. Furthermore, he will not be informed about any delivery details.

After acceptance of the delivery order, the supplier shall be notified of the secret storage location immediately.

My help requests and offers can be viewed by my friends and by citizens with at least the trust level gold. Before accepting the help request or offer, they are allowed to look at my complete name, not my address and only date and daytime of the preferred appointment.

My data will be permitted for scientific evaluation if all my personal data have been made anonymous.

### Privacy Level Orange

☐

When a merchant forwards my order data to an advertisement company, I forbid that.

Only my friends and deliverers with a trust level of at least gold may see and accept my delivery requests.

Figure 89: Example PAP Using View Module »Web«, Policy Vocabulary »Digital Villages« and Presentation Module »Security Levels«

## Appendix G Policy Specification Experiment

### G.1 Invitation Email

Hello,

Thank you for participating in our experiment!

You will find instructions attached to this mail. Please print them out, as you will need them several times.

On the first page you will find the link to the experiment website and your participation number.

You can use all standard browsers to open the link (Google Chrome, Firefox, Microsoft Edge, Internet Explorer).

Please open the link from a computer connected to a keyboard. (The website is not suitable for tablets and smartphones).

Please make sure your speakers are turned on.

Let's start the experiment now!

If you think after the experiment that you know someone else who would like to participate, I will gladly send you further invitations :)

Best Regards

## G.2 Experiment Handout



Figure 90: Policy Specification Experiment - Handout Page 1

Try to put yourself in the scenario described in the video!

Since you and many of your fellow citizens have a great sense of community and an urge to digitize, the presented services are now widely used. Many of your personal data are digitally processed. In addition to your name and address, these include details of your orders (ordered goods), deliveries (delivery address, location, delivery time window), exchanges (details of offers and enquiries) and complaints. Depending on your activities within the apps, other citizens of the village may be able to see this information partially or completely. You do not want that. The three services provide you with privacy settings that you can configure to suit your privacy needs. That is great, and you can start adjusting the settings right away.

There are several ways to set up your privacy settings on your computer. All these specification types have their advantages and disadvantages. In this experiment you will use and evaluate four different specification types. In the following, we give you concrete privacy demands, which you should configure all in each of the four specification types.

- When I place an order in the BestellBar app, I do not under any circumstances want to receive advertising from other providers that refers to the ordered product. They may not use my data.
- I do not like that all the citizens in my village know where I order goods. Therefore, only people who are considered to be as trustworthy as possible and my friends should be able to view my delivery requests in the LieferBar app.
- Before someone accepts my order in the LieferBar app, this person may know my name, but not exactly where I live. The name of my village with postal code would be ok. The potential deliverer may also know the dimensions of the package. However, further information on the address and the parcel should only be provided to the person after acceptance of the delivery request.
- If I am not at home, the delivery may be deposited at my house. If a person has accepted my delivery order, he/she is only allowed to find out via the App as close as possible to my front door where the storage location is.
- I want everyone in the DorfFunk to see my help requests and offers, but I don't want them to know that they are from me. Therefore, the other users should not be able to see my name or my exact address. My place of residence with zip code and the concrete day on which I need help should be sufficient. Further details, such as the exact address and the proposed time of day, can be seen after accepting my offer.
- I think it is important that scientists contribute to society through research. Therefore, I am willing to provide them my data for these purposes as long as they do not use my name.

Figure 91: Policy Specification Experiment - Handout Page 2



## G.3 Screenshots of Experiment

This section shows screenshots of all steps in the policy specification experiment.

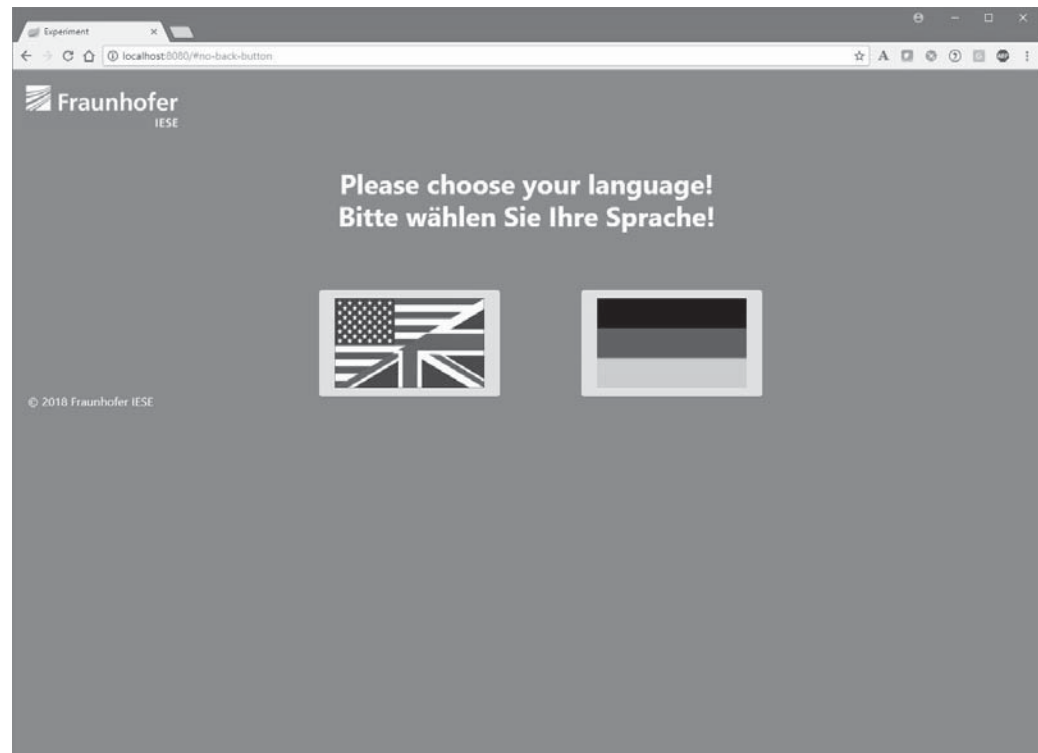


Figure 92: Screenshot - Language Selection

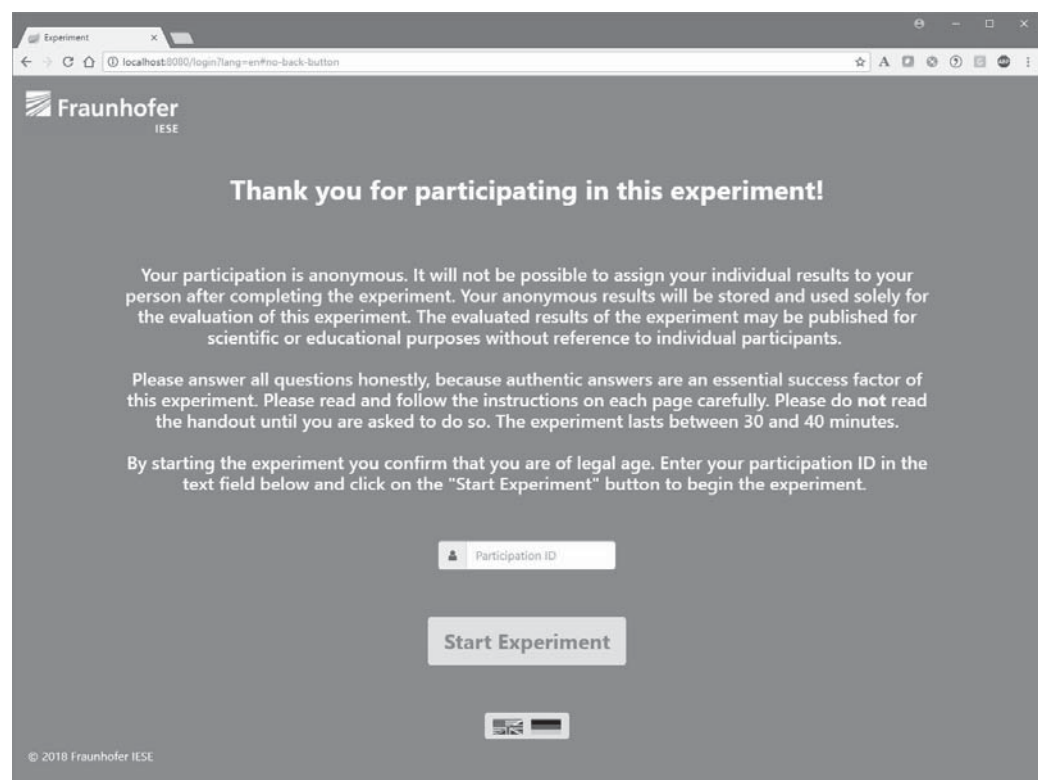


Figure 93: Screenshot - Login Page

The screenshot shows a web browser window with the URL `localhost:8080/p1?userId=stb6u7z7#no-back-button`. The page is titled 'Demographic questions' and features a progress bar at the top right. The form contains three sections: 'What is your gender?' with radio buttons for Male, Female, and Other; 'How old are you?' with a text input field labeled 'Enter your age'; and 'What is your highest level of education?' with radio buttons for Secondary school leaving certificate, Higher education entrance qualification, University degree, Doctoral degree, and Other (with a text input field for definition). A 'Next' button is at the bottom left. The footer reads '© 2018 Fraunhofer IESE'.

Figure 94: Screenshot - Demographic Questions

The screenshot shows a web browser window with the URL `localhost:8080/p2?userId=stb6u7z7#no-back-button`. The page is titled 'Relation to Fraunhofer IESE' and features a progress bar at the top right. The form contains three sections: 'I am a:' with radio buttons for Scientific employee at Fraunhofer IESE, Non-scientific employee at Fraunhofer IESE, Student working for Fraunhofer IESE or writing a thesis there, and Other; 'Do you have experience with IND<sup>2</sup>UCE?' with radio buttons for I have hands-on experience with IND<sup>2</sup>UCE, I know what IND<sup>2</sup>UCE is, and I do not know what IND<sup>2</sup>UCE is; and 'Have you ever specified a security policy with IND<sup>2</sup>UCE?' with radio buttons for Yes and No. A 'Next' button is at the bottom left. The footer reads '© 2018 Fraunhofer IESE'.

Figure 95: Screenshot - Relation to Fraunhofer IESE

Experiment

localhost:8080/p3?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

### Experience questions

**Please rate your knowledge of IT security measures!**  
IT security can be defined as the protection of your own systems (e.g., your computer) and data (e.g., your credit card number) against misuse and data loss. Various IT security measures are available, such as encryption (of emails and hard drives), secure passwords (individual passwords for each service), or antivirus scanner and firewalls. Most of those measures can be configured individually.

Please indicate your tendency to one of the two statements. If you strongly agree to one statement, click on the option closest to this statement.

I do not have knowledge in IT security ☐ ☐ ☐ ☐ ☐ I have expert knowledge in IT security

**How extensively are you using IT security measures for protecting your data and systems?**

Please indicate your tendency to one of the two statements. If you strongly agree to one statement, click on the option closest to this statement.

I do not voluntarily use IT security measures ☐ ☐ ☐ ☐ ☐ I use IT security measures very extensively

**How often do you use web services on average?**

|  | Every day             | On 4 to 6 days a week | On 1 to 3 days a week | On 1 to 3 days a month | Less frequently or never |
|--|-----------------------|-----------------------|-----------------------|------------------------|--------------------------|
| Social Networks (e.g. Facebook, Google+, Xing)         | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/>    |
| Communication Service (e.g. Email, Whatsapp, Snapchat) | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/>    |
| Search Engines (e.g. Google, Bing)                     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/>    |
| Online Banking   | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/>    |
| Online Retail (e.g. Amazon, Ebay)                      | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/>  | <input type="radio"/>    |

Next

Figure 96: Screenshot - Relation to Fraunhofer IESE

Experiment

localhost:8080/p4?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

### Motivation questions

**How often do you update the security and privacy settings of each online services on average?**

☐ Each time I use the online service

☐ Several times a month

☐ Several times a year

☐ About once a year

☐ Once, when using the online service the first time

☐ Less frequently or not at all

**What are the main reasons that prevent you from updating your security and privacy settings more often?**

Multiple answers can be chosen.

☐ Too time consuming

☐ Too complicated

☐ That does **not** interest me

☐ I did **not** know I had to do this

☐ I do **not** feel competent to do it appropriately

☐ I just forget to do it regularly

☐ I do **not** believe this is necessary

☐ I do **not** believe that my privacy settings are really enforced

☐ It is boring

☐ Other reasons:

Next

Figure 97: Screenshot - Motivation Question

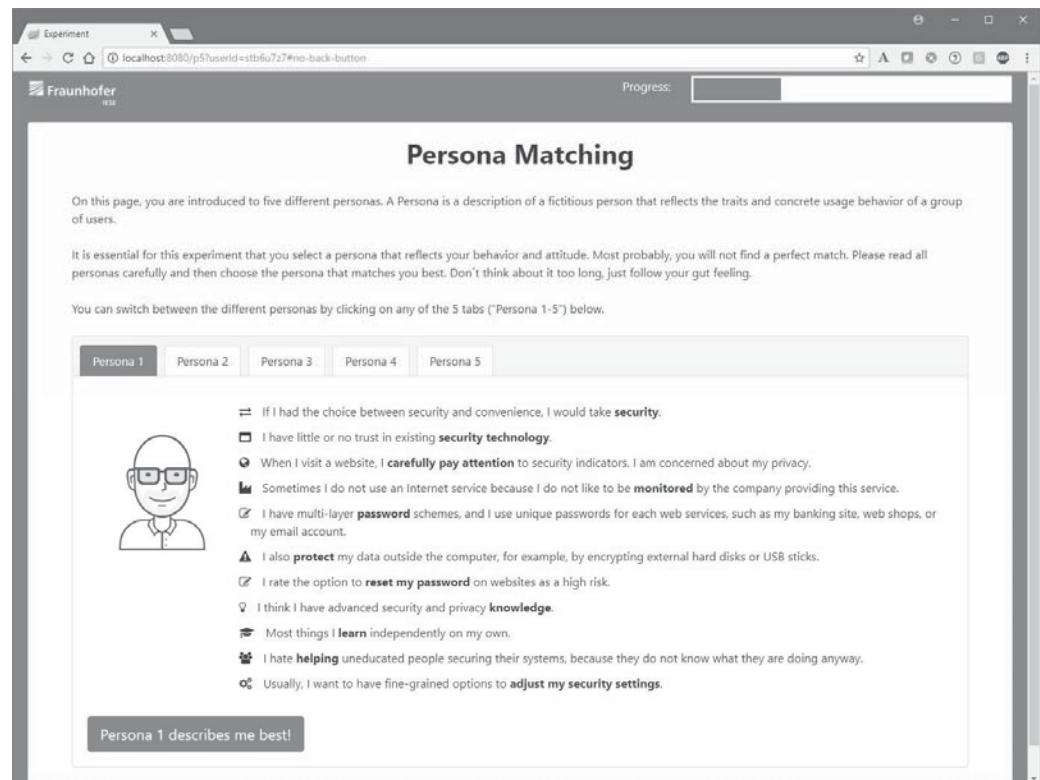


Figure 98: Screenshot - Persona Fundamentalist

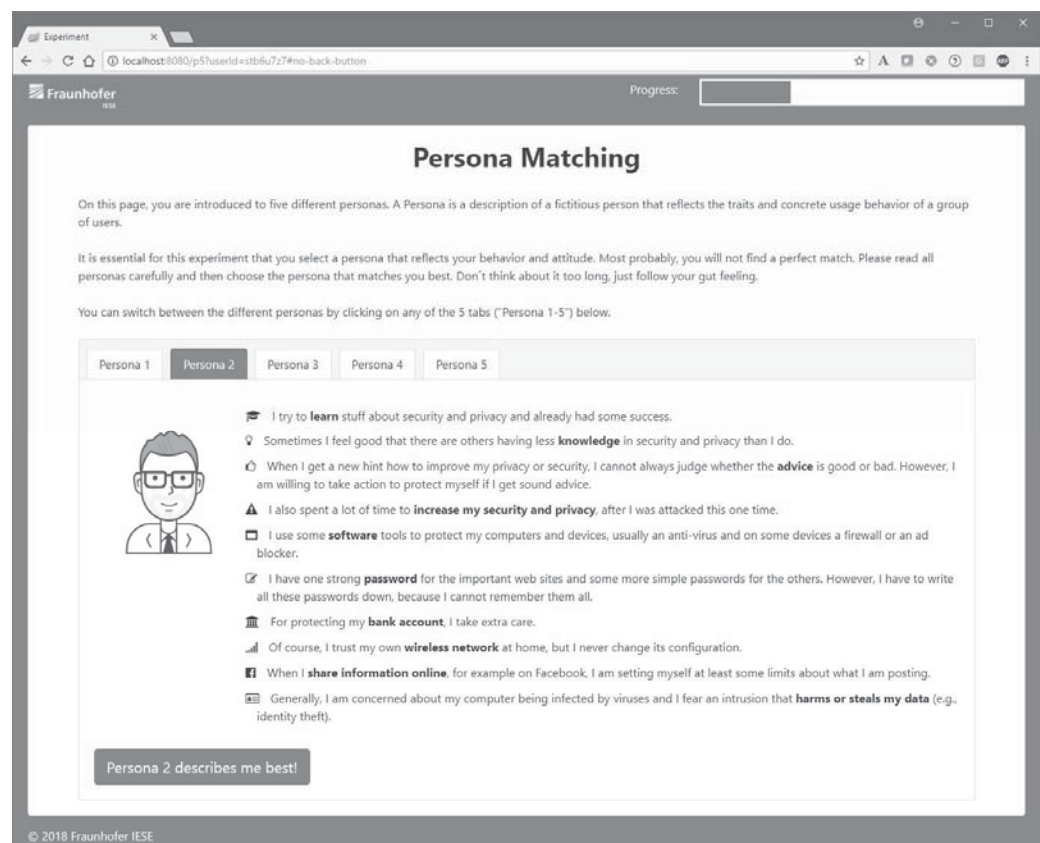


Figure 99: Screenshot - Persona Amateur

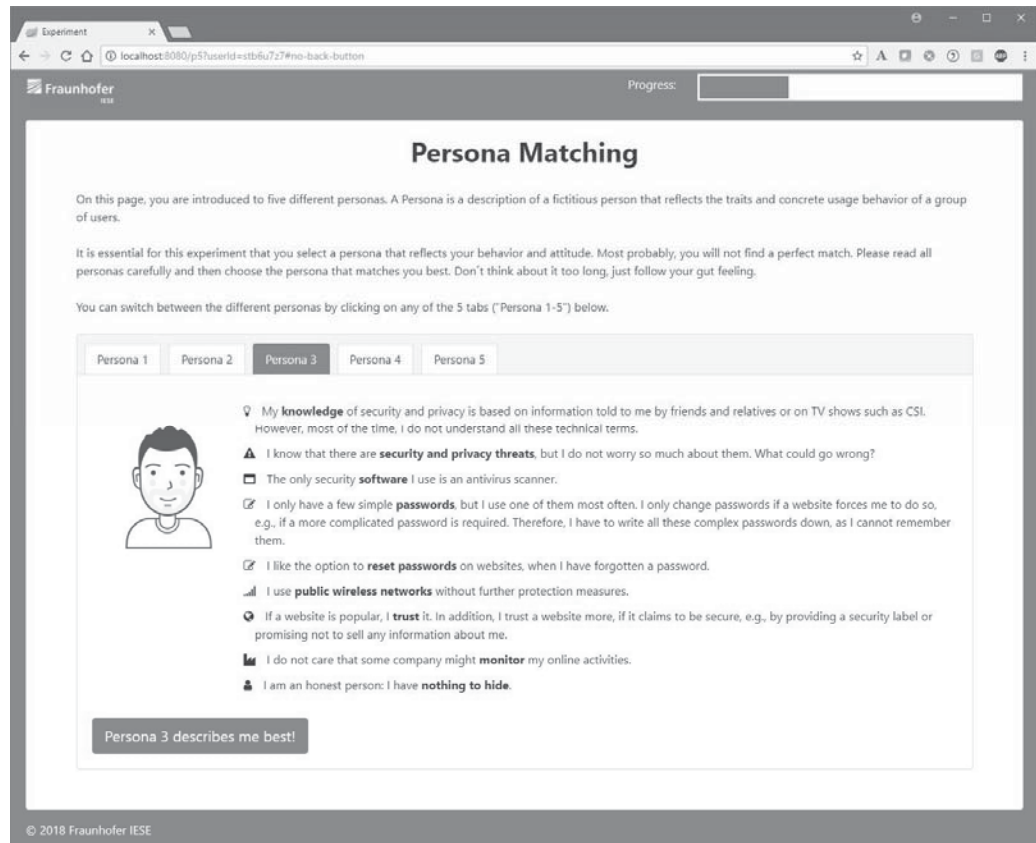


Figure 100: Screenshot - Persona Marginally Concerned

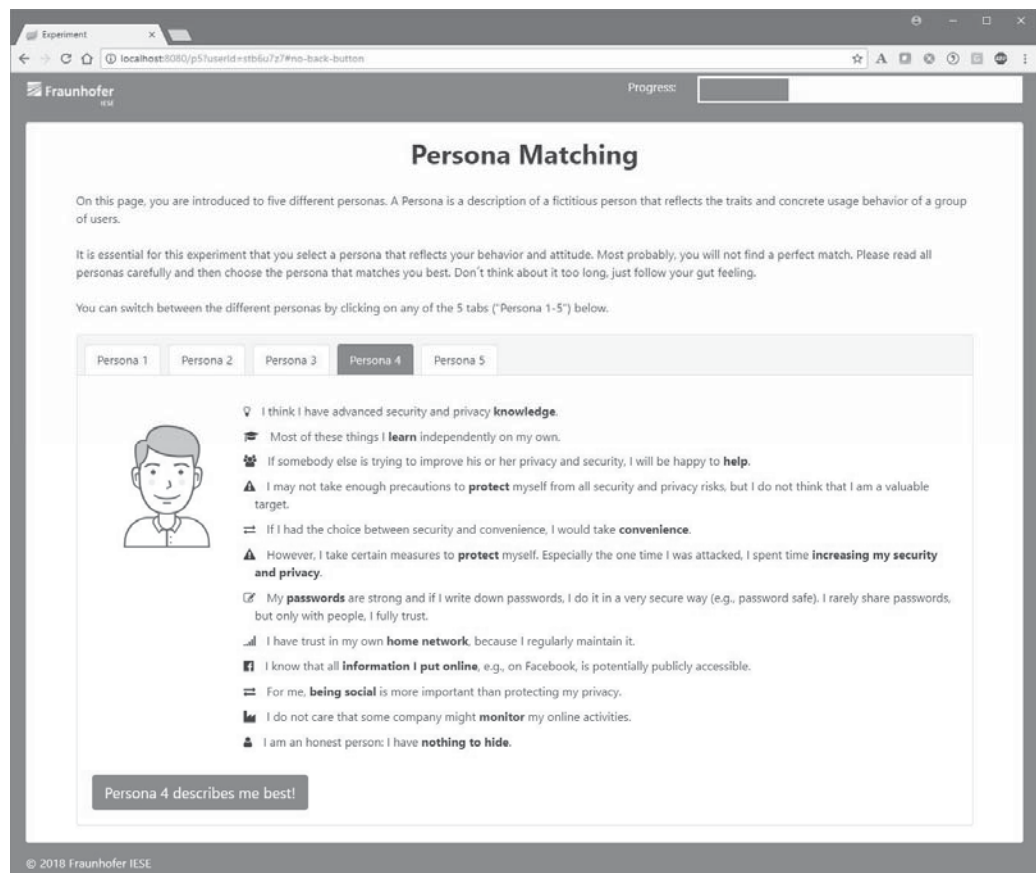


Figure 101: Screenshot - Persona Lazy Expert

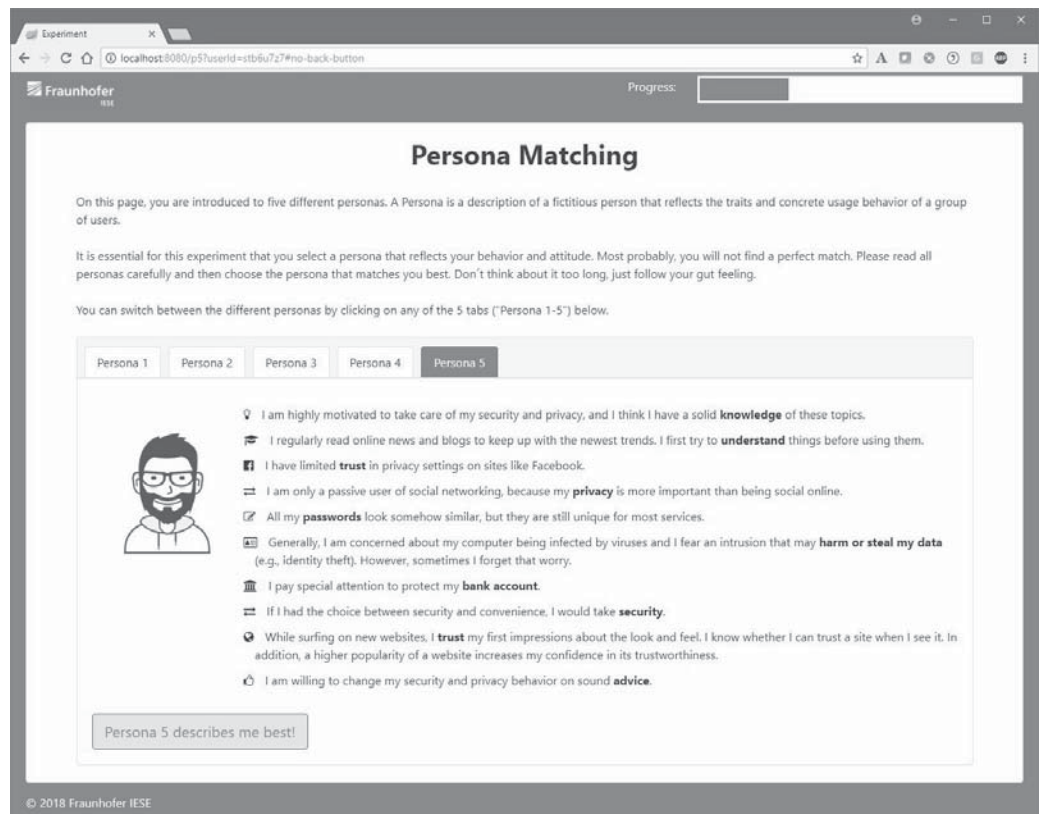


Figure 102: Screenshot - Persona Technician

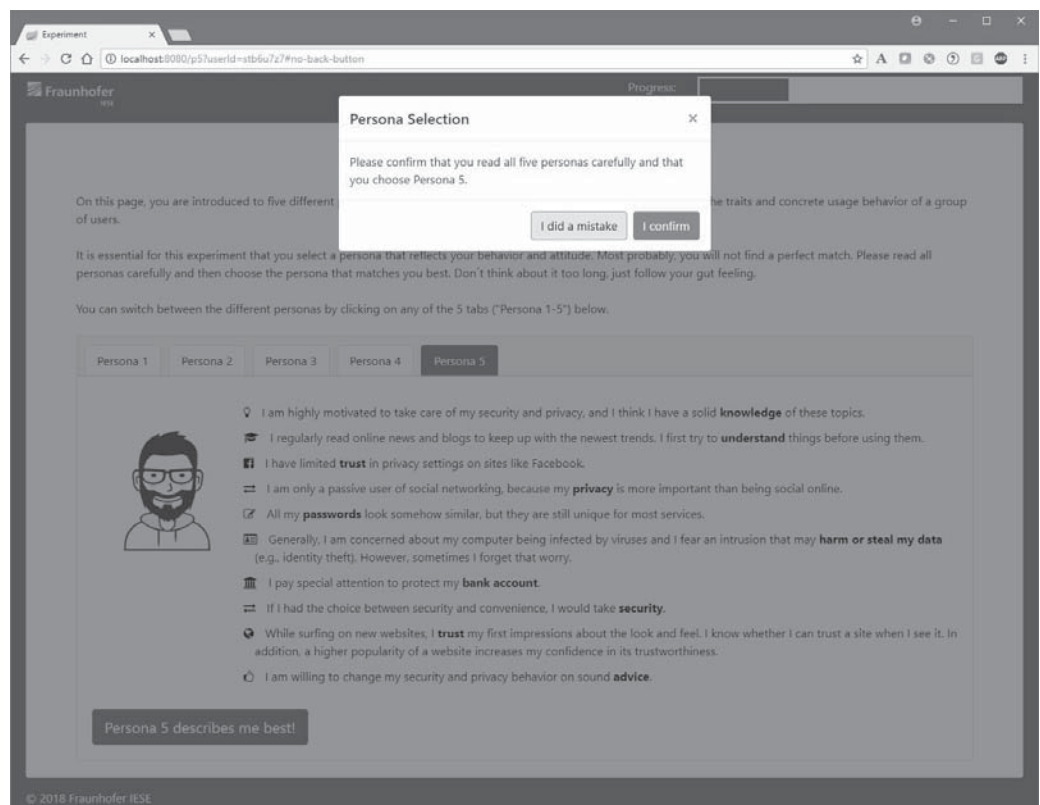


Figure 103: Screenshot - Persona Confirmation



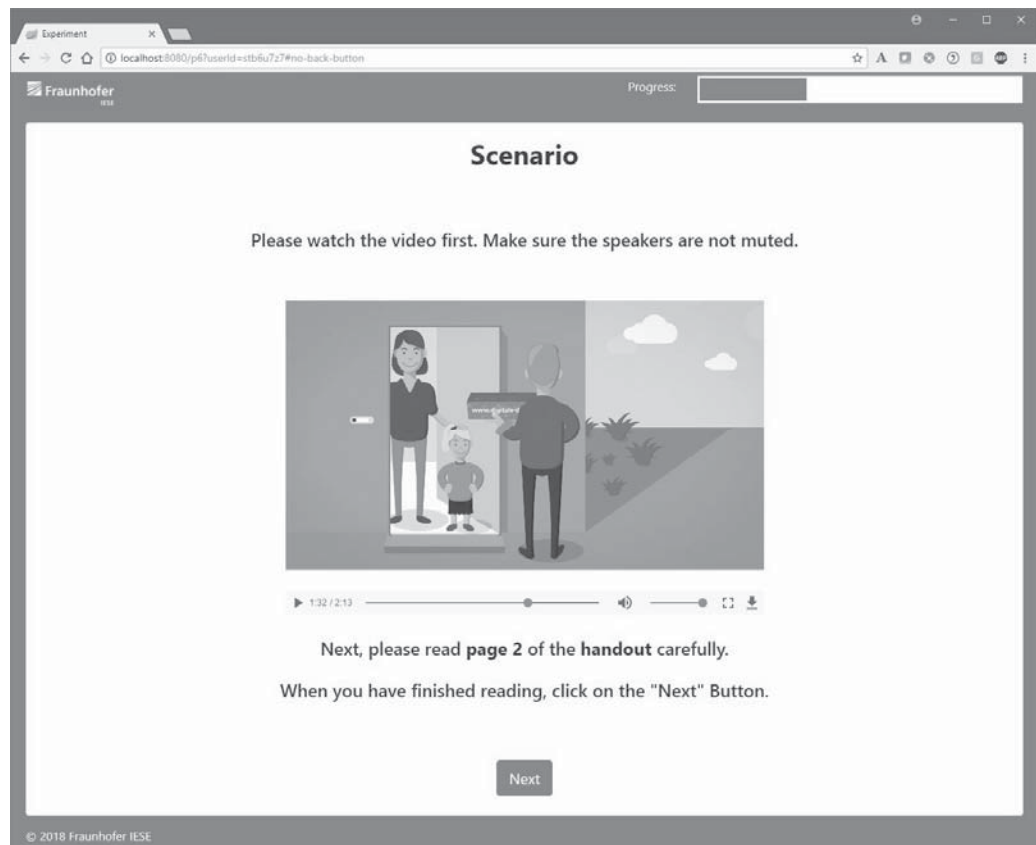


Figure 104: Screenshot - Scenario

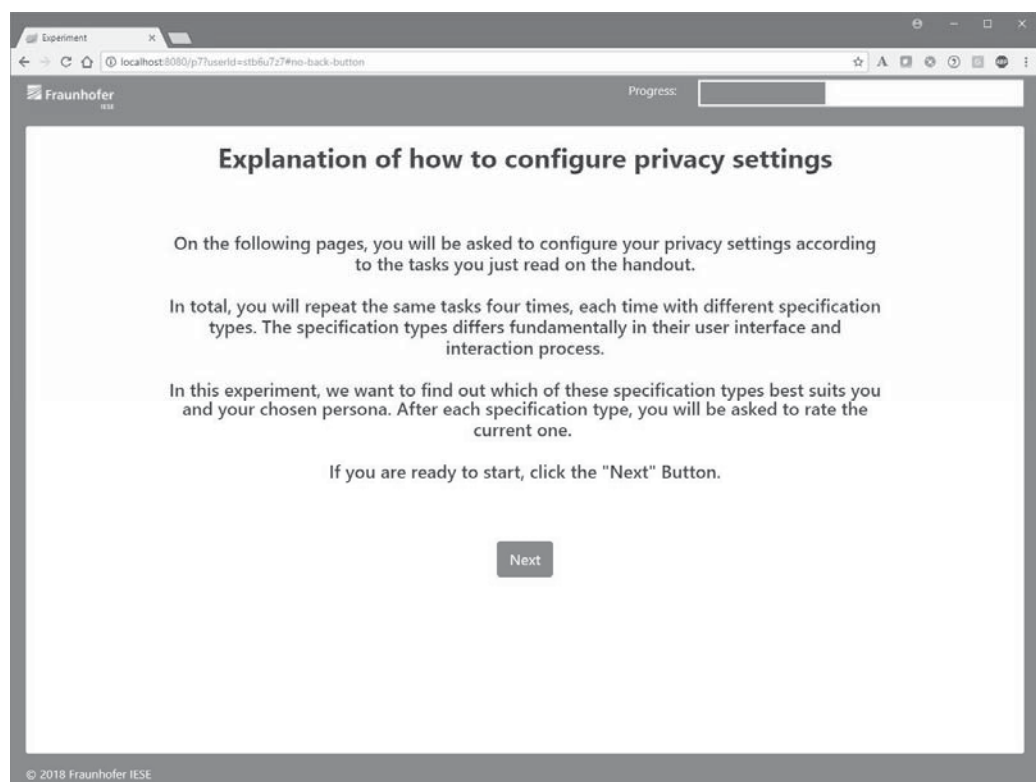


Figure 105: Screenshot - Specification Explanation

Experiment

localhost:8080/p8?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

## Specification Type 1: Templates

The blue box contains the first specification type. Please try to configure all six privacy demands of the handout.

- BestellBar: Forwarding of order data
- LieferBar: Acceptance of a delivery
- LieferBar: Information prior to acceptance of the delivery request
- LieferBar: Displaying the storage location for packages
- Dorffunk: Help requests and offers
- Scientific Evaluation

When a merchant forwards my order data to an advertisement company,

I want to be informed and my personal data needs to be anonymized

Select your option

I want to be informed

I want to be informed and my personal data needs to be anonymized

I forbid that

I configured all six privacy demands

© 2018 Fraunhofer IESE

Figure 106: Screenshot - Specification Type: Template 1

Experiment

localhost:8080/p8?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

## Specification Type 1: Templates

The blue box contains the first specification type. Please try to configure all six privacy demands of the handout.

- BestellBar: Forwarding of order data
- LieferBar: Acceptance of a delivery
- LieferBar: Information prior to acceptance of the delivery request
- LieferBar: Displaying the storage location for packages
- Dorffunk: Help requests and offers
- Scientific Evaluation

All citizens

Only

my friends

deliverers with a trust level of at least gold

may see and accept my delivery requests.

I configured all six privacy demands

© 2018 Fraunhofer IESE

Figure 107: Screenshot - Specification Type: Template 2



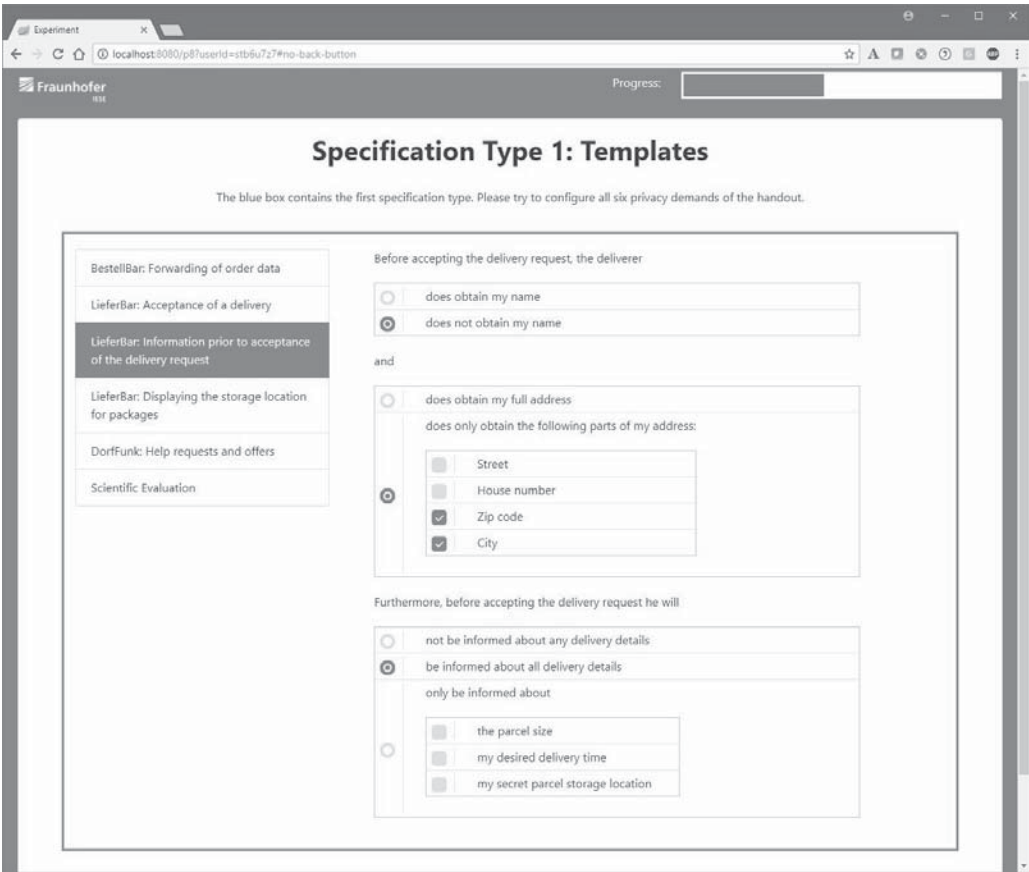


Figure 108: Screenshot - Specification Type: Template 3

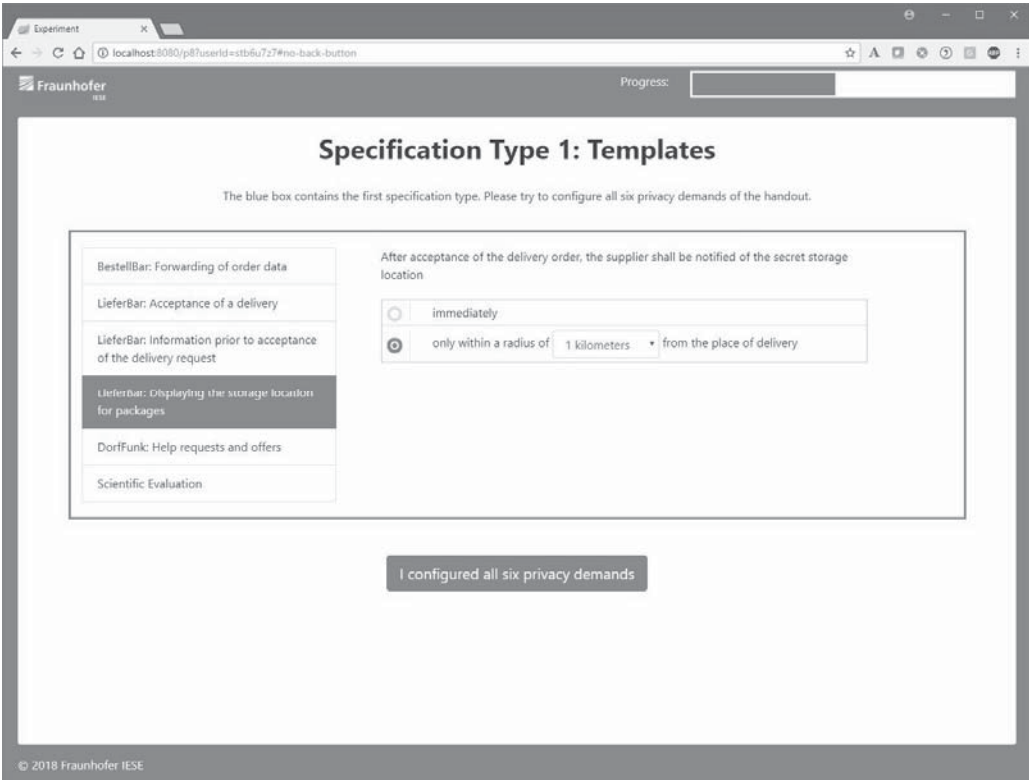


Figure 109: Screenshot - Specification Type: Template 4

Experiment

localhost:8080/p8?userId=stb6u7z7#no-back-button

Fraunhofer IES

Progress:

## Specification Type 1: Templates

The blue box contains the first specification type. Please try to configure all six privacy demands of the handout.

BestellBar: Forwarding of order data

LieferBar: Acceptance of a delivery

LieferBar: Information prior to acceptance of the delivery request

LieferBar: Displaying the storage location for packages

**Dorffunk: Help requests and offers**

Scientific Evaluation

My help requests and offers can be viewed

☐ by every citizen

☐ by my friends

☒ by citizen with at least the trust level

Before accepting the help request or offer, they are allowed to look at

and  and

I configured all six privacy demands

© 2018 Fraunhofer IES

Figure 110: Screenshot - Specification Type: Template 5

Experiment

localhost:8080/p8?userId=stb6u7z7#no-back-button

Fraunhofer IES

Progress:

## Specification Type 1: Templates

The blue box contains the first specification type. Please try to configure all six privacy demands of the handout.

BestellBar: Forwarding of order data

LieferBar: Acceptance of a delivery

LieferBar: Information prior to acceptance of the delivery request

LieferBar: Displaying the storage location for packages

Dorffunk: Help requests and offers

**Scientific Evaluation**

My data will be

☐ excluded from scientific evaluations

☒ permitted for scientific evaluation

☐ if my name has been made anonymous

☒ if all my personal data have been made anonymous

I configured all six privacy demands

© 2018 Fraunhofer IES

Figure 111: Screenshot - Specification Type: Template 6

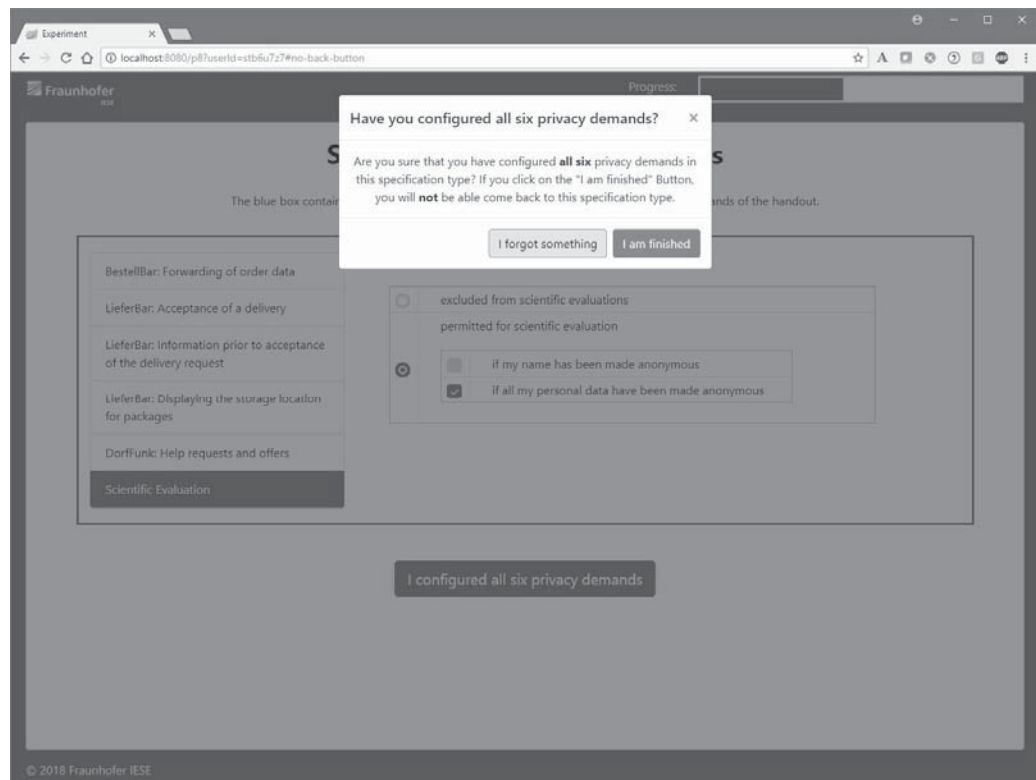


Figure 112: Screenshot - Specification Type: Template Confirmation

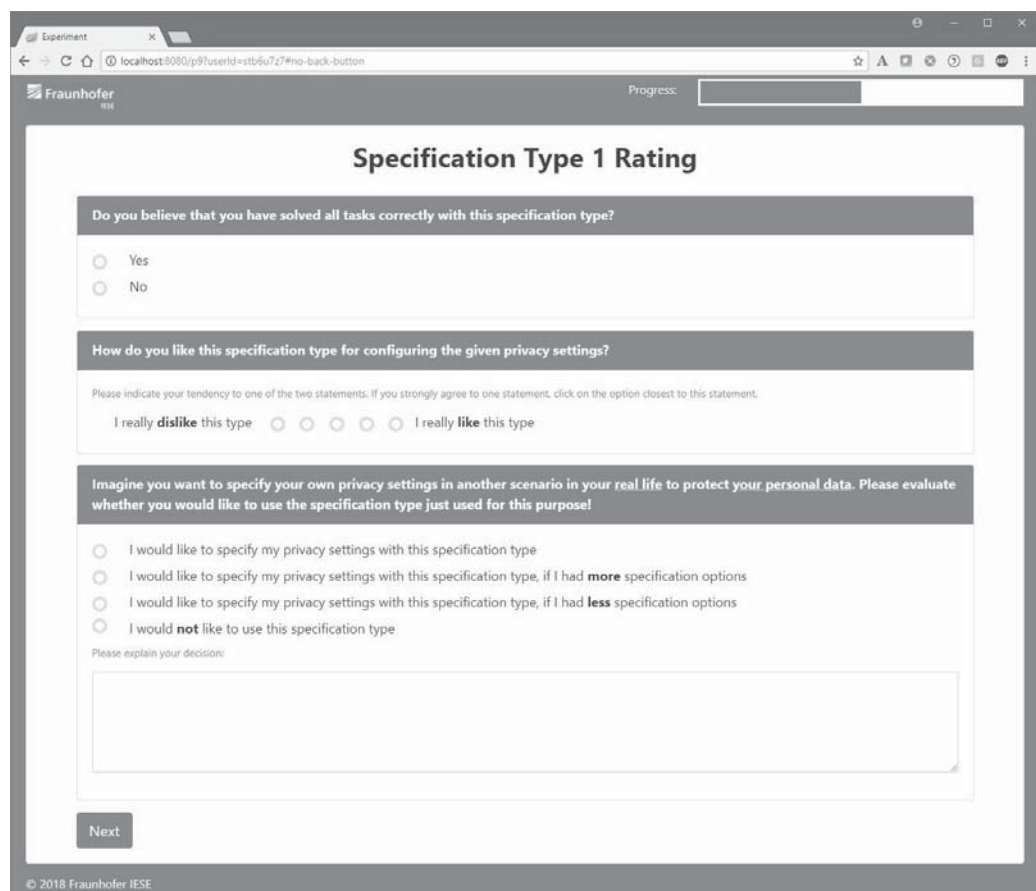


Figure 113: Screenshot - Specification Type Rating

Experiment

localhost:8080/p10?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

## Specification Type 2: Default Policies

The blue box contains the second specification type. Please try again to configure all six privacy demands of the handout.

|  |   |
|--|---|
| BestellBar: Forwarding of order data                               | <input type="radio"/> When a merchant forwards my order data to an advertisement company, I want to be informed.  |
| LieferBar: Acceptance of a delivery                                | <input checked="" type="radio"/> When a merchant forwards my order data to an advertisement company, I want to be informed and my personal data needs to be anonymized. |
| LieferBar: Information prior to acceptance of the delivery request | <input type="radio"/> When a merchant forwards my order data to an advertisement company, I forbid that.  |
| LieferBar: Displaying the storage location for packages            |   |
| DorfFunk: Help requests and offers                                 |   |
| Scientific evaluation  |   |

I configured all six privacy demands

© 2018 Fraunhofer IESE

Figure 114: Screenshot - Specification Type: Default Policies 1

Experiment

localhost:8080/p10?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

## Specification Type 2: Default Policies

The blue box contains the second specification type. Please try again to configure all six privacy demands of the handout.

|  |  |
|--|--|
| BestellBar: Forwarding of order data                               | <input type="radio"/> All citizens may see and accept my delivery requests.  |
| LieferBar: Acceptance of a delivery                                | <input type="radio"/> Only deliverers with a trust level of at least bronze may see and accept my delivery requests.                         |
| LieferBar: Information prior to acceptance of the delivery request | <input type="radio"/> Only my friends and deliverers with a trust level of at least silver may see and accept my delivery requests.          |
| LieferBar: Displaying the storage location for packages            | <input checked="" type="radio"/> Only my friends and deliverers with a trust level of at least gold may see and accept my delivery requests. |
| DorfFunk: Help requests and offers                                 | <input type="radio"/> Only my friends may see and accept my delivery requests.   |
| Scientific evaluation  |  |

I configured all six privacy demands

© 2018 Fraunhofer IESE

Figure 115: Screenshot - Specification Type: Default Policies 2

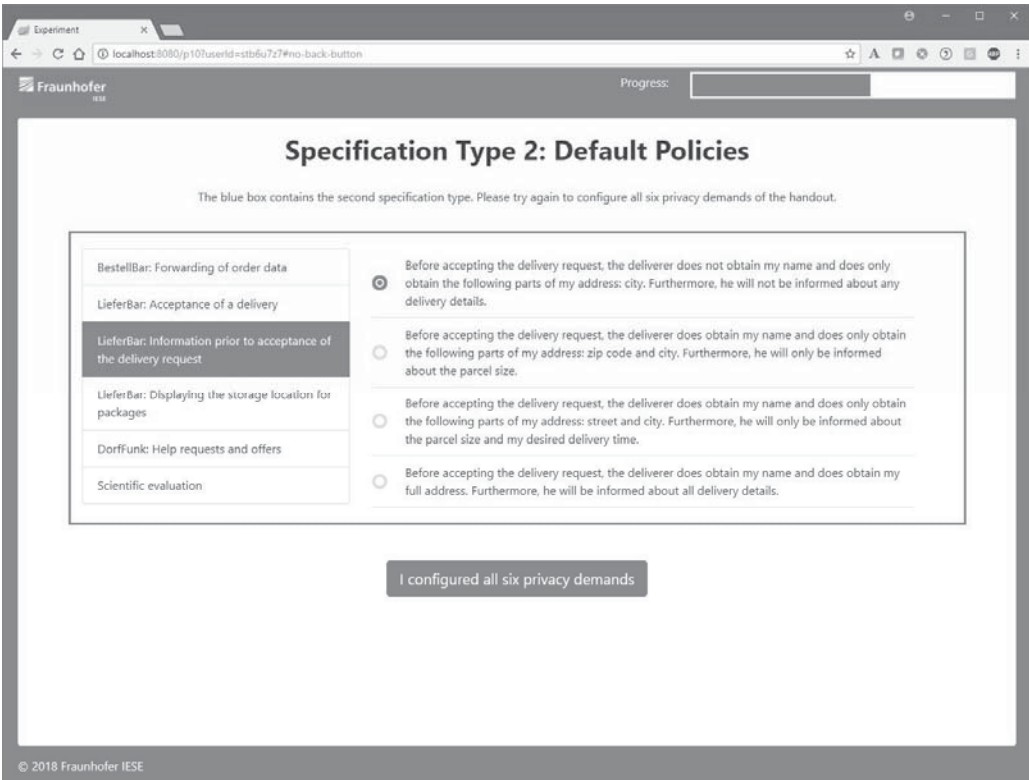


Figure 116: Screenshot - Specification Type: Default Policies 3

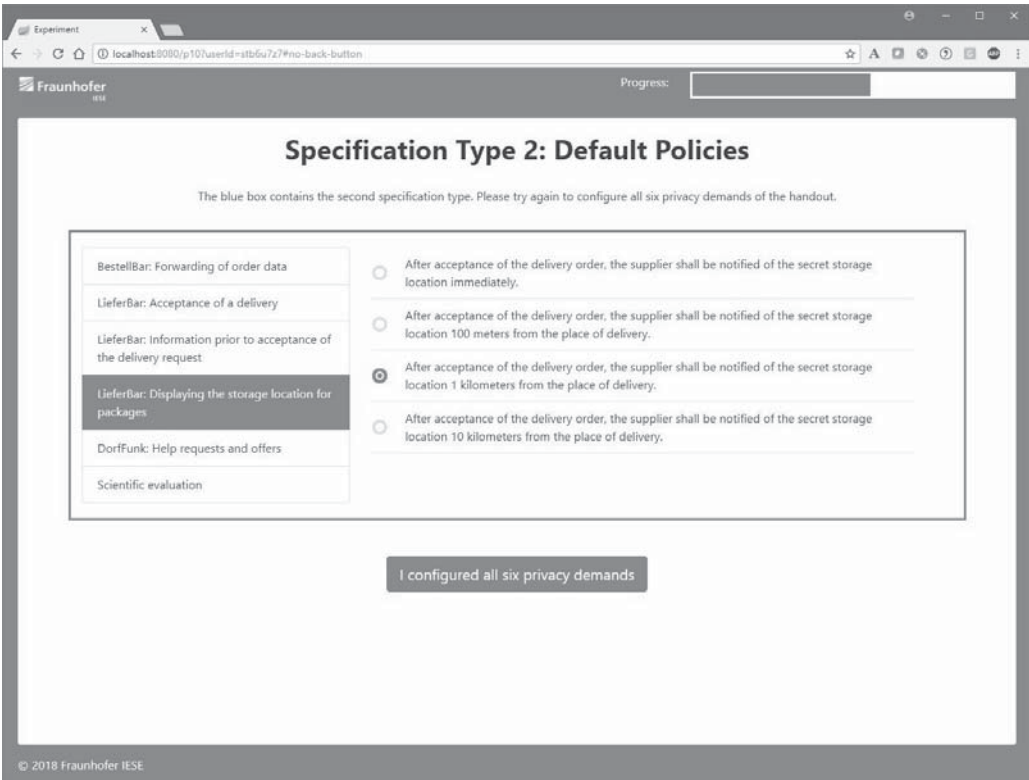


Figure 117: Screenshot - Specification Type: Default Policies 4

Experiment

localhost:8080/p10?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

## Specification Type 2: Default Policies

The blue box contains the second specification type. Please try again to configure all six privacy demands of the handout.

|  |   |
|--|---|
| BestellBar: Forwarding of order data                               | <input type="radio"/> My help requests and offers can be viewed by every citizen. Before accepting the help request or offer, they are allowed to look at not my name, only zip code and city of my address and only the date of the preferred appointment.   |
| LieferBar: Acceptance of a delivery                                | <input type="radio"/> My help requests and offers can be viewed by my friends. Before accepting the help request or offer, they are allowed to look at my complete name, my complete address and only date and daytime of the preferred appointment.  |
| LieferBar: Information prior to acceptance of the delivery request | <input type="radio"/> My help requests and offers can be viewed by my friends and by citizens with at least the trust level gold. Before accepting the help request or offer, they are allowed to look at my complete name, not my address and only date and daytime of the preferred appointment.      |
| LieferBar: Displaying the storage location for packages            | <input type="radio"/> My help requests and offers can be viewed by my friends and by citizens with at least the trust level gold. Before accepting the help request or offer, they are allowed to look at not my name, only street and city of my address and not the preferred appointment.            |
| <b>Dorffunk: Help requests and offers</b>                          | <input checked="" type="radio"/> My help requests and offers can be viewed by my friends and by citizens with at least the trust level gold. Before accepting the help request or offer, they are allowed to look at not my name, only street and city of my address and not the preferred appointment. |
| Scientific evaluation  |   |

I configured all six privacy demands

© 2018 Fraunhofer IESE

Figure 118: Screenshot - Specification Type: Default Policies 5

Experiment

localhost:8080/p10?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

## Specification Type 2: Default Policies

The blue box contains the second specification type. Please try again to configure all six privacy demands of the handout.

|  |   |
|--|---|
| BestellBar: Forwarding of order data                               | <input type="radio"/> My data will be excluded from scientific evaluations.   |
| LieferBar: Acceptance of a delivery                                | <input checked="" type="radio"/> My data will be permitted for scientific evaluation.                                       |
| LieferBar: Information prior to acceptance of the delivery request | <input type="radio"/> My data will be permitted for scientific evaluation if my name has been made anonymous.               |
| LieferBar: Displaying the storage location for packages            | <input type="radio"/> My data will be permitted for scientific evaluation if all my personal data have been made anonymous. |
| Dorffunk: Help requests and offers                                 |   |
| <b>Scientific evaluation</b>                                       |   |

I configured all six privacy demands

© 2018 Fraunhofer IESE

Figure 119: Screenshot - Specification Type: Default Policies 6

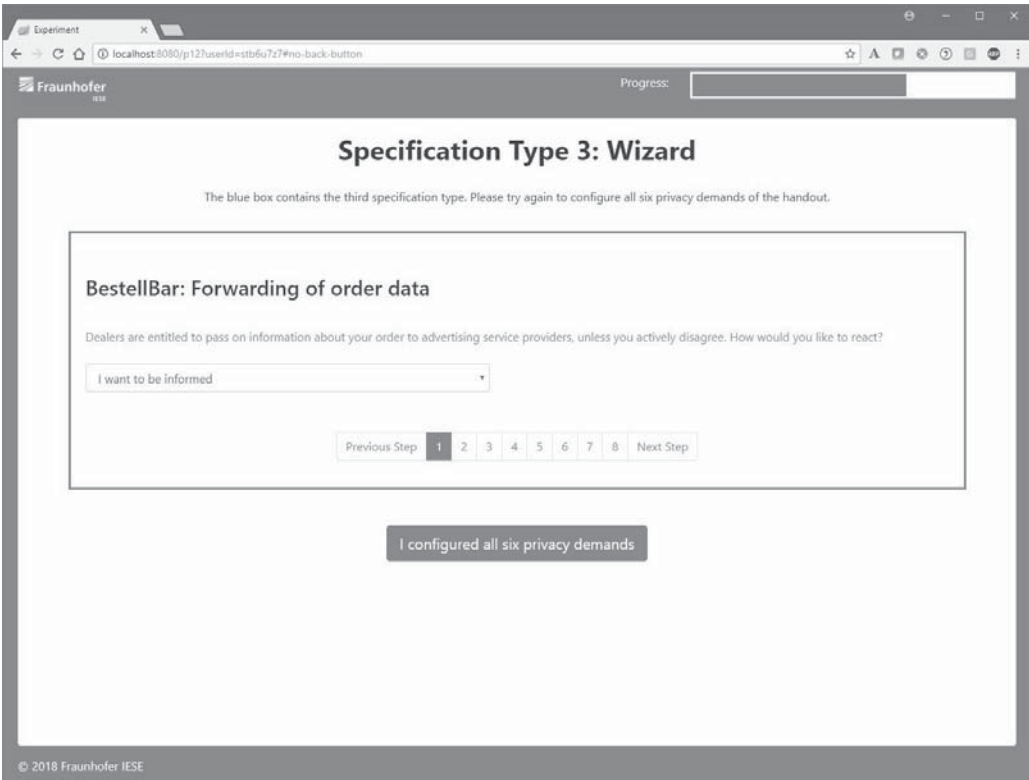


Figure 120: Screenshot - Specification Type: Wizard 1

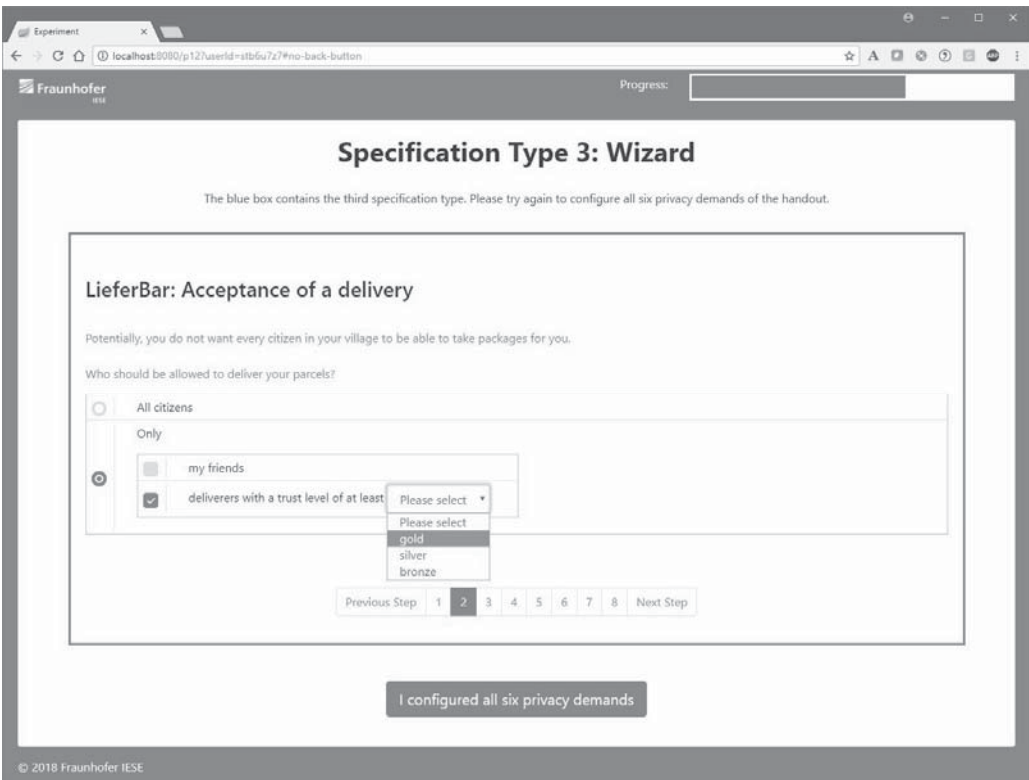


Figure 121: Screenshot - Specification Type: Wizard 2



Figure 122: Screenshot - Specification Type: Wizard 3

Figure 123: Screenshot - Specification Type: Wizard 4



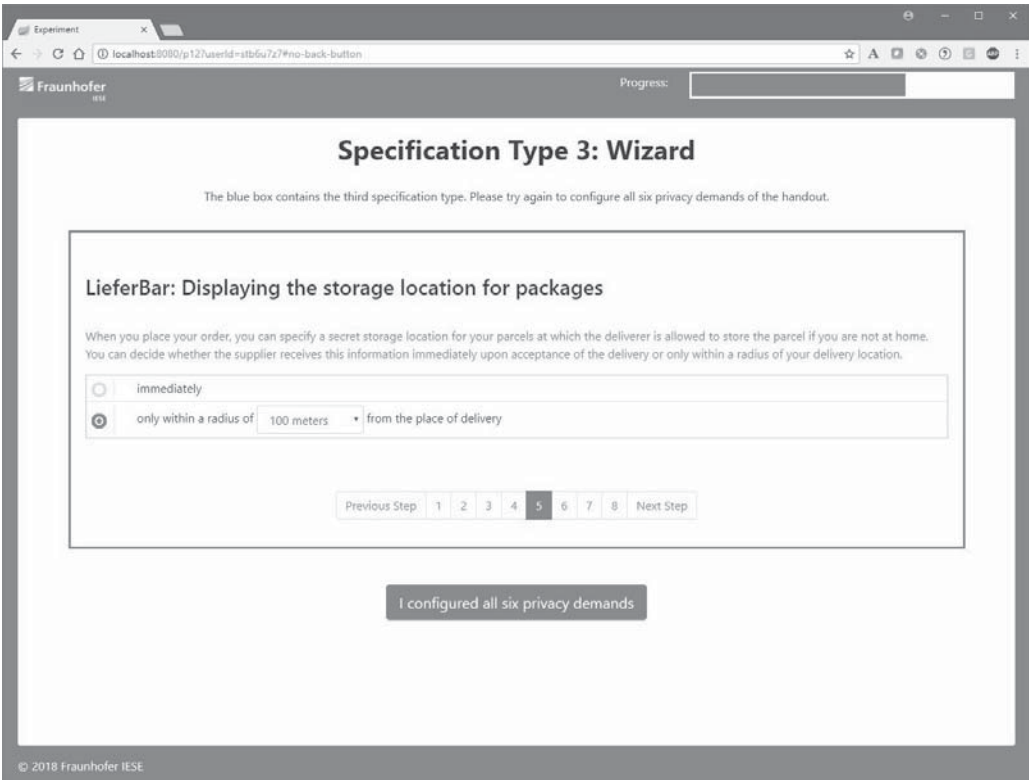


Figure 124: Screenshot - Specification Type: Wizard 5

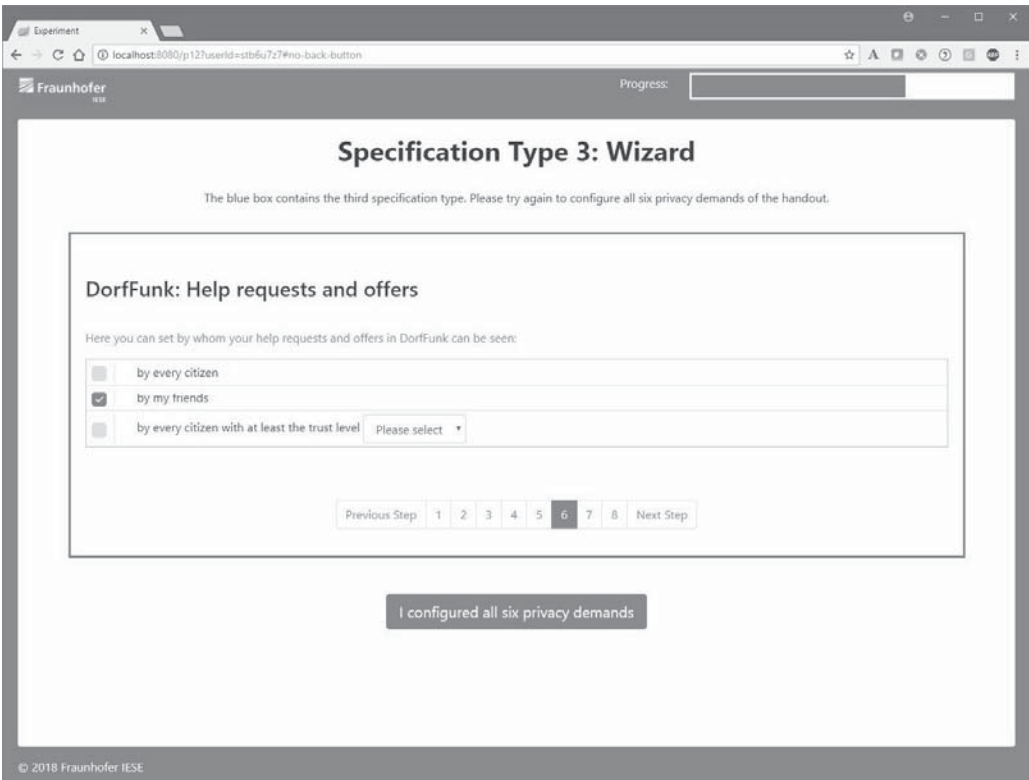
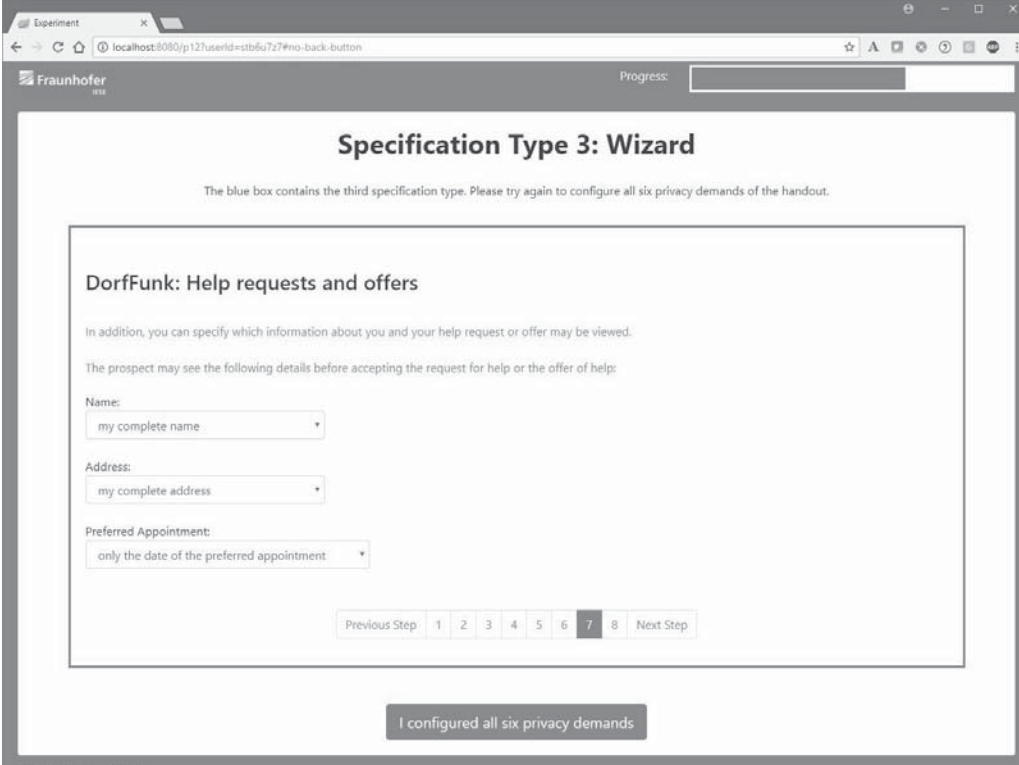
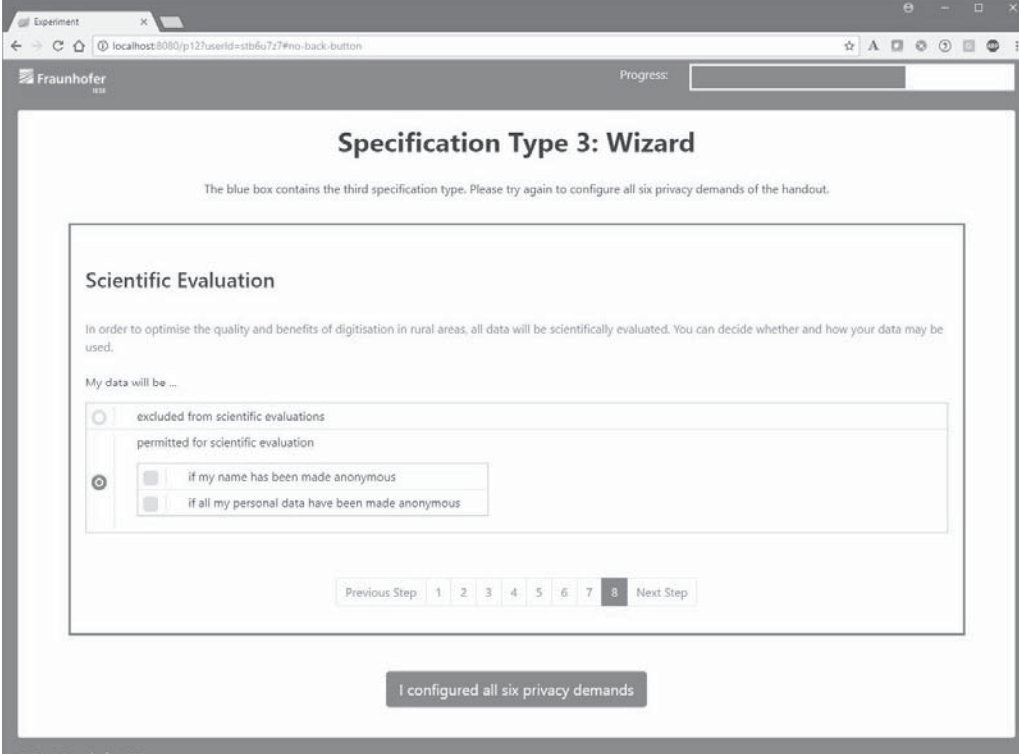


Figure 125: Screenshot - Specification Type: Wizard 6



The screenshot shows a web browser window with the URL `localhost:8080/p12?userId=stb6u7z7#no-back-button`. The page is titled "Specification Type 3: Wizard" and includes a progress bar. The main content area is titled "DorfFunk: Help requests and offers" and contains the following text: "The blue box contains the third specification type. Please try again to configure all six privacy demands of the handout." Below this, there is a form with three dropdown menus: "Name:" (selected "my complete name"), "Address:" (selected "my complete address"), and "Preferred Appointment:" (selected "only the date of the preferred appointment"). At the bottom of the form is a navigation bar with buttons for "Previous Step", "1", "2", "3", "4", "5", "6", "7" (highlighted), "8", and "Next Step". A button at the bottom of the page says "I configured all six privacy demands". The footer of the page reads "© 2018 Fraunhofer IESE".

Figure 126: Screenshot - Specification Type: Wizard 7



The screenshot shows the same web browser window as Figure 126, but at step 8 of the wizard. The main content area is titled "Scientific Evaluation" and contains the following text: "In order to optimise the quality and benefits of digitisation in rural areas, all data will be scientifically evaluated. You can decide whether and how your data may be used." Below this, there is a form with a section titled "My data will be ..." containing two radio buttons: "excluded from scientific evaluations" (selected) and "permitted for scientific evaluation". Under the "permitted for scientific evaluation" option, there are two checkboxes: "if my name has been made anonymous" and "if all my personal data have been made anonymous". At the bottom of the form is a navigation bar with buttons for "Previous Step", "1", "2", "3", "4", "5", "6", "7", "8" (highlighted), and "Next Step". A button at the bottom of the page says "I configured all six privacy demands". The footer of the page reads "© 2018 Fraunhofer IESE".

Figure 127: Screenshot - Specification Type: Wizard 8

Experiment

localhost:8080/p14?userId=stb6u7z7#no-back-button

Fraunhofer IESE Progress:

## Specification Type 4: Privacy Levels

The blue box contains the fourth specification type. Please try again to configure all six privacy demands of the handout.

**Privacy Level Blue**

When a merchant forwards my order data to an advertisement company, I want to be informed and my personal data needs to be anonymized.

All citizens may see and accept my delivery requests.

☐ Before accepting the delivery request, the deliverer does obtain my name and does only obtain the following parts of my address: street and city. Furthermore, he will only be informed about the parcel size and my desired delivery time.

After acceptance of the delivery order, the supplier shall be notified of the secret storage location 100 meters from the place of delivery.

My help requests and offers can be viewed by my friends and by citizens with at least the trust level gold. Before accepting the help request or offer, they are allowed to look at not my name, only street and city of my address and not the preferred appointment.

My data will be excluded from scientific evaluations.

**Privacy Level Purple**

When a merchant forwards my order data to an advertisement company, I want to be informed.

Only my friends and deliverers with a trust level of at least silver may see and accept my delivery requests.

☒ Before accepting the delivery request, the deliverer does not obtain my name and does only obtain the following parts of my address: city. Furthermore, he will not be informed about any delivery details.

After acceptance of the delivery order, the supplier shall be notified of the secret storage location immediately.

My help requests and offers can be viewed by my friends and by citizens with at least the trust level gold. Before accepting the help request or offer, they are allowed to look at my complete name, not my address and only date and daytime of the preferred appointment.

My data will be permitted for scientific evaluation if all my personal data have been made anonymous.

**Privacy Level Orange**

When a merchant forwards my order data to an advertisement company, I forbid that.

Only my friends and deliverers with a trust level of at least gold may see and accept my delivery requests.

☐ Before accepting the delivery request, the deliverer does obtain my name and does only obtain the following parts of my address: zip code and city. Furthermore, he will only be informed about the parcel size.

After acceptance of the delivery order, the supplier shall be notified of the secret storage location 100 meters from the place of delivery.

My help requests and offers can be viewed by every citizen. Before accepting the help request or offer, they are allowed to look at not my name, only zip code and city of my address and only the date of the preferred appointment.

My data will be permitted for scientific evaluation if my name has been made anonymous.

© 2018 Fraunhofer IESE

Figure 128: Screenshot - Specification Type: Privacy Levels

Experiment

localhost:8080/p16?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

## Personal preference of specification types

You have just used four different specification types. We are interested in your preferences regarding these specification types. Please sort the specification types in the order of your preference, starting with the specification type you prefer to use. Do not think about it for long: just follow your intuition.

**Please sort the specification types you just have used in the order of your preference, if you would have to use it in real life!**

Place the specification type you liked most at the top position and the specification type you liked least at the bottom position.

If you are no longer sure which paradigm had which name, move the mouse over the **i** after the name of the specification type.

Click on a specification type and hold down the left mouse button to move it to your preferred position.

|    |   |  |   |
|----|---|--|---|
| #1 | + | Specification Type 1: Templates        | i |
| #2 | + | Specification Type 2: Default Policies | i |
| #3 | + | Specification Type 3: Wizard           | i |
| #4 | + | Specification Type 4: Privacy Levels   | i |

**Please let us know why you chose this order!**

Please write your reason into the textbox.

Next

© 2018 Fraunhofer IESE

Figure 129: Screenshot - Specification Type Preference Ordering

Experiment

localhost:8080/p17?userId=stb6u7z7#no-back-button

Fraunhofer IESE

Progress:

## Identification with scenario and persona

Almost done. This is the last step of the experiment.

**Before we asked you to specify privacy settings, we presented a scenario to you, which is printed in the handout. How much can you put yourself into this scenario?**

Please indicate your tendency to one of the two statements. If you strongly agree to one statement, click on the option closest to this statement.

I could **not** empathize at all ☐ ☐ ☐ ☐ ☐ I perfectly empathize

**How well do you think the persona you have chosen matches your personality?**

Please indicate your tendency to one of the two statements. If you strongly agree to one statement, click on the option closest to this statement.

**Not very well**, but it matched best out of the five options ☐ ☐ ☐ ☐ ☐ I can identify myself **very well** with the persona

**Do you have any further comments about the scenario, the specification types, or the experiment?**

Please write your comments into the textbox.

Next

© 2018 Fraunhofer IESE

Figure 130: Screenshot - Identification with Scenario and Persona

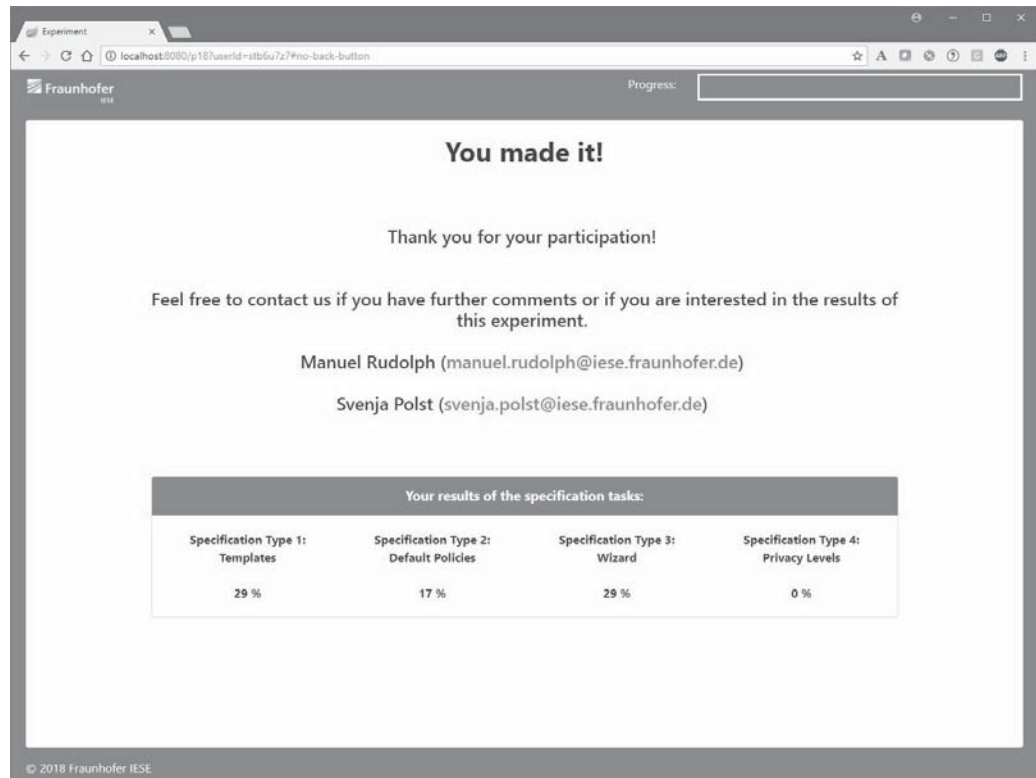


Figure 131: Screenshot - Final Page and Scores

## G.4 Sample Solution

### Template Instantiation and Wizard

The correct instantiations are:

- Template »Forwarding of order data«: »When a merchant forwards my order data to an advertisement company, I forbid that.«
- Template »Acceptance of a delivery«: »Only my friends and deliverer with a trust level of at least gold may see and accept my delivery requests.«
- Template »Information prior to acceptance of the delivery request«: »Before accepting the delivery request, the deliverer does obtain my name and does only obtain the following parts of my address: zip code and city. Furthermore, he will only be informed about the parcel size.«
- Template »Displaying the storage location for packages«: »After acceptance of the delivery order, the supplier shall be notified of the secret storage location 100 meters from the place of delivery.«

- Template »Help requests and offers«: »My help requests and offers can be viewed by every citizen. Before accepting the help request or offer, they are allowed to look at not my name, only zip code and city of my address and only the date of the preferred appointment.«
- Template »Scientific evaluation«: »My data will be permitted for scientific evaluation if my name has been made anonymous.“

## Wizard

The correct options on the wizard pages are:

- Page 1: »I forbid that«
- Page 2: »Only my friends and deliverer with a trust level of at least gold«
- Page 3: »does obtain my name« and »does only obtain the following parts of my address: zip code and city«
- Page 4: »only be informed about the parcel size«
- Page 5: »only within a radius of 100m«
- Page 6: »by every citizen«
- Page 7: »not my name« and »only zip code and city of my address« and »only the date of the preferred appointment«
- Page 8: »permitted for scientific evaluation if my name has been made anonymous«

## Default Policies

The correct default policies are:

- Category »Forwarding of order data« – Correct Option 3: »When a merchant forwards my order data to an advertisement company, I forbid that.«
- Category »Acceptance of a delivery« – Correct Option 4: »Only my friends and deliverer with a trust level of at least gold may see and accept my delivery requests.«
- Category »Information prior to acceptance of the delivery request« – Correct Option 2: »Before accepting the delivery request, the deliverer does obtain my name and does only obtain the following parts of my address: zip code and city. Furthermore, he will only be informed about the parcel size.«
- Category »Displaying the storage location for packages« – Correct Option 2: »After acceptance of the delivery order, the supplier shall be

notified of the secret storage location 100 meters from the place of delivery.«

- Category »Help requests and offers« – Correct Option 1: »My help requests and offers can be viewed by every citizen. Before accepting the help request or offer, they are allowed to look at not my name, only zip code and city of my address and only the date of the preferred appointment.«
- Category »Scientific evaluation« – Correct Option 3: »My data will be permitted for scientific evaluation if my name has been made anonymous.«

## Security Levels

The correct security level is: yellow

- »When a merchant forwards my order data to an advertisement company, I forbid that.«
- »Only my friends and deliverer with a trust level of at least gold may see and accept my delivery requests.«
- »Before accepting the delivery request, the deliverer does obtain my name and does only obtain the following parts of my address: zip code and city. Furthermore, he will only be informed about the parcel size.«
- »After acceptance of the delivery order, the supplier shall be notified of the secret storage location 100 meters from the place of delivery.«
- »My help requests and offers can be viewed by every citizen. Before accepting the help request or offer, they are allowed to look at not my name, only zip code and city of my address and only the date of the preferred appointment.«
- »My data will be permitted for scientific evaluation if my name has been made anonymous.«

## G.5 Detailed Results of Statistical Analyses

### Objective Correctness

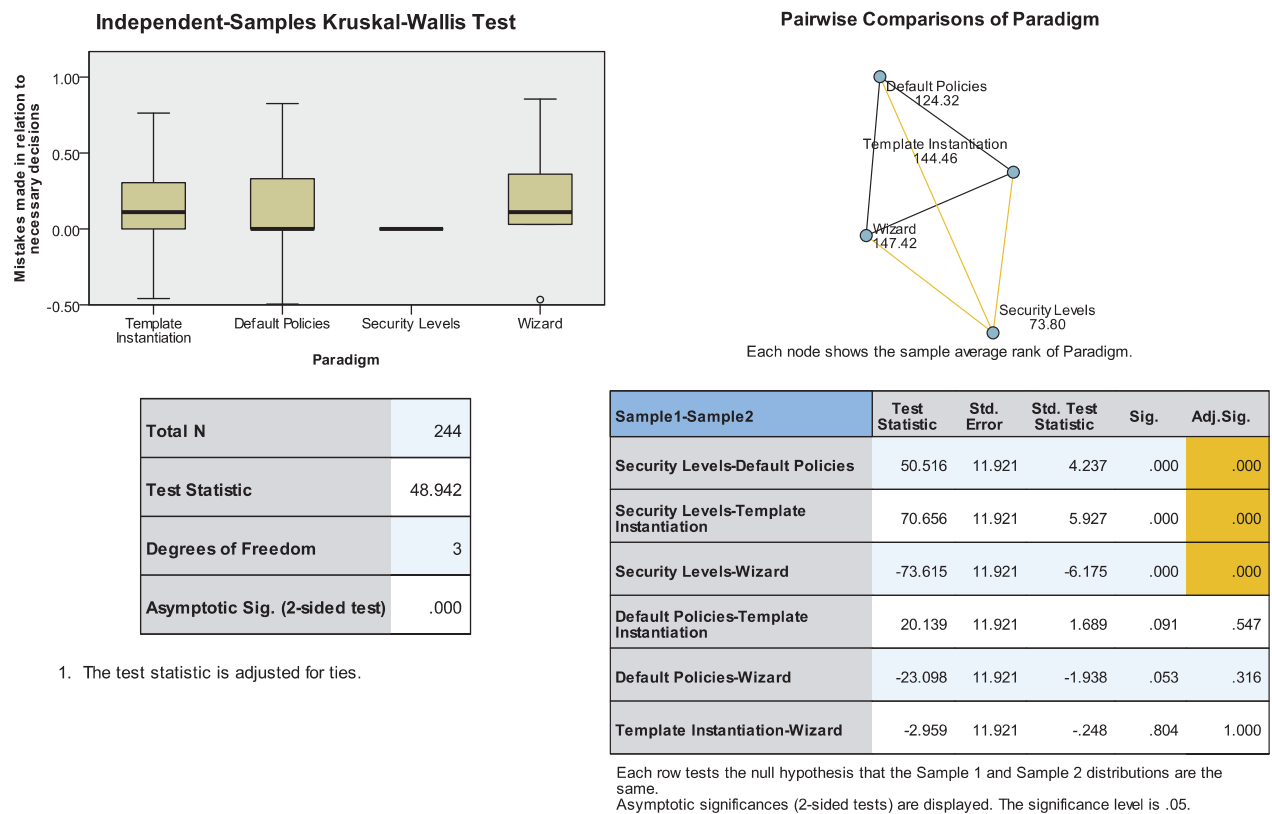


Figure 132: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes with Pairwise Comparison of Specification Paradigms (Q1.1.1)

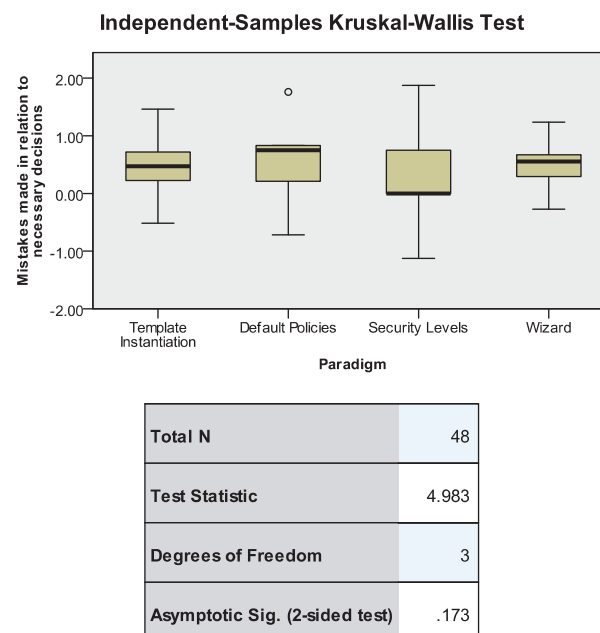


Figure 133: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Marginally Concerned (Q1.1.2)



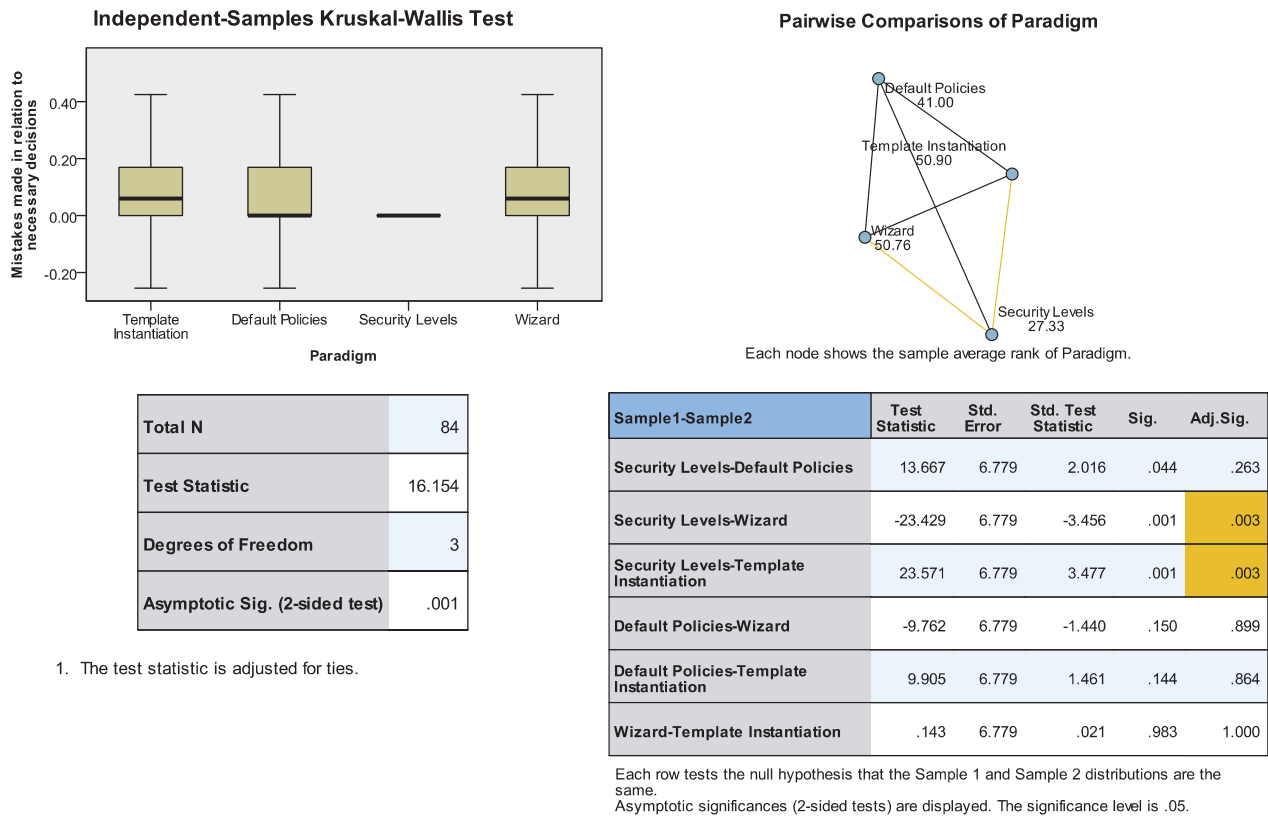


Figure 134: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Amateurs with Pairwise Comparison of Specification Paradigms (Q1.1.2)

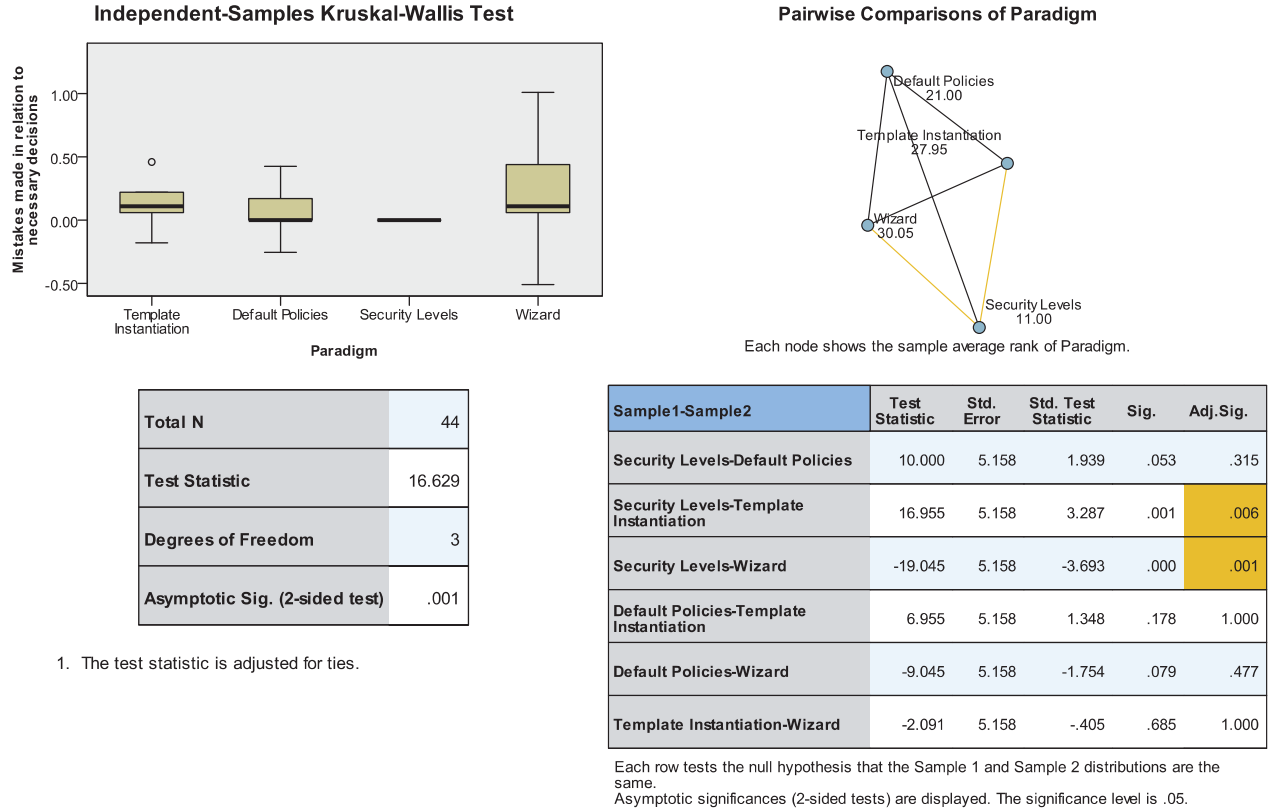


Figure 135: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Lazy Experts with Pairwise Comparison of Specification Paradigms (Q1.1.2)

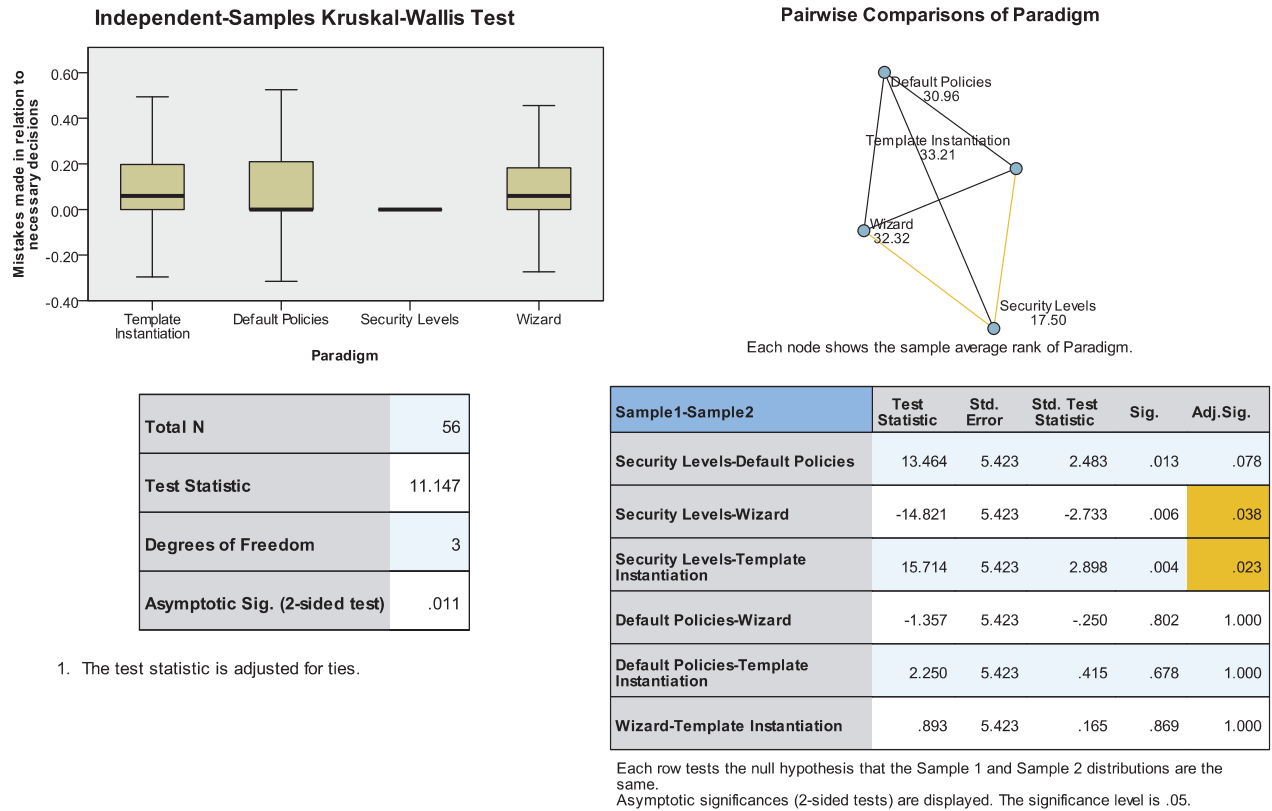


Figure 136: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Technician with Pairwise Comparison of Specification Paradigms (Q1.1.2)

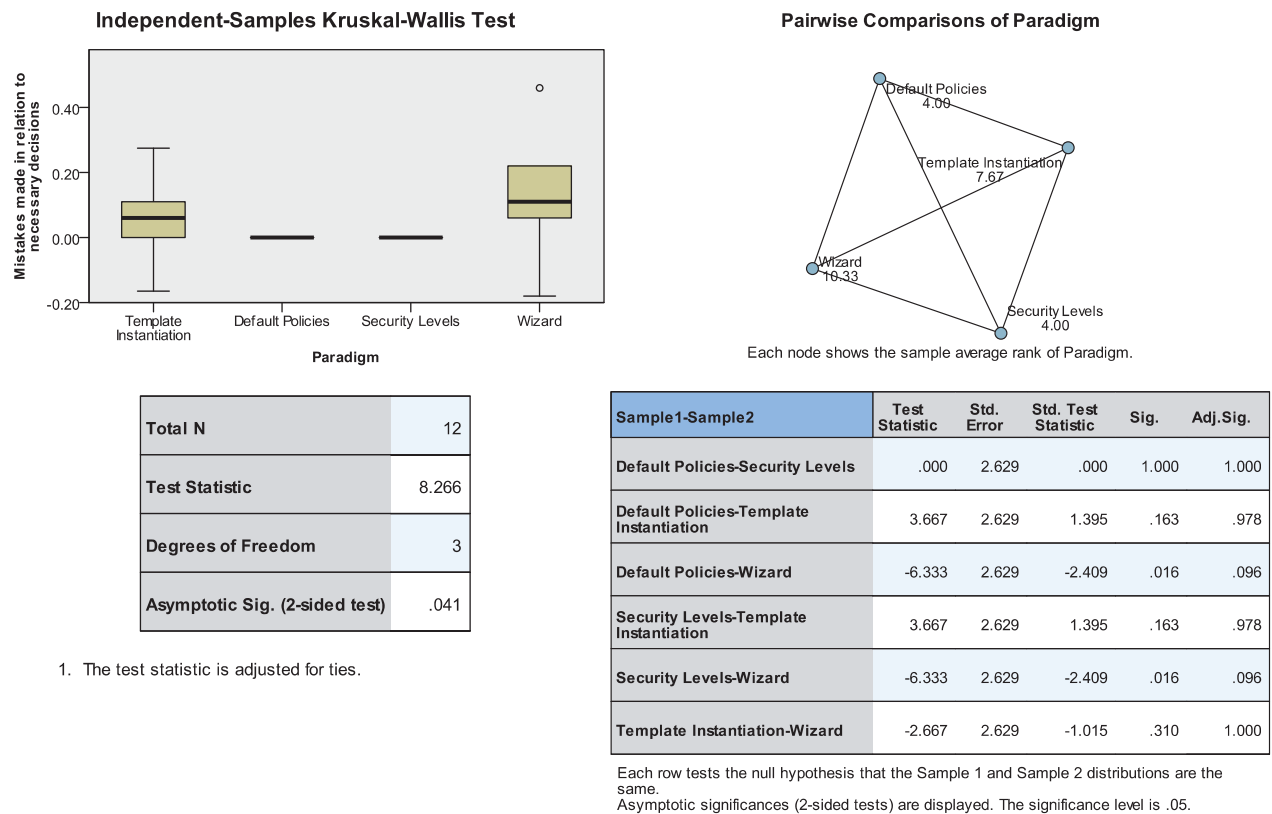


Figure 137: Kruskal-Wallis-Test on Influence of Specification Paradigms on Conducted Mistakes for Fundamentalists with Pairwise Comparison of Specification Paradigms (Q1.1.2)

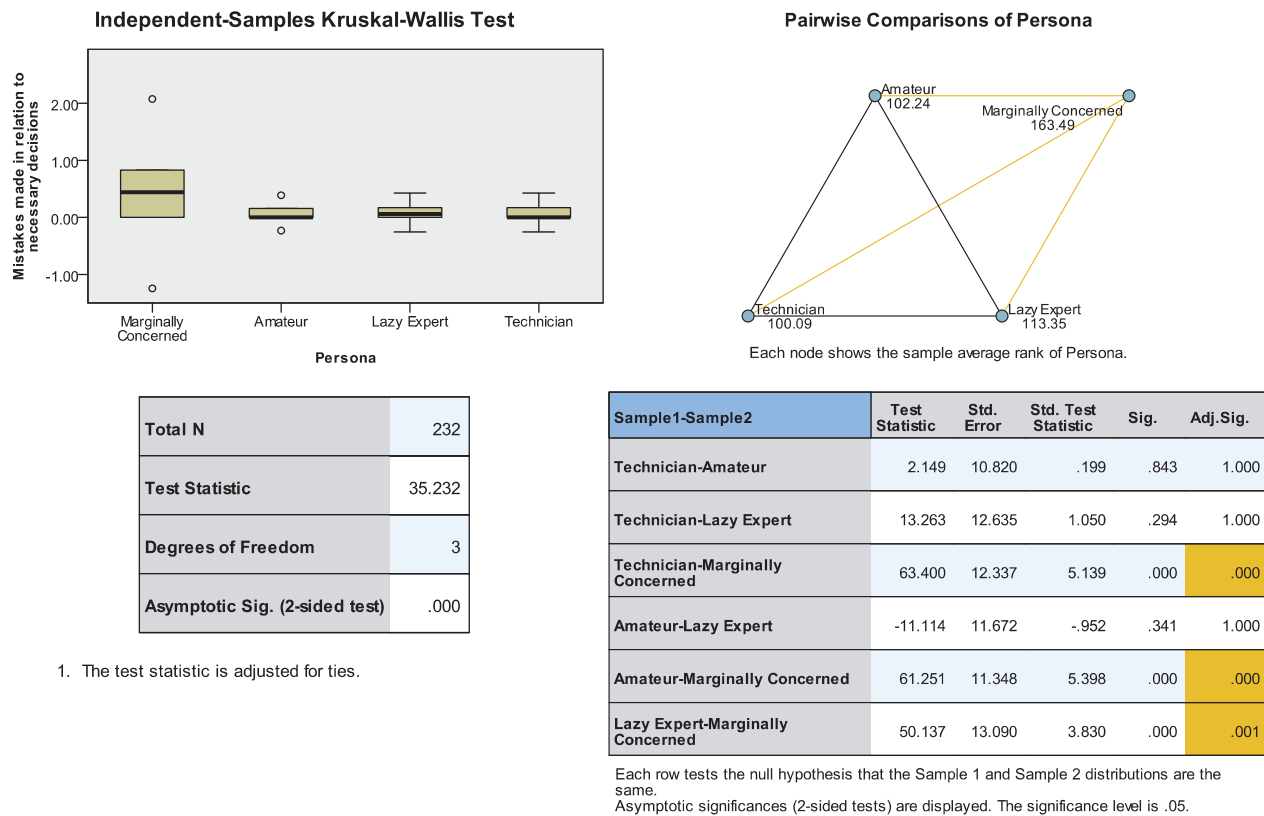


Figure 138: Kruskal-Wallis-Test on Influence of Persona Selection on Conducted Mistakes with Pairwise Comparison of Specification Paradigms (Q1.1.3)

## Self-evaluation regarding Objective Correctness

**Paradigm \* Correct Estimation Crosstabulation**

|          |                        | Correct Estimation   |                    | Total |
|----------|------------------------|----------------------|--------------------|-------|
|          |                        | Incorrect Estimation | Correct Estimation |       |
| Paradigm | Template Instantiation | 39                   | 22                 | 61    |
|          | Default Policies       | 24                   | 37                 | 61    |
|          | Security Levels        | 13                   | 48                 | 61    |
|          | Wizard                 | 43                   | 18                 | 61    |
| Total    |                        | 119                  | 125                | 244   |

**Chi-Square Tests**

|                              | Value               | df | Asymp. Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) | Point Probability |
|------------------------------|---------------------|----|-----------------------|----------------------|----------------------|-------------------|
| Pearson Chi-Square           | 37,711 <sup>a</sup> | 3  | .000                  | .000                 |                      |                   |
| Likelihood Ratio             | 39,361              | 3  | .000                  | .000                 |                      |                   |
| Fisher's Exact Test          | 38,692              |    |                       | .000                 |                      |                   |
| Linear-by-Linear Association | .003 <sup>b</sup>   | 1  | .954                  | 1,000                | .500                 | .045              |
| N of Valid Cases             | 244                 |    |                       |                      |                      |                   |

a. 0 cells (.0%) have expected count less than 5. The minimum expected count is 29,75.

b. The standardized statistic is -.057.

Figure 139: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness (Q1.2.1)

| Paradigm * Correct Estimation Crosstabulation |                        |                      |                    |       |  |  |
|---|------------------------|----------------------|--------------------|-------|--|--|
| Count   |                        | Correct Estimation   |                    |       |  |  |
|   |                        | Incorrect Estimation | Correct Estimation | Total |  |  |
| Paradigm                                      | Template Instantiation | 9                    | 3                  | 12    |  |  |
|   | Default Policies       | 9                    | 3                  | 12    |  |  |
|   | Security Levels        | 3                    | 9                  | 12    |  |  |
|   | Wizard                 | 11                   | 1                  | 12    |  |  |
| Total   |                        | 32                   | 16                 | 48    |  |  |

| Chi-Square Tests             |                     |    |                       |                      |                      |                   |
|------------------------------|---------------------|----|-----------------------|----------------------|----------------------|-------------------|
|                              | Value               | df | Asymp. Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) | Point Probability |
| Pearson Chi-Square           | 13,500 <sup>a</sup> | 3  | ,004                  | ,004                 |                      |                   |
| Likelihood Ratio             | 13,733              | 3  | ,003                  | ,007                 |                      |                   |
| Fisher's Exact Test          | 12,487              |    |                       | ,005                 |                      |                   |
| Linear-by-Linear Association | ,000 <sup>b</sup>   | 1  | 1,000                 | 1,000                | ,553                 | ,107              |
| N of Valid Cases             | 48                  |    |                       |                      |                      |                   |

a. 4 cells (50,0%) have expected count less than 5. The minimum expected count is 4,00.

b. The standardized statistic is ,000.

Figure 140: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Marginally Concerned (Q1.2.2)

| Paradigm * Correct Estimation Crosstabulation |                        |                      |                    |       |  |  |
|---|------------------------|----------------------|--------------------|-------|--|--|
| Count   |                        | Correct Estimation   |                    |       |  |  |
|   |                        | Incorrect Estimation | Correct Estimation | Total |  |  |
| Paradigm                                      | Template Instantiation | 14                   | 7                  | 21    |  |  |
|   | Default Policies       | 8                    | 13                 | 21    |  |  |
|   | Security Levels        | 4                    | 17                 | 21    |  |  |
|   | Wizard                 | 14                   | 7                  | 21    |  |  |
| Total   |                        | 40                   | 44                 | 84    |  |  |

| Chi-Square Tests             |                     |    |                       |                      |                      |                   |
|------------------------------|---------------------|----|-----------------------|----------------------|----------------------|-------------------|
|                              | Value               | df | Asymp. Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) | Point Probability |
| Pearson Chi-Square           | 13,745 <sup>a</sup> | 3  | ,003                  | ,003                 |                      |                   |
| Likelihood Ratio             | 14,430              | 3  | ,002                  | ,003                 |                      |                   |
| Fisher's Exact Test          | 13,781              |    |                       | ,003                 |                      |                   |
| Linear-by-Linear Association | ,151 <sup>b</sup>   | 1  | ,698                  | ,772                 | ,386                 | ,072              |
| N of Valid Cases             | 84                  |    |                       |                      |                      |                   |

a. 0 cells (,0%) have expected count less than 5. The minimum expected count is 10,00.

b. The standardized statistic is ,388.

Figure 141: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Amateurs (Q1.2.2)

| Paradigm * Correct Estimation Crosstabulation |                        |                      |                    |       |  |  |
|---|------------------------|----------------------|--------------------|-------|--|--|
| Count   |                        | Correct Estimation   |                    | Total |  |  |
|   |                        | Incorrect Estimation | Correct Estimation |       |  |  |
| Paradigm                                      | Template Instantiation | 8                    | 3                  | 11    |  |  |
|   | Default Policies       | 2                    | 9                  | 11    |  |  |
|   | Security Levels        | 3                    | 8                  | 11    |  |  |
|   | Wizard                 | 8                    | 3                  | 11    |  |  |
| Total   |                        | 21                   | 23                 | 44    |  |  |

| Chi-Square Tests             |                     |    |                       |                      |                      |                   |
|------------------------------|---------------------|----|-----------------------|----------------------|----------------------|-------------------|
|                              | Value               | df | Asymp. Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) | Point Probability |
| Pearson Chi-Square           | 11,205 <sup>a</sup> | 3  | ,011                  | ,012                 |                      |                   |
| Likelihood Ratio             | 11,802              | 3  | ,008                  | ,012                 |                      |                   |
| Fisher's Exact Test          | 10,855              |    |                       | ,012                 |                      |                   |
| Linear-by-Linear Association | ,018 <sup>b</sup>   | 1  | ,894                  | 1,000                | ,500                 | ,105              |
| N of Valid Cases             | 44                  |    |                       |                      |                      |                   |

a. 0 cells (,0%) have expected count less than 5. The minimum expected count is 5,25.

b. The standardized statistic is -,133.

Figure 142: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Lazy Experts (Q1.2.2)

| Paradigm * Correct Estimation Crosstabulation |                        |                      |                    |       |  |  |
|---|------------------------|----------------------|--------------------|-------|--|--|
| Count   |                        | Correct Estimation   |                    | Total |  |  |
|   |                        | Incorrect Estimation | Correct Estimation |       |  |  |
| Paradigm                                      | Template Instantiation | 6                    | 8                  | 14    |  |  |
|   | Default Policies       | 5                    | 9                  | 14    |  |  |
|   | Security Levels        | 2                    | 12                 | 14    |  |  |
|   | Wizard                 | 7                    | 7                  | 14    |  |  |
| Total   |                        | 20                   | 36                 | 56    |  |  |

| Chi-Square Tests             |                    |    |                       |                      |                      |                   |
|------------------------------|--------------------|----|-----------------------|----------------------|----------------------|-------------------|
|                              | Value              | df | Asymp. Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) | Point Probability |
| Pearson Chi-Square           | 4,356 <sup>a</sup> | 3  | ,226                  | ,302                 |                      |                   |
| Likelihood Ratio             | 4,735              | 3  | ,192                  | ,235                 |                      |                   |
| Fisher's Exact Test          | 4,443              |    |                       | ,235                 |                      |                   |
| Linear-by-Linear Association | ,000 <sup>b</sup>  | 1  | 1,000                 | 1,000                | ,549                 | ,098              |
| N of Valid Cases             | 56                 |    |                       |                      |                      |                   |

a. 0 cells (,0%) have expected count less than 5. The minimum expected count is 5,00.

b. The standardized statistic is ,000.

Figure 143: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Technicians (Q1.2.2)

| Paradigm * Correct Estimation Crosstabulation |                        |                      |                    |       |  |  |
|---|------------------------|----------------------|--------------------|-------|--|--|
| Count   |                        | Correct Estimation   |                    | Total |  |  |
|   |                        | Incorrect Estimation | Correct Estimation |       |  |  |
| Paradigm                                      | Template Instantiation | 2                    | 1                  | 3     |  |  |
|   | Default Policies       | 0                    | 3                  | 3     |  |  |
|   | Security Levels        | 1                    | 2                  | 3     |  |  |
|   | Wizard                 | 3                    | 0                  | 3     |  |  |
| Total   |                        | 6                    | 6                  | 12    |  |  |

| Chi-Square Tests             |                    |    |                       |                      |                      |                   |
|------------------------------|--------------------|----|-----------------------|----------------------|----------------------|-------------------|
|                              | Value              | df | Asymp. Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) | Point Probability |
| Pearson Chi-Square           | 6,667 <sup>a</sup> | 3  | ,083                  | ,240                 |                      |                   |
| Likelihood Ratio             | 8,997              | 3  | ,029                  | ,240                 |                      |                   |
| Fisher's Exact Test          | 5,999              |    |                       | ,240                 |                      |                   |
| Linear-by-Linear Association | ,978 <sup>b</sup>  | 1  | ,323                  | ,472                 | ,236                 | ,127              |
| N of Valid Cases             | 12                 |    |                       |                      |                      |                   |

a. 8 cells (100,0%) have expected count less than 5. The minimum expected count is 1,50.

b. The standardized statistic is -,989.

Figure 144: Cross Tables including Fisher's Exact-Test on Influence of Specification Paradigms on Correct Self-Evaluation regarding Objective Correctness for Fundamentalists (Q1.2.2)

| Persona * Correct Estimation Crosstabulation |                      |                      |                    |       |  |  |
|--|----------------------|----------------------|--------------------|-------|--|--|
| Count  |                      | Correct Estimation   |                    | Total |  |  |
|  |                      | Incorrect Estimation | Correct Estimation |       |  |  |
| Persona                                      | Marginally Concerned | 32                   | 16                 | 48    |  |  |
|  | Amateur              | 40                   | 44                 | 84    |  |  |
|  | Lazy Expert          | 21                   | 23                 | 44    |  |  |
|  | Technician           | 20                   | 36                 | 56    |  |  |
|  | Fundamentalist       | 6                    | 6                  | 12    |  |  |
| Total  |                      | 119                  | 125                | 244   |  |  |

| Chi-Square Tests             |                     |    |                       |                      |                      |                   |
|------------------------------|---------------------|----|-----------------------|----------------------|----------------------|-------------------|
|                              | Value               | df | Asymp. Sig. (2-sided) | Exact Sig. (2-sided) | Exact Sig. (1-sided) | Point Probability |
| Pearson Chi-Square           | 10,045 <sup>a</sup> | 4  | ,040                  | ,039                 |                      |                   |
| Likelihood Ratio             | 10,206              | 4  | ,037                  | ,041                 |                      |                   |
| Fisher's Exact Test          | 10,081              |    |                       | ,038                 |                      |                   |
| Linear-by-Linear Association | 6,345 <sup>b</sup>  | 1  | ,012                  | ,012                 | ,007                 | ,002              |
| N of Valid Cases             | 244                 |    |                       |                      |                      |                   |

a. 0 cells (,0%) have expected count less than 5. The minimum expected count is 5,85.

b. The standardized statistic is 2,519.

Figure 145: Cross Tables including Fisher's Exact-Test on Influence of Persona on Correct Self-Evaluation regarding Objective Correctness (Q1.2.3)

Efficiency

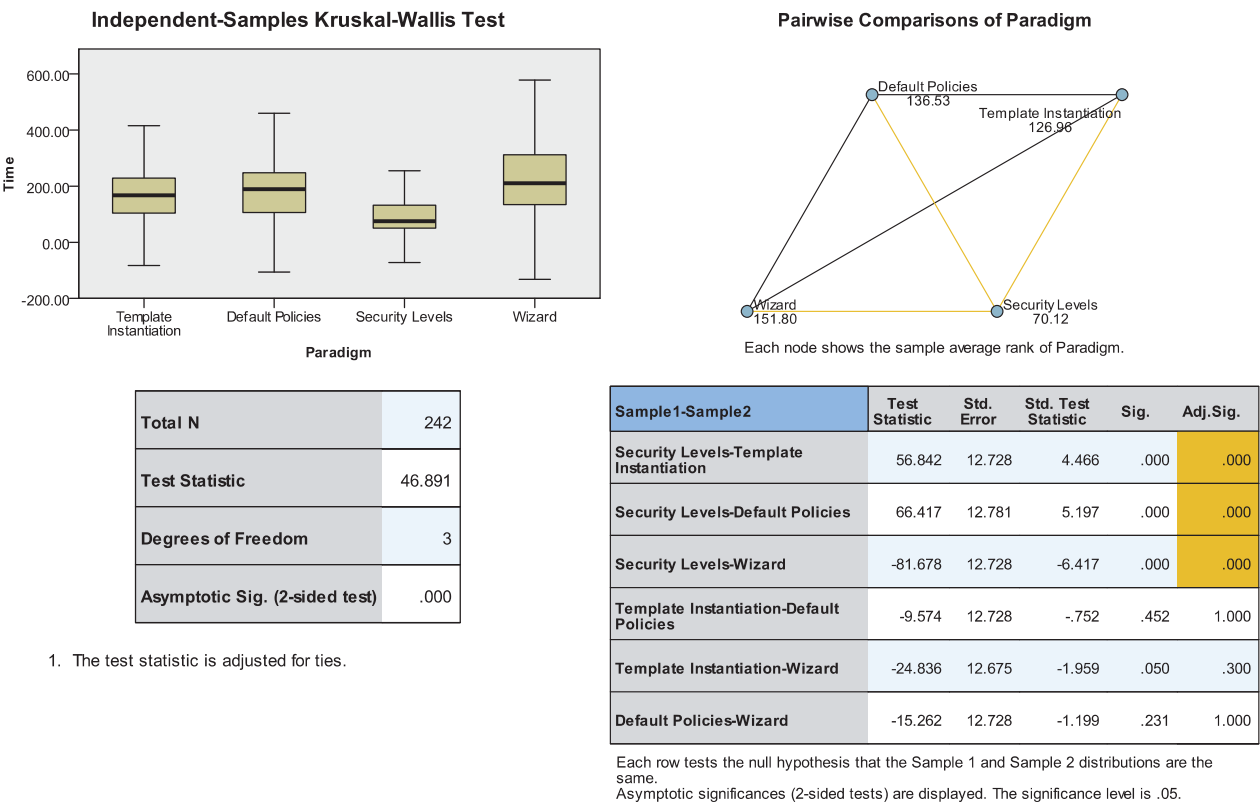


Figure 146: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time with Pairwise Comparison of Specification Paradigms (Q1.3.1)

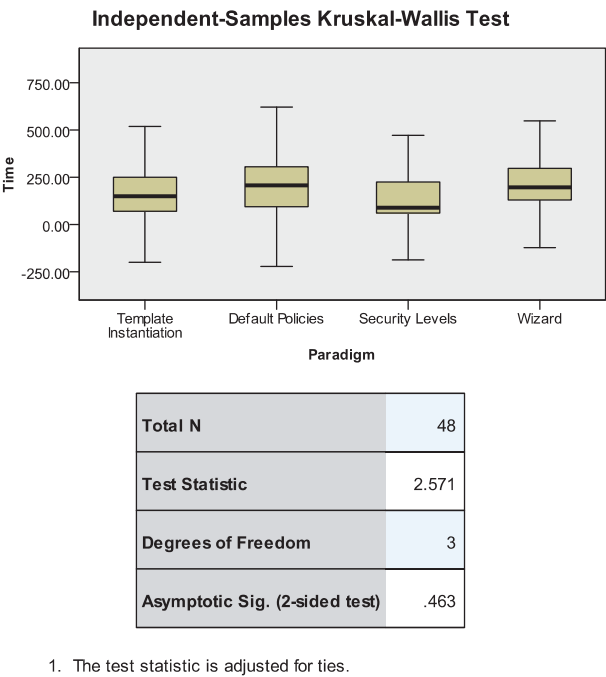


Figure 147: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Marginally Concerned (Q1.3.2)

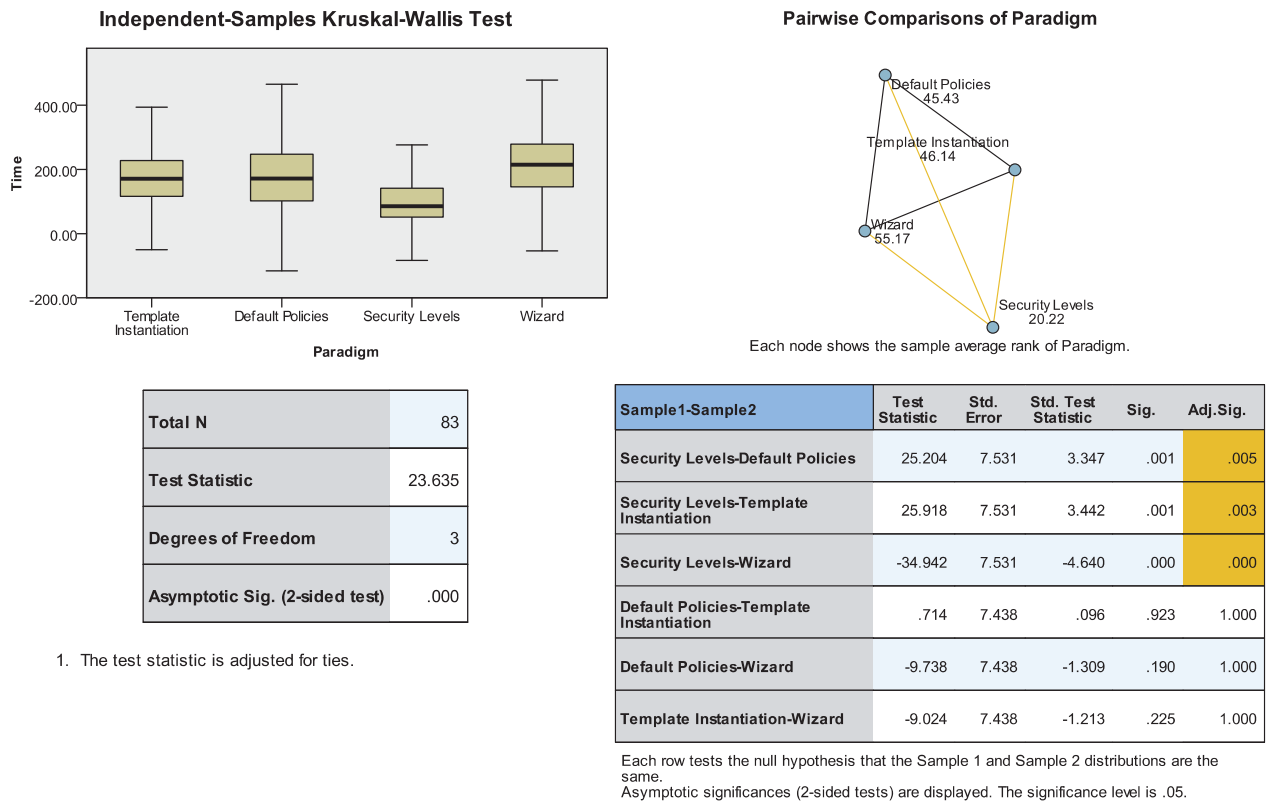


Figure 148: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Amateurs with Pairwise Comparison of Specification Paradigms (Q1.3.2)

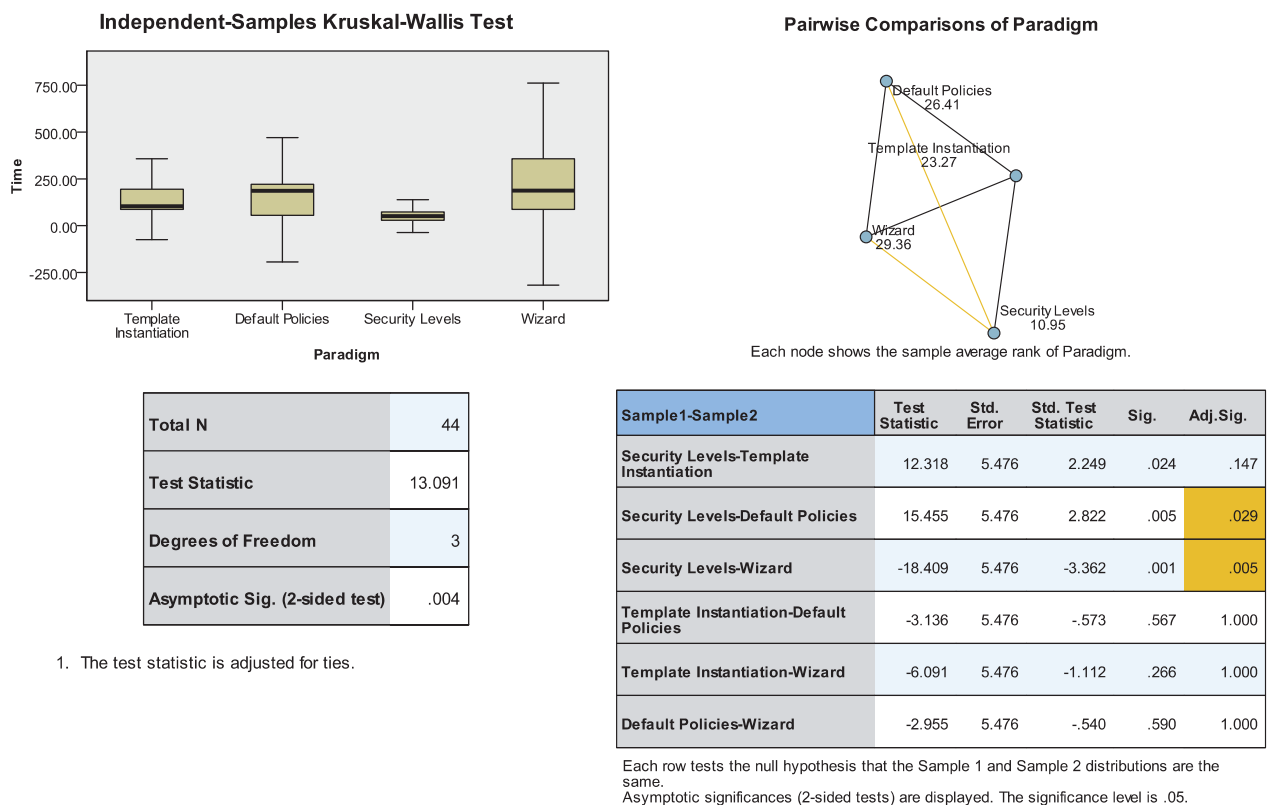


Figure 149: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Lazy Experts with Pairwise Comparison of Specification Paradigms (Q1.3.2)



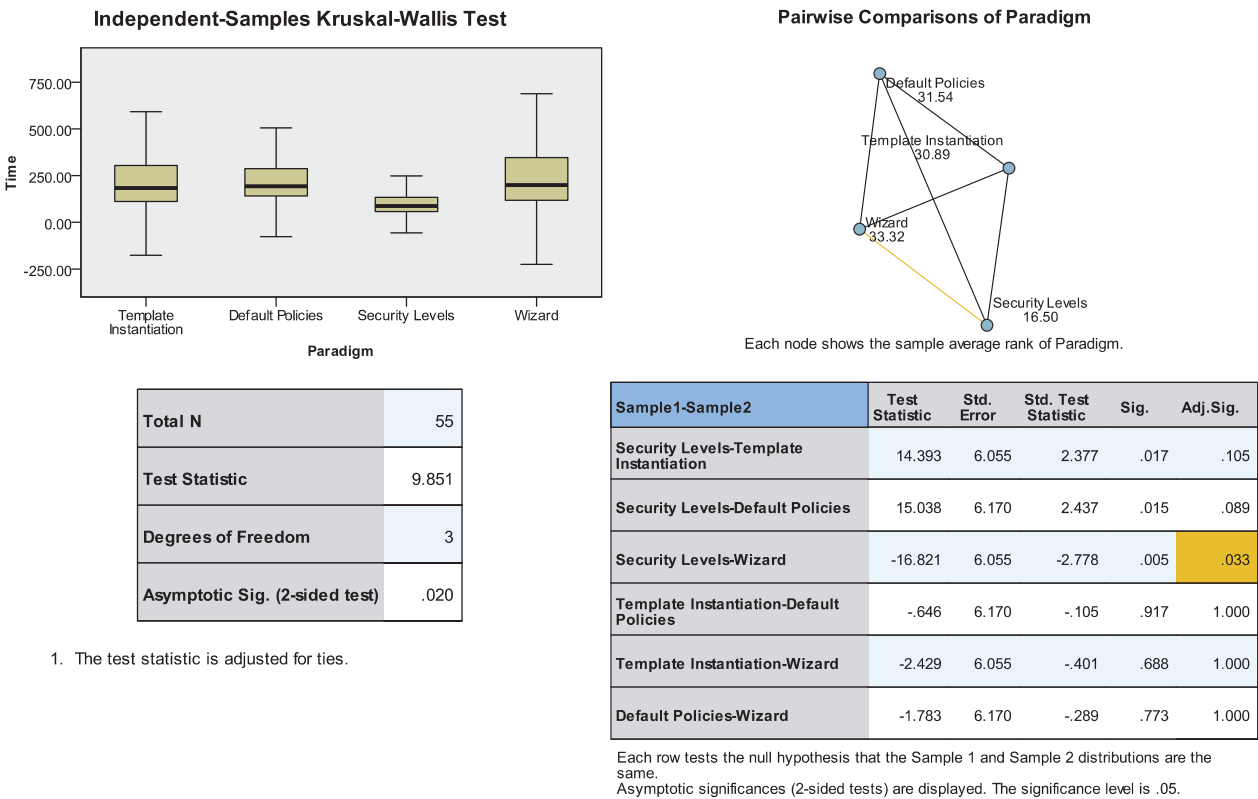


Figure 150: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Technicians with Pairwise Comparison of Specification Paradigms (Q1.3.2)

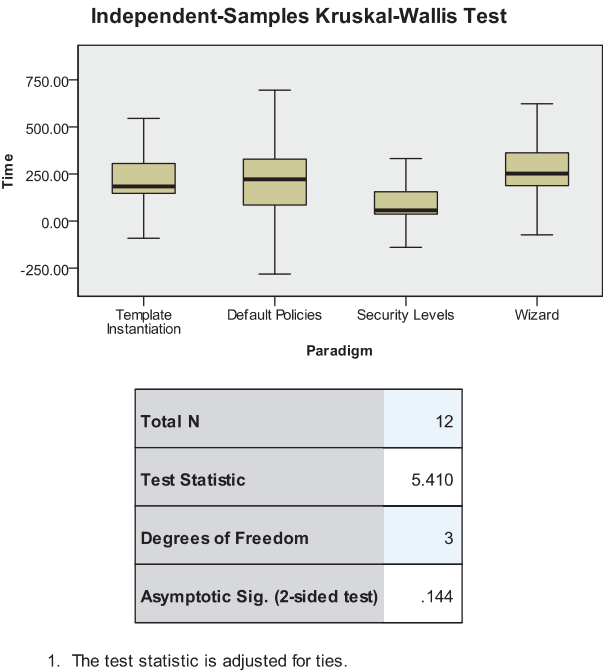
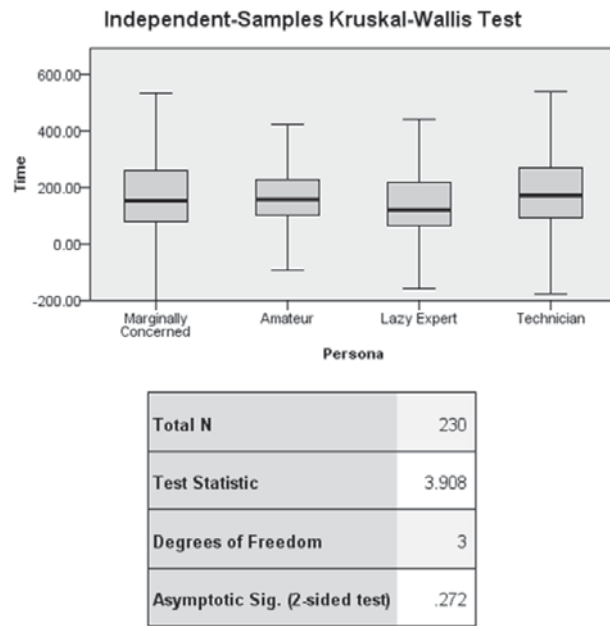


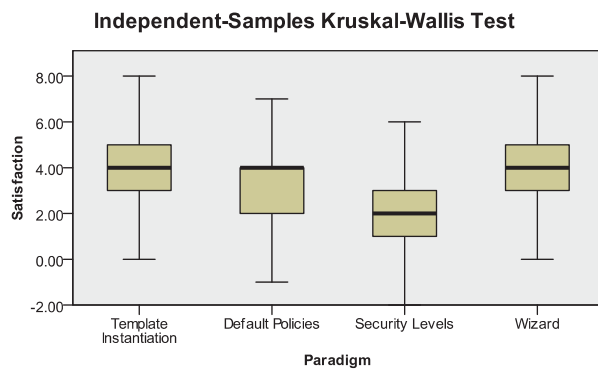
Figure 151: Kruskal-Wallis-Test on Influence of Specification Paradigms on Needed Time for Fundamentalists (Q1.3.2)



1. The test statistic is adjusted for ties.

Figure 152: Kruskal-Wallis-Test on Influence of Personas on Needed Time (Q1.3.3)

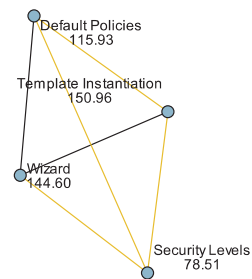
## Satisfaction



|                                       |        |
|---------------------------------------|--------|
| <b>Total N</b>                        | 244    |
| <b>Test Statistic</b>                 | 42.619 |
| <b>Degrees of Freedom</b>             | 3      |
| <b>Asymptotic Sig. (2-sided test)</b> | .000   |

1. The test statistic is adjusted for ties.

### Pairwise Comparisons of Paradigm



Each node shows the sample average rank of Paradigm.

| Sample1-Sample2                         | Test Statistic | Std. Error | Std. Test Statistic | Sig. | Adj. Sig. |
|---|----------------|------------|---------------------|------|-----------|
| Security Levels-Default Policies        | 37.426         | 12.400     | 3.018               | .003 | .015      |
| Security Levels-Wizard                  | -66.090        | 12.400     | -5.330              | .000 | .000      |
| Security Levels-Template Instantiation  | 72.451         | 12.400     | 5.843               | .000 | .000      |
| Default Policies-Wizard                 | -28.664        | 12.400     | -2.312              | .021 | .125      |
| Default Policies-Template Instantiation | 35.025         | 12.400     | 2.825               | .005 | .028      |
| Wizard-Template Instantiation           | 6.361          | 12.400     | .513                | .608 | 1.000     |

Each row tests the null hypothesis that the Sample 1 and Sample 2 distributions are the same. Asymptotic significances (2-sided tests) are displayed. The significance level is .05.

Figure 153: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction with Pairwise Comparison of Specification Paradigms (Q1.4.1)

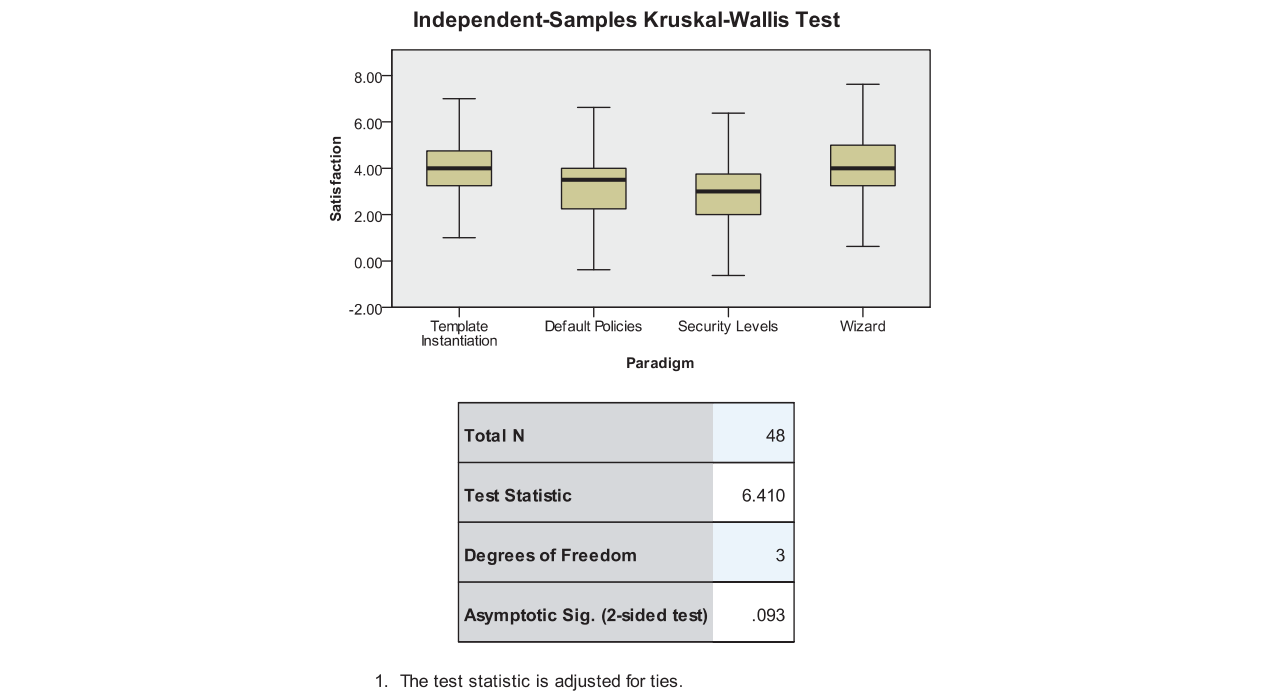


Figure 154: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Marginally Concerned (Q1.4.2)

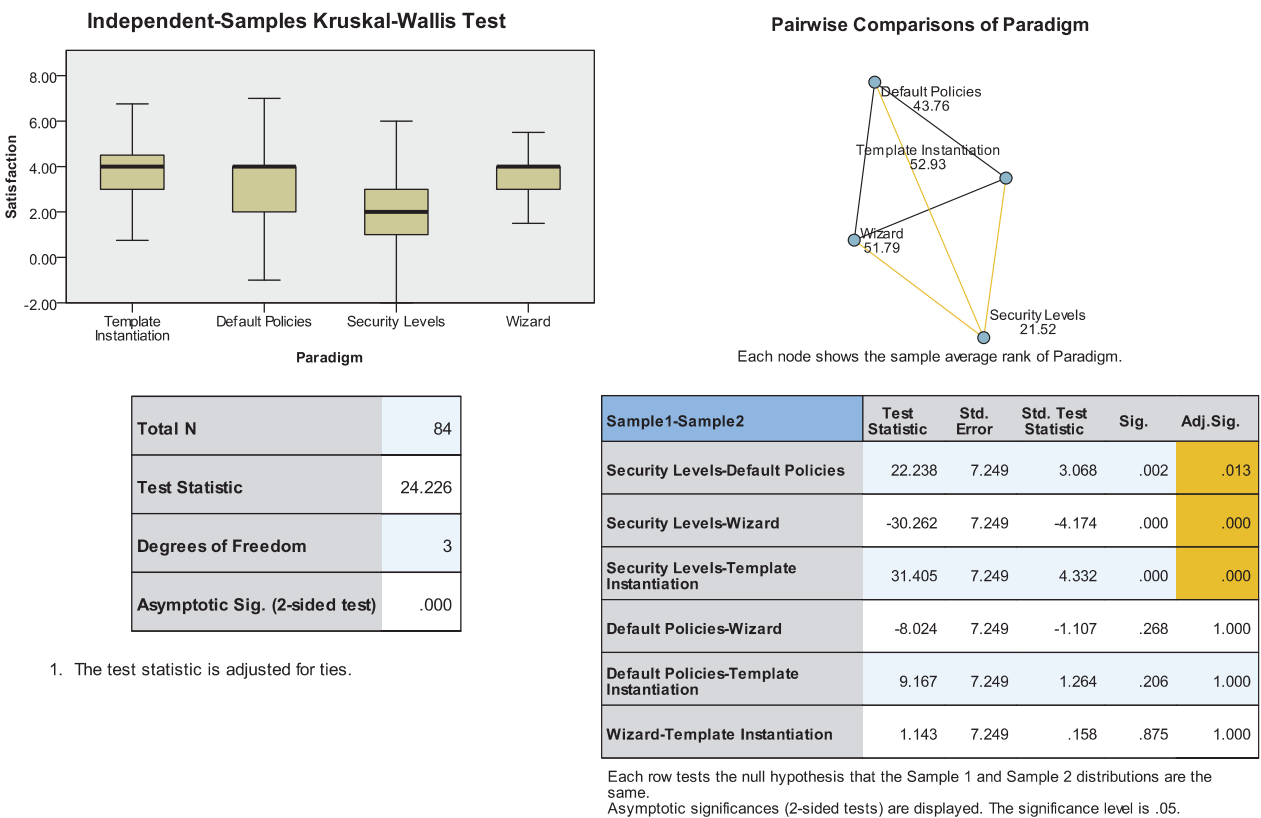


Figure 155: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Amateurs with Pairwise Comparison of Specification Paradigms (Q1.4.2)

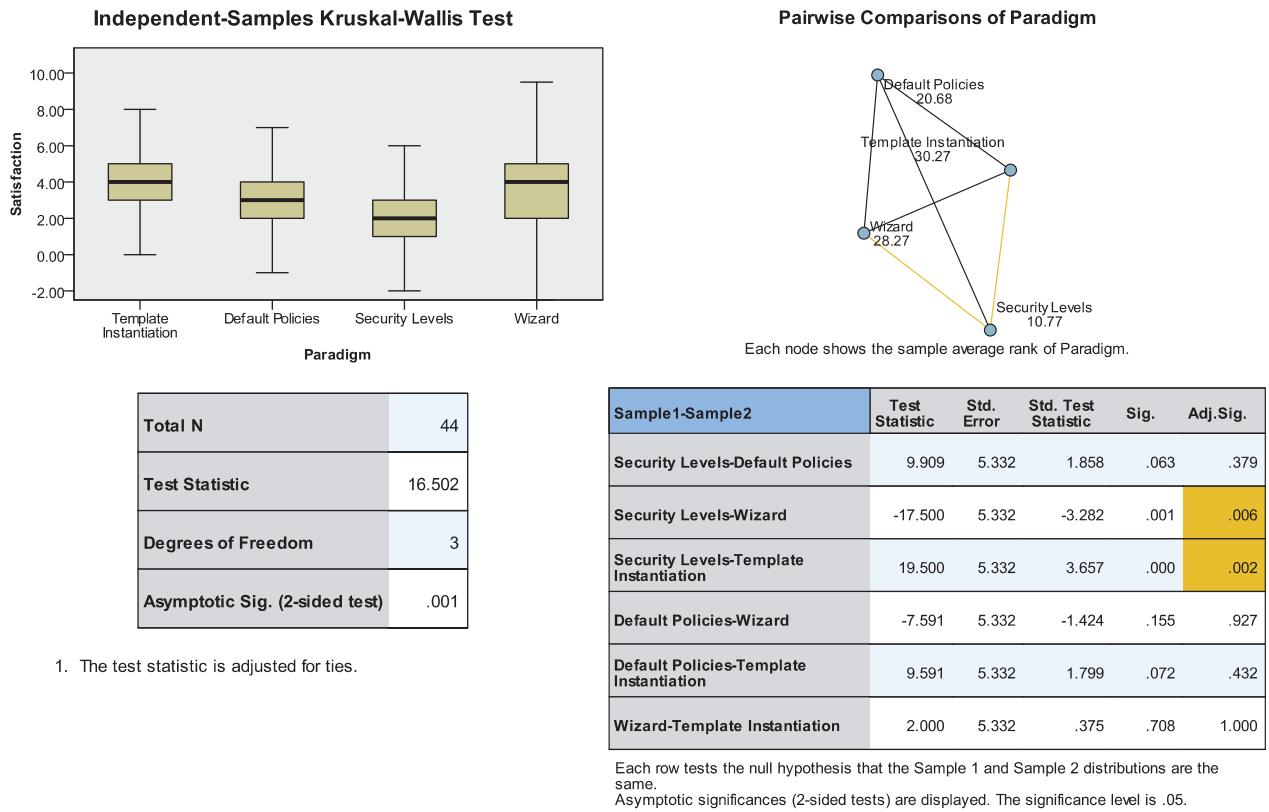


Figure 156: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Lazy Experts with Pairwise Comparison of Specification Paradigms (Q1.4.2)

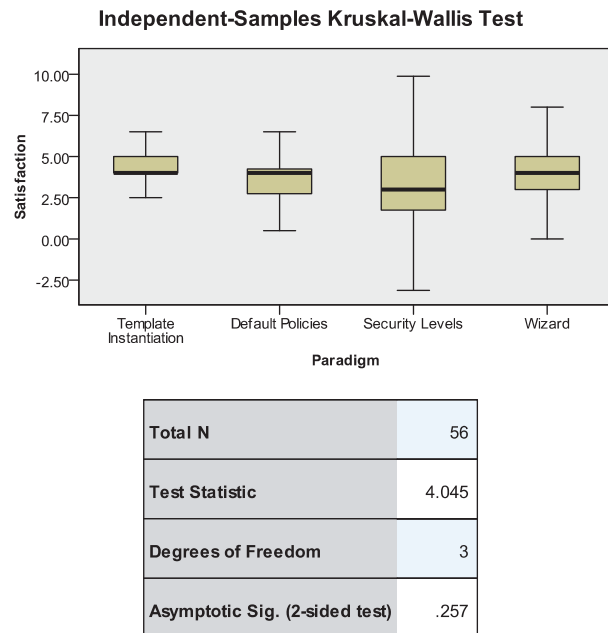
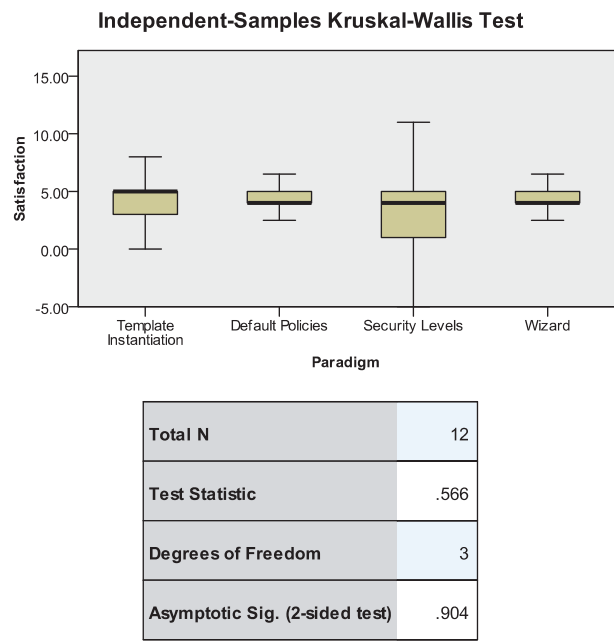
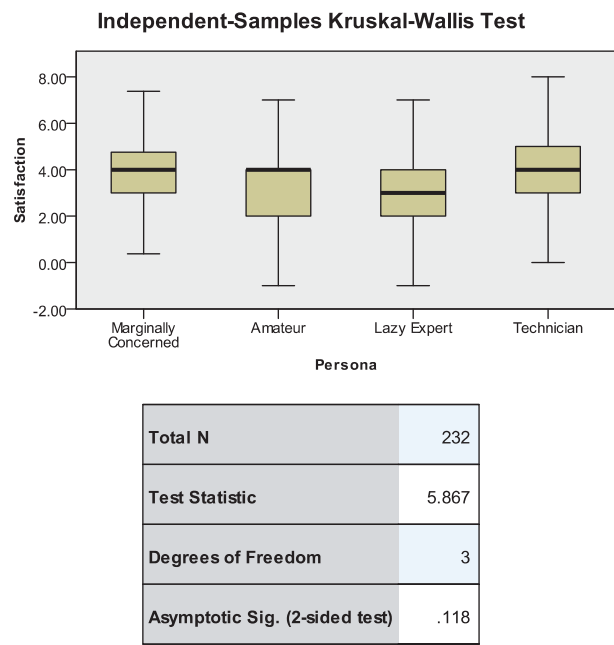


Figure 157: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Technicians (Q1.4.2)



1. The test statistic is adjusted for ties.

Figure 158: Kruskal-Wallis-Test on Influence of Specification Paradigms on Satisfaction for Fundamentalists (Q1.4.2)



1. The test statistic is adjusted for ties.

Figure 159: Kruskal-Wallis-Test on Influence of Personas on Satisfaction (Q1.4.3)

## **G.6 Raw Data**

Access to raw data of the policy specification experiment can be requested under <https://fordatis.fraunhofer.de/handle/fordatis/96> or via email ([primaerdaten@iese.fraunhofer.de](mailto:primaerdaten@iese.fraunhofer.de)). The email must contain full contact details of the requesting person, its institution, the reason for the request and the desired use of the data as well as the primary data identifier for the experiment data (PDI 53020).



# Lebenslauf

|                            |  |   |
|----------------------------|--|---|
| <b>Name</b>                | Manuel Rudolph                           |   |
| <b>Wohnort</b>             | Wallstadter Straße 18<br>68549 Ilvesheim |   |
| <b>Geburtsdatum</b>        | 16.11.1984                               |   |
| <b>Geburtsort</b>          | Heidelberg                               |   |
| <b>Familienstand</b>       | Verheiratet, 1 Kind                      |   |
| <b>Staatsangehörigkeit</b> | Deutsch                                  |   |
| <b>Schulbildung</b>        | 1991-1995                                | Graf-von-Oberndorff Grundschule in Edingen-Neckarhausen   |
|                            | 1995-2004                                | Elisabeth-von Thadden Gymnasium in Heidelberg<br>Abschluss: Abitur  |
| <b>Studium</b>             | 2004-2008                                | Bachelor Informatik - Hochschule Mannheim   |
|                            | 2008-2009                                | Master Informatik - Hochschule Mannheim   |
| <b>Berufstätigkeit</b>     | 2009-heute                               | Wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Experimentelles Software Engineering IESE, Kaiserslautern |

Kaiserslautern, den 11. Dezember 2019





# PhD Theses in Experimental Software Engineering

- Volume 1**      **Oliver Laitenberger** (2000), *Cost-Effective Detection of Software Defects Through Perspective-based Inspections*
- Volume 2**      **Christian Bunse** (2000), *Pattern-Based Refinement and Translation of Object-Oriented Models to Code*
- Volume 3**      **Andreas Birk** (2000), *A Knowledge Management Infrastructure for Systematic Improvement in Software Engineering*
- Volume 4**      **Carsten Tautz** (2000), *Customizing Software Engineering Experience Management Systems to Organizational Needs*
- Volume 5**      **Erik Kamsties** (2001), *Surfacing Ambiguity in Natural Language Requirements*
- Volume 6**      **Christiane Differding** (2001), *Adaptive Measurement Plans for Software Development*
- Volume 7**      **Isabella Wieczorek** (2001), *Improved Software Cost Estimation A Robust and Interpretable Modeling Method and a Comprehensive Empirical Investigation*
- Volume 8**      **Dietmar Pfahl** (2001), *An Integrated Approach to Simulation-Based Learning in Support of Strategic and Project Management in Software Organisations*
- Volume 9**      **Antje von Knethen** (2001), *Change-Oriented Requirements Traceability Support for Evolution of Embedded Systems*
- Volume 10**    **Jürgen Münch** (2001), *Muster-basierte Erstellung von Software-Projektplänen*
- Volume 11**    **Dirk Muthig** (2002), *A Light-weight Approach Facilitating an Evolutionary Transition Towards Software Product Lines*
- Volume 12**    **Klaus Schmid** (2003), *Planning Software Reuse – A Disciplined Scoping Approach for Software Product Lines*
- Volume 13**    **Jörg Zettel** (2003), *Anpassbare Methodenassistenz in CASE-Werkzeugen*
- Volume 14**    **Ulrike Becker-Kornstaedt** (2004), *Prospect: a Method for Systematic Elicitation of Software Processes*
- Volume 15**    **Joachim Bayer** (2004), *View-Based Software Documentation*
- Volume 16**    **Markus Nick** (2005), *Experience Maintenance through Closed-Loop Feedback*

- Volume 17**     **Jean-François Girard** (2005), *ADORE-AR: Software Architecture Reconstruction with Partitioning and Clustering*
- Volume 18**     **Ramin Tavakoli Kolagari** (2006), *Requirements Engineering für Software-Produktlinien eingebetteter, technischer Systeme*
- Volume 19**     **Dirk Hamann** (2006), *Towards an Integrated Approach for Software Process Improvement: Combining Software Process Assessment and Software Process Modeling*
- Volume 20**     **Bernd Freimut** (2006), *MAGIC: A Hybrid Modeling Approach for Optimizing Inspection Cost-Effectiveness*
- Volume 21**     **Mark Müller** (2006), *Analyzing Software Quality Assurance Strategies through Simulation. Development and Empirical Validation of a Simulation Model in an Industrial Software Product Line Organization*
- Volume 22**     **Holger Diekmann** (2008), *Software Resource Consumption Engineering for Mass Produced Embedded System Families*
- Volume 23**     **Adam Trendowicz** (2008), *Software Effort Estimation with Well-Founded Causal Models*
- Volume 24**     **Jens Heidrich** (2008), *Goal-oriented Quantitative Software Project Control*
- Volume 25**     **Alexis Ocampo** (2008), *The REMIS Approach to Rationale-based Support for Process Model Evolution*
- Volume 26**     **Marcus Trapp** (2008), *Generating User Interfaces for Ambient Intelligence Systems; Introducing Client Types as Adaptation Factor*
- Volume 27**     **Christian Denger** (2009), *SafeSpection – A Framework for Systematization and Customization of Software Hazard Identification by Applying Inspection Concepts*
- Volume 28**     **Andreas Jedlitschka** (2009), *An Empirical Model of Software Managers' Information Needs for Software Engineering Technology Selection  
A Framework to Support Experimentally-based Software Engineering Technology Selection*
- Volume 29**     **Eric Ras** (2009), *Learning Spaces: Automatic Context-Aware Enrichment of Software Engineering Experience*
- Volume 30**     **Isabel John** (2009), *Pattern-based Documentation Analysis for Software Product Lines*
- Volume 31**     **Martín Soto** (2009), *The DeltaProcess Approach to Systematic Software Process Change Management*
- Volume 32**     **Ove Armbrust** (2010), *The SCOPE Approach for Scoping Software Processes*

- Volume 33**     **Thorsten Keuler** (2010), *An Aspect-Oriented Approach for Improving Architecture Design Efficiency*
- Volume 34**     **Jörg Dörr** (2010), *Elicitation of a Complete Set of Non-Functional Requirements*
- Volume 35**     **Jens Knodel** (2010), *Sustainable Structures in Software Implementations by Live Compliance Checking*
- Volume 36**     **Thomas Patzke** (2011), *Sustainable Evolution of Product Line Infrastructure Code*
- Volume 37**     **Ansgar Lamersdorf** (2011), *Model-based Decision Support of Task Allocation in Global Software Development*
- Volume 38**     **Ralf Carbon** (2011), *Architecture-Centric Software Producibility Analysis*
- Volume 39**     **Florian Schmidt** (2012), *Funktionale Absicherung kamerabasierter Aktiver Fahrerassistenzsysteme durch Hardware-in the-Loop-Tests*
- Volume 40**     **Frank Elberzhager** (2012), *A Systematic Integration of Inspection and Testing Processes for Focusing Testing Activities*
- Volume 41**     **Matthias Naab** (2012), *Enhancing Architecture Design Methods for Improved Flexibility in Long-Living Information Systems*
- Volume 42**     **Marcus Ciolkowski** (2012), *An Approach for Quantitative Aggregation of Evidence from Controlled Experiments in Software Engineering*
- Volume 43**     **Igor Menzel** (2012), *Optimizing the Completeness of Textual Requirements Documents in Practice*
- Volume 44**     **Sebastian Adam** (2012), *Incorporating Software Product Line Knowledge into Requirements Processes*
- Volume 45**     **Kai Höfig** (2012), *Failure-Dependent Timing Analysis – A New Methodology for Probabilistic Worst-Case Execution Time Analysis*
- Volume 46**     **Kai Breiner** (2013), *AssistU – A framework for user interaction forensics*
- Volume 47**     **Rasmus Adler** (2013), *A model-based approach for exploring the space of adaptation behaviors of safety-related embedded systems*
- Volume 48**     **Daniel Schneider** (2014), *Conditional Safety Certification for Open Adaptive Systems*
- Volume 49**     **Michail Anastasopoulos** (2013), *Evolution Control for Software Product Lines: An Automation Layer over Configuration Management*
- Volume 50**     **Bastian Zimmer** (2014), *Efficiently Deploying Safety-Critical Applications onto Open Integrated Architectures*

- Volume 51**      **Slawomir Duszynski** (2015), *Analyzing Similarity of Cloned Software Variants using Hierarchical Set Models*
- Volume 52**      **Zhensheng Guo** (2015), *Safe Requirements Engineering: A Scenario-based Approach for Identifying Complete Safety-oriented Requirements*
- Volume 53**      **Bo Zhang** (2015), *VITAL – Reengineering Variability Specifications and Realizations in Software Product Lines*
- Volume 54**      **Norman Riegel** (2016), *Prioritization in Incremental Requirements Engineering*
- Volume 55**      **Pablo Oliveira Antonino de Assis** (2016), *Improving the Consistency and Completeness of Safety Requirements Specifications*
- Volume 56**      **Thomas Bauer** (2016), *Enabling Functional Integration Testing by Using Heterogeneous Models*
- Volume 57**      **Michael Kläss** (2016), *HyDEEP: Transparent Combination of Measurement and Expert Data for Defect Prediction*
- Volume 58**      **Liliana Katherine Guzmán Rehbein** (2017), *Empirically-based Method for Performing Qualitative Synthesis in Software Engineering*
- Volume 59**      **Michael Roth** (2017), *Qualitative Reliability Analysis of Software-Controlled Systems using State/Event Fault Trees*
- Volume 60**      **Hadil Abukwaik** (2017), *Proactive Support for Conceptual Interoperability Analysis of Software Units*
- Volume 61**      **Konstantin Holl** (2018), *Quality Assurance for Mobile Business Applications*
- Volume 62**      **Dominik Rost** (2019), *Task-Specific Architecture Documentation for Developers*
- Volume 63**      **Christian Jung** (2019), *Context-aware Security*
- Volume 64**      **Andreas Maier** (2019), *Identification and Specification of Hedonic Quality in User Requirements*
- Volume 65**      **Philipp Diebold** (2019), *Agile Practice Experience Repository for Process Improvement*
- Volume 66**      **Binish Tanveer** (2019), *Utilizing Change Impact Analysis for Improving Effort Estimation in Agile Software Development*
- Volume 67**      **Patrik Feth** (2019), *Dynamic Behavior Risk Assessment for Autonomous Systems*
- Volume 68**      **Manuel Rudolph** (2019), *Generation of Usable Policy Administration Points for Security and Privacy*

Software Engineering has become one of the major foci of Computer Science research in Kaiserslautern, Germany. Both the University of Kaiserslautern's Computer Science Department and the Fraunhofer Institute for Experimental Software Engineering (IESE) conduct research that subscribes to the development of complex software applications based on engineering principles. This requires system and process models for managing complexity, methods and techniques for ensuring product and process quality, and scalable formal methods for modeling and simulating system behavior. To understand the potential and limitations of these technologies, experiments need to be conducted for quantitative and qualitative evaluation and improvement. This line of software engineering research, which is based on the experimental scientific paradigm, is referred to as 'Experimental Software Engineering'.

In this series, we publish PhD theses from the Fraunhofer Institute for Experimental Software Engineering (IESE) and from the Software Engineering Research Groups of the Computer Science Department at the University of Kaiserslautern. PhD theses that originate elsewhere can be included, if accepted by the Editorial Board.

Editor-in-Chief: Prof. Dr. Dieter Rombach

Executive Consultant & Founding Director of Fraunhofer IESE and Head of the AGSE Group of the Computer Science Department, University of Kaiserslautern

Editorial Board Member: Prof. Dr. Peter Liggesmeyer

Director of Fraunhofer IESE and Head of the SEDA Group of the Computer Science Department, University of Kaiserslautern

Editorial Board Member: Prof. Dr. Frank Bomarius

Deputy Director of Fraunhofer IESE and Professor for Computer Science at the Department of Engineering, University of Applied Sciences, Kaiserslautern

ISBN 978-3-8396-1579-9

