An Ontology-Based Approach Supporting Adaptive and Context-Aware Information Provision in Ubiquitous Mobility Systems

Von der Fakultät Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Christine Keller

aus Stuttgart

Hauptberichter:	Prof. Dr. Thomas Schlegel
Mitberichter:	Prof. Dr. Niels Henze
	Prof. Dr. Thomas Ertl
Tag der mündlichen Prüfung:	24.03.2023

Institut für Visualisierung und Interaktive Systeme
der Universität Stuttgart

2023

Contents

List of Listings xiv

Abstract xxi

German Abstract — Zusammenfassung xxiii

1 Introduction 1
- 1.1 Problem Statement . . . 3
- 1.2 Research Questions . . . 4
- 1.3 Contributions . . . 5
- 1.4 Outline . . . 8

2 Theoretical Foundations 11
- 2.1 Intelligent and Ubiquitous Systems . . . 11
- 2.2 HCI in Ubiquitous Systems . . . 14
  - 2.2.1 Ubiquitous Interaction Paradigms . . . 14
  - 2.2.2 Ubiquitous Interactive Devices . . . 16
  - 2.2.3 Model-Based Adaptive User Interfaces . . . 17
  - 2.2.4 Usability in Ubiquitous Systems . . . 17
  - 2.2.5 Usability Evaluation of Ubiquitous Systems . . . 20
- 2.3 Semantic Web Technologies . . . 26
  - 2.3.1 Ontologies . . . 26
- 2.4 Context . . . 32
  - 2.4.1 Context Lifecycle . . . 34
- 2.5 Context-Awareness and Adaptive Systems . . . 42
  - 2.5.1 Definitions of Context-Aware Systems . . . 42
  - 2.5.2 What to Adapt . . . 43
  - 2.5.3 When to adapt . . . 46
  - 2.5.4 How to Adapt . . . 47

3 Requirements Analysis 51
- 3.1 Public Transport Systems . . . 52
- 3.2 Mobility Situations and Tasks . . . 56
- 3.3 Persona and Scenario Approach . . . 59
- 3.4 Requirements . . . 71
  - 3.4.1 *Why* Adapt? . . . 72
  - 3.4.2 *When* to Adapt? . . . 73
  - 3.4.3 *What* to Adapt? . . . 73

3.4.4 Adapt *To What*? 74
3.4.5 *Who* Adapts? 75
3.4.6 *How* to Adapt? 75
3.4.7 *Where* to Adapt? 77

4 Related Work 81
4.1 Architectural Approaches 82
4.1.1 General Architectural Approaches 82
4.1.2 Self-Adaptive System Approaches 84
4.2 User Interface Approaches 86
4.2.1 Model-Driven Adaptive User Interfaces 86
4.2.2 Plastic Interfaces 92
4.2.3 Models and Languages for Adaptive User Interfaces . . 98
4.3 Discussion 99

5 Concept 103
5.1 Objective 104
5.2 General Approach 105
5.2.1 Limitations of this Approach 106
5.2.2 Architecture 107

6 Ontologies 111
6.1 Context Analysis 113
6.1.1 Context Dimensions and Context Ontologies 113
6.1.2 Context Analysis 117
6.2 Public Transport Ontology 125
6.3 Task Ontology 132
6.4 Interaction Ontology 134
6.5 Device Ontology 137
6.6 Context Ontology 140

7 Autonomous Usability Assessment and Output Adaptation 145
7.1 Usability Analysis 147
7.1.1 Literature Review and Filtering 148
7.1.2 Clustering of Criteria and Identification of Characteristics 151
7.2 Ontologies for Usability 156
7.2.1 Message Specification Ontology 157
7.2.2 Representing Usability Qualities in an Ontology 158
7.3 Assessment Rules 169
7.4 The Decision Making Process 185
7.4.1 Determining Available Output Options 186

7.4.2 Usability Assessment . . . 188
7.4.3 Decision Process . . . 189

8 Implementation 195
8.1 Context Generation . . . 196
8.2 Usability Assessment . . . 198
8.2.1 Data Model . . . 198
8.2.2 Message Content Analysis . . . 198
8.2.3 Usability Assessments and Filters . . . 199

9 Evaluation 203
9.1 Proof of Concept Implementation . . . 204
9.2 Performance Evaluation . . . 205
9.3 Usability Evaluation . . . 207
9.3.1 Survey Setup . . . 209
9.3.2 Results . . . 212
9.3.3 Discussion . . . 221
9.4 Summary . . . 222

10 Discussion and Outlook 225
10.1 Summary . . . 225
10.2 Answers to the Research Questions . . . 228
10.3 Extendability and Generalizability of the Approach . . . 230
10.4 Limitations . . . 232
10.5 Outlook . . . 234

A Appendix Analysis Results 237
A.1 Hierarchical Task Analysis Results . . . 237
A.1.1 T1: Boarding a Public Transport Vehicle . . . 237
A.1.2 T2: Boarding a Motorized Vehicle . . . 239
A.1.3 T3: Boarding a Non-Motorized Vehicle . . . 240
A.1.4 T4: Boarding a Car Sharing or Rental Car . . . 241
A.1.5 T5: Boarding a Non-Motorized Sharing Vehicle . . . 241
A.1.6 T6: Boarding a Car as Passenger . . . 242
A.1.7 T7: In Transit in a Public Transport Vehicle . . . 243
A.1.8 T8: In Transit - Driving a Vehicle . . . 243
A.1.9 T9: Getting a Lift . . . 244
A.1.10 T10: In Transit - Walking . . . 244
A.1.11 T11: Alighting a Public Transport Vehicle . . . 244
A.1.12 T12: Alighting an Individual Transport Vehicle (as Driver) . . . 245

A.1.13 T13: Alighting from a Car as a Passenger 246
A.1.14 T14: Alighting from a Car Sharing or Rental Car 246
A.1.15 T15: Alighting from a Non-Motorized Sharing Vehicle . 247
A.1.16 T16: Wait at Station or Stop 248
A.1.17 T17: Waiting for a Lift 249
A.1.18 T18: Dealing with a Delay in Public Transport 249
A.1.19 T19: Dealing with a Delay in Individual Transport 251
A.1.20 T20: Dealing with a Disruption in Public Transport: 252
A.1.21 T21: Dealing with a Disruption in Individual Transport . 252
A.1.22 Context Analysis for Task Analysis Results 253

B Appendix: Usability Analysis and Rules 259
B.1 Usability Analysis 259
B.2 Usability Assessment and Filter Rules 261
B.2.1 Consistency Rules 262
B.2.2 Perceptibility Rules 262
B.2.3 Privacy Rules 265
B.2.4 Message Sensitivity Rules 267
B.2.5 Unobtrusiveness Rules 268
B.2.6 Relatability Rules 270
B.2.7 Relevance Rules 271
B.2.8 Urgency Rules 272
B.2.9 Message Requirement Rules 273
B.2.10 Perceptibility Filter Rules 273
B.2.11 Privacy and Sensitivity Filter 274
B.2.12 Relatability and Relevance Filter 276

C Appendix: Evaluation Results 279
C.1 Evaluation Scenarios 279
C.1.1 Scenario 1 279
C.1.2 Scenario 2 281
C.1.3 Scenario 3 282
C.1.4 Scenario 4 284
C.1.5 Scenario 5 285
C.1.6 Scenario 6 286
C.2 Performance Evaluation Results 288
C.3 Results Online Survey 288

Bibliography 303

List of Figures

Chapter 2

2.1 An Resource Description Framework (RDF) example graph. 28
2.2 A visual notation for ontologies. 32
2.3 Context lifecycle. 35
2.4 Options for adaptation. 45
2.5 Decision making techniques. 48

Chapter 3

3.1 System components in Public Transport. 53
3.2 An example of digital passenger information 55
3.3 A illustration of terminology 57
3.4 Trip phases in public transport. 58
3.5 Two personas. 61
3.6 Two personas. 62
3.7 Two personas. 62
3.8 Two personas. 63
3.9 Mockups to illustrate scenario 1. 65
3.10 Mockups to illustrate scenario 2. 66
3.11 Mockup to illustrate scenario 2. 67
3.12 Mockup to illustrate scenario 3. 68
3.13 Mockup to illustrate scenario 4. 70
3.14 Mockup to illustrate scenario 4. 70

Chapter 5

5.1 Adaptive ubiquitous mobility system architecture. 107

Chapter 6

6.1 Context dimensions after [Schlegel and Keller, 2011] and [Kühn et al., 2011] 114
6.2 HTA Task 1. 119
6.3 Hierarchical task analysis for riding a public transport vehicle. . . 120
6.4 HTA Task 11. 120
6.5 HTA Task 16. 121
6.6 HTA Task 18. 121

6.7 HTA Task 20. 122
6.8 Contexts in task 16. 122
6.9 Context factors in smart ubiquitous mobility. 123
6.10 Context ontologies. 124
6.11 Location ontology. 127
6.12 Trip class. 128
6.13 Timed trip leg class. 128
6.14 Continuous trip leg class. 129
6.15 Interchange trip leg class. 130
6.16 StopPlace class. 130
6.17 StopPoint class. 131
6.18 Vehicle class. 131
6.19 The task taxonomy. 133
6.20 Details of the task ontology. 134
6.21 Modality classes. 136
6.22 The device taxonomy. 138
6.23 Display class. 138
6.24 Functions of devices. 139
6.25 Output converter class. 139
6.26 Output option class. 139
6.27 Device context. 141
6.28 Specific smartphone context. 141
6.29 Vehicle context. 142
6.30 Static user context. 142
6.31 User location and task context. 143
6.32 Device user context. 143

Chapter 7

7.1 Analysis for usability ontology. 147
7.2 Clustering of usability attributes 152
7.3 Message specification. 158
7.4 Assessable elements. 159
7.5 Properties to express rating results. 159
7.6 BooleanRatingResult property. 160
7.7 CategoricalRatingResult property. 162
7.8 Usability qualities. 162
7.9 The `Perceptibility` class. 163
7.10 The `Unobtrusiveness` class. 164
7.11 The `Sensitivity` class. 165
7.12 The `Privacy` class. 167

7.13 The `Relatability` class. 167
7.14 The `Relevance` class. 168
7.15 The `Urgency` class. 169
7.16 Decision process. 185
7.17 Available output. 187
7.18 Usability assessment. 188
7.19 Message assessment. 189
7.20 Process for output decision. 190

Chapter 8

8.1 Architecture of the proof of concept prototype. 196

Chapter 9

9.1 Performance evaluation scenario 1. 206
9.2 Scenario page of the online questionnaire. 211
9.3 Second scenario page of the online questionnaire. 212
9.4 Demographic details: size of area of residency. 213
9.5 Participant's output choices for scenario 1. 216
9.6 Participant's output choices for scenario 5. 217
9.7 Agreement to statement 1. 219
9.8 Agreement to statement 4. 219
9.9 Agreement to statement 6. 220
9.10 Agreement to statement 7. 221

Chapter A

A.1 HTA task 1. 237
A.2 HTA task 2. 239
A.3 HTA task 3. 240
A.4 HTA task 4. 241
A.5 HTA task 5. 241
A.6 HTA task 6. 242
A.7 HTA task 7. 243
A.8 HTA task 8. 243
A.9 HTA task 10. 244
A.10 HTA task 11. 244
A.11 HTA task 12. 245
A.12 HTA task 13. 246
A.13 HTA task 14. 246

A.14 HTA task 15. 247
A.15 HTA task 16. 248
A.16 HTA task 17. 249
A.17 HTA task 18. 250
A.18 HTA task 19. 251
A.19 HTA task 20. 252
A.20 HTA task 21. 252
A.21 Context analysis for task T1. 253
A.22 Context analysis for task T2. 253
A.23 Context analysis for task T3. 254
A.24 Context analysis for task T4. 254
A.25 Context analysis for task T5. 254
A.26 Context analysis for task T6. 255
A.27 Context analysis for task T7. 255
A.28 Context analysis for task T8. 255
A.29 Context analysis for task T10. 255
A.30 Context analysis for task T11. 256
A.31 Context analysis for task T12. 256
A.32 Context analysis for task T14. 256
A.33 Context analysis for task T15. 256
A.34 Context analysis for task T17. 257
A.35 Context analysis for task T18. 257
A.36 Context analysis for task T19. 257
A.37 Context analysis for task T20. 258
A.38 Context analysis for task T21. 258

Chapter C

C.1 Questionnaire illustration of scenario 1. 280
C.2 Questionnaire illustration of scenario 2. 282
C.3 Questionnaire illustration of scenario 3. 283
C.4 Questionnaire illustration of scenario 4. 284
C.5 Questionnaire illustration of scenario 5. 286
C.6 Questionnaire illustration of scenario 6. 287
C.7 Performance evaluation scenario 1. 288
C.8 Performance evaluation scenario 2. 289
C.9 Performance evaluation scenario 3. 289
C.10 Performance evaluation scenario 4. 290
C.11 Performance evaluation scenario 5. 290
C.12 Performance evaluation scenario 6. 291
C.13 Demographic details: size of area of residency 291

C.14 Demographic details: local knowledge 292
C.15 Frequency of public transport usage. 292
C.16 Demographic details: public transport knowledge 292
C.17 Demographic details: trip purpose 293
C.18 Demographic details: passenger information media. 293
C.19 Output options scenario 1. 294
C.20 Output options scenario 2. 294
C.21 Output options scenario 3. 295
C.22 Output options scenario 4. 295
C.23 Output options scenario 5. 296
C.24 Output options scenario 6. 296
C.25 Agreement to statement 1. 297
C.26 Agreement to statement 2. 297
C.27 Agreement to statement 3. 298
C.28 Agreement to statement 4. 299
C.29 Agreement to statement 5. 299
C.30 Agreement to statement 6. 300
C.31 Agreement to statement 7. 300
C.32 Agreement to statement 8. 301

List of Tables

3.1 An overview over all requirements. 80

4.1 Related work comparison. 83
4.2 Related work comparison. 93
4.2 Related work comparison. 94
4.2 Related work comparison. 95

Chapter 5

5.1 A summary of design choices for the concept 105

7.1 Usability attributes. 150

9.1 Requirements overview and match. 224

A.1 HTA table for task 1. 238
A.2 HTA table for task 2. 239
A.3 HTA table for task 3. 240
A.4 HTA table for task 4. 241
A.5 HTA table for task 5. 242
A.6 HTA table for task 6. 242
A.7 HTA table for task 7. 243
A.8 HTA table for task 8. 243
A.9 HTA table for task 11. 244
A.10 HTA table for task 12. 245
A.11 HTA table for task 13. 246
A.12 HTA table for task 14. 246
A.12 HTA table for task 14. 247
A.13 HTA table for task 15. 247
A.14 HTA table for task 16. 248
A.15 HTA table for task 17. 249
A.16 HTA table for task 18. 249
A.16 HTA table for task 18. 250
A.17 HTA table for task 19. 251
A.18 HTA table for task 20. 252
A.19 HTA table for task 21. 253

B.1 Usability attributes. 259
B.1 Usability attributes. 260

Tables

B.1 Usability attributes. 261

List of Listings

2.1 A RDF statement. 27
2.2 Several RDF statements, expressing facts about children's books and authors. 27
2.3 A SPARQL Protocol and RDF Query Language (SPARQL) SELECT query for the data shown in listing 2.2 (extended by an example namespace). 29
2.4 A simple Resource Description Framework Schema (RDFS) ontology and RDF statements in Turtle format using an example namespace. 30
6.1 Namespaces and prefixes of the public transport ontologies (SPARQL notation). 126
6.2 Namespace and prefix of the task ontology (SPARQL notation). 132
6.3 Namespace and prefix of the interaction ontology (SPARQL notation). 134
6.4 Namespace and prefix of the device ontology (SPARQL notation). 137
6.5 Namespace and prefix of the context ontology (SPARQL notation). 140
7.1 The namespace and prefix of the message ontology in SPARQL notation. 157
7.2 The namespace and prefix of the usability ontology in SPARQL notation. 158

List of Abbreviations and Acronyms

IoT	Internet of Things
API	Application Programming Interface
OWL	Web Ontology Language
SWRL	Semantic Web Rule Language
RFID	Radio Frequency Identification
NFC	Near Field Communication
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
XML	Extensible Markup Language
SPARQL	SPARQL Protocol and RDF Query Language
OWL-DL	Web Ontology Language - Description Logic
DL	Description Logic
OSGi	Open Services Gateway Initiative
REST	Representational state transfer
MDA	Model-Driven Architecture
MDE	Model-Driven Engineering
UML	Unified Modeling Language
ER	Entity-Relationship
CC/PP	Composite Capabilities / Preference Profile
SGML	Standard Generic Markup Language
DOM	Document Object Model
SQWRL	Semantic Query-Enhanced Web Rule Language
GUI	Graphical User Interface
UI	user interface
GPS	Global Positioning System
ECA	Event-Condition-Action
HTML	Hypertext Markup Language
XHTML	Extensible Hypertext Markup Language
ISO	International Organization for Standardization
UX	User Experience
WWW	World Wide Web
W3C	World Wide Web Consortium
URI	Uniform Resource Identifier
IRI	Internationalized Resource Identifier

JSON	JavaScript Object Notation
Turtle	Terse RDF Triple Language
VDV	Verband Deutscher Verkehrsunternehmen
HCI	Human-Computer Interaction
CSV	Comma-separated values
PDA	personal digital assistant
CTT	ConcurTaskTree
WSDL	Web Services Description Language
XMI	XML Metadata Interchange
UIML	User Interface Markup Language
UsiXML	USer Interface eXtensible Markup Language
ERP	Enterprise Resource Planning
IFML	Interaction Flow Modeling Language
MBUID	Model-Based User Interface Development
SOA	Service-Oriented Architecture
QoS	Quality of Service
OWL-S	Web Ontology Language for Web Services
WIMP	Windows, Icons, Menus and Pointers
ITCS	Intermodal Transport Control System
CPS	Cyber Physical Systems
TRIAS	Travellor's Realtime Information Advisory Standard
NeTEx	Network Timetable Exchange standard
IFOPT	Identification of Fixed Objects in Public Transport

Confirmation

I confirm that I independently prepared this thesis with the title *An Ontology-based approach supporting adaptive and context-aware information provision in ubiquitous mobility systems* and that I used only the references and auxiliary means indicated in the thesis.

Karlsruhe, December 5, 2022

Dipl.-Inf. Christine Keller

Acknowledgements

This thesis would not have been possible without the opportunities my supervisor Thomas Schlegel provided. I thank him for his confidence in my abilities, his encouragement to pursue my ideas and his constant support. I thank him for his guidance and the possibilities to thrive and learn. I thank Tom Ertl for agreeing to also supervise this thesis, for his insights and the opportunity to submit this thesis at the University of Stuttgart.

I also thank my colleagues who accompanied and supported me during my journey towards finishing this dissertation. I thank Romina Kühn for her input regarding user experience and usability questions and especially for making me smile. I also thank Steffen Huber for discussing various questions about thesis writing and for listening to me talking about my work and asking very good questions. I thank Susann Struwe for discussing adaptation and especially usability of adaptation with me. I thank Nicola Fricke for enduring my questions on usability evaluations and especially on statistics. I am deeply grateful to all four of them for for reviewing parts of this thesis.

Special thanks go to all my colleagues, collaborators and co-authors Thomas Bettermann, Felix Böhm (now Felix Glück), Alexandra Funke, Carl-Friedrich Gerlach, Naemi Gerst, Jonas Hansert, Sarah Hausmann, Jakub Kuspiel, Diana Lemme, Björn Lunt, Tom Lutze, Mandy Korzetz, Florian Niebling, Rico Pöhland, Christian Rickert, Swenja Sawilla, Regina Schlindwein, Ronny Seiger, Hoa Tran and Mathias Trefzger. Thank you all for your support, your ideas and your enthusiasm for our research.

I also want to thank Werner Kohl for not only introducing me to the intricate details of public transport systems, but also for providing an opportunity to apply my experience after finishing my work in academia this year.

Additional thanks go to my friends who have been a constant source of support. Thanks for cheering me on, believing in me and for all these helping gestures, big or small, go to Tanja Blascheck, Dirk Coehne, Inga Gohn, Wiltrud Kessler, Alwine Lenzner, Claire and Loic Mouchard, Kathrin Ullrich and Javier López Carrasco, as well as Max and Anke Weidlich. Special thanks to those of you who made life in a pandemic a little easier.

I feel deep gratitude towards my family who never questioned my absurd idea of pursuing a doctoral degree. I thank my father, Kurt Keller, who not only inspired me to become a computer scientist but also read and reviewed some chapters of my thesis. And I thank my mother, Ulrike Keller, for having my back in all of my decisions.

Last but not least I sincerely and deeply thank my husband, Sören Brunk for being my colleague, co-author, reviewer, thought processor, teacher, my friend and my rock. Thank you for your patience, for being there, listening, questioning, pushing me forward and always extending a hand when I need one. I could not have done this without you. A special thank goes to my children, Paul and Pia, who taught me patience and the essence of life-work balance, for telling me dragon, unicorn and Pokémon stories and for being a constant source of joy and happiness.

Abstract

The vision in ubiquitous computing, context-aware computing and related fields conceives systems that surround us and adapt to diverse situations and settings. These systems comprise different types of devices, mobile or static, embedded systems, personal and public devices. Their main goal is to support users in their tasks in a natural and efficient way, utilizing various user interfaces and building on system intelligence. Based on recent technological innovations, concepts of ubiquitous computing are adopted in many application domains, for example in mobility, which is the application domain of this work.

Usability is a core aspect of ubiquitous systems. The usability of a device or interaction modality heavily depends on its context of use. In a ubiquitous system this context of use changes rapidly and constantly. Intelligent and ubiquitous systems should adapt to these context changes to preserve their usability. However, the specific context of a situation is not known until runtime and adaptations therefore need to be possible at runtime. Since such adaptation decisions affect the system's usability, measures need to be taken to preserve usability during adaptations.

Adaptive user interfaces in ubiquitous systems have been subject to several research efforts. Some approaches introduce architectural concepts for adaptive systems and apply them to user interfaces. These approaches focus on flexibility and do not consider the usability of the resulting user interfaces, however. Plastic user interfaces, as defined by Thevenin and Coutaz [1999] are model-based user interfaces that explicitly aim at preserving their usability while adapting at runtime. However, these approaches depend on complex and heavyweight model-based techniques and therefore lack flexibility and interoperability.

Smart ubiquitous mobility systems can not be designed and implemented from scratch. Ubiquitous technologies need to be integrated into existing system environments. However, in such legacy systems it is not possible to redesign and reimplement all components using model-based techniques. As a consequence, these approaches are not suitable to implement adaptive and usability preserving interaction in mobility systems.

An approach for adpative interaction is needed that works with existing components and architecture and that is more lightweight, more flexible and interoperable than previous extensive and monolithic approaches. Since mobility systems are constantly evolving and each regional or local mobility system has specific characteristics, such an approach needs to be extendable and parameterizable and provide a high level of abstraction for adaptation rules to be reusable and extendable, as well.

This thesis presents a novel approach towards an autonomous and context-aware usability assessment for the adaptation of interactive devices and modalities at runtime. This approach introduces usability factors into context representation and allows usability to be integrated in context-aware adaptation processes. It focuses on proactive information provision in mobility and on the adaptation of output devices and modalities. While it is designed for the application in mobility systems, it is extendable and therefore can be adopted in other domains, too.

This work introduces ontologies to model usability and usability qualities in context-aware ubiquitous systems. It also presents ontologies for the context of use for ubiquitous mobility systems. Based on these ontologies, this work introduces an approach to enable ubiquitous and context-aware systems to autonomously assess the usability of output devices and modalities and to adapt output to the context of use while preserving usability.

The developed ontologies allow to express abstract adaptation rules for a usability assessment and adaptation decisions at runtime. An assessment and decision process framework was developed that applies these rules and uses current context information to decide for usable adaptations. The framework enables a ubiquitous mobility system to choose an output device and modality whenever it needs to send information to a user. To facilitate this choice, the framework allows the system to assess the usability of its available output options for the user in a given situation. The framework has been implemented in a prototype as a proof of concept.

An evaluation has been done in two parts: a performance test of the framework shows its practicability and scalability. In a second part, the choices of the framework for six predefined scenarios were rated by test persons in an online questionnaire and compared to baseline scenarios. This evaluation shows that the framework is able to ensure adequate usability of device and modality adaptations in public transport scenarios.

The approach is more lightweight and flexible than previous approaches towards adaptive user interfaces in ubiquitous environments. Pre-existing user interfaces can be used with this approach and it can be implemented in systems that are not yet context-aware. It can be extended to include other domains and additional usability criteria or rules and decision parameters. Existing ontology-based context-aware systems can adopt this approach to include usability criteria in their adaptive applications.

German Abstract
—Zusammenfassung—

Die Vision des ubiquitären Computing, kontextsensitiven Computings und verwandter Forschungsfelder sind allgegenwärtige Systeme, die sich an verschiedenste Situationen und Umgebungen anpassen. Diese Systeme bestehen aus unterschiedlichen Arten von teils mobilen Geräten, eingebetteten Systemen, persönlichen und öffentlich zugänglichen Geräten. Ziel ist, Nutzende in deren Aufgaben auf natürliche und effiziente Weise zu unterstützen, indem verschiedenste Benutzungsschnittstellen und intelligente Methoden eingesetzt werden. Dank aktueller technologischer Entwicklungen können die Konzepte des ubiquitären Computing in vielen Anwendungsdomänen übernommen und genutzt werden, unter anderem in der Mobilität, der Anwendungsdomäne dieser Arbeit.

Usability ist ein Kernaspekt ubiquitärer Systeme. Die Usability eines Geräts oder einer Interaktionsmodalität hängt dabei stark vom jeweiligen Nutzungskontext ab. In einem ubiquitären System ändert sich dieser Nutzungskontext jedoch ständig und schnell. Intelligente und ubiquitäre Systeme müssen sich an diese Änderungen des Kontext anpassen, damit ihre Usability erhalten bleibt. Allerdings ist der spezifische Nutzungskontext einer Situation erst zur Laufzeit bekannt, weshalb Anpassungen ebenfalls zur Laufzeit möglich sein müssen. Da Anpassungsentscheidungen die Usability des Systems beeinflussen, müssen Maßnahmen ergriffen werden, um Usability trotz Anpassungen auch zur Laufzeit zu erhalten.

Adaptive Benutzungsschnittstellen für ubiquitäre Systeme sind Gegenstand verschiedener Forschungsaktivitäten. Einige Ansätze wenden Architekturkonzepte adaptiver Systeme auf Benutzungsschnittstellen an. Dabei liegt der Fokus allerdings auf Flexibilität und nicht auf der Usability der entstehenden Benutzungsschnittstellen. Sogenannte “plastic user interfaces”, wie sie Thevenin and Coutaz [1999] definiert haben, sind modellbasierte Benutzungsschnittstellen, die explizit darauf abzielen ihre Usability während Anpassungen zu erhalten. Diese Ansätze nutzen allerdings komplexe und schwergewichtige modellbasierte Methoden und sind daher wenig flexibel und nicht interoperabel.

Intelligente ubiquitäre Mobilitätssysteme können nicht von Grund auf neu designed und implementiert werden. Ubiquitäre Technologien müssen in dieser Domäne in existierende Systeme und Systemumgebungen integriert werden. In bereits vorhandenen Altsystemen ist es nicht möglich, alle Komponenten mit modellbasierten Methoden neu zu implementieren. Um adaptive Interaktion die ihre Usability aufrechterhält in Mobilitätssystemen umzusetzen, sind diese Ansätze daher ungeeignet.

Daher wird ein Lösungsansatz für adaptive Interaktion benötigt, der mit existierenden Komponenten und Architekturen funktionieren kann und dabei leichtgewichtiger, flexibler und interoperabler ist als bisherige Ansätze. Da sich Mobilitätssysteme ständig wandeln und regionale oder lokale Mobilitätssysteme jeweils sehr spezifische Eigenschaften haben, muss ein solcher Ansatz erweiterbar und parametrisierbar sein und sollte ein hohes Abstraktionslevel für Anpassungsregeln bieten, so dass diese ebenfalls wiederverwendbar und erweiterbar sind.

Diese Arbeit legt einen neuen Ansatz für eine autonome und kontextsensitive Usabilitybewertung zur Anpassung von interaktiven Geräten und Interaktionsmodalitäten während der Laufzeit vor. Der Ansatz bringt Usability-Faktoren in die Kontextrepräsentation ein und ermöglicht daher, Usability in kontextsensitive Anpassungsprozesse zu integrieren. Dabei steht die proaktive Bereitstellung von Information in der Mobilität im Fokus, ebenso wie die Anpassung von Ausgabegeräten und Ausgabemodalitäten. Obwohl der Ansatz für die Anwendung in Mobilitätssystemen konzipiert ist, ist er erweiterbar und kann auch in anderen Domänen eingesetzt werden.

Diese Arbeit beschreibt Ontologien, die Usability und Usability-Qualitäten für kontextsensitive ubiquitäre Systeme modellieren. Es werden Ontologien vorgelegt, die den Nutzungskontext von ubiquitären Mobilitätssystemen beschreiben. Aufbauend auf diesen Ontologien wird ein Ansatz vorgestellt, der es ubiquitären und kontextsensitiven Systemen ermöglicht, autonom die Usability von Ausgabegeräten und -modalitäten zu bewerten und ihre Ausgabe an den Nutzungskontext so anzupassen, dass die Usability erhalten bleibt.

Die entwickelten Ontologien ermöglichen es, abstrakte Anpassungsregeln für eine Usabilitybewertung und Anpassungsentscheidungen zur Laufzeit zu formulieren. Ein Framework für den Bewertungs- und Entscheidungsprozess wurde entwickelt, das diese Regeln anwendet und aktuelle Kontextinformation nutzt, um Entscheidungen für Anpassungen mit guter Usability zu treffen. Das Framework ermöglicht es einem ubiquitären Mobilitätssystem, ein Ausgabegerät und eine Ausgabemodalität auszusuchen, wenn es Informationen an Nutzende versenden möchte. Um diese Auswahl zu erleichtern, kann das System mit diesem Framework die möglichen Ausgabeoptionen, die für eine*n Nutzer*in in einer gegebenen Situation möglich sind, nach ihrer Usability bewerten. Das Framework wurde prototypisch als Proof of Concept implementiert.

Die Evaluation wurde in zwei Teilen durchgeführt: ein Performancetest zeigt die Umsetzbarkeit und Skalierbarkeit des Frameworks. In einem zweiten Teil wurden die Entscheidungen des Frameworks für sechs vordefinierte Szenarien durch Testpersonen in einem Online-Fragebogen bewertet und mit den Bewertungen einer Baseline verglichen. Diese Evaluation zeigt, dass das Framework adäquate Usability während der Anpassung von Geräten und Modalitäten im öffentlichen Verkehr erzielen kann.

Der Ansatz ist leichtgewichtiger und flexibler als frühere Ansätze für adaptive Benutzungsschnittstellen in ubiquitären Umgebungen. Bereits existierende Benutzungsschnittstellen können mit diesem Ansatz weiter genutzt werden und er kann in Systemen eingesetzt werden, die noch nicht kontextsensitiv sind. Er kann erweitert werden auf weitere Domänen und zusätzliche Usabilitykriterien oder Regeln und Entscheidungsparameter können ergänzt werden. Existierende ontologiebasierte kontextsensitive Systeme können diesen Ansatz einsetzen, um Usability-Kriterien in adaptiven Anwendungen zu integrieren.

CHAPTER 1

Introduction

Since Mark Weiser described his vision of Ubiquitous Computing in 1991 and set off the so-called third era of computing, many of the technologies he and his team envisioned have matured and became indispensable in our everyday lives [Weiser, 1991; Want, 2009]. Weiser imagined computing systems that support their user in fulfilling their tasks and that provide the computing service the user needs, in the situation, place and configuration they need it [Weiser, 1993; Abowd and Mynatt, 2000; Augusto et al., 2013]. Mobility is an application domain in which the user's situation is changing constantly while the user needs up-to date information and access to mobility services. Ubiquitous and mobile technology can provide the necessary information and access to services suitable to the user's situation. The application of ubiquitous and mobile technologies in mobility enables smart ubiquitous mobility systems.

The core goal of users in mobility is clear - it is to travel a distance between two points. However, mobility is diversifying and so are user needs. Recent developments of new transport concepts such as car sharing, peer-to-peer ridesharing and bike sharing change how people are mobile, particularly in urban areas. Aside from timeliness, predictibility and comfort, sustainability of transport plays a role in transport choices [Buehler, 2011]. Mobility, especially in urban spaces, is becoming multi-modal, meaning that people combine different transport modes to reach their destination [Kuhnimhof et al., 2012]. Subsequently, mobility is becoming more diverse and the information need of users increases.

This mobility calls for smart mobility systems that provide unified access to all kinds of different transport modes and mobility services. Following Weiser's vision of ubiquitous computing, where computing devices are getting out of the way, ubiquitous mobility systems support a user in their mobility by keeping them informed, guiding them and enable them to access means of transport, for example by providing electronic tickets or unlocking a rental car in such a way that the user can focus on other tasks.

In mobile and mobility systems that are used in many different surroundings and environments, the surrounding conditions change frequently. This affects the system's usability, since static user interfaces are rarely suitable for a large range of physical and social environments. Schmidt et al. [1999] wrote about context-awareness for ultra-mobile computing, stating that context-aware user interfaces which adapt to surrounding conditions can improve the utility of interaction styles and display modes. Blumendorf et al. [2010b] and Schwartze et al. [2010] argued that in a smart environment with changing context, the set of available devices and their interaction capabilities must be detected at runtime and therefore, the user interface must be distributed dynamically at runtime, too. Shafer et al. [2001] highlighted that, in contrast to traditional desktop systems, the user is not easily available in an intelligent context-aware environment. If the system needs to contact the user, it needs to adapt to current context. Shafer et al. [2001] wrote:

> "The use of contextual information is crucial to routing the message to the user in the most appropriate way." [Shafer et al., 2001]

Applying the principles and paradigms of ubiquitous computing, adaptive interaction and context-aware computing to mobility systems can improve accessibility and usability of several means of transport. A combination of various types of transport in one journey can increase the traveller's flexibility. It also facilitates sustainable mobility patterns, using less motorized individual transport and more public transport, bikes or at least encourage and support carpooling for a better utilization of individual transport. In order to achieve these goals, smart ubiquitous mobility systems must be able to guide the user every step on their way and they need to apply the principles of ubiquitous and context-aware computing. A smart ubiquitous mobility system is able to provide excactly the information the user needs in their situation. If the system is proactive and context-aware, it can plan ahead and dynamically react to disruptions, increasing reliability for the traveller. User interfaces in smart and ubiquitous mobility systems in particular must be adequately usable in order to reflect the reliability and flexibility of the system and to increase the acceptance of such systems, convincing travellers to actually use them.

1.1 Problem Statement

An important characteristic of ubiquitous and context-aware systems is that in contrast to classic computing applications, the system often initiates interaction as, for example, Carvalho et al. [2017] described. Such systems work proactively and contact the user, unlike in classic computing systems, where a user chooses to use a computing device. When the system initiates interaction, it is the system's responsibility to decide which interaction devices and modality are suitable.

Depending on the situation of the user, certain interaction modalities and devices are more suitable than others. Considering the information that a connecting train will depart from a different platform, the user's situation is highly relevant for the system's choice of interaction devices and modality to inform the user about the change of plans. In case the user is currently driving their car to the station, visual smartphone notifications are not the best way to indicate a schedule change, since it would distract the user from driving. Also, reading a notification on the smartphone screen while driving is not only unusable and unsafe, but also forbidden in many countries. To notify the user by speech output and to offer to read the information out loud would be a better alternative. If the user is currently riding a train, this choice is not preferrable - audio output would disrupt the privacy of the user. Instead, a display in the train could highlight relevant information or the smartphone of the user could vibrate and show a notification.

Many approaches that support adaptive interaction in ubiquitous systems do so by defining situations and the suitable selection of interfaces at design time. For example, Aquino et al. [2010] have introduced so-called transformation templates, that allow designers to easily shape the generation of user interfaces for specific end-user requirements. The requirements that result in the generation of a user interface as well as the properties of this user interface are all modeled at design time. In such approaches, the system's designer chooses the user interface layout and which interaction devices and modalities should be used and the system is then built according to these design decisions. However, there are several factors that indicate that this approach is not sufficient for intelligent environments and that a solution for adaptive interaction choices at runtime is needed. First, most intelligent environments and especially smart ubiquitous mobility systems are intended for a long operating time. A multitude of situations during that operating time makes it impossible to foresee all situations that may arise. Second, the mobility of many devices - from smartphones to wearables to devices in vehicles - results in unpredictable system configurations, where it is not always known which devices may be involved and available in the actual situation. Devices also fail and are sometimes replaced. Therefore, carefully and manually designed device choices for human-computer interaction are not applicable in all situations.

If a smart ubiquitous mobility system should be able to flexibly provide mobility information and interaction to the user, even if there are unforeseen situations and system configurations, the system needs to be able to decide at runtime which of the available interactive devices and modalities should be used. This decision should result in adequate choices that preserve the user's privacy, reach the user, are unobtrusive and ensure usability and safety.

1.2 Research Questions

This thesis focuses on information provision in mobility systems and therefore the output part of interaction. User input is out of scope. Adaptations of interaction for groups of users will not be examined. Furthermore, interaction modalities will be considered seperately, multimodal interaction is not in the scope of this work. The goal of this work is not to replace designers or to erase the design phase from the processes of engineering ubiquitous computing systems, but to augment ubiquitous systems by extending the system's capabilities to adapt information output at runtime.

The main research question of this thesis is:

Main Research Question

Can a smart ubiquitous mobility system adapt output devices and modalities autonomously and at runtime to the context of use while preserving adequate usability?

For a system that adapts output devices autonomously at runtime it is conceivably difficult to achieve a comparable degree of usability to an application that has been carefully designed for its intended use. However, the goal is not to reach for maximum usability but to enable a system to avoid clearly unusable choices and achieve better usability than a static approach can achieve. Current solutions are mostly static and use, for example, one dedicated output device at all times and in all situations, which is not in all cases a usable choice. In the research question described above, this goal is described as *adequate usability*. The term *autonomously* means, that the system makes a decision on its own and its decisions are not hardcoded for predetermined situations. The adaptation should take place *at runtime*, which means that the decision making process should not extend the system's response time too much.

In order to answer the Main Research Question, the following questions must be answered:

Research Question 1: Dimensions of the Context of Use

Which dimensions of the context of use are relevant as a basis for a usability assessment of device and modality adaptations?

Research Question 2: Representation of the Context of Use

How can the context of use be represented in order to be processed autonomously by a smart ubiquitous mobility system?

Research Question 3: Method for Autonomous Assessment

Which method is suitable for autonomously assessing the usability of an adaptation of device and modality in a given context of use?

Research Question 4: Decision Making

How can a system autonomously decide how to adapt device and modality while maintaining adequate usability?

1.3 Contributions

This work proposes an approach that enables context-aware and interactive smart ubiquitous mobility systems to choose output devices and modalities dynamically at runtime while preserving usability. The approach is designed for integration in ontology-based context-aware ubiquitous systems. The research that led to the solution approach described in this work resulted in several contributions to the current state of the art. Most of the publications I published as first author. Thomas Schlegel as my supervisor is co-author in all my publications and provided his insights, experience and advised me on my ideas and concepts. The contributions of this work are summarized in the following.

Contribution 1: Ontologies modeling the context of use for smart ubiquitous mobility systems

This work provides a thorough analysis of context dimensions in ubiquitous mobility systems. Parts of this analysis have been published in [Schlegel and Keller, 2011], where I contributed the discussion and outline of dimensions and contexts in public systems, while Thomas Schlegel contributed work on context-sensitive modeling of interaction cases. In [Kühn et al., 2011], we further structured a context taxonomy for public systems and Romina Kühn as first author provided her expertise on public interactive systems, while I contributed the context structure. The taxonomy was a collaborative effort. I further developed and refined this taxonomy in my own subsequent work. This thesis presents several ontologies that model these dimensions and can be used to describe the context of use in a ubiquitous mobility system. A version of these context ontologies is published in [Keller et al., 2020] as a passenger context model for adaptive passenger information. Waldemar Titov contributed to the identification of context facts using a bottom up approach and implemented the context management components in the course of the project "SmartMMI"[a], while I modeled the context model and developed the use cases for the application of this model. This contribution also includes a public transport ontology, partly described in [Keller et al., 2014a]. The ontology was developed in the project "IP-KOM-ÖV"[b] and my co-author Sören Brunk provided parts of the implementation of the prototype described in this paper. I modeled and documented the domain ontology for public transport. In later work, I added an extension to this ontology, as discussed in section 6.2. Additionally, a task ontology for public transport tasks, a device ontology for devices used in passenger information and a general context ontology that models user context, vehicle context and device context are a result of this work. A version of the device ontology is published in [Keller and Schlegel, 2019].

---

[a] "SmartMMI: Modell- und kontextbasierte Mobilitätsinformation auf Smart Public Displays und Mobilgeräten im Öffentlichen Verkehr", `http://smartmmi.de/`, last accessed October 12th, 2022

[b] "IP-KOM-ÖV: Internet Protokoll basierte Kommunikationsdienste im Öffentlichen Verkehr", `http://ip-kom.net`, last accessed October 8th, 2022

Contribution 2: A usability ontology modeling usability attributes for situations in smart ubiquitous mobility systems

This approach models usability attributes and interaction knowledge in a usability and an interaction ontology. It supports a smart ubiquitous mobility system to autonomously asess the usability of output devices and modalities for a given context of use. A part of the interaction ontology is published in [Keller and Schlegel, 2016] and [Keller and Schlegel, 2019]. The usability and interaction ontology builds upon insights from several of my works on adaptive passenger information systems. In [Keller et al., 2011], we published a study on the visualization of passenger information on smartphones which is a result of a bachelor's thesis of Mandy Korzetz under my supervision. Romina Kühn contributed to the implementation of the study. Insights on the reception of passenger information and the need of personalization were used in this work. An adaptive semantic mobile application using public transport information was published in [Keller et al., 2014b]. Rico Pöhland and Sören Brunk provided the implementation of this mobile application, while I contributed the context model and adaptation concept. Several usability studies of the smart window prototype from the project "SmartMMI"[a] are published in [Keller et al., 2019b]. Waldemar Titov contributed to the implementation of the studies and the prototypes and Swenja Sawilla contributed to the eyetracking study and the data analysis. My contribution was the concepts for several of the prototypes as well as structuring the studies we performed and interpreting the results. Insights on preferences of users regarding the display of types of passenger information on smart windows or other devices were used in this work. In [Keller et al., 2019a] we published a study performed as an online questionnaire about acceptance and usability of output adaptations in public transport. Susann Struwe contributed intelligibility features and participated in the data analysis while Waldemar Titov contributed the persona adaptation approach. My contribution to this work includes the adaptation concept for adapting output devices, the respective scenarios and the implementation of the study as well as data analysis. Collaborative work with Waldemar Titov and Hoa Tran on passenger information on smart windows and privacy concerns was published in [Titov et al., 2020]. Waldemar Titov provided the prototypes used in this work as well as the pseudonymization concept for personal information on smart windows. Hoa Tran contributed the legal background and the categorization of personal data on legal grounds. Waldemar Titov and I collaboratively developed the concept of information areas on the smart window and the distribution of personal data on these areas. Part of the privacy and sensitivity model described in 7.2.2 is based on this work.

---

[a] "SmartMMI: Modell- und kontextbasierte Mobilitätsinformation auf Smart Public Displays und Mobilgeräten im Öffentlichen Verkehr", `http://smartmmi.de/`, last accessed October 12th, 2022

Contribution 3: A usability rule framework to express rules for the assessment of usability criteria.

Based on the usability and interaction ontologies, this work proposes a rule framework that enables the expression of abstract usability rules for the assessment of usability criteria. The rules are described on a high abstraction level, referencing the usability and interaction ontologies, but also the context ontologies. They therefore enable the expression of usability qualities related to the context of use. The abstract rules that reference the usability ontology are easier to read than hardcoded and specific rules, they are reusable and extendable.

Contribution 4: A process for a usability assessment of device and modality options in a given context of use

In this work, a process for usability assessment is proposed and implemented. A first version of this process has been published in [Keller and Schlegel, 2019]. It is applicable to all aspects of usability that are modeled in the aforementioned usability ontology. The process uses a high abstraction level to be easily modifiable.

Contribution 5: A decision making process for the adaptation of output devices and modalities in smart ubiquitous mobility systems

The usability assessment process results in a rating of available output devices and modalities. The next step is to reach a decision about an adaptation of output devices and modalities based on this rating. This work presents and implements a decision making process that uses abstract rules to decide which options to use. It is loosely coupled and flexible. Part of its architectural approach is published in [Keller and Schlegel, 2016]. A version of the decision making process is published in [Keller and Schlegel, 2019]. A proof of concept implementation of the usability assessment process as well as the decision making process shows their feasibility and was used to conduct a performance evaluation as well as a usability evaluation of this work.

1.4 Outline

The foundations for this work are described in chapter 2. In the following chapter 3, a requirements analysis is detailed and results in requirements towards an approach for the research questions. In chapter 4, related research and approaches towards adaptive interactive and ubiquitous systems are reviewed and compared, using the requirements from chapter 3. Two research gaps are identified that are adressed by the remainder of this work. In chapter 5, the approach of this work is described and several design decisions are outlined. The following chapter 6 details the analysis process to identify the dimensions of the context of use in smart ubiquitous mobility systems. Based on these, the ontologies developed to allow a representation of the context of use are described. Chapter 6 therefore reports the answers to research questions 1 and 2 and it describes contribution 1. Chapter 7 presents the usability assessment framework for adaptive output in ubiquitous public transport. In this chapter, the analysis process towards a usability ontology and assessment rules is outlined and the usability ontology is documented. Chapter 7 also presents the developed rules and the assessment and decision making process. In this chapter, research questions 3 and 4 are answered. It also describes contributions 2, 3, 4 and 5. In chapter 8 follows the description of a prototype implementation of the usability assessment framework.

The evaluation of this prototype, and evaluation results, are detailed in chapter 9, presenting a performance evaluation and a usability evaluation. A summary of the presented concepts and approach as well as a discussion and outlook follows in chapter 10.

Copyright

In this thesis, images from the following publications are reused, with permission of the copyright holder:

- [Keller et al., 2014a] is under copyright of Springer International Publishing
- [Keller et al., 2020] is under copyright of Springer Nature

Additionally, images are used in some figures obtained from Freepik[1] and Flaticon[2] that are free to use with attribution:

- Transport icons, designed by Vitaly Gorbachev from Flaticon
- Ways of transport icons, designed by Freepik
- Sale posters, designed by Macrovector_Official / Freepik

---

[1] `http://www.freepik.com`

[2] `www.flaticon.com`

- Bakery poster, designed by Macrovector_Official / Freepik
- Map background vector, created by Freepik
- Digital device mockup, designed by rawpixel.com / Freepik
- Heavy rain icon, designed by Freepik

The pictures used in the online questionnaire during the usability evaluation were created by Lars Erber and based upon graphics by Nadine Vollers. They are used with kind permission by the creators.

Chapter 2

Theoretical Foundations

This work tackles a research question concering ubiquitous systems and intelligent environments. It also covers aspects of Human-Computer Interaction (HCI) and usability as well as context-awareness and adaptivity. As such, there are many foundations in each of these areas that play a role in answering the research question. This chapter covers the foundational aspects necessary for the development of a solution to the given research question.

2.1 Intelligent Environments, Ubiquitous and Pervasive Computing

The term *ubiquitous computing* was first used in 1988 by Mark Weiser and his research group at the Palo Alto Research Center (PARC) of Xerox [Weiser, 1991; Takayama, 2017]. They described ubiquitous computing as the intention of bringing a variety of devices in the background, creating a computing environment that is not distracting users, so that they can focus on their tasks and not on how to operate a specific device. In their research efforts, they worked on various aspects of ubiquitous computing, developing prototype devices in different sizes for individual or multi-user use as well as network protocols for wireless connectivity and new interaction concepts [Weiser, 1993]. The ubiquitous computing paradigm was widely adopted in the following years. In 2010 and 2012, Schmidt [2010] and Schmidt et al. [2012] look back on 20 years of ubiquitous computing research. They noted that due to the development of smartphones, tablets and embedded systems, computing everywhere as Weiser envisioned had made significant progress. However, many issues remain unsolved, such as the application of artificial intelligence for everyday problems as well as privacy issues, for example. Schmidt [2010] also point out that the understanding of "invisible" computing that Weiser formed, evolved since 1991. Smartphones as inherently personal devices are not a part of Weiser's vision, who strived towards less personal devices and more actual invisibility. While smartphones are very much not invisible, Schmidt [2010] argue that they have become powerful tools for many people, where the users ignore the technology behind it and focus on the tasks they perform using the devices.

The term *pervasive computing* was also introduced in the 1990s, parallel to the development of ubiquitous computing. Pervasive computing comprises research on wireless network protocols, sensing technologies and embedded devices. Satyanarayanan [2001] highlights the embeddedness of pervasive computing, a capacity that is not so much focused on in ubiquitous computing. Lyytinen and Yoo [2002] discuss the relationship between ubiquitous computing, pervasive computing and mobile computing. *Mobile computing*, as they describe it, is concerned with the physical mobility of either devices themselves or of computing services. Lyytinen and Yoo [2002] characterize pervasive computing as focused on the sensing and processing of information using sensors and embedded devices. On top of that, these devices build models of their surroundings and their computing in that environment. The authors concluded that ubiquitous computing includes the challenges and characteristics of both mobile and pervasive computing.

Context-aware computing also collect information using sensors and other sources. They process the user's location, environment, activities and other context factors to adapt a system's behavior Schilit and Theimer [1994]; Dey and Abowd [2000]. Salber et al. [1998] name context-awareness a "functional service for ubiquitous computing". They argue that ubiquitous systems are used in dynamic contexts and must adapt to these contexts, especially when these systems have a high degree of mobility [Salber et al., 1998].

Ambient intelligence unites approaches from Artificial Intelligence with pervasive and ubiquitous computing research. About ten years after Weiser's vision about ubiquitous computing, a report of the IST Advisory Group (ISTAG) of the European Commission identified several use cases for the application of artificial intelligence, sensor technologies, miniaturized devices, ubiquitous communication infrastructure and natural interfaces, among others [Weiser, 1991; K. et al., 2001]. Artificial intelligence methods are, for example, applied for activity recognition or facial analytics [Gams et al., 2019]. The main application domains are smart homes and ambient assisted living [Cook et al., 2009; Yun and

Yu-Hua Gu, 2017]. However, transportation was also identified as a worthwile application domain by the ISTAG in 2001, as well as learning and smart offices [K. et al., 2001; Cook et al., 2009].

The paradigm of an *Internet of Things* (IoT) is based on technologies for close range data transmission using radio-frequency identification (RFID) tags that enable easy identification of objects combined with sensor network technologies and common internet protocols. Everyday objects are linked together and either are equipped with processors, transforming them into computing devices or creating virtual copies of them that are accessible over the internet. A large number of such linked objects that serve some useful objective forms the Internet of Things (IoT) [Whitmore et al., 2015]. Whitmore et al. [2015] identify context-awareness and ubiquitous computing as key requirements for a successful Internet of Things (IoT) that provides Ambient Intelligence. *Cyber Physical Systems* (CPS) developed as a term describing the integration of the virtual and physical world. Physical entities have digital representations and impact the virtual world, as in the Internet of Things (IoT). Using actuators, virtual entities can in turn impact the physical world in a Cyber Physical Systems (CPS) [Wu et al., 2011]. Cyber Physical Systemss (CPSs) have been applied to various application domains, including transportation, but have been particularly successful in industrial systems Lu [2017].

Reasearch on *Intelligent Environments* is combining research approaches from ubiquitous and pervasive computing as well as Ambient Intelligence and the Internet of Things (IoT) and focuses on user experience [Coen, 1998; Shafer et al., 2001; Augusto et al., 2013]. Therefore, research in Intelligent Environments is, for example, concerned with reliability and the incorporation of user preferences and expectations, as well as ethical aspects [Augusto, 2009; Jones et al., 2015; Corno, 2018; Augusto and Muñoz, 2019]. Context-awareness and proactivity are key characteristics of Intelligent Environments [Bidot et al., 2011; Augusto et al., 2013].

Ubiquitous computing, in turn, has been influenced by the development of these computing paradigms, technologies and fields of research. The evolution of ubiquitous computing is shaped by further miniaturisation of devices - smartphones and tablets being the most ubiquitous - further progress in networking, context-aware systems as well as a diversification of user interfaces towards more natural interaction.

2.2 Human-Computer Interaction in Intelligent Environments and Ubiquitous Systems

Ubiquitous computing involves various aspects of human-computer interaction from the beginning. This includes natural, multimodal and implicit interaction techniques on the one hand and interaction design, usability and user experience on the other hand [Abowd and Mynatt, 2000; Fitton et al., 2005; Quigley, 2010; Resnick, 2013]. There are numerous examples for innovative interaction techniques that were developed for ubiquitous computing or intelligent environments [O'Neill et al., 2006; Katsuragawa et al., 2016]. However, most of them are out of scope of this work. The following sections briefly review key concepts of human-computer interaction in ubiquitous computing which are related to the research question in this work.

2.2.1 Interaction Paradigms for Ubiquitous Computing

In a description of the evolution of user interfaces, Nielsen [1993a] describes classic command-based interfaces as "function-oriented", using text-based commands that execute a function. The generation of interfaces after these are user interfaces implementing the WIMP paradigm (Windows, Icons, Menus and Pointers). Nielsen [1993a] classifies them as "object-oriented", where objects are directly manipulated and modified. He calls the next generation of user interfaces after WIMP interfaces *noncommand user interfaces* and sketches them as "user-oriented" and "task-oriented", a characterization that goes well with the ubiquitous computing vision that puts the user in focus and dedicates computing to supporting the user in their task.

Schmidt [2000] indicates implicit interaction as a core of ubiquitous computing systems. He describes *implicit interaction* in contrast to explicit interaction, which is embodied in classic *Graphical User Interface* (GUI) based systems. In explicit interaction, a user explicitly takes actions as input towards the system, while in implicit interaction the system perceives actions of a user that are not directed towards the system and processes them as input [Schmidt, 2000]. This type of interaction is realized using sensor and other context data that can inform the system about the user's intention and needs [Quigley, 2010].

Ju and Leifer [2008] categorize interaction in the dimensions of attentional demand by a user and the system's initiative. A classic interface relying on direct manipulation is a reactive system that demands attention from the user, where implicit interaction relies on a proactive system and reduces the attention that is required from the user. Implicit interaction therefore is a step towards Weiser's vision of computers "getting out of the way" [Weiser, 1991, 1993].

Central to implicit interaction is adequate knowledge of the user's context, in order to correctly interpret the user's actions. A context-aware system is a prerequisite for implementing implicit interaction. A system implementing implicit interaction needs to adapt interaction autonomously, a goal that aligns with the Main Research Question of this work.

The goal of *natural interfaces* is to shape interaction according to interaction between humans, so that a user does not need to adapt to the interaction styles a computer provides, such as a keyboard or mouse. Natural interfaces are considered as easy to learn and use, enabling a user to focus on their task rather than operating a device. They are thereforen seen as an important part of ubiquitous computing [Abowd and Mynatt, 2000]. Speech-based interfaces as a type of natural interfaces were adapted and used in ubiquitous computing from early on [Juang, 2001]. Leong et al. [2005] present a context-aware speech-based ubiquitous interface, for example. Recent developments in natural speech processing have been very successful and led to several speech-based assistants, such as Siri[1] or Alexa[2], as compared by López et al. [2018].

Gestural interaction is also considered a natural interaction style. It is realized in multitouch interfaces that enable direct touch and manipulation of objects. Multitouch interfaces have, similar to speech-based interfaces, matured significantly over the last years and evolved to a de facto standard interaction technique [Rekimoto and Saitoh, 1999; Schlegel, 2013]. Other gestural interfaces use gestures without touch, either based on video-processing, infrared sensors or accelerometer and other sensors on small devices, handheld or wearable [Grandhi et al., 2011; Francese et al., 2012; Silpasuwanchai and Ren, 2014]. Tangible user interfaces are another type of natural interface. Physical, tangible objects are traced and detected by a system and can then be used for interaction [Ishii and Ullmer, 1997; Shaer and Hornecker, 2010]. Many more interaction modalities support natural interaction and are in different stages of maturity and usability. Considering information output as is the focus of this work, speech output and tactile output or tactile feedback are the most relevant modalities that can be seen as natural, aside from visual output [Azenkot et al., 2011; Lee et al., 2013].

A ubiquitous system providing implicit and natural interaction uses a variety of devices and modalities and proactively considers contextual information to shape interaction. This is the area of application for the results of this work.

---

[1] `http://www.apple.com/ios/siri/`, last accessed October 12th, 2022

[2] `https://developer.amazon.com/de-DE/docs/alexa/alexa-voice-service/api-overview.html`, last accessed October 12th, 2022

2.2.2 Ubiquitous Interactive Devices

Ubiquitous computing, pervasive computing and similar research trends have contributed to and driven the development of numerous new interactive devices and have pushed the evolution of existing devices. Weiser [1991] had the vision of "Pads, Tabs and Boards" as ubiquitous devices of different sizes. He described Pads as very small notepad-like devices, Tabs as handheld devices of page-size and boards as interactive large displays. With Smartphones, Tablets, Smartboards and other interactive Surfaces, these types of devices are no longer a vision.

Schmidt et al. [1999] prototyped an orientation-aware handheld device in 1999 that rotated its user interface based on device orientation, a feature that is nowadays standard in tablets and smartphones. Wearable technology has been applied in many different ubiquitous settings [Rhodes et al., 1999; Lapinski et al., 2011; Zhang and Sawchuk, 2012]. Contemporary smartwatches are often used for ubiquitous applications and in ubiquitous environments, as they offer a personal interface that is always accessible by the user but can also provide various sensor data to the system [Guiry et al., 2014; Vilarinho et al., 2015; Katsuragawa et al., 2016; Chow et al., 2016].

Since the perception of boards by Mark Weiser, large interactive displays were an active field of research. Very often, large displays were combined with mobile or smartphones, where the phones could be used as interaction devices for ubiquitous displays [Uemukai et al., 2002; Kühn et al., 2013; Di Geronimo et al., 2017; Horak et al., 2018]. With multitouch enabled displays, a direct interaction technique for large displays became available [Kim et al., 2010; Jacucci et al., 2010]. Other forms of interaction with ubiquitous displays are handgestures or full-body gestures as well as tangible objects [Ardito et al., 2015]. Tabletops are displays in a horizontal orientation and used in a multitude of application domains [Müller-Tomfelde and Fjeld, 2012].

Public displays are large and often interactive displays that are installed in public spaces, such as libraries, town squares, railway stations or bus stops and others [McCarthy, 2002; Rogers and Brignull, 2002; Müller et al., 2010; Hörold et al., 2015]. Very early on, social implications of public displays were discussed, such as how people could be convinced to interact with a display [Russell et al., 2002; Brignull and Rogers, 2003]. Public displays were also developed as context-aware displays that adapted content or interaction possibilities based on context data [Favela et al., 2004; Lemme et al., 2014; Taniguchi, 2018; Parker et al., 2018].

2.2.3 Model-Based Adaptive User Interfaces

A specific type of user interfaces are model-based user interfaces. In the research area of model-based user interfaces, adaptive or context-sensitive user interfaces have been explored [Clerckx et al., 2005b; Motti and Vanderdonckt, 2013]. Meixner et al. [2011] present a review of four generations of Model-Based User Interface Development (MBUID), following a categorization by Paternò et al. [2009]. In the 1990s, researchers started to develop model-based approaches towards user interfaces, for example Foley and Piyawadee [1995]. Paternò et al. [2009] and Meixner et al. [2011] describe these as first generation approaches. In a second generation, task models were developed that made it possible to describe the user's tasks in an abstract way. Paternò [1999] introduces the ConcurTaskTree (CTT) notation to formalize user tasks in a hierarchical structure. As soon as mobile computing devices with different types of user interfaces became prevalent, a third generation emerged. Several approaches used model-based user interfaces to enable the development of mostly Graphical User Interfaces for various platforms, for example by Eisenstein and Puerta [2000], Nylander and Bylund [2003] or Gajos et al. [2010]. A fourth generation of model-driven approaches targets ubiquitous environments, supporting access to Web Services with different user interfaces, often supporting multimodality [Paternò et al., 2009; Gajos et al., 2010]. Model-driven approaches towards adaptive user interfaces are discussed in greater detail in section 4.2 in chapter 4.

2.2.4 Usability in Ubiquitous Systems and Intelligent Environments

Usability has long been an important topic in human-computer interaction research. It is a term that stems from ergonomics and replaces the former used description of user friendly systems [Nielsen, 1993b].

The International Organization for Standardization (ISO) published a standard for ergonomics of human-system interaction that contains a definition for usability, which reads:

> **Usability**: *"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."* [International Organization for Standardization, 2018]

This definition centers around the ability of a user to finish a task correctly using the system (*effectiveness*), doing so using little resources (*efficiency*) and being content with the usage of the system (*satisfaction*). Usability greatly depends on the context of use of a system. The context of use involves the user's skills and attributes, the tasks they want to use the system for, as well as the setting in which the system will be used including the technical and physical environment. Calvary et al. [2001] argue that user interfaces should adapt to the context of use. They define *plastic user interfaces* as user interfaces that adapt to changes in the context of use, focussing on retaining their usability in the process Calvary et al. [2001, 2003]. In order to achieve this, a system needs to implement methods to asses and rate the usability of its adaptations, which is the subject of research question 3. Approaches towards plastic user interfaces will therefore be discussed in chapter 4 in greater detail.

In human-computing interaction research, the possibilities of developing, assessing and measuring usability have been and are widely researched and discussed. Nielsen [1993b] presents *usability engineering* as an engineering approach for the user-centric development of usable systems. Key to such development processes is assessing and testing usability of a prototype or a product. An overview over classic usability measuring and evaluation methods can be found, for example in the book by Lazar et al. [2017], or in a more practical approach, by Rubin et al. [2008]. Section 2.2.4 discusses usability evaluation methods specifically for ubiquitous systems.

After some time of usability research, the notion of User Experience (UX) has been discussed and then defined. UX includes usability but additionally comprises the beauty, appeal, desirability of a system, and the user's emotional response to the system [Forlizzi and Battarbee, 2004; Hassenzahl and Tractinsky, 2006] The ISO has also defined UX from a user centric point of view, where UX is concerned with a user's perception of a system and their anticipation towards a system [International Organization for Standardization, 2019]. UX factors are a very important for ubiquitous systems and environments. Väänänen-Vainio-Mattila et al. [2015] have reviewed UX research specifically in ubiquitous computing, for example. While UX qualities beyond usability are relevant in ubiquitous computing, they are out of the scope of this work and therefore will not be discussed in more detail.

To achieve ubiquitous computing goals, interacting with ubiquitous computing environments must be efficient, effective and satisfactory, which translates to highly usable [Weiser, 1991]. However, ubiquitous systems have several characteristics that differ from those of graphical user interfaces and of applications for desktop computers. Thomas and Thimbleby [2002] therefore write, for example that a "new usability" is necessary for the design and engineering of ubiquitous applications. The following sections reviews relevant of the characteristics of ubiquitous systems from a usability point of view.

Calm Computing

Calmness was one of the key values that Mark Weiser and his team associated with ubiquitous computing [Weiser and Brown, 1996]. Ubiquitous technology should, in the Xerox PARC vision, inform people and support them in their tasks, but at the same time it should not overload and stress its users. Weiser and Brown [1996] discussed that *calm technology* adresses not only the center of one's attention, but also its periphery, at the same time or alternately. They argue that technology in the periphery does bind less of the user's attention but still is able to deliver information the user can process. The user is in control if and when they focus their attention on a piece of technology that acts in the periphery. In ubiquitous computing, the user's attention may often not be focused on the interface of these computing applications [Abowd and Mynatt, 2000]. The idea of calm computing moves computing technology into the background and is one approach to enhance a system's usability, taking the user's attention into account. Satyanarayanan [2001] refers to this idea as *invisibility*, for which a useful practical goal is *minimal user distraction*. An important usability attribute for ubiquitous systems is to avoid information overload. An awareness and possibly adaptation to the user's attention can benefit the system's usability [Okoshi et al., 2017; Anderson et al., 2018].

Context of Use and User Mobility

In ubiquitous systems, the context of use is much more complex and dynamic than for desktop computing [Abowd and Mynatt, 2000]. Other than the more versatile tasks an everyday computing system supports, the environment such systems are used can be much more diverse than that of classic computing systems. Physical environmental factors such as lighting conditions, loudness or temperature influence the usage of such a system. The social situation is very different and more varied for ubiquitous systems than for desktop systems. It is a part of the context of use that affects the system's usability. How many people are in the room or near to the user is relevant for privacy and security, for example when logging into a device [Zakaria et al., 2011; Kühn et al., 2016].

Other than the complexity of context that must be considered for ubiquitous systems, the context is also changing rapidly, often due to the mobility of users. Salber et al. [1999] have discussed the mobility of the user as a central attribute of ubiquitous computing. They refer to user mobility as the possibility of the user to interact with the system while moving freely. *Full mobility* allows a user to use computing services at any location, unrestricted from GPS or cell phone coverage [Salber et al., 1999]. A mobile user can be interrupted at any time during their task, requiring the application to enable resuming a task [Abowd and Mynatt, 2000]. When a user is mobile, the system needs to adapt to different system configurations, because new computing devices can become available, while others move out of reach. Mobility entails changes to the context of use, which impacts usability [Moran and Dourish, 2001].

Context-Awareness

For classic computing systems, the context of use is considered at the design stage to improve a system's usability. The complexity and variability of the context of use in ubiquitous systems, however, requires ubiquitous systems to respond to context at runtime. A system that adapts to changes can mitigate their effect on the system's usability. But context-awareness entails usability challenges on its own [Bellotti and Edwards, 2001; Barkhuus and Dey, 2003]. A system that changes based on internal decision making processes can confuse its users when they can not make sense of the changes. This affects the user's trust, satisfaction, sense of control and overall learnability. Intelligibility features can mitigate or avert this effect by explaining adaptations or offering information about software decisions [Bellotti and Edwards, 2001; Lim and Dey, 2009, 2010].

2.2.5 Usability Evaluation of Ubiquitous Computing Environments

Evaluating and testing usability in ubiquitous computing environments is a difficult task. As described above, usability in ubiquitous computing environments involves factors that differ from classic desktop computing. Therefore, common usability evaluation methods need to be adapted and complemented to measure all meaningful factors. Ubiquitous systems are also more complex to set up and test than desktop applications. Per definition, ubiquitous computing takes place in environments, some of them in closed spaces such as offices or homes, but others in open spaces, as in smart city and smart mobility research. Setting up laboratories and developing matured prototypes for such spaces is costly and complex, or, considering the size of the target system, nearly impossible. As those systems are context-aware, they need context from their sensors and other context data sources to adapt to. As described before, for most ubiquitous systems, not all situations to which the system should adapt can be foreseen at design time of the system. This makes the evaluation of such a system harder, since testing can therefore not cover all possible test cases. It also complicates usability assessments a system needs to do while planning an adaptation, because simple rules defined at design time might not be applicable to the situations it encounters. This complexity calls for abstraction, as is explained further in section 3.4 and in requirement 7.

The purpose of this work is to enable ubiquitous systems to assess and compare interaction options in a given situation. Furthermore, the resulting prototype of this work will be tested using an appropriate usability testing method. Therefore, I will shortly review and discuss evaluation methods, heuristics, factors, and metrics in the following paragraphs. A detailed analysis of usability factors and heuristics specifically for ubiquitous systems can be found in section 7.1 in chapter 3.

Usability Evaluation Methods

Usability evaluation methods can be categorized as expert methods, usability testing with users or as automated testing methods. Expert methods use preferably independent usability experts that inspect and assess a user interface. In the following, heuristic evaluation as one expert method will be introduced shortly. Usability testing with users takes a prototype of a system or a final software and actual users as test persons. Two types of usability tests will be discussed briefly in the following: field tests and lab tests. Automated usability testing uses software to automatically test predefined use cases.

Heuristic Evaluation: In a heuristic evaluation of a user interface, experts take usability heuristics, meaning usability guidelines and rules, and carefully inspect the user interface with respect to these heuristics. The inspection is done systematically. Nielsen and Molich [1990] describe heuristic evaluation as an informal evlauation method and stress that the use of several experts is important, since individual evaluators are not able to find many usability errors, but the aggregation of findings from three or more experts could identify a satisfactory amount of usability problems. The characteristics of ubiquitous systems call for an extension of traditional metrics [Scholtz and Consolvo, 2004; Rocha et al., 2017]. Mankoff et al. [2003], for example, have described their derivation of heuristics specifically for ambient displays. Since such heuristics and metrics are relevant to the research questions of this work, specific metrics for ubiquitous computing are presented in greater detail in section 2.2.5.

User-based Testing: With user-based testing, a system or a prototype is evaluated with test persons that use the software to carry out given tasks. These tests are performed in a laboratory or as field tests. Often, questionnaires are used to inquire details about the test person that may be relevant for the analysis of test results and to assess user satisfaction. During the test, measurements are taken and usability metrics are applied to identify usability problems the test persons do encounter. Usability tests involving users are useful in early stages of development, as exploratory and formative evaluation, but also in late stages of development as summative evaluation.

Lab-based usability tests, user-based tests can be conducted in early development stages with paper-based prototypes or design sketches [Landay and Myers, 2001]. In wizard-of-oz studies, a prototype is used that is controlled by a moderator, creating an experience for the test person mimicking a fully functional system [Maulsby et al., 1993; Dow et al., 2005].

A field test is a user-based usability test in the actual context of use. The system that will be tested is deployed in its intended target environment [Kjeldskov et al., 2004]. In this version of test, environmental factors affect the user as they would in later use. Just as in lab tests, test persons are given tasks they are supposed to perform using the system Rowley [1994]. Measurements are taken to observe usability metrics and the user's behavior as well as the system's responses. Questionnaires are used as well.

Remote usability tests are user-based tests that users perform on their own computer and at home instead of in a lab or in the field [Hammontree et al., 1994; Thompson et al., 2004]. Remote usability tests can use online questionnaires, video conferencing tools or recordings of usage data to evaluate the usability of a software the user tests on their own computer. As Hammontree et al. [1994] describes, early prototypes such as storyboards can also be tested remotely.

With respect to user-based evaluation, there has been an ongoing discussion of appropriate user study methods for ubiquitous computing. Abowd and Mynatt [2000] note that for evaluation, a system needs to be reliable enough and the innovative nature of ubiquitous systems makes it hard to develop reliable prototypes. Therefore, they advocate iterative and exploratory, user-centric evaluation in the development of ubiquitous systems. The authors also argue that such a system should be evaluated in an authentic setting, where the context of use of the system is given. According to the authors, the methods that are used for evaluation in such an authentic setting should, however, not be the traditional task-centric methods. Since ubiquitous computing systems support everyday computing, there are no specific tasks that can be considered in isolation to evaluate the system's usability.

Consolvo et al. [2002] also conclude that field and lab tests are not suitable for ubiquitous systems. They argue that, for lab and field tests alike, test persons do behave differently when they know they are observed. Additionally, they discuss that situations in an usbaility test are artificial and therefore the results of such a test are insufficiently transferable to the real world. The authors agree with Abowd and Mynatt [2000] that everyday computing can not be evaluated with a task-centric approach. Instead, Consolvo et al. [2002] apply a *Lag Sequential Analysis* which is an observational technique that uses video recordings and the identification and evaluation of events of interest on video. However, Consolvo et al. [2007] later describe several methods of data collection in situ for the evaluation of ubiquitous systems. They argue that ubiquitous systems are around the user in their everyday situations with changing context and therefore traditional lab-based evaluation methods are not suitable for the evaluation of ubiquitous computing, since several context factors would be missing from the evaluation. They present data collection techniques that are focused on evaluation in early stages of prototypical development and discuss their experiences with those methods [Consolvo et al., 2007].

Neely et al. [2008] discuss the outcomes of several workshops about evaluation of ubiquitous systems. They also note that for several application domains, lab-based testing would not be appropriate, for example for applications involving special equipment, specific contexts, or mobility. They also mention that qualitative methods and analysis are more suitable for ubiquitous systems that work in everyday situations.

However, not all ubiqiuitous computing approaches have an everyday computing approach that prevents a task-centric usability evaluation and user-based field studies, and different types of lab studies can frequently be found for ubiquitous and for mobile systems. Hörold et al. [2014] discuss usability field tests specifically for public transport and propose the usage of several data collection methods. They also argue that because of the changing context of use in public transport, lab-based usability tests are not sufficient to evaluate all usability aspects of passenger information systems. The authors describe some difficulties of in situ evaluations in public transport and present guidelines for such evaluations. In Mayas et al. [2014], the same research group describes equipment and its application in field tests in public transport, including eye tracking and interaction cameras.

de Amorim et al. [2019] report a usability test of a mobile public transport ticketing solution in the field, testing their working prototype in context. Participants were asked to travel a specific route and were then interviewed for qualitative feedback. Reis et al. [2015] have systematically reviewed several approaches towards the evaluation of ubiquitous mobile applications. Their analysis found that case studies, field studies and lab experiments were used very frequently and that case studies were used the most, followed by field studies. Kjeldskov and Paay [2012] have found similar results, focusing on mobile systems, where lab studies have been the most common approach, followed by field studies.

Dhouib et al. [2016] discuss several evaluation methods, specifically for adaptive interactive systems and present their advantages and disadvantages. They mention the high costs of task-based experiments as a disadvantage of user-centered lab studies. The authors also argue that methods should be chosen individually for a given prototype and the specific goals of the evaluation.

Automated Usability Tests: Automated usability evaluation started with automatically evaluated metrics of graphical user interfaces, such as layout appropriateness, based on the frequence of use of used widgets and the distance a mouse cursor would have to move between widgets or the balance of widgets on the screen [Sears, 1993, 1995]. Ivory and Hearst [2001] present an extensive survey on automated usability evaluation methods and categorize 132 methods in their own taxonomy. Results of automated evaluation methods are intended to be interpreted by designers and not by systems. One type of methods the authors describe are those that can suggest solutions to usability problems. Most of these systems generate feedback that a designer must interpret. The authors stress that automated usability evaluation methods are not able to capture subjective and qualitative information and therefore argue that adequate usability testing methods are still necessary. Tools applying such methods can help user interface designers to detect usability problems in early stages of user interface design, such as the MeMo workbench by Jameson et al. [2007], for example. It can only be applied to graphical user interfaces and simulates the usage of an interface by modeled users and tasks. This way, usage errors can be identified during user interface design.

Halbrügge [2018] reviews several automated usability evaluation methods with respect to ubiquitous and multi-target user interfaces. The author discusses that model based user interfaces are particularly suitable for automated evaluation, since the models modeled during user interface development can be used for evaluation and no additional models are necessary. An example for the usage of automated usability tests for ubiqutous systems is presented by Feuerstack et al. [2008]. For their Multi-Access Service Platform (MASP) that supports model-based multimodal user interfaces they developed an approach for automated usability evaluation. They adapted the MeMo workbench by Jameson et al. [2007] to simulate user interaction and trigger the runtime adaptation of the user interface. The result is written in log files for designers to evaluate.

In summary, many authors argue for field testing specifically for ubiquitous systems to include real context. However, this requires a working prototype, as for example de Amorim et al. [2019] tested. As Abowd and Mynatt [2000] argued, for the innovative nature of ubiquitous systems, an iterative and exploratory approach is advisable, in which early prototypes can also be tested and then developed further. This, however, means that there is a need of evaluation methods for early prototypes of ubiquitous systems. The selection of a suitable usability evaluation method for such early prototypes depends on the individual characteristics of a system and the usability factors that are to be measured.

Usability Heuristics and Metrics for Ubiqutous Computing

Usability factors and metrics help in the design and conception of ubiquitous systems, and in the evaluation of a system's usability as well. Usability factors can be used to derive measurable criteria, for example as *time behavior*, which then can be used in user-based tests to measure aspects of the usability of a system [Seffah et al., 2006]. Deciding which usability factors are relevant essentially means deciding what to measure and test in an usability test. Phrased as usability principles or rules, usability factors can be used as heuristics for the heuristic evaluation of a system [Nielsen, 1993b].

Often, such principles are based on basic attributes of usability, such as *learnability*, *efficiency*, *memorability*, *errors*, and *satisfaction*, as Nielsen [1993b] described. He derived principles for a usabilty heuristic, for example *simple and natural dialogue*, *speak the user's language* and *minimize user memory load*, among others Nielsen [1993b]. Ben Shneiderman et al. [2018] have defined eight golden rules of interface design, that can also serve as a heuristic for an evaluation. Those rules include *strive for consistency*, *cater to universal usability* and *design dialogs to yield closure*, for example.

Because of the differences between ubiquitous systems and Windows, Icons, Menus and Pointers (WIMP)-based, Graphical User Interface (GUI)-centric systems, usability factors and usability metrics that apply for GUIs and desktop applications do not necessarily apply to ubiquitous applications and some factors may be new in ubquitous environments. These differences have been frequently discussed for mobile systems, but in some works also for ubiquitous and context-aware systems.

Scholtz and Consolvo [2004] propose a framework for evaluating ubiquitous computing applications. The authors argue that a framework of ubiquitous computing usability attributes and metrics can enable heuristic evaluations. They identified nine ubiquitous computing evaluation area and propose several metrics for each of these areas, for example *privacy*, *awareness*, and *control* as metrics for *trust* [Scholtz and Consolvo, 2004].

Rocha et al. [2017] have developed a set of heuristics specifically for the usability of ubiquitous systems. They evaluated and refined their heuristics with expert evaluations. Their set of heuristics comprises fifteen heuristics that extend

Nielsen's heuristics by specific ubiquitous heuristics, for example *privacy and safety*, *context awareness and adaptive interfaces* and *sensors and data input* [Rocha et al., 2017; Nielsen, 1993b].

A more detailed analysis of usability factors and metrics for mobile and ubiquitous systems can be found in chapter 3, in section 7.1.

2.3 Semantic Web Technologies

Berners-Lee et al. [2001] described the Semantic Web as a vision of providing machine-readable semantics to the World Wide Web (WWW), making it more intelligent. In this vision, semantics enables intelligent agents that can utilize the vast information on the web. This vision sparked numerous research efforts resulting in several innovations which had an impact on research fields far beyond the WWW. In the course of this research efforts, the World Wide Web Consortium (W3C) established several standards for the formalization and the sharing of knowledge in ontologies [W3C, 2012; Group, 2013; Sch, 2014]. These standards ensure the reusability of knowledge and knowledge models. Based on them, a multitude of tools and technologies have been developed that enable ontology modeling and handling of semantic data. Ontologies are widely used in context-aware and ubiquitous systems to express knowledge and to reason on this knowledge [Chen et al., 2003a; Gil and Pelechano, 2017]. The following sections therefore will describe the technical foundations for ontology modeling and handling based on the W3C standards.

2.3.1 Ontologies

Ontologies are a abstraction method that stems from philosophy, where an ontology is a systematization of everything that exists [Gruber, 1993]. In computer science, ontologies have been adopted as a means of knowledge modeling and sharing. Ontologies were adopted for artificial intelligence to formalize knowledge, standardize terminology and share this knowledge between systems or parts of systems. Gruber [1993] described an ontology as "*an explicit specification of a conceptualization*". It makes knowledge *explicit* and explicitly describes the *conceptualization* of entities, relationships and concepts that belong to the specified knowledge. Uschold and King [1995] describe the term *conceptualization* as "*an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality*".

Studer et al. [1998] later focused the definition of Gruber into the definition "*An ontology is a formal, explicit specification of a shared conceptualisation*". This definition included the formalization that is necessary to utilize an ontology in knowledge-based systems, making it machine-readable. It also adds the notion that an ontology is a shared conceptualization, meaning that a shared understanding of the formalized knowledge must be achieved. Ontology engineering as a discipline has developed several methods, processes and tools that structure the development of an ontology and support the process of finding a shared understanding of concepts, relationships and entities [Mizoguchi and Ikeda, 1998; Sure et al., 2009; Mizoguchi and Kozaki, 2009]. Central to all efforts of ontology and knowledge engineering are standardized representation formats.

Resource Description Framework - RDF

The Resource Description Framework (RDF) is a data model for representing resources and the relationships between resources [Sch, 2014]. First conceived as a metadata format, RDF was later utilized as a basic data format for the Semantic Web [Hitzler et al., 2008].

RDF knows resources and relationships between *resources*. A RDF relationship is called a *property*. RDF data is expressed in the form of triples. A triple consists of a *subject*, a *predicate* and an *object*. The subject and object are resources, while the predicate is the property. A property is always directional. With these three elements, an RDF statement can be expressed as shown in listing 2.1.

```
<subject> <predicate> <object>
```

Listing 2.1 — A RDF statement.

Using several statements, facts about resources can be expressed. Then, one resource is referenced in several triples. An example is given in listing 2.2.

```
<The Gruffalo> <is a> <children’s book>.
<The Gruffalo> <was written by> <Julia Donaldson>
<The Gruffalo> <was drawn by> <Axel Scheffler>.
<The Gruffalo> <was published> <1999>
<Room on the Broom> <is a> <children’s book>.
<Room on the Broom> <was written by> <Julia Donaldson>.
<Room on the Broom> <was drawn by> <Axel Scheffler>.
<Room on the Broom> <was published> <2001>.
<Julia Donaldson> <is a> <writer>.
<Axel Scheffler> <is a> <illustrator>.
```

Listing 2.2 — Several RDF statements, expressing facts about children's books and authors.

With various statements about resources, these statements can be represented as a graph, as shown in figure 2.1. Resources are visualized as nodes and properties are shown as the edges connecting the resources. Arrows show the direction of a property.

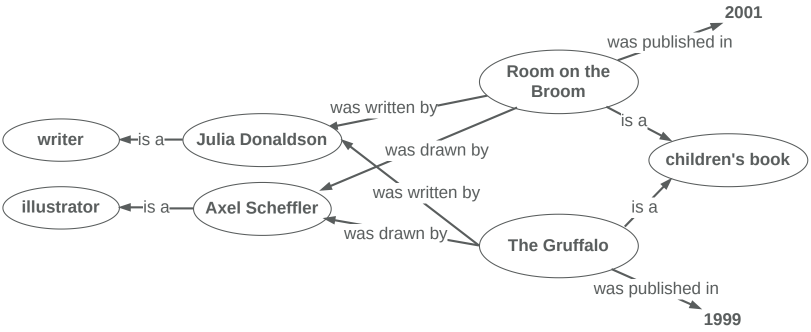


Figure 2.1 — An example RDF graph, expressing the data from listing 2.2.

For reuse of RDF formalizations and graphs, unique identifiers are necessary. With Semantic Web technologies, Uniform Resource Identifiers (URIs) are used as unique identifiers for resources and properties [Berners-Lee et al., 2005]. In newer documents, Internationalized Resource Identifiers (IRIs) are used as a generalization of URIs that allow the use of more Unicode characters [Duerst and Suignard, 2005]. As in Extensible Markup Language (XML), namespaces are used in RDF documents to abbreviate the unique identifiers [Bra, 2009].

RDF defines the use of *literals* that are not resources with IRIs but have a datatype. Since they represent basic values, they can only be objects of a triple. Available datatypes and their use are described in the RDF standardization documents [Cyg, 2014].

Several formats are available for the serialization of RDF. An XML syntax was standardized, as well as a JavaScript Object Notation (JSON) serialization [Gan, 2014; Kel, 2020]. A simpler format that is more human-readable is Terse RDF Triple Language (Turtle) [Beckett et al., 2014]. The listings 2.1 and 2.2 are written in basic Turtle format. For this document, Turtle will be used in listings.

SPARQL Protocol and RDF Query Language

Data that is represented in RDF can be queried by the SPARQL Protocol and RDF Query Language (SPARQL) [Har, 2013]. SPARQL defines a request format that uses triples to match RDF data in an RDF graph. It uses the Turtle syntax to define request patterns. SPARQL is able to return results in several query result formats, such as the "SPARQL Query Results XML Format"or the "SPARQL Query Results JSON Format", both specified by the W3C [Haw, 2013; Sea, 2013]. Other formats, for example Comma-separated values (CSV) are also available.

SPARQL defines several types of queries: `ASK`, `CONSTRUCT`, `DESCRIBE` and `SELECT` [DuCharme, 2013]. With SPARQL Update, the query language was extended by `DELETE`, `INSERT` and `UPDATE` queries that allows the modification of RDF data in the graph. The `SELECT` query is the most frequently used query and it's syntax is shown in listing 2.3. Further explanations and definitions of SPARQL can be found in the language specification or in the book "Learning SPARQL" by DuCharme [2013], for example.

```
PREFIX ex: <http://iums.eu/ns/example#>

SELECT ?author WHERE
{ ex:TheGruffalo ex:wasDrawnBy ?author.}
```

Listing 2.3 — A SPARQL SELECT query for the data shown in listing 2.2 (extended by an example namespace).

RDF Schema

Just as XML Schema is a schema definition for XML, Resource Description Framework Schema (RDFS) provides a syntax and a formal semantics to express a vocabulary for RDF [Mal, 1999; Bri, 2014]. With RDFS , RDF documents can model a shared knowledge that can be interpreted by different components in the same way. RDF allows to express facts about resources, that are individuals or instances. RDFS enables modeling concepts by introducing *types* for resources and therefore adds a conceptual layer to RDF. RDF can thus be used to express ontologies [Hitzler et al., 2008].

With the definition of classes, RDFS introduces the concept of inheritance. This concept is also applied to properties so that a RDF Schema can define class hierarchies as well as property hierarchies. RDFS allows simple reasoning, since the inheritance relationships `rdf:subClassOf` and `rdf:subPropertyOf` are transitive and reflexive [Hitzler et al., 2008].

```
<ex:Book> <rdf:type> <rdfs:Class>.
<ex:ChildrensBook> <rdf:type> <rdfs:Class>.
<ex:ChildrensBook> <rdfs:subClassOf> <ex:Book>.
<ex:TheGruffalo> <rdf:type> <ex:ChildrensBook>.

<ex:Person> <rdf:type> <rdfs:Class>.
<ex:drawnBy> <rdf:type> <rdf:Property>.
<ex:drawnBy> <rdfs:domain> <ex:Book>.
<ex:drawnBy> <rdfs:range> <ex:Person>

<ex:writtenBy> <rdf:type> <rdf:Property>.
<ex:writtenBy> <rdfs:domain> <ex:Book>.
<ex:writtenBy> <rdfs:range> <ex:Person>.

<ex:hasName> <rdf:type> <xsd:String>.

<ex:AxelScheffler> <rdf:type> <ex:Person>
<ex:AxelScheffler> <ex:hasName> "Axel Scheffler"^^xsd:string.
<ex:JuliaDonaldson> <rdf:type> <ex:Person>
<ex:JuliaDonaldson> <ex:hasName> "Julia Donaldson"^^xsd:string.

<ex:TheGruffalo> <ex:drawnBy> <ex:AxelScheffler>.
<ex:TheGruffalo> <ex:writtenBy> <ex:JuliaDonaldson>.
```

Listing 2.4 — A simple RDFS ontology and RDF statements in Turtle format using an example namespace.

For properties, RDF Schema allows the definition of domain and range in form of classes or datatypes using the properties `rdfs:domain` and `rdfs:range`. Listing 2.4 shows an example of a small RDFS ontology and expresses some of the statements of listing 2.2 using the defined vocabulary. The class `ex:ChildrensBook` is defined as a subclass of `Book`, for example. Furthermore, the properties `ex:drawnBy`, `ex:writtenBy` and `ex:hasName` are defined with domain and range.

The formal semantics of RDF Schema allows reasoning over RDF data. In the example of listing 2.4, reasoning can deduce that `ex:TheGruffalo` is a `ex:Book` as well as a `ex:ChildrensBook`, for example. However, RDF Schema does have some drawbacks and lacks the level of expressiveness that is sometimes necessary in order to model ontologies.

Web Ontology Language

The Web Ontology Language (OWL) is an ontology language developed for the Semantic Web that is based on first order predicate logic [W3C, 2012]. It is standardized by the W3C and currently available in version 2 as OWL 2. It was influenced by several other ontology languages, DAML+OIL being the most important. OWL is more expressive than RDFS, meaning that in OWL several expressions are possible that cannot be expressed in RDFS, for example negation [Hitzler et al., 2008]. Instance data of a given OWL ontology is frequently represented as RDF data.

Several partial languages of OWL exist: OWL Lite, which is a part of OWL DL which is itself a part of OWL Full. The languages are different in expressiveness and consequently in complexity. OWL Full as the most expressive and complex language is, in contrast to the other two dialects, undecideable [Hitzler et al., 2008].

For OWL, a RDF syntax exists that makes documents defining OWL ontologies machine-readable. An OWL/XML syntax also exists [Mot, 2012]. However, as RDF and RDFS, OWL can be serialized in Turtle, too.

OWL is declarative, which means it describes facts. An ontology statement is also called an *axiom*. OWL uses the same basic components as RDFS: *classes*, *individuals* or *entities*, *properties* and *literals*. In addition to these components, OWL further specifies properties and distinguishes between `owl:DatatypeProperty` that only uses a datatype as range, which can be specified using the `rdfs:range` property. `owl:ObjectProperty` in contrast relates entities with other entities. These are both subclasses of `rdf:Property`.

With OWL, it is possible, among other things, to define classes as disjoint, as equivalent, as an intersection or as a complement of each other [Hitzler et al., 2008]. Furthermore, all instances of a class can be directly specified, implying that no other individuals can be an instance of this class. Other cardinality restrictions on classes are also possible [Hit, 2012].

Further language specifics can be found in the official OWL specification and documentation or, for example, in the book “Semantic Web” Hitzler et al. [2008], [W3C, 2012; Hit, 2012].

For the remainder of this document, the Turtle syntax will be used for the presentation of RDF, RDFS and OWL statements. Further, a graphical notation for graphs expressing RDF, RDFS or OWL is defined in figure 2.2. Classes will be shown in white ovals, while individuals will be depicted in blue ovals. The `rdf:type` property is also shown in blue to distinct it from other properties that will be

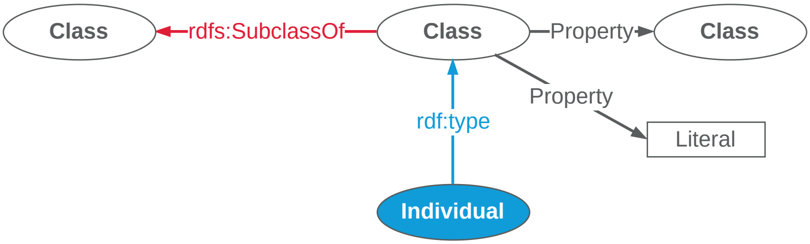


Figure 2.2 — A graphical notation for RDF, RDFS or OWL graphs.

shown in dark grey instead. The `rdfs:subClassOf` and `rdfs:subPropertyOf` properties will be printed red and literals will be shown in rectangles.

Semantic Web Rule Language

To express rules on facts expressed in OWL ontologies, the Semantic Web Rule Language (SWRL) was developed and submitted as a proposal to the W3C in 2004 [Horrocks et al., 2004]. SWRL is based on RuleML, a specification for the expression of rules in XML [Boley et al., 2001]. SWRL builds on OWL Lite or OWL DL and is used to express rules that have an *antecedent* and a *consequent*, both consisting of atoms that are statements over OWL class membership, data range or properties, variables or individuals, or some built-in relations of SWRL.

$$drawnBy(?book, ?person) \implies illustratorOf(?person, ?book) \tag{2.1}$$

A rule such as rule 2.1 expresses that if the *antecedent* evaluates as true, the *consequent* is true, as well. The example rule 2.1 is written in a human-readable syntax. There also exist a formal EBNF syntax as well as serializations in XML and in RDF [Horrocks et al., 2004]. In the remainder of this document, the human-readable syntax will be used for SWRL rules.

SWRL rules have been used in several research efforts and bindings to several existing rule engines exist [O'Connor et al., 2005; Rigas et al., 2012].

2.4 Context

As discussed before, context plays an important role in ubiquitous systems. However, it is a very broad concept and therefore it is hard to grasp and many definitions are very general. Context must always be closely analyzed and discussed specific to an application domain and system's design. Picking a basic definition of context sets a starting point for this analysis. The handling of context in a system then requires a context model or data structure, a storage solution, decisions about context interpretation or reasoning and the provision of context data to the context-aware application, as for example discussed by Perera et al. [2014] in their Context Lifecycle. Definitions and concepts for handling context are discussed in the following sections.

In a survey on engineering context-aware systems, Alegre et al. [2016] write that context acquisition and the utilization of context should be separated from each other, for reusability and to benefit from loose coupling. Many approaches toward context-aware architectures implement this principle, for example of Dey et al. [2001], Hong and Landay [2001] or Dockhorn Costa et al. [2005]. Following this separation, the next section discusses context and tasks for handling context, while context-awareness and adaptation is covered in section 2.5.

Definitions of Context

The first context-aware ubiquitous or mobile systems were essentially location-aware. One of the earliest examples is the Active Badge System by Want et al. [1992]. Schilit et al. as one of the first authors defining and describing context-aware systems, defined context as "*Where you are, who you are with and what resources are nearby*", focusing on location and spatial relations [Schilit and Theimer, 1994]. Mobile computing research examined the usage of location information for mobile devices and developed location-aware services, for example by Leonhardt et al. [1996]. Mobile tourist guides are a type of mobile and ubiquitous applications that are related to applications for mobility and often used location as context information. Several prototypes of context-aware mobile tourist guides were developed by Cheverst et al. [2000], among others. In the NEXUS project, the NEXUS platform for location-based context-aware applications was developed [Hohl et al., 1999; Nicklas et al., 2001]. It provides an augmented world model that connects modeled representations of real-world objects and can manage their locations to be accessed for location-aware applications Nicklas and Mitschang [2004]; Lehmann et al. [2004].

With further research on mobile and ubiquitous computing, context definitions were extended and more aspects of context were discussed, for example by Schmidt et al. [1999], arguing that "*There is more to context than location*". Schmidt et al. prototyped systems that used ambient lighting and device orientation as context data for user interface adaptation.

In a survey on context-aware systems, Chen and Kotz [2000] discuss context definitions and add the time categorgy to the definition of Schilit and Theimer

[1994]. They refer to the time of day, the weekday and other time aspects that are relevant to context.

One of the most widely used context definitions was given by Dey and Abowd [2000], who write:

> *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."* [Dey and Abowd, 2000]

They identify that persons, places or objects can be entities whose situation is described by context. Otherwise, their definition is very general. They identify categories of context information that are highly relevant to context-aware systems, which are location, identity, activity and time and refer to these categories as *primary* context [Dey and Abowd, 2000]. These basic context categories have later been supplemented with other categories, for example relations, added by Zimmermann et al. [2007].

2.4.1 Context Lifecycle

Context can be acquired from several sources and for most context-aware systems, context sources are very heterogeneous. Dey et al. [2001] argue for a separation of concerns, separating context processing from applying context in a context-aware system, therefore reaching higher reusability of components. Taking separation of concerns one step further, the tasks for handling context information can be divided into several phases.

A context-information lifecycle is one way to describe the steps in which a context-aware system handles context information. In a survey on context-awareness for the Internet of Things, Perera et al. [2014] have reviewed several context lifecycles from the literature and proposed a unified model. They differentiate the following steps in their lifecycle:

1. Context Acquisition
2. Context Modeling
3. Context Reasoning
4. Context Dissemination

The modeling step in Perera's context lifecycle covers the development of a context model and data structure for context, as well as the conversion of context

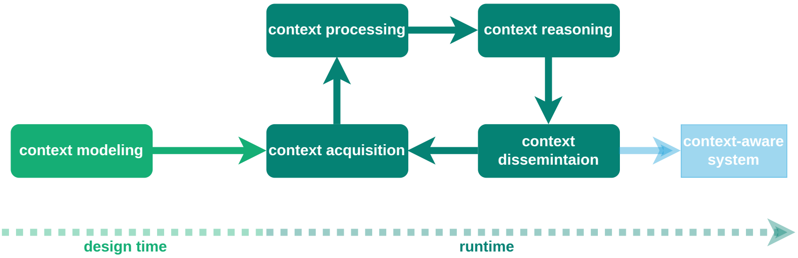


Figure 2.3 — Modified context lifecycle after [Perera et al., 2014].

data into this model or data structure. This mixes the tasks for the design phase of a system with tasks at runtime. In the design phase, the decision for a context modeling technique, and for concepts to store context data must be made and context must be modeled. At runtime, context data must be transformed into the context model and stored. I therefore extend the context lifecycle by the context processing step and consider context modeling as a step that is done at the design time of a system, as shown in figure 2.3, separate from the additional context processing step during runtime, which includes the transformation of context data from sources into the context model.

Context Modeling

A context model provides a data structure for context data, but in many cases also models relationships between context facets. Many context models support the inference of new context facts from collected context data, applying rules that are expressed in the context model. Strang and Linnhoff-Popien [2004] have surveyed and discussed context modeling techniques. They identified several requirements towards context models for ubiquitous computing environments that apply to context modeling methods. These requirements can be structured as follows:

- requirements that stem from the distributed nature of context-aware ubiquitous systems:
  - can the context model or context data be distributed?
  - can a context model be partly validated?
- requirements concerning the data and metadata that is represented in the model

  - if and how are data richness and data quality modeled?
  - does the model handle ambiguity or incompleteness of data?
  - how high is the level of formality?
- how applicable is the context model to already existing systems?

Key-Value Pairs: The simplest representation of context data is in key-value pairs. A context value is assigned to a context. This approach lacks formalization, reusability is not very high and the model does not provide an underlying semantics that would enable reasoning of higher level context. However, it is a simple and lean approach. Several approaches have implemented this modeling technique, for example Petrelli et al. [2000] or De Virgilio and Torlone [2005]; De Virgilio et al. [2006].

Markup Scheme Models: These models utilize markup schemes, such as variants of the Standard Generic Markup Language (SGML), Composite Capabilities / Preference Profile (CC/PP) or XML. Their hierarchical structure can provide more formalization than key-value pairs do. Several approaches applied CC/PP and dialects thereof for context modeling, but also uncovered its limits, as described by Strang and Linnhoff-Popien [2004]. Schemes can be used for consistency and validity checking on syntax level. While they enable a shared understanding of the syntax, markup schemes do not provide means to make semantics machine-processible. Especially XML based approaches can benefit from many tools that are available for handling XML documents. Knappmeyer et al. [2010] have developed ContextML, which is a context representation and management schema based on XML. It is specifically built to be used in Representational state transfer (REST)-interfaces, supporting distributed context-aware systems.

Graphical Models: Graphical models are mostly used to represent context models for human consumption. Depending on the syntax that is used, some can be transformed into more machine-oriented formats. Some graphical models can be converted into Entity-Relationship (ER) diagrams and then into relational database schemes. Unified Modeling Language (UML) diagrams have also been used. UML based context models represent a programming-language specific data structure and can be used to derive code. The level of formality is, in most cases, relatively low. An example was presented by Henricksen et al. [2002], who are reporting they have tried UML and ER modeling, but found them lacking expressiveness. They propose a different graphical modeling technique, specially designed for context modeling. Their approach explicitly models various types of associations between entities or attributes, for example a *sensed* *association* or a *temporal association*. The authors also provided a mapping from this model to a context management system, based on Object-Role Modeling (ORM) by Halpin [2001], [Henricksen et al., 2003].

Object-Oriented Models: Object-oriented data structures are frequently used as context models. Context models can benefit from the object-oriented paradigm, specifically from inheritance and encapsulation, providing reusability. Context objects can hide the details of context processing and provide a uniform interface for context access. Strang and Linnhoff-Popien [2004] note, that such context models are extensible and can be used in a distributed environment, but the approach may be ressource-intensive. Object-oriented context models are used very often, for example in the GUIDE project, a context-aware tourist guide by Cheverst et al. [2000]. Another example is the NEXUS project, where an augmented world model was developed that manages the location and spatial relations of real-world objects Nicklas et al. [2001].

Logic-Based Models: Logic-based models use some kind of logic which applies rules and axioms to expressions and facts to derive new expressions or new facts. Besides supporting reasoning, logic-based models have a high formality, but reasoning is often ressource-intensive. McCarthy [1993] applied logic to context as early as 1993. Gray and Salber [2001] proposed the application of first-order predicate logic to context modeling. Fahy and Clarke [2004] describe another example for an approach applying a logic-based model and utilizing an inference engine to reason about context.

Ontology-Based Models: As a very expressive knowledge representation, ontologies are often used for context modeling. They provide reasoning support, validation mechanisms, a high degree of formalization and high reusability. Most ontologies support an object-oriented approach towards knowledge modeling, offering the advantages of the object-oriented approach. Since ontology languages are mostly programming-language independent, they overcome one of the drawbacks of object-oriented models. Bettini et al. [2010], among others, argue that the explicit and formalized semantics of ontologies enables knowledge sharing and therefore is beneficial in heterogeneous computing environments and supports interoperability. OWL, especially in its dialect Web Ontology Language - Description Logic (OWL-DL) and RDFS are examples for ontology language standards that are widely used. Ontologies are, however, complex and modeling is extensive and costly. Bettini et al. [2010] stress that graphical modeling tools like Protegé compensate this disadvantage to a certain degree since they provide extensive modeling support. Reasoning with ontologies or instance data can be expensive and not very efficient, which can lead to problems in highly distributed and realtime environments. Some projects follow a hybrid approach. In the NEXUS project, an extension of the object-oriented location model using ontologies for knowledge representation and reasoning was proposed, for example [Becker and Nicklas, 2004]. Several context ontologies for application in ubiquitous context-aware systems were developed, for example the context ontology (CONON) by Wang et al. [2004]. It used OWL to model context and utilize the reasoning capabilities of OWL in a service-oriented middleware for context-awarenes in the Smart Home domain [Gu et al., 2005]. Chen et al. [2004] proposed SOUPA, a "standard ontology for ubiquitous and pervasive applications", which evolved from the CoBrA project, implementing a context-aware meeting room.

Strang and Linnhoff-Popien have come to the conclusion, that ontologies are the most appropriate modeling technique for context-aware ubiquitous computing environments. However, depending on the type of application or system, the decision for a type of context models might be made differently.

After the context modeling technique is decided, the content of a context model must be modeled. Taking a context definition as a basis or using an own context definition is a starting point for context analysis. Context models of high complexity often contain a more general and an application- and domain-specific part. For the general context model, the primary context categories of basic context definitions can be a point of origin, for example choosing location, identity, activity and time. The general structure of a context model is also an important aspect. Mostly depending on the type of application, some approaches model the ambiguity, quality or confidence of context data directly or as context metadata. Dey et al. have identified the modeling or handling of ambiguity in context data as an important research question and Bolchini et al. have considered how context models handle ambiguity, incompleteness and data quality in their review of context models, for example [Dey et al., 2001; Bolchini et al., 2007].

The type of storage that is used to store context data at runtime can either influence the general structure of a context model or is influenced by it. It depends on the modeling technique - key-value pairs or object-oriented context models can be mapped in code, for example, XML-based context management uses files and ontology-based models can be used with any of the existing frameworks for managing and utlizing ontologies and ontology-described instance data. Keeping a full context history requires storage solutions to manage such a history and enable querying of this history. Depending on wether ambiguity, quality or completeness of context data is modeled and depending on how it is modeled, there are additional infrastructure requirements.

Context Acquisition

At runtime, the context lifecycle starts with the acquisition of context data. Different researchers summarize different system tasks in this step. Data gathering at the source can be one of them, but also data preprocessing, sensor fusion or the deduction of new context data from one or several sources. Here, I will concentrate on the process of data gathering from sources and will consider all further processing of context data in the following step, discussed in section 2.4.1. Context sources can be manifold. Sensors are used very often, but Web services and other software services can also provide context data, such as a user's schedule or calendar, for example. Indulska and Sutton [2003] discussed different types of sensors as context sources, distinguishing hardware sensors that physically sense context data from virtual and logical sensors. Their analysis is focusing on location as context, but Perera et al. [2014] have used this distinction for other types of context data as well. Virtual sensors are software sensors that access data sources, such as applications or Web Services, while logical sensors combine sensor data from several sensors and infer new context data.

Apart from the types of sensors, the sensor access mechanisms are relevant in the context acquisition task. Data can be accessed either using a push or pull pattern. A sensor pushes context data towards a data sink or the data sink pulls data from the sensor via request. Both methods are discussed by Perera et al. [2014] or Alegre et al. [2016], for example. Dey et al. [2001] also discussed the push/pull access to sensor data, calling it notification/request mechanisms. The distinction between accessing sensors in context acquisition and accessing context information in the context dissemination phase is sometimes not clear and can depend on the chosen architecture. In the context acquisition phase, a component that gathers and then stores context data is accessing the data, while in context dissemination, an application that uses context information for adaptation purposes is accessing the data. Parallel with the question of push vs. pull comes the question of when to access sensor data. This question is often depending on the type of data the context provides. Some data is continuous in nature, which is true for most physical conditions, for example temperature. Some data is event-based or singular, such as the detection of a person in a room. Data access for continuous data can be provided as a continuous data stream. Kwon et al. [2010] have developed a system that processes and provides spatial context data from data streams in realtime, for example. Most applications, however, do not need streamed data in realtime and therefore can access continuous data in intervals depending on the application's requirements on data resolution. For the processing of singular or event-based data, data access can be instantly when the event occurs, for example when a threshold of temperature is crossed, or also in intervals, which can be necessary when the event is going on for some time, for example when a presentation is held in a room. Perera et al. [2014] have discussed these questions in greater detail.

Context Processing

Context processing is the task of making raw data useful for a context-aware system. The goal of this step is to provide context information in the common data structure or model of the system. Components in this step provide an abstraction of the possibly heterogeneous context data sources in a context-aware system. Sanchez et al. [2006] have differentiated between raw data and context information. They described raw data as data that is directly obtained from sensors, in contrast to context information that is generated by processing this raw data. Transforming data into context information can include adding metadata about the data source, time of sensing and other relevant information. Information about data quality, incompleteness or confidence can also be added here. Data mining approaches can be used to clean the raw data, for example by filling missing data points or removing outliers [Perera et al., 2014].

Besides transforming raw data into context information conforming to a context model, new context data can be generated in this step. In many approaches, this process is referred to as creating high level context from low level context or interpretation, as Dey et al. [2001] call it. Some authors refer to all mechanisms for this process as context reasoning, which is a separate step in this context lifecycle. I differentiate deriving new context data from raw context data, which takes place in the context processing step, from inferring new context information based on information in the context model, which I discuss in the context reasoning step.

Sensor fusion is one concept that takes data from (physical) sensors and generates context data from it. Schmidt et al. [1999]. Sensor fusion can be used when data from one sensor alone is not precise enough. Combining data from several sensors could enhance the resolution of a context fact. Another example of sensor fusion for context-aware systems is presented by Padovitz et al. [2005], determining activities of a user in a smart meeting room [Padovitz et al., 2005].

Context Reasoning

The application of rules or logic-based inference on context information takes place in this phase of the context lifecycle. As mentioned before, in contrast to the processing steps operating on raw data, I refer to methods using context information in a given context model as well as the characteristics of the context model as *context reasoning*. The details of available methods for context reasoning depend on the context model that was used. Perttunen et al. [2009] highlight logic programming, ontology-based reasoning and case-based reasoning as relevant reasoning strategies. Perera et al. [2014] list slightly different categories, including supervised learning, which contains decision tree classifiers but also artificial neural nets, unsupervised learning, rule-based approaches, fuzzy logic, ontology-based and approaches applying probabilistic logic.

Logic-based reasoning applies rules on facts to infer new rules. Most approaches use first order logic and some logic programming language to implement reasoning. An example is the situation model using predicate logic by Henricksen and Indulska [2006].

The reasoning capabilities of ontologies based on description logic is one of their biggest advantages for context modeling. For the standard ontology languages there are several reasoners available. Ontologies use a description logic for modeling that can be used to reason, for example, which classes an individual is instance of. Several authors have argued, that reasoning based on the description logic of ontologies is not expressive enough for context reasoning and is very expensive and ressource intensive, especially when context is changing rapidly [Perttunen et al., 2009; Bettini et al., 2010; Machado et al., 2019].

The class of rule-based approaches is very diverse. Simple realizations use IF THEN ELSE rules, for example the ECA-rules approach by Daniele et al. [2007]. Rule engines can be used to apply rules during runtime, for example in the MiRE system by Choi et al. [2008]. Rules also can be combined with ontological approaches, for example using SWRL as a rule language. The system presented by Ricquebourg et al. [2006] is one example for combining SWRL rules with an ontology for context-awareness.

Case-based reasoning is an approach where cases represent situations or problems and predefined or already solved cases are stored, with their solutions, for later retrieval. For new cases, similiar cases are searched, using a similiarity measure. The solution of the best fitting case is then applied to the new case. After solving the new case, it is persisted together with the generic part of its solution. Different types of data structures are used. Zimmermann [2003] presents an approach to apply case-based reasoning for context-aware systems.

Since most of the reasoning approaches do only satisfy some of the requirements towards context reasoning, hybrid reasoning strategies have emerged [Perttunen et al., 2009; Bettini et al., 2010]. Machado et al. [2019] have very recently reviewed several hybrid reasoning approaches and found that most of them use different reasoning strategies in parallel, but not combined.

Context Dissemination

In this phase, the gathered and refined context information is made accessible to applications that use it to implement context-awareness. Analogous to the context acquisition phase, one of the central questions in this phase is, if context information is delivered to the application or if the applications requests it. The push mechanism for context dissemination is mostly implemented using the publish/subscribe pattern, while the pull mechanism is realized using a query interface for applications. The publish/subscribe pattern utilizes asynchronous communication, while a query interface implements synchronous request/response communication. In their Context Toolkit, Dey et al. [2001] implemented callbacks in a publish/subscribe manner for their context widgets, but also interfaces for requesting context from widgets. The CoBrA framework by Chen et al. [2003b] and Chen [2003] also implements a query protocol, but a publish/subscribe protocol was planned, too. The system by Ricquebourg et al. [2006] and Ricquebourg et al. [2007] depends on a publish/subscribe communication utilizing an event bus to connect sensors and actuators to application services. In the NEXUS project, an approach was developed that specifically targets the requirements of large scale systems [Grossmann et al., 2005].

[Bellavista et al., 2012] have provided a more detailed view on context dissemination and discuss context data delivery from a routing standpoint. They review several dissemination strategies for context-aware systems, ranging from direct sensor access for applications to flooding-based algorithms or even gossip-based strategies that use a probabilistic approach. The authors propose to investigate the adaptation of the context data distribution approach itself, based on context.

2.5 Context-Awareness and Adaptive Systems

Designing a context-aware system requires decisions on *what* to adapt, *when* to adapt and *how* to adapt. The following sections first present definitions of context-aware systems and then discuss several possible answers to these questions.

2.5.1 Definitions of Context-Aware Systems

Schilit and Theimer [1994] defined context-aware computing in 1994 in the context of mobile computing. They described *context-aware software* as software that *examines* the computing environment and *reacts* to changes to this environment. This definition contains two steps a context-aware system would perform repeatedly: examine the environment, in an implementation gathering and interpreting context data, and then reacting to it, which would be implemented as adaptive actions the system performs. In a prototypical system they present, Schilit et al. described *proximate selection* as one such adaptation, where the nearest input or output device is chosen for a user, for example printers, displays or speakers [Schilit and Theimer, 1994].

Dey and Abowd discussed these and other definitions of context-awareness and added their own definition:

> *"A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."*
> [Dey and Abowd, 2000]

This definition focuses adaptation as the provision of relevant information or services and introduces their relevancy as the selection criteria for services and information. Dey and Abowd stress that centering context-awareness on the adaptation of an application's behavior excludes systems that display context to the user, enhancing visualizations. Their definition therefore explicitly includes those systems.

2.5.2 What to Adapt in a Context-Aware System

Definitions of context-awareness agree that context-aware systems adapt, select or change system-related functions, but this is still a very general description. Several authors have tried to describe categories of adaptation in context-aware systems. Pascoe [1998], for example, described four capabilities of context-aware systems: "*sense, react, interact and augment*". Pascoe [1998] calls displaying gathered context information to the user "*contextual sensing*" and mentions locating the user on a map using location context and a map marker as an example. The category "*contextual adaptation*" describes applications that "*integrate more seamlessly with the user's environment*" using context information, for example adapting the color scheme of a visualization at night. "*contextual resource discovery*" is a category for applications that use context to discover resources in the computing environment and utilize them, according to context, for example by proximity. The author illustrates this category with the example of a wearable computer using a nearby, unused display to display additional information, an example that fits within the scope of this work, redirecting the system's output. The fourth category is "*contextual augmentation*", which comprises augmented reality approaches that augment the environment using context data or other relevant data and the augmentation of the virtual with sensed context data.

Schmidt et al. [1999] outline three possibilities of applying context in ultra-mobile computing. They describe the adaptation of user interfaces as one possibility, where *interaction styles or display modes* are changed based on context, specifically environmental conditions. As another category, they describe applications that adapt their *communication* to context conditions in a highly mobile environment, including urgency and interruptibility as factors that influence the system's usability. As a third category, Schmidt et al. [1999] mention systems that *pre-select applications* based on context for improved assistance. This pre-selection could be task-based or situation-based, for example.

Based on a review of these and other classifications, Dey and Abowd [2000] categorized three possible features of a context-aware application. They described these features as follows:

- *Presentation* of information and services to the user: the presentation of context information to the user would fall in this category, as well as the presentation or highlighting of information or services to the user that are relevant to the user's context.
- Automatic *execution* of a services: actions that are triggered by context and executed by the system, equivalent to the "*contextual adaptation*" category by Pascoe [1998].
- *Tagging* of context information for later retrieval: this category includes the "*contextual augmentation*" by Pascoe [1998]. Information is automatically enhanced with context data.

In their survey on engineering context-aware systems, Alegre et al. [2016] discussed these categories and modified them. They emphasized the difference in interactivity levels of context-aware systems and distinguished between active and passive system behavior, which can be described as the degree of automation. An active system realizes an adaptation proactively and acts autonomously. In a passive system, the user is offered a selection of possible actions and selects one to be executed. The authors adopt the categories of Dey and Abowd [2000] and split the presentation category in two. The result are the following adaptation options in context-aware systems:

- Presentation of information to the stakeholders: the authors extended the scope of the user to stakeholders, including primary users and secondary users or system's engineers. The situations of interest in which context information can be presented or can be used to present information may vary for different users.

- Active execution of a service: the system acts actively, meaning it decides autonomously, based on current context, to execute a certain service. So-called self-adaptive systems fall into this category, cf. Krupitzer et al. [2015], for example.
- Passive execution of a service: in this category, the user "is in the loop" and decides about the adaptation the system performs. This category therefore includes systems that Dey and Abowd [2000] have categorized in their first category, where the system presents options to choose from to the user.
- Active configuration of a service: in this category, the system actively configures itself and adapts its parameters. This can include systems that learn the user's preferences and evolve their adaptations at runtime. The category also includes personalization systems that automatically learn and modify personalization rules.
- Passive configuration of a service: systems that modify their behavior based on context at runtime and propose possible modifications to the user.
- Tagging context to information: this category is adopted from Dey and Abowd [2000].

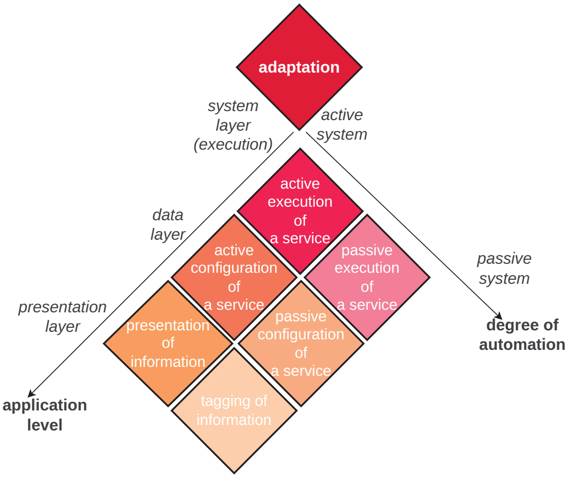


Figure 2.4 — Options for adaptation in context-aware systems after Alegre et al. [2016].

Apart from the different degree of automation, these categories can be described with respect to the application level on which the adaptation takes place. The execution of a service happens on a system layer, whereas the configuration of a service concerns the data layer (manipulating parameters). The presentation of information, but also the tagging of information leading to augmented data or reality is implemented on the presentation layer. Figure 2.4 shows all options of adaptation in a context-aware system categorized along the two dimensions: the degree of automation and the application level of the adapation. The goal of the approach developed in this work is to achieve active execution of a service. A passenger information system should proactively choose device and modality to provide information for the user.

2.5.3 When to Adapt in a Context-Aware System

The question when an adaptation is decided in a context-aware system is closely related to design questions about the context model and storage of context as well as the overall architecture of the system. There are two main possibilities: adaptations can be triggered by context changes or by system actions. Knappmeyer et al. [2013] describe these options in degrees of context-awareness . They categorize synchronous context distribution on demand by applications as "*context-based adaptation*" and asynchronous and event-based context diffusion that requires the identification of events of interest as "*context-aware adaptation*". The authors describe "*situation-aware adaptation* as the highest degree of context-awareness. It uses asynchronous context distribution as well, but for situations that are inferred as higher level context. Apart from the context abstraction levels that differ in this categorization, the first two categories have different answers to the question when to adapt. Either the applications request context data and therefore initialize adaptation or a context management component does.

In the first category of synchronous, or request-based adaptation, applications or the application layer requests context from the context management component or layer and initializes adaptation based on the requested context. The timing of adaptation is set by the application.

The second category involves the constant monitoring of context data and a mechanism that decides if a change in context is meaningful. This approach is usually implemented event-based. Certain context changes fire events that then can be processed by the responsible components. A system that consistently implements this approach was, for example, presented by Ricquebourg et al. [2006]. They developed a context-aware system that uses an Open Services Gateway Initiative (OSGi) event bus for communication between sensors, context handling services and services on the application layer, for example. Sensors generate an event if new context data was sensed and an ontology service handles context processing and inference. If an action is necessary based on the context, this service triggers an event for actuators.

Which schema is applicable in a context-aware system is determined by the use case. Often, both approaches are implemented to support a variety of use cases.

2.5.4 How to Adapt in a Context-Aware System

How a context-aware systems implements and executes adaptation is a question with a variety of answers. The question itself can be divided into two parts: first, the system needs to *decide* which kind of adaptation should be executed. The second part is concerned with the implementation details of the adaptation. Adaptations based on context take place at runtime and therefore must be implemented as an application flexibility at runtime. Many architecture approaches and programming paradigms have been discussed and implemented to achieve this kind of flexibility.

Decision Making

The decision making part of context-awareness has not been discussed as extensively as context modeling or context dissemination. Sometimes, adaptation decisions are covered with context reasoning techniques, for example by Bikakis et al. [2008] or Bettini et al. [2010]. Rule-based approaches are often used as variants of Event-Condition-Action (ECA) rules that have a simple if-then logic. The system presented by Daniele et al. [2007] is an example that provides context-awareness using ECA rules, as well as the context-aware notification service by Etter et al. [2006]. Context-aware adaptation of business processes using ECA rules was, for example, proposed by Mejia Bernal et al. [2010]. Semantic rules that are used with ontologies are more complex and are, for example, used by Ricquebourg et al. [2006], Barbosa and Andrade [2009] and Skillen et al. [2013]. These approaches use SWRL rules together with OWL ontologies. Case-based reasoning can be applied for decision making, especially because the approach uses problem descriptions with solution templates, which persists possible and successful actions during runtime [Kofod-Petersen and Mikalsen, 2005].

Logic-based approaches are also common for decision making. Ranganathan and Campbell [2003] used first order logic for context modelling in the *Gaia* system. Logical expressions are used to define application logic in specific configuration files. The description logic that is used in OWL-DL provides reasoning capabilities for ontologies, which is sometimes also used to determine application behavior. Naganuma and Kurakake [2005] presented a system, for example, that supports a user by recognizing their task and selecting appropriate services in the mobility domain. They used an OWL-S ontology to model possible tasks and processes.

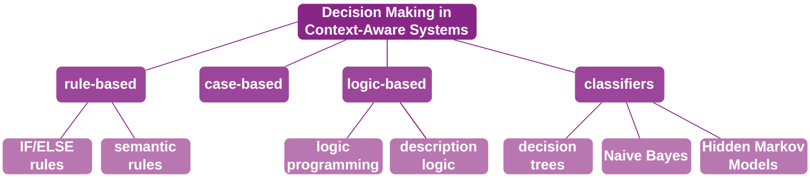


Figure 2.5 — Techniques commonly used for decision making in context-aware systems.

However, there are also approaches that use other techniques to decide about adaptations. Lim and Dey [2010] have reviewed several decision models for context-awareness. They also found rule-based approaches, but additionally approaches that use decision tree classifiers, Naïve Bayes classifiers and Hidden Markov Models for decision making. Figure 2.5 gives an overview over these and the above-mentioned approaches.

Realization of adaptation

Depending on the type of adaptation, different implementation methods are applicable. In the categorization shown in figure 2.4, adaptations are categorized in those at presentation level, the data level and the system layer.

Adaptations at presentation level can change or enhance the information that is displayed but they can also influence the whole user interface. An example for a system that generates a user interface based on context information is the SECAS project, presented by Chaari et al. [2008]. It is capable also of content adaptation and services adaptation, but the user interface generation is a feature that stands out. The system is able to construct a graphical user interface using a XML-based language. The context-aware tourist guide GUIDE by Cheverst et al. [2000] is an example for a system that uses context to select the information that is shown to the user. The authors implemented a web-based interface that enabled the system to adapt based on customized Hypertext Markup Language (HTML) tags and tailored HTML sites.

The Active Badge System is an example of adaptation on the data level, implementing the active configuration of a service, where the calls for a person in the office is automatically routed to the phone nearest to them [Want et al., 1992].

In case that services are executed, the adaptation takes place on a system level. Several programming paradigms and architectural approaches are used to achieve the needed runtime flexibility. Many approaches implement a service-oriented approach for context-awareness and use dynamic service binding to execute contextually appropriate services. The adaptation management framework by Attou and Moessner [2007] is an example for a service-based context-aware system. Based on context, a Content Adaptor component identifies necessary adaptations and suitable services that can be invoked to execute these adaptations.

Agent-based approaches use software agents to achieve loose coupling of components and execution flexiblity at runtime. Chen et al. [2004b] have presented a system that uses the CoBrA system and the Vigil ubiquitous computing environment [Chen et al., 2004a; Undercoffer et al., 2003]. The system uses intelligent agents and services to realize a smart meeting room. The MAPIS system presented by Petit-Rozé and Grislin-Le Strugeon [2006] handles personalization in an agent-based information system. Intelligent agents are used to handle proactive personalization of information for users.

Other approaches use model-driven development to implement adaptation in a modular and component-based system. Hallsteinsen et al. [2012] presented their MUSIC framework that supports the model-driven development of adaptive applications.

There are more techniques and paradigms that can be applied to achieve adaptivity on a system level. Many approaches leverage the advantages of more than one method in hyprid approaches. Alegre et al. [2016] have discussed programming paradigms that have been used for the development of context-aware adaptive applications in greater detail.

CHAPTER 3

Requirements Analysis

This analysis comprises several different parts that consider the problem of adaptive information provision in mobility systems from several perspectives. The goals of this analysis are to elicit requirements towards an approach for adaptive and usable information provision and to provide insight into the user's tasks and possible contexts that are relevant for the usability assessment of the approach. As such, the results of this analysis are a basis for the modeling step described in chapter 6. The resulting requirements are used in the following chapter 4 to develop a scheme for the comparison of existing approaches and to identify gaps in research.

This analysis looks at adaptive information provision in mobility systems from a system perspective and from a user perspective. Existing components and architecture properties in mobility systems influence the applicability of adaptation approaches, which is why the system perspective includes an analysis of existing system environments. As described in chapter 1.1, this work focuses on the public transport domain as application domain. Section 3.1 therefore analyzes the characteristics of existing public transport systems, focusing on passenger information systems that are relevant for information provision. An examination of architectural aspects extracted from related work on adaptive user interfaces complements the system perspective in section 3.4.

Section 3.2 and 3.3 comprise the user perspective of this analysis. In section 3.2, situations and tasks in mobility systems are analyzed and structured. On this basis, a persona an scenario approach further evaluates user requirements. At the same time, this user perspective analysis will provide a foundation for the ontologies described in chapter 6.

3.1 Characteristics of Public Transport Systems

Since public transport systems are running 24/7 and evolve over a long period of time, there are several characteristics that must be taken into account when designing extensions for future smart public transport systems. This section will therefore discuss public transport systems in more detail. In the course of this section, I will present devices and modalities that are currently in use for passenger information as well as some that are still subject of research. The analysis of passenger information devices is the foundation for the device ontology which is described in section 6.5. This analysis focuses on german public transport systems, since the developed prototype will use standards and data from german public transport systems.

Public transport systems comprise a number of subsystems that handle planning and operating public transport. They form a system of systems in the sense that complex subsystems work independently from each other and are managed each for their own purpose, as described by Maier [1998]. The combination of the subsystems can then provide additional features. Digital passenger information systems are a part of this system of systems and utilize several other subsystems. As Maier [1998] wrote, communication interfaces are the most important parts that shape a system of systems, as each subsystem may implement different architectural styles, but communication interfaces between them enable their collaboration.

Figure 3.1 shows a generalized overview of typical system components of a german public transport system. In this generality, it is probably applicable to many public transport systems from other European countries, but since these systems have evolved over several decades and often reflect national regulations, there can be substantial differences in their implementations. Each rectangle in figure 3.1 represents several subsystems that have a common task. The figure focuses on systems that are part of passenger information or have an impact on passenger information. A multitude of standards shape the data structures and interfaces of many of these systems, some of them European standards, others national standards. However, each public transport provider has their own implementation of these systems and some of them do not fully adhere to these standards. Therefore, public transport systems are characterized by great heterogeneity of system components, data and interfaces.

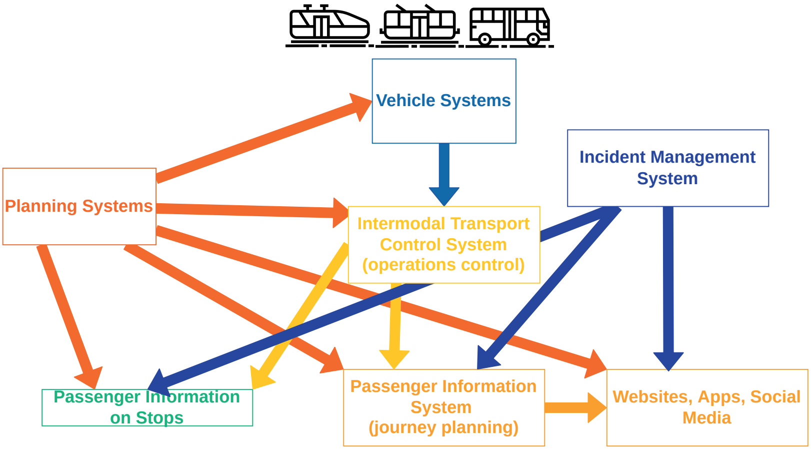


Figure 3.1 — An abstract architecture of involved system components of a public transport system.

Planning Systems: Planning systems are used to plan all aspects of public transport. For passenger information, the timetable information is most relevant.

Intermodal Transport Control System (ITCS): An Intermodal Transport Control System is handling realtime data for a public transport system. It collects vehicle positions and status, disruptions and delays. Realtime data from the ITCS is fed into passenger information systems.

Vehicle Systems: Several system components and, in some cases, separate systems are present in a public transport vehicle. The extent of these systems differs, based on the type of vehicle, but also varies for vehicle manufacturers as well as vehicle providers and operators. In most public transport systems, a system is included in each vehicle that informs the driver of their route, timetable and other information. In current systems, the vehicles determine their own position, using a Global Positioning System (GPS) and odometers, for example. They report their position and additional data to the Intermodal Transport Control System (ITCS). Many vehicles have speakers for passenger information and in most cases, they are operated by the driver. The IP-based standard for communication of in-vehicle systems, VDV 301-2 also allows speakers to be fed from external systems [Weißer et al., 2014]. Some vehicles also are equipped with vehicle information displays that are installed at the vehicle ceiling inside the vehicle. Using VDV 301-2, the content of these displays can also be fed from external systems. There are, however, still many systems in use that do not allow a dynamic change of content on vehicle displays. A new research topic are semi-transparent, interactive displays that replace vehicle windows [Keller et al., 2019b; Titov et al., 2020]. These so-called Smart Windows are multitouch enabled and their content is managed by an external data platform that is integrated in the public transport system. These Smart Windows allow users to either look through them, just like a window or to see and explore passenger information, using multitouch.

Incident Management Systems: If there are incidents, like disruptions they are handled in a control center. An incident management system is used to manage incidents and responses to incidents. Some information about incidents is usually conveyed to passenger information, for example on websites or in apps.

Journey Planning Systems: A central system for passenger information is a journey planning system that can compute suitable journeys for customers, based on timetables, realtime information and given origin and destination. It also provides information about upcoming departures at stops, among other information. The journey planning system often is accessed by customers via websites, mobile applications or social media. Mobile applications are generally used on smartphones, but smartwatch applications are also used [Chow et al., 2016].

Passenger Information on Stops: At stops, there can be different types of passenger information systems. Small stops, wich are very common in german public transport, especially in areas outside of inner cities, very often have no digital passenger information systems. However, digital displays for passenger information are becoming more common. An example is shown in figure 3.2. In cities, digital displays are established at many public transport stops. In their simpler form, these displays are not interactive, as the example in figure 3.2. Interactive displays, so-called public displays, are subject of current research. The deployment of interactive displays was, for example, evaluated by Mayas et al. [2018]. Public displays currently are not used for personalized passenger information, however research on this subject exists [Keller et al., 2019a]. As with Smart Windows, speakers on public displays for localized announcements are conceivable. At larger stops, speakers on platforms are currently widely in use for general passenger information.

Figure 3.2 — A digital passenger information display at a bus stop in the small township Schwieberdingen. Schwieberdingen is rural community near the capital city Stuttgart of Baden-Württemberg, Germany. It had 11,576 residents on 31th december 2021 (source: `https://www.schwieberdingen.de/`, last accessed October 8th, 2022). Its public transport network consists of several bus lines as well as one suburban railway line.

The digitization of public transport progressed over the course of several decades and is still progressing. Existing systems and infrastructure were extended over time by additional and newer systems, vehicles and technologies. Old technology is replaced only slowly and therefore, current public transport systems are very heterogeneous systems with older parts that sometimes use outdated technology and new extensions that use current technology. An example is the vehicle fleet of a public transport provider. Public transport vehicles are used for 30 and up to 40 years [VDV, 2019]. Public transport in Germany is a state affair, meaning cities, counties or federal states commission the implementation of public transport to transportation companies or transportation associations and they render co-operative services. Transportation companies receive public compensation. Purchases of new vehicles, for example, are therefore subject of tendering. These processes are time-consuming. Additionally, after an order is placed, vehicles will have to be produced. Years go by from the initial decision to buy new vehicles to their actual delivery [Barros, 2019]. This means that vehicles that are in use have several different levels of technological equipment, sometimes within one vehicle, due to later modernization. The organization of transportation companies in Germany, Verband Deutscher Verkehrsunternehmen (VDV), issued an IP-based standard for communication of in-vehicle systems in 2014, which was developed in the research project "IP-KOM-ÖV"[1] [Weißer et al., 2014]. Before this standard was issued, vehicle systems used either proprietary solutions or the older standard, VDV-300, which utilized transmission speeds of 0.0012 MBit/s between vehicle system components [Wehrmann et al., 2012; VDV, 1984]. These limit their application for passenger information and the types of content that are possible to display. Due to the longevity of public transport vehicles and their continuous operation in an public transport company, vehilces using outdated equipment can not be taken out of service and be replaced easily. Therefore, the fleet of a public transport company in Germany usually uses very heterogeneous technology. Expanding and extending the public transport system means that this heterogeneous technology levels must be considered in order to keep the fleet operating. The same is also true for other systems, such as passenger information on stops, for example. An approach that introduces adaptive passenger information therefore must be flexible enough to handle heterogeneous data and heterogeneous passenger information devices. The devices themselves and their user interfaces cannot be developed and implemented from scratch. Such an approach must integrate different existing devices and their existing user interfaces and it should not require major interventions in existing components.

3.2 Situations and Tasks in Ubiquitous Mobility Systems

This section describes the analysis of situations and user tasks in individual mobility. This analysis is the basis for the task context model and the overall mobility context model that are detailed in chapter 6 and contributes to the requirements in section 3.4.

---

[1] "IP-KOM-ÖV: Internet Protokoll basierte Kommunikationsdienste im Öffentlichen Verkehr", `http://ip-kom.net`, last accessed October 8th, 2022

As a first step in the analysis, situations and tasks in individual mobility are identified and structured. In a second step, the user tasks are further analyzed applying a hierarchical task analysis as a foundation for the task model. The hierarchical task analysis is described in section 6.1.2.

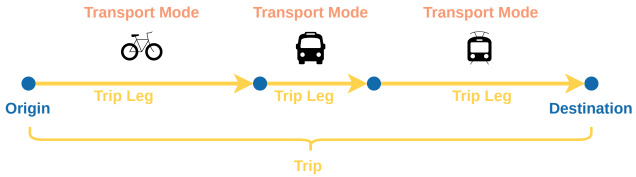


Figure 3.3 — A illustration of terminology

Situations in mobility of individuals can be characterized by the state of a trip the user is in and which transport mode they use. A trip is defined as covering a distance between two points that is not zero. These points are identified as origin and destination of the trip. A trip might have several trip legs and each of those can be covered using a different mode of transport. The mode of transport is defined as the type of movement and possibly vehicle that helps covering that distance. The terminology is illustrated in figure 3.3.

For the task and situation analysis, tasks are distinguished by user activity and transport mode. Transport modes can be differentiated in public transport modes and individual transport modes. The public transport category covers public transport in cities and rural areas as well as long-distance traffic by train or bus. Vehicles are provided by a public transport agency or similar services and mostly operate on a given schedule. The individual transport category includes modes of transport that do not follow a schedule and can be individually navigated.

The task analysis was performed involving most transport modes. Air transport as well as ships were excluded as a means of transport, because of fundamental organizational differences to transport modes on the ground. Autonomous vehicles were also excluded.

Camacho et al. [2013] define *pre trip* and *on board* as phases of a trip, distinguishing trip planning and activities before boarding a vehicle from activities on board a vehicle. Hörold et al. [2013] describe a *journey chain* or *travel chain* specifically for public transport, that distinguishes the following phases: *preparation for the journey*, *starting the journey*, *waiting for the vehicle*, *entering the vehicle*, *travel with the vehicle*, *transfer to another vehicle*, *alighting from vehicle* and *heading towards destination*.

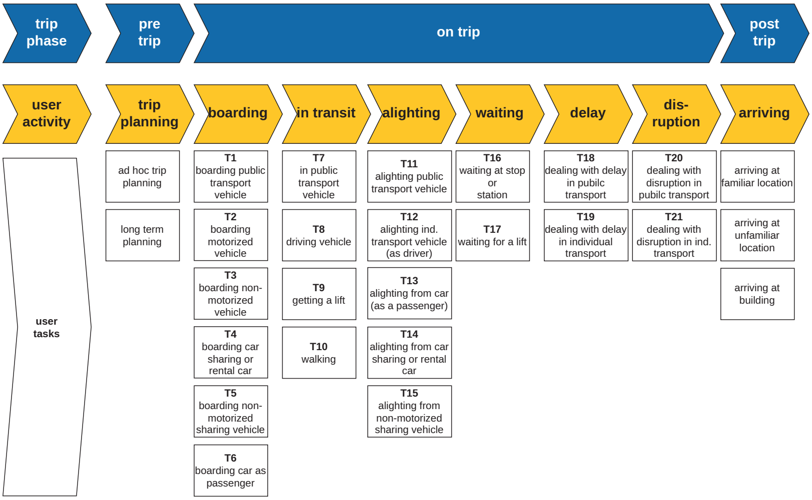


Figure 3.4 — Phases of a trip, user activities and transport mode specific user tasks

Taking the concepts of Camacho et al. [2013] and Hörold et al. [2013] into account, I define *pre trip*, *on trip* and *post trip* as the three main phases of a journey. In each of these phases user activities are defined that can be part of an individual travel chain. For each activity, there are several different user tasks that a user can perform. The specific tasks differ based on the transport mode that is used. Figure 3.4 shows the phases of a journey, the activities and specific tasks. The activities are:

- **Trip planning:** Before starting a trip, a user plans it. Trip planning can be an ad hoc activity or take place ahead of the trip. Trip planning may also be needed during a trip, when rescheduling is necessary. The result of trip planning is an itinerary that includes all necessary information for the trip. An itinerary may be implicitly determined by the user and therefore only existing in their mind, but can also exist on paper, in a digital form or in a combination of these.
- **Boarding:** As long as a user uses any kind of vehicle as a means of transport, they have to board this vehicle somehow. The tasks a user performs while boarding a vehicle are depending on the transport mode. A requirement for boarding is an itinerary that can determine the vehicle

location, time and mode of transport. Boarding tasks begin at the location a vehicle is parked or halted, e.g. parking lots, train stations or stop places and are not concerned with reaching those. If the user needs to reach this location first, a new trip leg is needed in the itinerary. Very often, this is a trip leg walked.

- **In Transit:** The experience of a user in transit differs for different transport modes. Very often, the user monitors their progress on their itinerary.
- **Alighting:** When alighting, a user normally is oriented towards their final destination or, if there is another trip leg to be covered, towards their next vehicle and direction.
- **Waiting:** If the user is not driving the vehicle they use for a trip leg themselves or they are driving and have one or more passengers, they might be waiting at some point.
- **Delay:** Itineraries often have a specific timing and delays often happen. The reasons can be congestions or vehicle malfunctions, delays at boarding and many others. At some points, delays have an impact on arrival times or on connections and a re-planning of the trip might be necessary. Therefore, there are user tasks associated with delays. This is a deviation from the original plan.
- **Disruption:** Disruptions are situations in which a trip leg changes or terminates unexpectedly and the user task can not be completed as planned. If their trip is affected, users will have to reschedule, therefore there are user tasks concerned with handling disruptions. A disruption is also a deviation from the original plan.
- **Arriving:** Post-trip, a user is arriving at their destination. This activity might involve some orientation towards needed factilities, for example hotels or restaurants.

During a journey, a user can engage in several of these activities, not necessarily in all of them. Some of them might repeat. For each activity, a specific user task is performed. These tasks are further analyzed as a basis for the task ontology in section 6.1.2 of chapter 6. The user activities and tasks are used to structure the persona and scenario analysis and they show core properties of mobility tasks that are very relevant for the requirements determined in section 3.4.

3.3 Persona and Scenario Approach

The persona method aims at analyzing and describing the target audience of a prospective software system using stereotypical persona that represent important characteristics of the target audience. Based on data collection and analysis, central charactaristics and user needs are worked out [Nielsen, 2019]. Personas are then identified and described. The persona analysis is commonly used to elicit core user requirements [Nielsen, 2002]. In this work, I also used it to identify user characteristics and context facts for a user context model. Persona are often used as a starting point for scenario-based design [Carroll, 2000].

Scenarios describe the usage of a possible software in an illustrative manner and are used as a way to elicit and describe user requirements as well as to support the design and development of interactive software [Sutcliffe et al., 1998; Rosson and Carroll, 2002]. Scenarios describe goals and needs of users, the interaction between user and system and the usage context, which is particularly important in ubiquitous systems. Scenarios can be used to explore relevant context facts that influence system adaptation, since they involve a rich description of system usage.

Based on the persona descriptions, I developed scenarios for context-aware information provision in ubiquitous mobility systems. The scenarios illustrate possible smart ubiquitous mobility systems and support the user perspective for requirements elicitation. The scenarios were also used to analyze context facts of mobility situations as a basis for further analysis for the context model in section 6.1.

The following analysis of the target audience of ubiquitous mobility systems is based on persona analyses from three research projects in public transport that I contributed to. The basis persona were researched and developed in the research project “IP-KOM-ÖV”[2], funded by the German Federal Ministry for Economic Affairs and Energy. The key charactaristics used to describe these personas for public transport, were local knowledge, system knowledge (of the public transport network), type of ticket, preferences and special needs [Radermacher et al., 2012]. In following research projects, the personas were developed further. A persona using a wheelchair and several persona characteristics were added as well as some charactaristics. These research projects were “DynAPsys”[3], also funded by the German Federal Ministry for Economic Affairs and Energy and “SmartMMI”[4], funded by the German Federal Ministry of Transport and Digital Infrastructure.

---

[2] “IP-KOM-ÖV: Internet Protokoll basierte Kommunikationsdienste im Öffentlichen Verkehr”, `http://ip-kom.net`, last accessed October 8th, 2022

[3] “DynAPsys: Dynamisches Agendaplanungssystem”

[4] “SmartMMI: Modell- und kontextbasierte Mobilitätsinformation auf Smart public displays und Mobilgeräten im Öffentlichen Verkehr”, `http://smartmmi.de/`, last accessed October 8th, 2022

Eight personas were the result of our work on these research projects. I extended and modified these persona descriptions in order to describe the target audience of ubiquitous mobility systems. The results are shown in figures 3.5, 3.6, 3.7 and 3.8 and will be discussed briefly in the following.

	Michael Baumann	Maria Ziegler
Age	34	22
Profession	management consultant	college student
Local Knowledge	***	****
System Knowledge in Public Transport	***	****
Technical Affinity	****	***
Car Sharing Membership	yes	no
Bike Sharing Membership	yes	yes
Own Car	yes	yes
Own Bike	yes	no
Smartphone	yes	yes
Smartwatch	yes	no
Uses Headphones with Smartphone	yes	yes
Special Needs	none	none
Interaction Abilities	visual, haptic, acoustic	visual, haptic, acoustic
Interaction Preferences	acoustic › visual › haptic	visual › acoustic › haptic
Transport Mode Preferences	pt › bike › walking › bike sharing › car › car sharing	pt › bike › bike sharing › walking › car › car sharing

Figure 3.5 — Central characteristics of two of the personas.

The local knowledge attribute describes how well a person knows their local area. The system knowledge of public transport indicates how familiar a user is with the local public transport system. Together, these charactaristics affect how much information these users need to understand their itinerary or messages regarding their mobility. The technical affinity of a person characterizes their familiarity with new technologies. It also indicates if a person is inclined to use new technologies, for example an interactive public display at a bus stop. Their membership with bike and car sharing agencies is listed, as well as their ownership of a car or a bike. Their usage of a smartphone and smartwatch as well as their habit of using headphones with their smartphone, for example to listen to music or podcasts, give an indication of devices and interaction modalities that can be used.

Special needs are noted when they affect the person's mobility or interaction abilities. The persona Martina Grundler, for example, is often on her way with her little children and uses a stroller. She is therefore not able to use a steep flight of stairs. Christian Peters is using a wheelchair and can not use stairs or escalators at all. He also does not use bikes or car sharing cars. Hildegard Krause is walking slowly and easily looses her balance, she therefore needs a

	Martina Grundler	Kevin Schubert
Age	42	15
Profession	housewife and mother	pupil
Local Knowledge	***	**
System Knowledge in Public Transport	***	***
Technical Affinity	**	****
Car Sharing Membership	no	no
Bike Sharing Membership	no	yes
Own Car	yes	no
Own Bike	no	yes
Smartphone	yes	yes
Smartwatch	no	no
Uses Headphones with Smartphone	no	yes
Special Needs	stroller and small child	none
Interaction Abilities	visual, haptic, acoustic	visual, haptic, acoustic
Interaction Preferences	visual › haptic › acoustic	visual › audio › haptic
Transport Mode Preferences	pt › car › walking › car sharing › bike › bike sharing	bike › pt › bike sharing › walking › car sharing › car

Figure 3.6 — Central characteristics of two of the personas.

seat in buses or trams. Bernd Lorenz wears hearing aids and therefore does not hear announcements on speakers very well.

	Hildegard Krause	Bernd Lorenz
Age	69	51
Profession	pensioner	manager in marketing
Local Knowledge	***	***
System Knowledge in Public Transport	**	*
Technical Affinity	*	**
Car Sharing Membership	no	no
Bike Sharing Membership	no	no
Own Car	yes	yes
Own Bike	no	yes
Smartphone	no	yes
Smartwatch	no	no
Uses Headphones with Smartphone	no	no
Special Needs	walking slowly	hearing aid
Interaction Abilities	visual, acoustic	visual, haptic, acoustic
Interaction Preferences	visual › acoustic	visual › haptic › acoustic
Transport Mode Preferences	car › walking › pt › car sharing › bike › bike sharing	car › pt › walking › bike › car sharing › bike sharing

Figure 3.7 — Central characteristics of two personas.

To capture these charactaristics, I added interaction abilities and preferences to the persona description. Interaction abilities describe if a person is able

	Carla Alvarez	Christian Peters
Age	29	31
Profession	tourist	product designer
Local Knowledge	*	****
System Knowledge in Public Transport	*	****
Technical Affinity	****	****
Car Sharing Membership	no	no
Bike Sharing Membership	no	no
Own Car	no	yes
Own Bike	no	no
Smartphone	yes	yes
Smartwatch	yes	yes
Uses Headphones with Smartphone	yes	yes
Special Needs	none	wheelchair
Interaction Abilities	visual, haptic, acoustic	visual, acoustic
Interaction Preferences	acoustic > visual > haptic	visual > acoustic
Transport Mode Preferences	pt > walking > car > car sharing > bike > bike sharing	car > pt > walking > car sharing > bike > bike sharing

Figure 3.8 — Central characteristics of the last two personas.

to perceive *visual*, *haptic* or *acoustic* output. Interaction preferences describe the persona's preferred interaction modalities. In the figures 3.5, 3.6, 3.7 and 3.8, these preferences are shown in a sequence, where the first is the favourite interaction modality and the last is the least preferred one. The color also highlights preferences from light blue as preferred to dark blue as not preferred. Transport mode preferences are displayed in the same way. Each persona has different transport mode preferences. They are linked to availablity and abilities. The pupil, for example does not have a driver's license and therefore driving a car is not an option.

The persona are chosen so that they differ from each other in several characteristics, but each charactaristic is represented at least once. As an exception, there is no persona that does not have the ability for visual interaction, since systems for blind or visually impaired users differ profoundly from systems that use, among others, visual interaction. Such systems should therefore be considered separately and more carefully. The local knowledge, system knowledge and technical affinity charactaristics are ranging from *no knowledge / affinity* (in the table noted as one star: *) to *high expertise / affinity* (noted as four stars in the table: ****).

Scenarios for the usage of smartphone-based information in public transport were, for example, developed in the IP-KOM-ÖV project, featuring the developed personas [Radermacher et al., 2012]. I developed separate scenarios for the scope of this work and used an approach based on the task analysis in section

3.2. The scenarios are chosen so that every user task is covered and appears in at least one scenario.

Scenario 1 - A Delayed Train: *This weekend, Maria has planned to visit her friend Anna in her home in a nearby town. They want to finish a presentation for their studies and go out on saturday evening. Maria has already planned her trip and chosen a train. She transferred the trip details and her train ticket to her smartphone. Her friend will pick her up at the station with her car. Saturday morning, after breakfast, Maria packs her backpack, while listening to music on her laptop. Her smart mobility manager uses the audio output on this laptop to remind Maria that her train will be leaving at the station in 30 minutes and she will have to leave soon. Maria checks the mobility manager on her smartphone. It presents several options to reach the station, taking her transport mode preferences into account, as shown in figure* 3.9*. There is no direct bus or tram line from her flat to the station, so her first option is to use her own bike or, as a second option, the local bike sharing service. She could also walk, but then she would have to leave in five minutes. Maria needs to finish packing and feed her cat before leaving, so she doesn't want to walk. Since she will stay the night at Anna's house, she also doesn't want to use her own bike and leave it at the station overnight. She chooses the bike sharing service. The next available bike is only 50m away in her street. By choosing this option, her mobility manager books this bike, so that it will be available for her when she needs it. A booking confirmation is given using the audio output on her laptop while she finishes packing and feeds her cat. She then leaves the house and walks towards the bike. She is wearing her headphones and continues to listen to her music on her smartphone. Her mobility manager gives her general directions towards the booked bike over her headphones. Maria can use her smartphone to unlock the bike, using NFC. She then proceeds towards the station. She knows her way around the city and doesn't need directions. At the station, she locks the bike and then checks her smartphone, to see at which platform her train will leave. At the platform she still has some time left and therefore checks her social media timeline on her phone. She is distracted and does not notice the announcement on speakers at the platform, nor the information on the displays that her train will be delayed. Her mobility manager registers the usage of her smartphone and therefore displays a notification about the delay. Maria notices and opens her mobility manager. It shows that the train is 25 minutes late and that she has two options, as also shown in figure* 3.9*. She could take another train and change trains once and arrive probably 15 minutes after her original time of arrival. She could also wait for the delayed train, not change trains and arrive 25 minutes late. She chooses the latter option, stays at the platform and informs her friend that she will arrive later. After 25 minutes, her train arrives. Maria boards the train and searches for a seat. Since she has no seat reservation, she walks through the train until she finds a free seat, next to a Smart Window. During her one hour trip, Maria reads a book and listens to music*

Mobility Manager

At: Home
To: Freiburg, Hbf
Departure: 09:15
Arrival: 10:42

Next Step: Reach Karlsruhe Hbf
Choose Option

Your Bike
7 min
1,7 km

Bike Sharing
7 min
1,7 km
0,75 €

Walk
20 min
1,6 km

Mobility Manager

At: Karlsruhe Hbf
To: Freiburg, Hbf
Departure: 10:03
Arrival: 11:07

Next Step: Reach Freiburg Hbf

Your train is currently 25 minutes late.

Choose Option

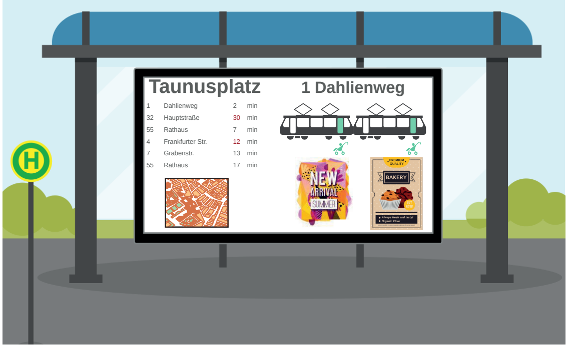
Train
Departure: 10:03
Arrival: 11:07
Karlsruhe Hbf
Freiburg Hbf

Train
Departure: 09:40
Arrival: 10:14
Departure: 10:19
Arrival: 10:47
Karlsruhe Hbf
Offenburg Hbf
Freiburg Hbf

Figure 3.9 — Mockup examples for a mobility manager app, scenario 1. Icons and map graphics by draw.io.

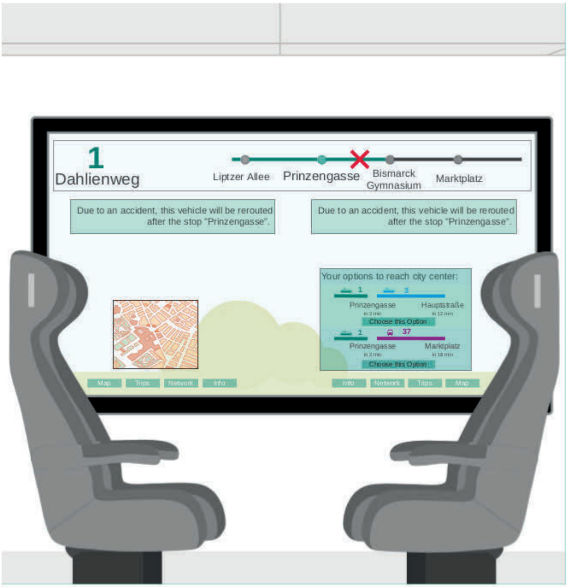
on her smartphone, using her headphones. After 45 minutes, her mobility manager notifies her via audio over her headphones that the train was able to mitigate some of its delay and she will arrive only 15 minutes after her original time of arrival. She sends her new time of arrival to her friend. Five minutes before arrival, Maria stows away her headphones, her smartphone and her book. She looks out of the window and sees the station approaching. Her mobility manager then displays directions towards the parking lot in the back of the station on the Smart Window in front of Maria, because this is where she will meet her friend. After alighting the train, Maria proceeds to the parking lot. She finds her friend waiting for her next to her car. They board the car and her friend drives them to her house.

Scenario 2 - Disrupted Service: *Martina Grundler is a housewife and mother of three children, aged one, three and seven. Martina and her family live on the outskirts of a medium-sized city. Today, she is taking the children downtown for shoe shopping. She takes a look at her mobility manager app for suggestions how to reach the city center. Her app shows her three options: taking her car would take her 30 minutes due to heavy traffic and the app also notes that there are not many parking spaces left, which is why she dismisses this option. She can also use public transport. The mobility manager lists two options, one that takes less time but requires to change twice and one that* *takes longer, but is without change. Both options are accessible with a stroller, because the app saved this as a mobility requirement for Martina. Martina chooses the option without change, because she doesn't mind its length. As her children are getting ready, she needs to change the diapers of her youngest child shortly before they are due to leave. The app notifies her via voice output that she should leave now. Since she does not leave for the tram stop for several minutes, the mobility manager automatically chooses the next tram, which leaves 15 minutes later. Again, Martina is informed via voice output on her smartphone. When Martina and her children leave the house, she checks her app and approves the new itinerary. They walk to the tram stop and wait there until the tram arrives. Martina has put her smartphone in her pocket and is busy watching her children. The mobility manager connects to the public display at the stop and one minute before their tram arrives, the public display shows where the tram will stop and where one can find the doors next to the space designated for strollers or wheelchairs (cf. figure 3.10). Martina notices the display and moves to the correct section of the platform. As the tram arrives, they enter and Martina can park the stroller in the designated area. She finds seats for herself and her children next to the stroller. The kids are fascinated by the Smart Window next to their seats and try all of its functions. After some time, Martina notices an announcement via the tram's speakers. But since the children are fighting over who gets to operate the Smart Window, Martina doesn't get the message. Shortly after, the message appears on the Smart Window: There has been an accident on the track some distance ahead and the service of the tram is disrupted. It will change its route after the next stop. Martina's smartphone has connected to the Smart Window as she sat down next to it. Therefore, the Smart Window now*



1	Dahlienweg	2	min
32	Hauptstraße	30	min
55	Rathaus	7	min
4	Frankfurter Str.	12	min
7	Grabenstr.	13	min
55	Rathaus	17	min

Figure 3.10 — Mockup examples for mobility information on a Public Display, scenario 2. Background graphic by Nadine Vollers, icons and map graphic by draw.io, posters designed by Macrovector_Official / Freepik.



Due to an accident, this vehicle will be rerouted after the stop "Prinzengasse".

Due to an accident, this vehicle will be rerouted after the stop "Prinzengasse".

Figure 3.11 — Mockup example for mobility information on a Smart Window, scenario 2. Background graphic by Nadine Vollers, icons and map graphic by draw.io

shows several options for Martina to reach her destination, as shown in figure 3.11. She can alight at the next stop and take a bus, or she can change into another tram three stops later. Martina chooses the second option, since traveling by tram is more comfortable for her and her children. She can do that on the Smart Window and the option is transmitted to her smartphone. A short confirmation message on the Smart Window assures her of that. Shortly before they need to change, a notification appears on the Smart Window and announces the next stop and which trams currently can be reached at this stop. Martina notices that they should alight here and they get off the tram. At the stop, her smartphone connects with the public display. The display shows the next departures for the city center and Martina can see that they can simply use the next tram that is arriving shortly after. In the next tram, there is no Smart Window nearby. They take some seats and after four stops, Martina's smartphone vibrates. As she checks her smartphone, she sees a notification to alight at the next stop. All four get off the tram and have arrived at the city center. A local shoe shop is just 20 meters away. Martina is relieved that they arrived easily despite the disruption.

Scenario 3 - Engine Failure: *Today, Bernd has an appointment with a new graphic design studio he wants to hire for his company's next campaign. The studio is in a small, nearby town. After breakfast, he leaves the house and sits in his car. His phone automatically connects to the car's navigation system. His mobility manager app is accessing his calendar and extracts the appointment and address. His navigation systems shows two options of reaching this destination, one using the motorway and one using country roads. By choosing the faster motorway route on the multi-touch display of the navigation system, Bernd also acknowledges his destination. Bernd can't hear well and is using hearing aids. The mobility manager app and the navigation system of his car are adjusted to this. As long as Bernd is alone in his car, audio output is set to a high volume he can hear well, so that in addition to visual navigation, Bernd can use audio navigation support. After some time driving on the motorway, there is suddenly a strange noise coming from his car's motor. The service lights of the car indicate that something is wrong. Bernd exits at the next motorway service area. After a short look at the motor, he decides that he can't fix this by himself. He calls a repair and towing service. They arrive after a short time and the mechanic takes a look at the motor. She tells Bernd that his car has to be towed and repaired in the workshop. Bernd has been searching for options to leave with his mobility manager app. His options are taking a taxi or using a carpooling service. He sees that a driver with his local carpooling service is already on the motorway, not far away from him. The destination of the driver is the next village. There, bernd can take a bus that takes him to his appointment, only 30 minutes late. As this is the fastes option, Bernd chooses it and the driver is notified immediately. As soon as the driver acknowledges, Bernd's smartphone vibrates. He*

Figure 3.12 — Mockup example for a public display, scenario 3. Background graphic by Nadine Vollers, icons by draw.io, background vector of map created by freepik, posters designed by Macrovector_Official / Freepik.

looks at it and sees the confirmation. He then waits at the parking space and informs his business partners at the design studio that he will be approximately 30 minutes late. The driver arrives only ten minutes later. Bernd greets him and gets into the car. The driver directly drives him to the next village, where Bernd gets out of the car at the nearest bus station. He checks his smartphone and his mobility manager app shows him, that he needs to cross to the other side of the street for the correct bus stop, where the next bus will arrive in five minutes. In the bus, Bernd checks his app several times to not miss the stop where he has to exit. To his relief, his smartphone vibrates to notify him that he should exit at the next stop. Once arrived, he sees a public display at the stop. His smartphone connects to the display and once Bernd is standing directly in front of the display, the walking route to the address of the graphic design studio is displayed, as shown in figure 3.12*. Bernd sees that it is only two blocks away and easily reaches his destination. Bernd arrives at the studio and is relieved that he is only around 30 minutes late after his car broke down.*

Scenario 4 - Home from Work: *Michael is at work and finishes his report punctually to leave at his regular time. His smartwatch vibrates and shows a notification that there is a change of plans for his way home, as shown in figure* 3.13*. Normally, Michael uses the same tram every day, but today his mobility manager updated his plan, because of bad weather. A view out of the window shows him that there is a thunderstorm going on outside and rain is pouring down. Michael looks at his smartphone and checks the options the app is showing him. He still could take his regular tram, but he usually changes at a small stop that has no rain shelter. He could use a different connection, but due to a delayed tram, he would be at home 30 minutes later than usual. But this evening, Michael wants to go to handball practice and needs to be at home on time to get his things and then leave towards the gym. He therefore chooses the third option to use a car-sharing car.*

His mobility manager shows him that one is available in the underground parking of his office building and that it is one he can park and leave anywhere in the city, instead of returning it to a specific station. Michael is pleased that the local car sharing company now provides such flexible cars as well and chooses this option. His smartwatch vibrates to confirm the booking of the car. At the entrance of the underground parking, there is a public display that shows the different levels and areas of the parking garage. His smartphone connects to the display as Michael approaches it and the display then indicates where exactly the car is parked, as shown in figure 3.14*.*

Michael walks to the car. He opens it using NFC and his smartwatch. The watch also shows him, that one scratch on the left front door on the car is already registred as damage. He checks the car for any new damages, but finds none besides this scratch. Michael then drives towards home. It is the evening rush hour and after driving some time, Michael is stuck in traffic. The rain has stopped now and the sun came out. The

Figure 3.13 — Mockup example for a smartwatch notification, scenario 4. Device mockup designed by rawpixel.com. Rain icon made by freepik from www.flaticon.com.

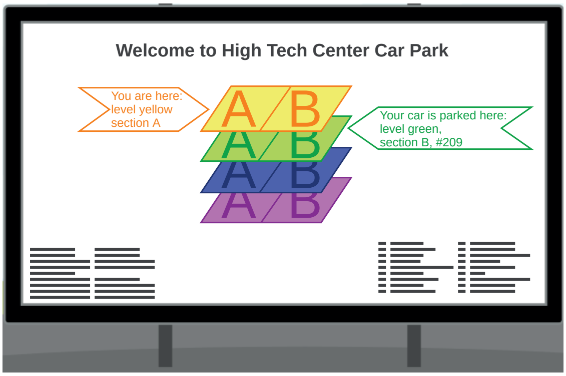


Figure 3.14 — Mockup example for a public display at a parking garage, scenario 4. Background graphic by Nadine Vollers.

mobility manager app uses audio output on the car's system to inform Michael that he will now be at least 25 minutes late due to heavy traffic - apparently, more people than usual have decided to take the car, due to the rain. Michael asks for other options via voice input and the mobility manager gives him two other options. Parking the car at the next car park and taking a tram would still mean a delay of 15 minutes, because he would have to wait for the tram. But next to the car park, there also is a bike sharing station and some bikes are available. Since the rain stopped, Michael is choosing the bike option. He parks the car and locks it. The bike was already reserved by his mobility manager and its lock is opened via NFC and Michael's smartwatch. Michael hurrys *home and arrives ten minutes late. He parks and locks the bike and goes inside to grab his training outfit, shoes and drinking bottle. Normally, Michael would walk to the gym, but since he is still late, he now grabs his own bike and arrives in time to change for his handball training.*

These persona and scenarios show the very broad range of contexts of use for smart ubiquitous mobility systems as well as user properties, abilities and preferences. Depending on a multitude of context factors, different options of information provision are available and only some of them are usable. Using only one device and modality for information would lead to users missing crucial information about their trip in these scenarios. The scenarios also show that information provision in mobility should not only focus on Graphical User Interfaces, but include various modalities. The personas and scenarios inform the requirements that are determined in the next section from a user perspective.

3.4 Requirements for Adaptive and Smart Public Transport

The analysis presented in the previous sections provides a basis to elicit requirements for the approach to adapt output in passenger information systems developed in this work. The requirements determined in this section allow a classification and comparison of approaches towards adaptive user interfaces in chapter 4. Adaptive user interfaces have been discussed since the 1980s and long before ubiquitous computing emerged [Norcio and Stanley, 1989]. Several researchers have proposed structuring this research and have discussed dimensions of adaptive user interfaces to do so, often organized as design spaces [Karagiannidis et al., 1996; McKinley et al., 2004; Motti and Vanderdonckt, 2013; Bouzit et al., 2017]. As Bouzit et al. [2017] write, a design space should serve, among other purposes, to make adaptive user interfaces comparable using the criteria that are defined in the design space. Structures proposed by these researchers therefore serve as a starting point for structuring the requirements in this section.

Karagiannidis et al. [1996] focus on intelligent multimedia presentation systems and present an adaptivity strategy to abstract and structure an adaptation process, in order to make it more re-usable. They propose adaptivity as explicitly defined rather than implicitly defined and hard-coded. The authors present four dimensions of adaptivity, each led by a question: *what* to adapt, *when* to adapt, *why* to adapt and *how* to adapt. Motti and Vanderdonckt [2013] describe a unified design space for context-aware adaptive user interfaces as a tool to compare adaptive applications and to inform designers of these applications. Vivian Motti presents this approach in greater detail in her PhD thesis [Motti, 2013]. They assessed the context dimensions of several works on design spaces of adaptive application. After this assessment they split the dimensions and concepts of context-aware adaptive user interfaces and grouped several in an “Context-Aware Reference Framwork” (CARF) and others in their design space, called the “Context-Aware Design Space” (CADS). In their Context-Aware Reference Framework, Motti and Vanderdonckt [2013] list the following important questions: *what* to adapt, *why* to adapt, *how* to adapt, adapt *to what*, *who* adapts, *when* to adapt and *where* to adapt.

Taking these questions as a foundation and considering the results of the analysis presented in this chapter, I determined requirements for adaptive passenger information in smart public transport as described in the following.

3.4.1 *Why* Adapt?

According to Karagiannidis et al. [1996], the question *why* to adapt is determined by *adaptivity goals*. Such goals can be performance criteria, for example. For Motti and Vanderdonckt [2013] the dimension *Why* defines the goals of adaptations, and the authors refer to software qualities, such as usability, as possible goals.

The main goal of this work is to ensure adequate usability of smart ubiquitous mobility systems. In theses systems, the context of use of is changing frequently and is very diverse. The scenarios in section 3.3 illustrate these context changes.

The context of use has a great impact on the system’s usability. In systems where the context of use changes often, adaptive user interfaces can maintain the system’s usability, as was proposed by Thevenin and Coutaz [1999] for plastic user interfaces, for example. In the scenarios in section 3.3, the system chooses different output devices if the passengers have their smartphones in their pockets, for example. The main adaptation goal of an approach towards adaptation in smart ubiquitous mobility systems therefore should be usability, see requirement 1.

Requirement 1: Adaptation Goal Usability (AG)

The main adaptation goal of an approach towards an adaptive smart ubiquitous mobility information system should be usability. An approach should support usability.

3.4.2 *When* to Adapt?

Motti and Vanderdonckt [2013] discuss *when* to adapt as the decision at what stage of the development and operation process the system is able to adapt a user interface. They identify three different timespans for adaptation. At design time, several user interfaces can be designed to match different contexts of use. Another possibility is the adaptation of the user interface at compilation time, often based on generative approaches using user interface models. These two alternatives are frequently used in approaches that support the development of user interfaces for different computing platforms, such as the approaches of Eisenstein and Puerta [2000] and Meskens et al. [2008], for example. The third category of systems adapts at runtime. The adaptation at runtime is often motivated for ubiquitous and pervasive systems, where the need for adaptation may arise unexpectedly, for example due a change of the context of use or because of the failure of a service or the detection of a new service. In these circumstances, adaptation can not be forseen in all aspects and planned for at design time [Khan, 2010; Schwartze et al., 2010].

The scenarios in section 3.3 show the multitude of situations and context factors that arise for users of mobility systems. These situations and the adaptation of the system to these situations can not be forseen at design or compilation time. The adaptation to new contexts of use needs to be dynamic and performed at runtime, as the following requirement expresses.

Requirement 2: Adaptation at Runtime (AR)

An approach towards adaptive smart ubiquitous mobility systems must support dynamic adaptation at runtime.

3.4.3 *What* to Adapt?

Karagiannidis et al. [1996] discuss *adaptivity constituents* as the subjects of adaptation. The authors note differences in semantic, syntactic and lexical aspects of interaction, where semantic aspects refer to content, syntactic aspects are interaction sequences and lexical aspects are interaction techniques or interface objects. According to this structuring, adaptation can either adapt content, interaction sequences or interaction techniques and interface objects. Motti and Vanderdonckt [2013] mention the navigational flow of an application, its presentation or its content as possible subjects of adaptation.

Both definitions of Karagiannidis et al. [1996] and Motti and Vanderdonckt [2013] focus predominantly on GUIs. However, as the scenarios in section 3.3 illustrate, the situations in mobility systems do not necessarily call for adaptations of GUIs. Often, the context of use in mobility systems prevents the user of perceving information on a given device and GUI, for example when a user has no access to their smartphone at the moment they need to see urgent information. At the same time, a multitude of devices and modalities is available in mobility situations, in public transport as well as in cars, for example. Other interfaces than GUIs are also affected by changes in the context of use, for example if a noisy environment prevents passengers from perceiving announcements in a vehicle, as illustrated in scenario 3.3.

Adaptations of GUIs such as changes in the navigational flow, for example, are therefore not the most suitable instrument to adapt to changes in the context of use in mobility systems. The selection of the used output device and output modality is much more relevant with regard to the types of context encountered in mobility systems, leading to requirements 3 and 4.

Requirement 3: Adaptation of Device Choice (DC)

An approach towards adaptation in smart and ubiquitous mobility should support the adaptation of the output device.

Requirement 4: Adaptation of Modality Choice (MC)

An approach towards adaptation in smart and ubiquitous mobility should support the adaptation of the output modality used.

The choice of output devices should be differentiated from the adaptation to a device type which is a common use case for adaptive GUIs, using user interface description language for device-independent GUI development. Chmielewski et al. [2016] review and compare eight different user interface description languages for device-independent GUIs, for example. However, in these cases, adaptivity aims at the ability of a system to deploy a user interface to multiple devices, where in this work, adaptivity aims at choosing a device for information provision suitable for a situation. Similarly, the support of multiple modalities and adaptation of modality choice in this work does not aim at multimodal user interfaces that support various modalities, but at enabling a system to choose a modality based on information about the context of use.

3.4.4 Adapt *To What*?

Motti and Vanderdonckt [2013] identify the categories user, platform and environment to which a system may adapt, answering the question *To What?*. Karagiannidis et al. [1996] discuss this topic differently. They write that *adaptivity determinants* are the factors that are assessed to then decide about adaptations and they discuss these using the question *when* to adapt. In both cases, the factors that justify adaptation are described as an important dimension of adaptation.

Some systems use only specific factors to adapt to, for example the target device of a user interface [Nylander and Bylund, 2003]. Others support a range of context factors that lead to adaptation [Hussain et al., 2018]. As discussed before and illustrated in the usage scenarios, context is very diverse in mobility systems, which is why a broad range of factors should be supported.

Requirement 5: Context-Awareness (CA)

An approach towards adaptivity in smart ubiquitous mobility systems should be context-aware and support a rich variety of context factors to inform adaptation.

3.4.5 *Who* Adapts?

The dimension *Who* refers to the trigger of an adaptation and Motti and Vanderdonckt [2013] name the end user, the system and third parties as possible values. Considering information provision in smart ubiquitous mobility systems, adaptation is needed to sustain usability during changes of the context of use. In instances when the user seeks information, the user themselves most likely adapt to the context of use by choosing a device and modality that suits their needs. In mobility systems, the system often needs to proactively provide information to the user, which is a central feature that is illustrated in the scenarios in section 3.3. In this case, the adaptation must be triggered by the system that needs to inform the user.

Requirement 6: System Triggered Adaptations (STA)

The smart ubiquitous mobility system needs to be able to trigger adaptation.

3.4.6 *How* to Adapt?

The question *how* to adapt sheds light on the techniques and methods that are implemented to realize adaptation. Motti and Vanderdonckt [2013] describe adaptation techniques and their implementation in detail, for example the adaptation of font size or the change of color balance in images. They structure these techniques in three categories of methods for adapting the navigation of the application, adapting its content or its presentation. As argued before, the focus on Graphical User Interfaces is too narrow for mobility systems and these methods are not applicable for the focus on device and modality choice in this work.

Karagiannidis et al. [1996] consider adaptivity rules as adaptation technique. In answering the question *how* to adapt, they detail that adaptivity rules assign adaptivity constituents to adaptivity determinants. This means the question *how* adaptation is executed is answered in specifying *what* is adapted *to what*, expressed in rules.

A closer examination of adaptation rules reveals that there are several different approaches towards formalizing and implementing adaptation rules. Geihs et al. [2009a] list three types of approaches towards adaptation mechanisms, roughly distinguished by their abstraction level. The first category they name are *situation-action approaches* that have the lowest abstraction levels. Rules specify the exact actions that should be taken in a given situation [Geihs et al., 2009a]. Not only the actions, but also the situations must be explicitly described in great detail. This leads to very verbose rule collections. Considering the diversity of context factors and situations in mobility systems, such an extensive definition of rules is neither feasible nor maintainable. A higher abstraction level is achieved by *goal-oriented* approaches that set a specified goal and are implemented using a system component that computes which actions should be taken to reach that goal. The goal of adaptation in smart ubiquitous mobility systems has been expressed as increasing usability.

The authors mention that goals in such an approach can be in conflicts that can not be resolved by the system. As Geihs et al. [2009a] discuss, the category of *utility-based* approaches further extends such goal-based approaches by introducing utility functions that formalize certain values contributing to a goal. Such utility functions then are computed for each adaptation option and used to compare them and then decide. However, the utility functions the authors describe for their approach are still on a very explicit and detailed level. These utility functions compose several concrete properties that are put in relation to each other to compute a utility score. An example is a function adding one point to the utility if the user has free hands and the application supports hands free interaction and adding another point if the response time of the user is higher than the response the application can sustain based on bandwith [Geihs et al., 2009a]. This abstraction level of adaptation rule expression is still hard to maintain and to adapt to different types of system. Since in public transport systems, each system is very different, this type of adaptation rules is still too complex and specific. Adaptation rules should therefore have a high abstraction level to be easy to modify and to allow the support of the large variety of situations and adaptations needed in these situations without requiring to explicitly specify situations and adaptations in great detail. This is expressed as the following requirement.

Requirement 7: High Abstraction Level of Adaptation Rules (HAL)

Adaptation rules for the formalization of adaptations based on context factors must be able to cover a large variety of situations and adaptations. They must also be easily extendable. Adaptation rules must therefore be expressed using a high abstraction level.

In order to enable such adaptation rules, the overall goal of maintaining usability during changes of the context of use should be broken down into more specific goals that formalized adaptation rules can target. In order to do that, usability criteria that shape the usability of the system should be expressed and referenced in adaptation rules. In many approaches on adaptive interfaces, usability criteria are only formalized implicitly in very specific adaptation rules, for example in the work of Hussain et al. [2018]. An example rule in this work switches the GUI to night mode when specific lux values are reached. This rule refers to visibility. It implicitly defines low visibility based on surrounding light. The work of Akiki et al. [2016], for example, implements a rule that switches a Combo-Box to a Radio-Group, if the Combo-Box has less than three items. The rule implicitly formalizes that for three or less items, a Combo-Box is less usable than a Radio-Group. Such rules are very specific and in a ubiquitous mobility system a great number of rules would be necessary to account for a great variety of devices, modalities and contexts. The following requirement reflects the need for explicitly modeled usability criteria that help formalize highly abstract adaptation rules.

Requirement 8: Explicit Modeling of Usability Criteria (EM)

Usability criteria should be explicitly modeled so that they can be referenced in highly abstract adaptation rules.

3.4.7 *Where* to Adapt?

Finally, Motti and Vanderdonckt [2013] ask *where* adaptation should take place. With this question, they refer to the location of adaptation in the architecture of a system. The authors mention client, proxy and server as possible values. This question calls for a closer look to architectural requirements towards adaptation in smart ubiquitous mobility systems. As discussed in section 3.1, the mobility domain and specifically the public transport domain have unique characteristics that influence the applicability of adaptivity approaches.

Different applications and user interfaces for passenger information already exist and many rely on legacy technology. As described before, these systems can not be replaced or terminated easily. An approach that introduces adaptivity to a public transport information system should support these legacy systems and should not result in replacement and reimplementation of system parts, which is expressed in the following requirement.

Requirement 9: Support for Legacy Systems (SLS)

An approach towards adaptive passenger information must be able to utilize and integrate legacy systems.

As argued in section 3.1, public transport systems function as a system of systems. In such systems, each subsystem operates independently and collaboration works based on communication interfaces. In order to integrate adaptivity in such a system of systems without reengineering several subsystems, adaptation itself should be regarded as a subsystem and operate autonomously from other system parts, also indicating adaptation on a proxy or server rather than on client-side. Additionally, adaptation in the clients is difficult to realize because most of the client applications already exist and reimplementation would be very costly. An adaptation component should be an autonomous subsystem or component, as requirement 10 describes.

Requirement 10: Autonomous Usability Assessment (AUA)

The usability assessment of adaptations must be realized in an autonomous adaptation component.

In the public transport system of systems, each subsystem has independent tasks and works independently. If a new subsystem or component is added or components are altered, these changes should not or only slightly affect all other subsystems. Therefore, an adaptation component should be loosely coupled, only using communication interfaces to interact with other system components, as expressed in the following requirement.

Requirement 11: Loose Coupling (LC)

An approach extending public transport systems must be loosely coupled.

An approach is highly prescriptive, if it strictly prescribes the architecture, data model or other implementation details of many or all system parts. Since in public transport, many subsystems are already existing and are running around the clock, they can not be easily reengineered according to a new, highly prescriptive approach towards adaptivity, for example. The following requirement expresses the need for low prescriptiveness.

Requirement 12: Low Prescriptiveness (LP)

A newly introduced approach to a public transport system must have a low prescriptiveness and not require extensive changes in existing parts of the system.

Public transport systems show a great variety. They differ in the types and amount of system components that are used, but also in the types of services that they offer and in their size. Passengers in rural areas use public transport differently than in urban areas. An approach towards adaptive passenger information should therefore be extendable and configurable for different public transport systems. Extensibility in this case is specifically relevant concerning the context model, adaptation rules and user interfaces, resulting in requirements 13,14 and 15.

Requirement 13: Extensibility of Context (EC)

Context must be represented and processed in a way that is extensible and customizable for specific public transport systems.

Requirement 14: Extensibility of Adaptation Rules (EAR)

The adaptation rules that shape the adaptation must be extendable to allow for adjustment and fine-tuning of the adaptation behavior suitable to the public transport usage and priorities of passengers in different public transport systems.

Requirement 15: Extensibility of User Interfaces (EUI)

An approach towards adaptive passenger information systems must allow the easy integration of different types of user interfaces.

A last architectural charactaristic of smart ubiquitous mobility systems is that the device configuration of the system a user uses in a given situation changes very often. As the scenarios in section 3.3 illustrate, devices are either personal devices, traveling with the user, or public devices, which can be stationary or mobile. At public transport stops, stationary displays can be used for passenger information, where in vehicles, displays or speakers are mobile. Since the user is mobile, too, their surroundings change and therefore, available devices change as well. The following requirement specifies that an adaptation approach should support device mobility.

Requirement 16: Support for Device Mobility (DM)

An approach towards adaptive smart ubiquitous mobility should support device mobility and enable the system to cope with devices disappearing or appearing.

The requirements described in this section are summarized in table 3.1. They will serve as a basis to compare and assess the state of the art in adaptive user interfaces and self-adaptive applications. They are also a foundation for the concept developed in this work and for its evaluation.

Table 3.1 — An overview over all requirements.

Nr.	Requirement	Acronym	Short Description
1	Adaptation Goal Usability	(AG)	Usability should be the main adaptation goal.
2	Adaptation at Runtime	(AR)	Dynamic adaptation at runtime must be supported.
3	Adaptation of Device Choice	(DC)	Adaptation of the output device should be supported.
4	Adaptation of Modality Choice	(MC)	Adaptation of the output modality used should be supported.
5	Context-Awareness	(CA)	Aadaptation should be context-aware.
6	System Triggered Adaptation	(STA)	The system needs to be able to trigger adaptation.
7	High Abstraction Level of Adaptation Rules	(HAL)	Adaptation rules must be expressed using a high abstraction level.
8	Explicit Modeling of Usability Criteria	(EM)	Usability criteria should be explicitly modeled.
9	Support for Legacy Systems	(SLS)	Legacy systems must be integratable.
10	Autonomous Usability Assessment	(AUA)	The usability assessment of adaptations must be an autonomous component.
11	Loose Coupling	(LC)	An approach extending public transport systems must be loosely coupled.
12	Low Prescriptiveness	(LP)	An approach must have a low prescriptiveness.
13	Extensibility of Context	(EC)	Context must be represented and processed in an extensible way.
14	Extensibility of Adaptation Rules	(EAR)	The adaptation rules that shape the adaptation must be extendable.
15	Extensibility of User Interfaces	(EUI)	Different types of user interfaces must be integratable.
16	Support for Device Mobility	(DM)	The system must be able to cope with devices disappearing or appearing.

Chapter 4

Related Work

This chapter presents the state of the art for adaptive and context-aware user interfaces and review related work with regard to the requirements described in section 3.4.

Adaptive user interfaces require flexibility at a certain point in their implementation. That flexibility can be approached from different angles. Two main viewpoints are considering flexibility from an architectural point of view or as a property of the user interface itself. Related works will be discussed following these perspectives. Research from an architectural perspective is discussed in section 4.1. The perspective focusing on user interfaces and user interface development is presented in section 4.2.

Several of the described approaches mix methods and can be viewed from both perspectives. They are categorized according to their main aspect. In tables 4.1 and 4.2, all presented approaches are listed and their compliance with the requirements from section 3.4 is depicted using symbols. An overview of all requirements as a reference can be found in table 3.1. The requirements can either be fully met, shown by a ●, partly met, represented by a ◐ or not met, shown by a ○. Not all works address all requirements. In some cases, a requirement is not applicable, because implementation details are not defined, for example. In other cases, the details necessary to assess a requirement are not described in the publications about an approach. In these instances, a - indicates that a requirement is not specified for this approach. In section 4.3, the properties of the presented works are discussed and the identified research gaps are summarized. Research gaps identified in this inspection lead to the proposal of the concept described in chapter 5.

4.1 Approaches Focusing on Architecture

Pervasive and ubiquitous computing and specifically context-aware computing deal with flexibility of system components and adaptation from the beginning [Weiser, 1991; Want et al., 1992; Schilit and Theimer, 1994]. Researchers have developed techniques and methods to distribute computing to various heterogeneous devices and adapt to the user's context [Garlan et al., 2002; Helal et al., 2005]. Building on this foundation, several works have targeted adaptive user interfaces. Table 4.1 shows an overview over the works discussed in the following.

4.1.1 General Architectural Approaches

One type of adaptation is concerned with the adaptation of content presentation, for example in the work of Malandrino et al. [2010]. They present the MIMOSA framework for context-aware adaptation of Web content. They use adaptation services for transcoding Web content and describe an adaptation module to determine suitable adaptation services based on context data. The context data is represented in CC/PP. The work of Malandrino et al. [2010] is not very prescriptive, since it focuses on HTML and XML content adaptation, which only requires a client capable of displaying this content. However, device and modality choice is not supported, because of the focus on content. Usability is also not considered as an adaptation goal or in adaptation criteria.

The adaptation of modality choice and content is considered by Skillen et al. [2013]. They present an approach towards the personalization of mobile applications using SWRL rules to model adaptation. They aim at personalizing help services for older users and chose a travel scenario to illustrate their approach. Based on a user model, help services are personalized using the adaptation rules that are expressed in SWRL. The approach of Skillen et al. [2013] uses a high abstraction level of adaptation rules, but does not explicitly model usability criteria. The application is constrained to one personal device and device choice as well as device mobility is not supported.

Table 4.1 — A comparison of related works from an architectural perspective sorted by date. ●: meets the requirement, ◐: partly meets the requirement, ○: does not meet the requirement, -: not specified.

	Why?	When?	What?		To What?	Who?	How?			Where?						
Requirements:	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
	AG	AR	DC	MC	CA	STA	HAL	EM	SLS	AUA	LC	LP	EC	EAR	EUI	DM
[Rouvoy et al., 2009; Hallsteinsen et al., 2012]	◐	●	●	●	●	●	◐	●	○	●	●	○	●	●	●	◐
[Malandrino et al., 2010]	○	●	○	○	●	●	◐	○	●	●	●	●	●	●	●	◐
[Skillen et al., 2013]	◐	●	○	●	●	●	●	○	○	●	●	○	●	●	◐	○
[Evers et al., 2014]	●	●	●	●	●	●	◐	●	○	●	●	○	●	●	●	●
[Di Mauro and Cutugno, 2016; Di Mauro et al., 2017]	○	●	◐	◐	●	●	○	○	○	○	●	◐	●	○	●	●
[Gil and Pelechano, 2017]	◐	●	◐	●	●	●	◐	◐	●	●	●	◐	●	◐	●	○
[Schipor et al., 2019]	○	●	○	○	●	●	○	○	○	○	●	○	◐	○	●	○

Di Mauro and Cutugno [2016] propose a framework for multimodal interaction in intelligent environments. They developed a highly flexible and distributed approach, where a set of independent nodes provides some services, specifically including multimodal input or output. A node automatically connects to other nodes when it cannot process a request and forwards that request [Di Mauro et al., 2017]. Adaptation of input or output therefore only occurs when a node is not able to process an output or input request, adaptation rules are not implemented. While this approach provides loose coupling and supports device mobility, it does not consider usability criteria.

Schipor et al. [2019] present EUPHORIA, an event-based multi-layered architecture for smart environments using various heterogeneous devices. According to the authors, this approach supports easy design and implementation of new interaction techniques that may involve several interaction devices. It uses event-based adaptation using JSON message formats and a modular architecture. The application of this approach requires the implementation of all system components using the EUPHORIA architecture, message formats and protocols. It is therefore highly prescriptive. Rules for adaptation are not explicitly modeled and usability is not considered as an adaptation goal.

4.1.2 Self-Adaptive System Approaches

A category of adaptive systems using an architectural approach are self-adaptive systems for autonomic computing [Kephart and Chess, 2003]. These systems are designed to autonomously adapt to external context and to internal changes. Salehie and Tahvildari [2009] present several properties for such systems, which they call self-* properties: self-adaptiveness, self-configuring, self-healing, self-optimizing, self-protecting, self-awareness and context-awareness. Akiki et al. [2014] discuss the application of these properties to adaptive user interfaces and conclude that self-optimizing, self-configuring and context-awareness are relevant for this domain. Several approaches extend methods for self-adapting systems to adapt user interfaces.

Hallsteinsen et al. [2012] present a model-driven approach for self-adapting and context-aware applications, specifically for ubiquitous environments. This approach was developed in the MADAM project and its successor project MUSIC [Rouvoy et al., 2009; Geihs et al., 2009b]. It supports the development of self-adaptive and context-aware applications and includes an adaptation middleware and context management [Geihs and Wagner, 2013]. Adaptation is extended to user interfaces, including distribution and migration of user interfaces. The approach is illustrated using a public transport scenario, among others, which introduces a TravelAssistant application. Hallsteinsen et al. [2012]

stress that their modular approach supports a clear separation of concerns for robustness as well as long and continuous system runtimes. The middleware uses a MAPE (Monitor, Analyse, Plan, Execute) loop and reacts to context changes at runtime with a planning process using a runtime adaptation model. This adaptation model includes Quality of Service (QoS) dimensions and can be used in Web Ontology Language for Web Services (OWL-S) descriptions of services, enabling quality-aware service discovery [Geihs et al., 2009b; Rouvoy et al., 2008, 2009]. Available configurations are evaluated with regard to their utility in a given context. For this evaluation, utility functions are implemented [Geihs et al., 2009a]. Utility is considered as both meeting user preferences and optimizing the use of resources [Rouvoy et al., 2008]. User interfaces are used as pre-built interface options. As drawbacks, the authors remark that in experimentation their methods had a steep learning curve for developers unfamiliar with the approach and models [Paspallis, 2009; Floch et al., 2013], illustrating its complexity. This approach depends on a heavyweight model-based architecture and is highly prescriptive. Legacy systems are not supported. While extendable and describing qualities for adaptation assessment, the QoS ontology Geihs et al. [2009b] present and implemented in their prototypes does not include usability properties. Usability considerations are implemented on a specific, properties-based level.

Evers et al. [2014] address usability concerns of self-adaptive software and argue that the user's preferences about interactions as well as their intentions should be considered. They complement the MUSIC approach focusing on controllability by including the user in the adaptation process either allowing them to trigger adaptation or to modify or reject proposed adaptations. In this version of the MUSIC approach, the adaptation rules are stored in a separate component, increasing maintainability. The user's focus and their implicit or explicit feedback on adaptations are used to avoid usability problems during adaptation. While Evers et al. [2014] introduce a focus on the usability of adaptations, their system does not evaluate usability of adaptation autonomously. The approach focuses on user feedback for adaptations.

Gil and Pelechano [2017] developed a self-adapting system specifically to manage the obtrusiveness of interactions in mobile computing systems. They argue that, as Weiser and Brown [1996] conceptualized, calm computing should be implemented in mobile and pervasive computing environments in order to not distract users from their tasks and not interrupt them. In this system, an OWL ontology for user context and formalized rules are used to decide which level of obtrusiveness would be acceptable in a situation. A feature model using XML Metadata Interchange (XMI) models modalities and their properties. A set of activated features from this feature model forms an interaction configuration and describes used modalities or combinations of modalities as well as interaction components and their properties on a technologically independent level [Gil and Pelechano, 2017]. Formal rules define, which interaction configuration to use in a certain context. A Service Manager component registers all services and stores the current interaction configuration for each service. When context changes, interaction configurations for a service can change. If a service needs to notify a user and provide information, the Service Manager component informs the device to activate the appropriate interaction configuration via push notification. Gil and Pelechano [2017] evaluated the approach as correct and efficient and the participants of the user study rated its user experience as higher as the non-adaptive approach. While it is very modular and not prescriptive for user interfaces due to the service approach, it focuses only on one aspect of usability and other usability aspects are not easy to integrate.

4.2 Approaches Focusing on User Interfaces

Akiki et al. [2016] categorize user interface (UI)-based approaches towards adaptive user interface into three categories. The first category realizes adaptation in the window manager component, which is a complex and costly approach. Approaches in the second category were developed in the attempt to address adaptivity more efficiently. It involves widget toolkits, as for example in the Ubiquitous Interactor approach by Nylander and Bylund [2003]. Both types of approaches target Graphical User Interfaces. The third category developed in the scope of Model-Driven Engineering (MDE). Introducing an abstraction level to the user interface allows to develop more flexible user interfaces [Akiki et al., 2016]. Based on abstract user interface models, concrete user interfaces are generated.

4.2.1 Model-Driven Adaptive User Interfaces

The application of model-driven approaches introduces an abstraction layer to user interfaces as a basis for user interface generation. Generating user interfaces dynamically and at runtime can be used to adapt the user interfaces to external parameters. In the following sections, I will review several approaches with regard the requirements described in section 3.4. Table 4.2 summarizes the results.

Eisenstein and Puerta [2000] developed an approach towards a meta UI model. Their abstract UI model can be used to derive consistent user interfaces for several mobile devices [Eisenstein et al., 2000, 2001]. The authors mention that their method can be used to dynamically adapt graphical user interfaces to the context of the mobile device and user but do not provide any details. They focus on generating user interfaces for various devices and platforms at design time. The approach uses decision trees to implement design choices for the transformation of an abstract UI to an actual GUI implementation. The work of Eisenstein and Puerta [2000] stands as a representative of several comparable approaches that generate GUIs for various devices from an abstract user interface description. These approaches developed the foundation for following approaches, but do not support runtime adaptation and therefore are not applicable to answer the research question of this work.

Nylander and Bylund [2003] and Nylander et al. [2005] present the Ubiquitous Interactor, an approach to create tailored user interfaces for mobile services on various devices, using a widget-based approach. A service they call *interaction acts*, combined with specific presentation information for devices and users generates suitable interfaces from an abstract representation [Nylander and Bylund, 2003]. Possible interactions with a service are modeled in an interface-independent way using the Interaction Specification Language [Nylander et al., 2005]. Device specifics that influence the generation of an interface are specified in so-called customization forms. A customization form influences user interface generation and can be used to achieve a certain look-and-feel for all user interfaces of one service [Nylander and Bylund, 2003]. An interaction engine generates the interface dynamically at runtime. While the authors write that they want to extend their systems adapting the user interfaces also to user preference, the system does not consider environmental context or take quality measures into account while generating user interfaces [Nylander et al., 2005].

Vandervelpen and Coninx [2004] define one-way communication channels, called interaction resources. An interaction resource describes either one input or one output channel using one modality. Following this definition, one device can provide several interaction resources. Clerckx et al. [2005a] present the DynaMo-AID design process and runtime system that uses a model-based approach, based on the work of Vandervelpen and Coninx [2004]. The runtime system utilizes task models to support context-sensitive user interfaces for pervasive computing. The task model is based on the CTT notation that is extended to express contextual changes to the tasks. They apply location context as well as service context, mapping the availability of services in the environment. This work uses a context model with several types of context and supports runtime adaptation of GUIs. The authors do not consider different modalities in their work. However, it depends on several models for user interfaces, generating interfaces at runtime and is therefore highly prescriptive. The approach defines no adaptation rules and no usability criteria.

Based on this work, Luyten and Coninx [2005] describe an approach towards distributing user interfaces that are modeled using CTT and the User Interface Markup Language (UIML). User interfaces for different platforms are described in UIML documents. In a distribution process, the UIML description that suits best is chosen and then rendered for the chosen device. The authors write that they have chosen completeness and continuity as usability metrics for this decision process. Completeness means that a user should always have access to all user interface components that they need to fulfil their tasks. The authors also write that continuity is ensured when a user can always interpret the internal state of the system. Luyten and Coninx [2005] propose rules that can trigger the redistribution of the user interface if continuity can not be guaranteed any more. Luyten et al. [2005] propose a design platform for task-based user interfaces in ambient intelligent environments, building on this work. They focus on the design phase and do not implement runtime adaptation, but integrated in the DynaMo-AID runtime architecture from earlier work, runtime adaptation could be realized [Clerckx et al., 2005a]. The representation of the mentioned usability metrics is not described and if they could be extended is therefore unclear.

Further advancing this approach, the work of Clerckx et al. [2008] supports multimodal interfaces. They extend the task model so that designers can use modality interaction constraints to define which modalities are possible for a user task. The runtime system uses SPARQL to query information about the modality interaction constraints during the mapping process that maps tasks to devices and modalities. The decision process chooses devices that provide as many of the defined modalities as possible. The queries also statisfy the CARE properties for multimodal interfaces defined by [Coutaz et al., 1995]. These properties relate modalities to each other in multimodal interfaces. The CARE properties are: complementarity, assignment, redundancy and equivalence. In this extended work of Clerckx et al. [2008], adaptation rules are implemented on a more abstract level using SPARQL and usability metrics are modeled for adaptation decisions, based on the CARE properties for multimodal interfaces.

Gajos et al. [2010] introduce the SUPPLE framework that aims at interface generation for ubiquitous environments. Their java-based framework supports adaptive rendering of user interfaces for different types of devices and applications. The SUPPLE framework implements interface generation as an optimization problem. The cost to minimize is the user's effort and the authors use user and device specific cost functions, taking device and user context into account [Gajos and Weld, 2004]. Gajos and Weld [2004] use declarative functional user interface specifications as well as a device and user model for their optimization algorithm. In their device model, they use functions that compute the user effort necessary to use specific UI widgets in given context and on given devices. Their user model depends on so-called user traces that capture previous sequences of usage by a user [Gajos and Weld, 2004; Gajos et al., 2005a]. In a later extension of the framework, Gajos et al. [2007] developed a cost function that incorporated the user's ability to control the input device, to factor in motor impairment or other atypical abilities [Gajos et al., 2010]. Gajos et al. [2005b] also extend the cost function to incorporate *presentation consistency* to achieve a similarity of generated user interfaces for the same application across several devices. The SUPPLE framework is able to generate user interfaces for a multitude of devices and supports several input modalities. However, modality can not be adapted at runtime. The SUPPLE framework adapts to device characteristics and usage, but does not take context of the user's environment or situation into account.

Duarte and Carriço [2006] present FAME, a Framework for Adaptive Multimodal Environments. The authors specifically address multimodal and natural interfaces and argue that these interfaces should choose the modality best suitable to a given context. Additionally, they write that devices leaving and entering an environment should be supported. Their framework is intended to support the development of model-based adaptive multimodal user interfaces. They introduce several context models that capture user characteristics and preferences, environmental factors and platform facts. An interaction model is used to describe interactive components and templates for their usage with respect to the context. The input and output behavior of an application then adapts based on user input and context facts. For each adaptive interface component, a behavioral matrix is defined that not only models the adaptation rules but also how they evolve, for example by recording the user's behavior and preferences. This matrix is intended for analysis purposes but the authors write that it can also be implemented for runtime adaptation decisions. The rules they describe are very specific and the framework mostly supports the conceptual work of analysis and evaluation. They do not specify how adaptation at runtime could be implemented and realized.

Blumendorf et al. [2008] developed executable user interface models for context-aware adapatation at runtime. They implemented their approach in a project researching Ambient Assisted Living Environments [Blumendorf and Albayrak, 2009]. Blumendorf et al. [2008] present a so-called Multi-Access Service Platform (MASP), that uses executable user interface models to adapt and generate multimodal user interfaces at runtime. The rules guiding the transformation of user interfaces in this approach are expressed as abstract model transformations, but depend heavily on user interface models and are therefore not easy to extend. In order to include their approach of executable models into their User Interface Management System (UIMS), the authors write, illustrating the complexity and prescriptiveness of their approach:

> *"Utilizing executable models as the underlying concepts for the approach lead to a complete redesign of the system"* [Blumendorf et al., 2008].

Based on this work, Blumendorf et al. [2010b] developed an algorithm that autonomously distributes a user interface to available devices. The algorithm uses information about the location of user and device, user preferences as well as requirements and constraints that were defined by the application designer at design time, to identify suitable interaction resources. Such interaction resources are "one-way interaction channels" using one modality on one device for either input or output. The algorithm is able to create multimodal interfaces. Relations between devices and modalities are modeled using the CARE properties. Considering output, they aim at finding the most suitable combination of output resources, but the specification of "most suitable" is not defined. The distribution algorithm and components are integrated in the MASP. Schwartze et al. [2010] describe an end user development approach for the user interface adaptations by the platform that involves the user in adaptations. They also introduce usability criteria using the CARE properties, but their evaluation at runtime remains unclear. The approach is as prescriptive as before.

In the work of López-Jaquero et al. [2008], the authors present an agent-based framework for adaptive Graphical User Interfaces. The authors use the USer Interface eXtensible Markup Language to support user interface generation in the adaptation process. Their system uses adaptation rules in a knowledge base to decide for an adaptation based on the context of use. The authors do not specify the formalization of adaptation rules and usability criteria are not defined. This approach is loosely coupled due to the agent-based architecture. The adaptation process is realized in an autonomous component, but the approach is limited to Graphical User Interfaces.

Peissner et al. [2012] propose a user interface generation approach that uses multimodal design patterns and adaptation rules that are organized in an open repository. The authors focus on accessible user interfaces that adapt at runtime. Their approach enables user feedback for UI adaptations and is self-learning to optimize personalization over time. The design patterns they propose have a machine-readable and a human-readable part and are intended to foster shared UI adaptation knowledge. The open repository approach is an interesting concept for the mobility domain, since mobility applications are implemented or replicated in various instances, for various transport companies. While the authors discuss adaptation of modalities, they only implement adaptation and generation of GUIs. Usability criteria are, however, not modeled explicitly.

Miñón et al. [2013] provide adaptation rules to adapt user interfaces for people with disabilities based on context. They designed these rules to be usable with several types of model-based languages but propose a normalization step that transforms the rules to use the MARIA language [Paternò et al., 2009]. The approach uses abstract Event-Condition-Action rules as adaptation rules, but does not include explicitly modeled usability criteria. The work of Miñón et al. [2013] is also limited to Graphical User Interfaces.

Motti and Vanderdonckt [2013] describe a framework for context-aware user interfaces that is able to adapt modality and relies on ECA rules to guide adaptation. The approach is conceptual and needs to be implemented in a suitable runtime environment. Usability criteria are not explicitly modeled and the approach does not support device choice or device mobility.

Akiki et al. [2016] discuss the application of model-driven adaptive interfaces for existing systems of high complexity, such as Enterprise Resource Planning (ERP) systems. The authors propose an adaptation technique with tool support to specifically integrate adaptive user interfaces in legacy software. The adaptation goal is the simplification of a GUI by applying feature-set minimization. Their user interface adaptation technique is called Role-Based UI Simplification (RBUIS) and implemented in the Cedar architecture [Akiki et al., 2012, 2013]. The RBUIS approach is based on roles that are assigned to tasks in the task model and resources, including user interfaces, are only available to specific roles [Akiki et al., 2013]. A role can be shaped by the user goals and tasks, but also by current context. The adaptive behavior of their method is defined in rules in a relational database that stores adaptation steps related to context data. The authors propose an architecture that uses Web Services to provide adaptation services for legacy software. They also propose a technique to reverse-engineer existing user interfaces, in order to develop the necessary user interface models. This approach explicitly supports legacy systems. However, existing user interfaces need to be reengineered and suitable UI models need to be developed. The approach focuses on Graphical User Interfaces and does not support device or modality choice or device mobility.

Hussain et al. [2018] describe a model-based approach towards adaptive user interfaces that takes current context and user experience into account. User model, context and device models are modeled as OWL ontologies. Adaptation rules are realized as SWRL rules. In their user model, the authors model several context facts related to user experience, including a hedonistic quality and pragmatic quality as well as the user's emotional state. This information is collected implicitly by monitoring the user' behavior and explicitly by including a user questionnaire in the target application. They state that the perceived context is used to adapt the user interface during runtime, according to the adaptation rules that are edited at design time. The adaptations are realized by generation of a new user interface. The authors have evaluated their approach and report a positive impact on user experience. The user experience criteria applied in their adaptation rules are not explicitly modeled. The adaptation rules follow the ECA scheme and are relatively specific, for example stating that if the user has low vision, the text size should be increased to 16pt. The approach targets a user interface on only one device, supporting adaptation of modalities.

4.2.2 Plastic Interfaces

Thevenin and Coutaz [1999] introduce the notion of *plastic interfaces* that preserve their usability while adapting to changing context of use. This concept sparked a new line of research in the field of adaptive user interfaces. A selection of works from this research will be reviewed in this section. Table 4.2 shows the results of this review, as well.

Thevenin and Coutaz [1999] and Calvary et al. [2001, 2003] introduce a unifying reference framework for plastic and multi-target user interfaces. Multi-target user interfaces aim at supporting several platforms and devices, while plastic user interfaces focus on preserving usability while adapting to context. The framework considers user, platform and environmental context. It identifies several levels of abstraction and suggests suitable models for each level to support generation of adaptive GUIs. Calvary et al. [2003] also introduce a runtime process for adaptation. In this framework, adaptation is triggered by context changes. Upon a change of context, candidate solutions for the user interface are computed and the best is chosen. The choice is based on criteria of persistence and migration cost. This approach is a *utility-based* approach following the categorization by Geihs et al. [2009b]. It focuses on GUIs and while it supports platform and device changes, it does not support a device or modality choice for adaptation.

Balme et al. [2004] introduce the CAMELEON-RT architecture reference model for distributed, migratable and plastic user interfaces. The CAMELEON-RT architecture suggests a layered approach, separating the interface systems layer from a middleware layer and a concrete platform layer. As a central component, an open adaptation manager triggers UI adaptation and realizes the adapted interface. The authors define usability as a set of properties, where for each property a metric with a domain of values is defined. The usability of a user interface is defined as set of properties and their values. In case of adaptations, these properties need to remain within given limits, to ensure usability. As example for such properties, Balme et al. [2004] name observability and predictability. Balme et al. [2004] support device migration and introduce explicit usability metrics to compute the usability of an adaptation.

Table 4.2 — A comparison of related works from an user interface perspective sorted by date. ●: meets the requirement, ◐: partly meets the requirement, ○: does not meet the requirement, -: not specified.

	Why?	When?	What?		To What?	Who?	How?		Where?							
Requirements:	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
	AG	AR	DC	MC	CA	STA	HAL	EM	SLS	AUA	LC	LP	EC	EAR	EUI	DM
[Calvary et al., 2001, 2003]	●	●	○	○	●	●	◐	○	○	◐	○	○	●	○	●	◐
[Eisenstein and Puerta, 2000; Eisenstein et al., 2001]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
[Nylander and Bylund, 2003; Nylander et al., 2005]	○	●	○	○	◐	●	◐	○	○	○	◐	○	○	◐	●	◐
[Balme et al., 2004]	●	●	○	○	●	●	●	●	○	○	○	○	●	●	●	●
[Vandervelpen and Coninx, 2004; Clerckx et al., 2005a]	◐	●	○	○	●	●	○	○	○	○	○	◐	●	◐	●	●
[Gajos and Weld, 2004; Gajos et al., 2005a, 2010]	◐	●	○	○	●	●	◐	●	○	○	○	○	○	◐	●	◐
[Luyten and Coninx, 2005; Luyten et al., 2005]	◐	◐	◐	◐	●	●	◐	◐	○	○	○	○	◐	◐	●	●

Table 4.2 — A comparison of related works from an user interface perspective sorted by date. ●: meets the requirement, ◐: partly meets the requirement, ○: does not meet the requirement, -: not specified.

	Why?	**When?**	**What?**		**To What?**	**Who?**	**How?**		**Where?**							
Requirements:	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
	AG	AR	DC	MC	CA	STA	HAL	EM	SLS	AUA	LC	LP	EC	EAR	EUI	DM
[Calvary et al., 2005]	●	●	◐	○	●	●	◐	●	○	○	○	○	●	◐	●	●
[Duarte and Cariço, 2006]	●	●	◐	●	●	●	◐	○	○	●	○	○	●	◐	●	●
[Sottet et al., 2006, 2007]	●	●	○	○	●	◐	●	●	○	●	○	○	●	●	●	◐
[Bachvarova et al., 2007]	◐	-	-	●	-	●	-	○	-	-	-	-	-	-	-	-
[Demeure et al., 2007, 2008]	●	●	○	○	●	●	◐	○	○	○	○	◐	●	◐	●	◐
[Clerckx et al., 2008]	◐	●	●	●	●	●	●	◐	○	◐	◐	○	●	●	●	●
[Blumendorf et al., 2008; Blumendorf and Albayrak, 2009]	◐	●	◐	◐	●	●	◐	○	○	○	○	○	●	◐	●	●
[López-Jaquero et al., 2008]	◐	●	○	○	●	●	●	○	○	●	●	○	●	●	●	○
[Paternò et al., 2009]	○	●	○	○	-	-	◐	○	○	-	-	○	-	●	●	-

Table 4.2 — A comparison of related works from an user interface perspective sorted by date. ●: meets the requirement, ◐: partly meets the requirement, ○: does not meet the requirement, -: not specified.

	Why?	When?	What?		To What?	Who?	How?		Where?							
Requirements:	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
	AG	AR	DC	MC	CA	STA	HAL	EM	SLS	AUA	LC	LP	EC	EAR	EUI	DM
[Blumendorf et al., 2010a,b; Schwartze et al., 2010]	◐	●	●	●	●	●	●	◐	○	○	○	○	●	◐	●	●
[Calvary et al., 2011]	●	●	○	○	●	●	○	◐	○	●	○	○	●	○	●	◐
[Peissner et al., 2012]	◐	●	○	◐	●	●	●	○	○	○	○	○	-	●	●	○
[Miñón et al., 2013]	◐	●	○	○	●	●	●	○	○	●	●	○	●	●	●	○
[Motti and Vanderdonckt, 2013]	●	-	○	●	●	-	●	○	○	-	-	○	●	●	●	○
[Castillejo et al., 2014]	○	-	-	-	●	-	●	○	○	-	-	-	●	●	-	-
[Akiki et al., 2016]	◐	●	○	○	●	●	◐	○	●	◐	◐	◐	●	●	●	○
[Hussain et al., 2018]	●	●	○	●	●	●	◐	○	○	●	◐	○	●	●	●	○
[Yiğitbaş et al., 2017a, 2019, 2020]	●	●	○	●	●	●	◐	○	○	●	●	○	●	●	●	○

Calvary et al. [2005] widen the definition of plasticity to include not only usability qualities but also other software quality properties based on ISO models for software product quality. The authors include two additional quality models in their reference framework, a quality in use model as well as a model for external and internal quality. In these quality models, designers can explicitly model quality requirements that are used in the generation process of user interfaces. Calvary et al. [2005] introduce COMETs (COntext Mouldable widgET) as plastic UI widgets. Their approach supports the migration of UIs to other devices. It is GUI-centric and uses transition models to guide adaptation, which tend to be verbose and are difficult to expand.

Sottet et al. [2006] target the open issue of how the adapted user interface can review its compliance with explicitly modeled requirements at runtime to be plastic. Sottet et al. [2007] present an approach combining Model-Driven Engineering (MDE) and Service-Oriented Architecture (SOA) for plastic interfaces. The authors describe the necessary models for the user interface as a graph of models, where models are connected by mappings [Sottet et al., 2006]. During runtime, this graph of models is modified using model transformations. The transformations are expressed using transformation rules and the authors stress that the transformation model can in turn be transformed at runtime, as well. Transformations can be executed manually, semi-automated or automatically. Transformations of models are used as the actions of Event-Condition-Action (ECA) rules and the usability properties the actions statsify are explicitly modeled. Thus, the responsible system component can apply adaptation policies, rating the ECA rules based on their usability properties. This approach allows for explicitly modeled usability properties, where developers can choose the usability framework to work with. Regarding the generation of user interfaces, it is very prescriptive, however.

Demeure et al. [2007] address runtime plasticity of user interfaces introducing a semantic network that describes concepts, UI elements and their relationships on all levels of the CAMELEON reference framework. The authors introduce "plasticity questions" as a tuple of a plasticity goal and a set of plasticity solutions, that are ways to reach that goal. They argue that plasticity solutions can be computed using the semantic network and that this approach introduces more flexibility for adaptations at runtime. They utilize COMET widgets and describe a tool called the COMETs inspector that lets a designer inspect and also modify a user interface using proposed adaptation operations based on the semantic net [Calvary et al., 2005; Demeure et al., 2007]. Demeure et al. [2008] present an architectural framework applying COMETs for plastic user interfaces. The framework is event-based and realizes a strong separation of concerns between its components. It addresses multimodal user interfaces and supports redundancy and equivalence as two of the CARE properties for automatic reasoning about multimodality. Specific conditions that support redundancy and equivalence can be expressed in a formalized way. Their architecture supports a strong separation of concerns, especially within the COMET widgets, while strong dependencies on several models remain. The conditions and transformation rules that shape adaptation in this approach are expressed on a relatively low abstraction level, which implies an extensive effort at design time.

Calvary et al. [2011] discuss transport and mobility as a worthwile application domain for plastic user interfaces. In this conceptual work, they consider a wider concept of quality for plasticity than usability only. Such an extension beyond usability was first discussed by Cockton [2004] and includes the expectations of the user and their goal. Calvary et al. [2011] discuss that notion for transport and conclude that value criteria for users in transport can be saving time or money, for example. They stress that a variety of context factors need to be taken into account, including disabilities or weather and the information need of the user about their chosen route and mode of transport. The authors identify situations and reasons for UI adaptation as well as possible adaptation goals specifically for transport scenarios. Calvary et al. [2011] present a conceptual work and an important discussion of plastic user interfaces for transport and mobility. Their approach targets GUIs only and while they discuss extended quality criteria for GUIs in the mobility domain, these criteria are not formalized and their application remains unclear.

Yiğitbaş et al. [2020] propose a model-based approach to self-adaptive user interfaces. Their approach allows not only to model and generate user interfaces but also model and generate adaptation logic and context providers [Yiğitbaş et al., 2017a]. They argue for a separation of adaptation and context modeling from user interface modeling. In their approach, they employ a domain model, expressed as a UML class diagram and a user interface model using the Interaction Flow Modeling Language (IFML) [Brambilla and Fraternali, 2015]. The user interface is then generated using these two models. Adaptation rules are modeled in a rule modeling language presented by Yiğitbaş et al. [2017b], called AdaptUI. The adaptation rules bind to entities from the UI models and the context model. In a generation step, adaptation logic is generated from the AdaptUI model in form of an adaptation service. This service monitors context, evaluates context changes and triggers the adaptation of the generated user interface. The authors implemented two case studies to show the feasability of their approach. They also conducted an on-the-fly usability test, using user feedback questions during application usage, triggered by UI adaptation and could see an increase in end-user statisfaction [Yiğitbaş et al., 2019]. The modeling language for adaptation rules creates a certain level of abstraction. However, the authors do not model criteria used for these rules, wether for usability or other qualities. The rules are still very specific, using concrete sensor values, for example. The approach targets GUIs only.

4.2.3 Models and Languages for Adaptive User Interfaces

In the ongoing research about model-based user interfaces and adaptive UI, a large variety of modeling languages and models have been developed. In contrast to the approaches discussed above, these models and modeling languages mostly do not specify their implementation and usage in concrete systems.

Bachvarova et al. [2007] describe a modality ontology that is intended to support the choice of an appropriate modality as well as a suitable combination of modalities for output generation. The ontology models modalities on a content and property level. The content level is used to describe the information that is meant to be expressed. The authors integrate the existing MPEG7 ontology on that level [Hunter, 2005]. On the property level, the authors model the nature of modalities in three categories, describing the information presentation properties, the perception properties and structural properties of a modality very specifically. On the perception level, the authors distinguish visual, auditory and haptic modalities and specify the properties in these classes in more detail. Structurally, the authors use relations between modality classes to express that modalities are often used together, to support the combination of modalities. The ontology proposed by Bachvarova et al. [2007] is specifically intended to support modality choice, while device choice is not supported. It provides very specific content and information models that differentiate between several types of graphs and their applications, for example. How modality is chosen in a running system is not detailed.

Paternò et al. [2009] propose MARIA, a user interface description language specifically for service-oriented ubiquitous environments. The language was later submitted to the W3C for standardization [Paternò et al., 2012]. MARIA can be used to annotate the Web Services Description Language (WSDL) description of a Web Service, providing information about the user interface that is needed to access this Web Service's operations. The language can be used to describe dynamic changes of the user interface, changing presentation as well as navigation. The authors write that the composition of an interface based on MARIA can take place statically, at design time as well as dynamically at runtime [Paternò et al., 2012]. It supports the generation of user interfaces using several modalities and devices, including graphical user interfaces, digital TV, voice-based interfaces and others. Paternò et al. [2009] also demonstrate that

MARIA can support migrating user interfaces by implementing a prototype. In case of migration, a new interface is generated that suits the target device. MARIA allows the specification of dynamic changes of user interface and also supports multimodal UI descriptions [Paternò et al., 2009]. However, usability criteria are not included.

Castillejo et al. [2014] present an ontology that integrates a user model, context model and device model specifically to support adaptive user interfaces. Their model takes the capabilities of users into account and models restrictions to these capabilities that stem from stressful situations or activities in general. An example are activities that restrict the usage of the hands, for example driving. The authors also describe two types of SWRL rules that their ontology supports. Preadaptation rules infer higher level context data and adaptation rules deduce properties for the adaptation of the user interface. The work of Castillejo et al. [2014] allows the specification of adaptation rules in SWRL. Usability knowledge is not modeled explicitly in this ontology.

4.3 Discussion

The previous sections have discussed several related works from two main perspectives. Approaches that have a strong relation to ubiquitous or pervasive computing and an architectural background are all loosely coupled. In these approaches, adaptation is mostly implemented in modular, autonomous components, as can be seen regarding requirements 10 and 11 in table 4.1. All architectural approaches support adaptation at runtime and are context-aware. Notably, device mobility is also supported often, a feature which is straightforward to support in a modular and loosely coupled architecture. These approaches also often support the choice of device or modality. Such features are described in visions for ubiquitous or pervasive computing, where a distribution not only of computing but also of user interfaces is a central element [Weiser, 1991; Satyanarayanan, 2001].

However, in these approaches, usability is mostly no concern. Skillen et al. [2013] consider usability as a part of service personalization. Usability measures are only an implicit factor in their approach, though. They provide abstract adaptation rules using SWRL but target only one application on one personal device. Results of the MUSIC project show flexible adaptation including user interfaces and the authors explicitly discuss usability during adaptation [Hallsteinsen et al., 2012]. In this work, an explicit model for usability factors exists, but the adaptation rules are using a low abstraction level. Evers et al. [2014] complement this work with further considerations of usability affected by adaptations. However, the MUSIC approach is a heavyweight model-based framework that is very prescriptive and not applicable to legacy systems. The work of Gil and Pelechano [2017] is less prescriptive and is explicitly concerned with usability, but only focuses on obtrusiveness as one factor of usability. It is not clear from this work, if other usability factors could be extended.

Most of the architectural approaches, while flexible and modular, are highly prescriptive and do not support legacy systems. Their implementation for mobility systems would require a reengineering of many existing components, which is not feasible.

The approaches discussed from the user interface perspective can also achieve high flexibility for adaptation at runtime. Usability is mentioned as adaptation goal more frequently in comparison to architectural approaches and approaches to plastic interfaces are explicitly designed to retain their usability during adaptation. They achieve flexibility due to a model-based development approach. This means, they rely on several user interface models that often are only applicable to GUIs. They accomplish a high diversity of adaptation outcomes, ranging from layout optimization to customization for different screen sizes or other device characteristics. These types of adaptations are not relevant for the scope of this work, as is illustrated in the scenarios in section 3.3. Device or modality choice is rarely supported by user interface focused approaches. Regarding the high effort necessary for modeling abstract and concrete user interface models, model-based approaches are also highly prescriptive and lack support of legacy systems. The work of Akiki et al. [2016] stands out, specifically supporting legacy systems. However, it still requires reengineering of these legacy user interfaces, which is quite complex and expensive. Usability criteria are explicitly modeled in a few of these works, but they are often tightly integrated with the user interface models, which suggest expensive modeling. The high prescriptiveness and and their lack in support of legacy systems shows that solutions following MBUID approach are not appropriate for the application in the mobility and transport domain.

In summary, most of the discussed approaches from both categories support adaptation criteria only on a very specific level that requires extensive modeling at design time. Highly dynamic context changes and a high number of situations or unknown situations at runtime can not be all conceived and modeled at design time. This means that a deeper knowledge of usability factors must be made available to the system, since usability related decisions can't be anticipated at design time to an adequate degree. This is the first research gap identified.

Research Gap 1: Usability Model

There is a lack of models for usability factors that are able to provide a level of abstraction for adaptation rules to separate them from concrete implementations, ensuring extensibility and maintainability.

At the same time, high flexibility and a wide range of adaptation features is in most approaches linked with a highly prescriptive approach that places many and various demands on its implementation. These demands can not be met in a system environment as it is present in mobility, and in public transport specifically, including legacy systems. Considering the complexity of some adaptations, such as adapting GUI navigation based on screen size or distributing all elements of a Graphical User Interface to several devices, a prescriptive and heavyweight approach is certainly warranted. However, as the scenarios in section 3.3 show, such elaborate adaptations are not necessary to keep users informed in a usable way during their trip. The second research gap is therefore the following:

Research Gap 2: Non-prescriptive device and modality adaptation

There are no approaches that allow adaptations of devices and modalities as well as device mobility while supporting legacy systems and not being highly prescriptive.

Chapter 5

Concept

In this chapter, the concept for this work will be described and explained. It is intended to answer the Main Research Question and subsequent research questions. This concept aims at enabling an adaptive ubiquitous mobility system to adapt its information provision to the user's situation, preserving usability of output device and modality. The concept results in a framework for autonomous usability assessment and adaptation in ubiquitous public transport systems. In this chapter, the design choices and the resulting architecture of the framework are presented.

This chapter is in part based on the following publications:

- C. Keller and T. Schlegel. Model Based and Service Oriented Interaction for Ubiquitous Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, page 429–434, New York, NY, USA, 2016. Association for Computing Machinery; [Keller and Schlegel, 2016]
- C. Keller and T. Schlegel. How to get in Touch with the Passenger: Context-Aware Choices of Output Modality in Smart Public Transport. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers*, UbiComp/ISWC '19 Adjunct, New York, NY, USA, 2019. ACM; [Keller and Schlegel, 2019]

5.1 Objective

The objective of this work is to enable passenger information systems to adapt information output to the passenger's situation. Following Vandervelpen and Coninx [2004], an output option is defined as an output-related interaction resource, a combination of a device and a modality, forming a one-way communication channel to the user. Adaptation, for the course of this work, is defined as the choice of an output option for the delivery of a message to the user, following requirements 3 (adaptation of device choice) and 4 (adaptation of modality choice). Consequently, an approach towards this objective needs to be able to establish which output devices and modalities exist, to choose from these options and then to output information using the chosen device and modality.

Section 3.1 discusses that a diverse and complex infrastructure for passenger information already exists in public transport. It would not be feasible to redesign and implement all user interfaces for passenger information systems from scratch with a new adaptation technology. This means that adding adaptivity must work with the existing systems. Requirements 9 and 12 express that legacy systems must be supported and a solution should be not prescriptive, in order to work well with existing systems and interfaces. Research gap 2 points out that there is a lack of approaches that can support device and modality choice supporting legacy systems. A concept addressing this research gap must be able to integrate existing user interfaces. Loose coupling and a modular architecture can achieve this, facilitating easy integration into an existing system as expressed in requirements 10 and 11.

Requirement 1 defines the adaptation goal usability. This goal influences the choice of adaptations. Usability of available output options must be known and must be a factor in deciding which option should be used. The usability of output options depends on the situation they will be used in. This situation forms the context of use for an interaction and must be considered when determining its usability. Requirement 5 expresses that an approach needs to be context-aware. A context-aware system can access information about the context of use of output devices and modalities in a given situation. Since adaptation must take place at runtime, as stated by requirement 2, usability needs to be determined at runtime as well. As discussed before, the context factors influencing usability are changing frequently in a mobility system and the usability of output options can not be determined for every possible situation at design time. An approach needs to be able to assess usability of output options at runtime, based on the current context.

Such a usability assessment and decision process should be flexible enough

Design choices for the concept.	
An approach needs	**Design Choices**
… knowledge about devices and modalities	►Ontologies modeling devices and modalities
… the ability to determine, which devices and modalities are available	►Device and modality knowledge as context data
… to choose from output options	►Decision making process
… to be loosely coupled and modular	►Service-oriented approach
… to determine the usability of output options at runtime	►A usability assessment process
… to take the context of use into account determining the usability of output options	►A context ontology and context-aware assessment and decision process
… a high abstraction level for adaptation rules	►Adaptation rules built on ontologies
… an explicit model of usability criteria	►A usability ontology that can be referenced in adaptation rules

Table 5.1 — A summary of the core design choices for the concept

to be applicable in unforseen situations. As described in section 2.5.4 and as can be seen in related work in chapter 4, some type of adaptation rules are commonly implemented to realize adaptive behavior. As requirement 7 expresses, these rules should be formalized on a high abstraction level to be reusable, widely applicable and to provide the needed flexibility. Since those rules need to support assessment of usability and decisions based on the assessment results, usability factors should be explicitly modeled, expressed in requirement 8. Research gap 1 shows that there are no approaches that explicitly model usability criteria to assess usability at runtime.

The concept presented in this and the following chapters therefore aims at realizing adaptation rules that utilize the modeled usability criteria for the implementation of an autonomous usability assessment on an abstract level.

5.2 General Approach

This concept provides an approach for assessing the usability of output devices and modalities considering the current situation of the passenger. It allows the adaptation of output device and modality for passenger information. The core design choices of this concept and the requirements and prerequisites they satisfy are shown in table 5.1.

The approach aligns with ontology-based context-aware ubiquitous systems as discussed in section 2.5. The approach uses ontologies expressing knowledge about devices, modalities, the context of use and usability and therefore address research questions 1 and 2. The approach comprises a usability assessment and a decision making process. The usability assessment can determine the usability of output options and addresses research question 3. Based on this assessment, a decision making approach then reaches a decision for a usable output option, which addresses research question 4.

At runtime, using context data about devices and modalities is used to determine, which output options are available. Context data representing the context of use is used during the usability assessment.

Knowlege about usability criteria modeled in the usability ontology is used in the usability assessment and decision making process. The assessment of the usability of a device or a modality, as well as the decision which option should be used, are shaped by adaptation rules. The rules reference the ontologies and can therefore be expressed at a high level of abstraction. The context-aware usability assessment and decision making process is implemented using a service-oriented approach, which provides loose coupling and autonomy. After the service reaches a decision, which output option should be used, a system can send the information to the chosen device and request the output of this information using the given modality. This way, existing user interfaces do not have to be changed and the approach is not prescriptive.

The framework implementing this approach is concerned with identifying, assessing and deciding for one or more output options based on usability criteria and context information.

5.2.1 Limitations of this Approach

Existing passenger information systems generate information for users and some of them also can personalize these messages on a certain level. Personalized public transport systems are also currently researched, see for example Beutel et al. [2016]; Wienken et al. [2017]; Keller et al. [2019a]. Therefore, this approach is not considering the generation of messages for the user. Similarly, passenger information systems currently are implemented for several types of devices, as described in section 3.1. New user interface technologies and devices for passenger information are also subject of recent research, see, for example Chow et al. [2016]; Mayas et al. [2018]; Keller et al. [2019b]. In the context of this work, the implementation of user interfaces for several types of devices will therefore not be considered. Additionally, the acquisition, processing and reasoning of context data is out of scope for this work. This work will provide a context model, since the usability assessment must use context data.

5.2.2 Architecture

Figure 5.1 shows the general architecture for the framework. The modules and interfaces encircled with the dashed line are part of the framework, while all other elements are out of scope. The public transport system must implement the generation of messages and the output of messages on a chosen output device, using the chosen output modality provided by the framework.

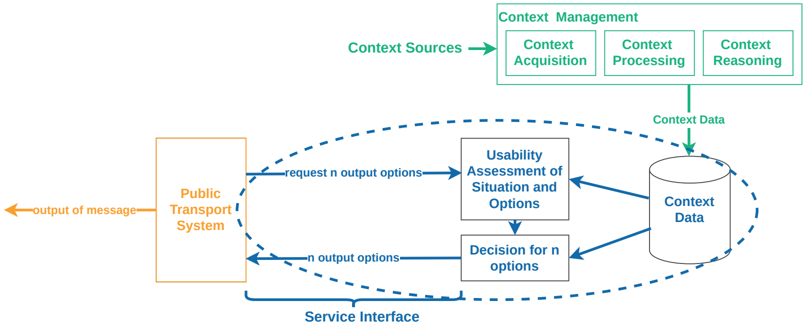


Figure 5.1 — Modules and data flow for an adaptive ubiquitous mobility system.

In this concept, the framework provides an interface for a public transport system to request a number of output options for a given message and user. Since the type and content of the message can influence the usability of output options, the message can be submitted with the request and then will be considered in the usability assessment. The response to this request will be the requested number of output options, chosen based on usability criteria. This simple interface can be realized as a service interface and thus requires minimal customization of the public transport system. The public transport system just needs to be extended by some module that is able to request an output choice, to process the response of the asessment framework and to initiate the output of the message with the chosen option.

The service interface for the public transport system as well as the interface for access to context data can be kept simple. Other interfaces are not necessary. They make the framework independent and autonomous from the system that requests a usability assessment on one side and the context management system on the other side.

The models and components that are part of the framework for autonomous usability assessment and adaptation in ubiquitous public transport systems will be outlined in the following paragraphs.

Context Models and Data: As described in section 2.4.1, ontologies are a widespread approach towards context modeling that is very expressive and extendable, supporting requirement 13. An extendable solution for the context model allows to incorporate additional context facts when, for example, new context sources can be exploited. In chapter 6, I present ontologies for several types of context data in ubiquitous public transport. They provide a fundamental structure for the organisation of context data and can be extended as needed for the implementation of the framework in any target system. The ontology that represents public transport data is based on standards that are used for public transport systems in Germany and is therefore compatible with data from german passenger information systems.

Usability Assessment of Situation and Options: The necessary knowledge for a usability assessment of the user's situation and available output options is modeled specifically in a usability ontology that is described in chapter 7. Adaptation rules that utilize this knowledge and form the basis of the assessment are also specified there. Chapter 7 then details the modules in which these rules are executed. The result are rated output options and a rated message, if given, that are the basis for a decision.

Decision for n Options: Several filters that apply additional filter rules allow the framework to choose a given number of output options. These filters will be detailed in chapter 7 as well. The adaptation component provides a request-based interface for any application that requires an output choice and only uses the interface of the context data store. It therefore can be easily used in existing systems. This leaves the extension of existing applications to request output choices from the adaptation component and to implement these choices. The approach does not require any specific user interface technology in order to adapt. As long as the application is able to use a specific device and specific modalities on this device for information output, these can be modeled in the ontology accordingly and become available as output options. The usability assessment rules can be adapted to new output options or even a new domain, in order to reflect the usability values for the target system.

This concept focuses on public transport rather than mobility in general. The ontologies provide a basic structure that can be extended by any other transport mode. This approach is easily extendable, lightweight and reusable. It can be extended and be applied to a mobility system covering several transport modes. The details of the developed ontologies and the framework components are described in the following chapters 6 and 7. Section 9.1 describes a proof of concept implementation of the framework.

CHAPTER 6

Ontologies for Context-Aware Public Transport Information Systems

The context of use of a system directly impacts its usability. A context-aware system can acquire, process and interpret context data. This context data can then provide the context of use as a foundation for a usability assessment at runtime. The modeling of context data is highly relevant as a basis for this work. Establishing context models that can represent the context of use is necessary for the development of a usability assessment that must consider this context of use. As described in chapter 5, an ontology-based approach provides the necessary flexibility, expressivity and extensibility. The context models are therefore modeled as ontologies.

In this chapter, the context analysis is detailed in section 6.1. It identifies the dimensions of the context of use that are relevant to adaptive smart ubiquitous systems and therefore answers research question 1. Based on this analysis, ontologies are modeled and described in the following sections. Knowledge about different domains is separated into different ontologies in order to allow easy reusability and extendability. A public transport ontology provides a domain model for public transport. The ontology presented in section 6.2 is compatible with public transport data standards. A task ontology for public transport tasks builds on the analysis of situations in section 3.2 and the more detailed hierarchical task analysis in section 6.1.2. An interaction ontology provides a vocabulary to describe interaction modalities and is detailed in section 6.4. Section 6.5 presents the device ontology for devices used for passenger information. A context ontology described in section 6.6 integrates each ontology to express context facts from different dimensions. This representation of the context of use adresses research question 2. All ontologies are published as a dataset [Keller, 2023].

This chapter is in part based on the following publications:

- T. Schlegel and C. Keller. Model-based Ubiquitous Interaction Concepts and Contexts in Public Systems. In *Proceedings of the 14th International Conference on Human-Computer Interaction*, 2011; [Schlegel and Keller, 2011]
- R. Kühn, C. Keller, and T. Schlegel. A Context Taxonomy Supporting Public System Design. In *Proceedings of the 1st International Workshop on Model-based Interactive Ubiquitous Systems*, Pisa, Italy, 2011; [Kühn et al., 2011]
- C. Keller, S. Brunk, and T. Schlegel. Introducing the Public Transport Domain to the Web of Data. In B. Benatallah, A. Bestavros, Y. Manolopoulos, A. Vakali, and Y. Zhang, editors, *Web Information Systems Engineering – WISE 2014*, pages 521–530, Cham, 2014a. Springer International Publishing; [Keller et al., 2014a]
- C. Keller, R. Pöhland, S. Brunk, and T. Schlegel. An Adaptive Semantic Mobile Application for Individual Touristic Exploration. In M. Kurosu, editor, *Human-Computer Interaction. Applications and Services*, page 434–443, Cham, 2014b. Springer International Publishing; [Keller et al., 2014b]
- C. Keller and T. Schlegel. Model Based and Service Oriented Interaction for Ubiquitous Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp ’16, page 429–434, New York, NY, USA, 2016. Association for Computing Machinery; [Keller and Schlegel, 2016]
- C. Keller and T. Schlegel. How to get in Touch with the Passenger: Context-Aware Choices of Output Modality in Smart Public Transport. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers*, UbiComp/ISWC ’19 Adjunct, New York, NY, USA, 2019. ACM; [Keller and Schlegel, 2019]
- C. Keller, W. Titov, and T. Schlegel. A Passenger Context Model for Adaptive Passenger Information in Public Transport. In H. Krömker, editor, *HCI in Mobility, Transport, and Automotive Systems. Driving Behavior, Urban and Smart Mobility*, pages 238–248, Cham, 2020. Springer International Publishing; [Keller et al., 2020]

6.1 Context Analysis for Adaptive and Smart Public Transport

As described in section 2.4, context is very diverse. For any type of context-aware system, relevant context must be carefully analyzed and modeled. This section describes the analysis of relevant context factors for the context of use in adaptive and smart public transport systems. The goal of this analysis is to identify dimensions of context and context factors that form the context of use of passenger information output to enable a usability assessment for output options. The context analysis is based on my work in the research projects "IP-KOM-ÖV"[1] and "SmartMMI"[2].

6.1.1 Context Dimensions and Context Ontologies

As a first step in this context analysis, a top-down analysis of context dimensions structures context into several categories and led to the development of a context taxonomy. As described in section 2.4, early context-aware systems focused on location context [Want et al., 1992; Schilit and Theimer, 1994]. For mobility systems, location information is of course a relevant context dimension [Tumas and Ricci, 2009; Ferris et al., 2010]. Schmidt et al. [1999] discuss the extension of location context by environmental context and device context. The dimension of time is discussed, for example by Chen and Kotz [2000] and time is also highly relevant to smart and ubiquitous mobility systems [Cheverst et al., 2000; Tumas and Ricci, 2009]. Dey and Abowd [2000] introduce the user's identity and activity as additional, primary context dimensions. Considering the context of use that is relevant to assess the usability of an adaptive user interface, Calvary et al. [2005] refer to environment, user and platform as important aspects in their work on plastic user interfaces. By platform they understand the hardware and software configurations of devices. They include physical and social usage conditions in their conception of environmental context.

In our work on both research projects IP-KOM-ÖV and SmartMMI, we used these primary context dimensions as a starting point and detailed them into a taxonomy. The taxonomy based on these dimensions was developed in several iterations (cf. [Schlegel and Keller, 2011; Kühn et al., 2011] and [Keller et al., 2020]). Figure 6.1 shows the context dimensions we derived.

---

[1] "IP-KOM-ÖV: Internet Protokoll basierte Kommunikationsdienste im Öffentlichen Verkehr", `http://ip-kom.net`, last accessed October 8th, 2022

[2] "SmartMMI: Modell- und kontextbasierte Mobilitätsinformation auf Smart public displays und Mobilgeräten im Öffentlichen Verkehr", `http://smartmmi.de/`, last accessed October 8th, 2022

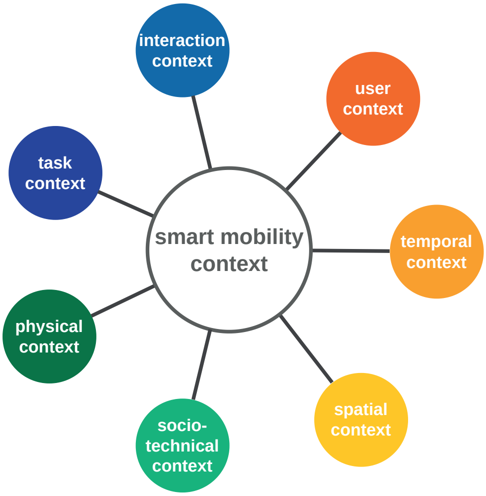


Figure 6.1 — Context dimensions for smart ubiquitous mobility systems, after [Schlegel and Keller, 2011] and [Kühn et al., 2011]

Task Context: The task context is related to user activities, described by Dey and Abowd [2000] as a primary context dimension. Another example of task context is used in the AURA system, by Garlan et al. [2004] to achieve task-based self-adaptation. The authors argue that the user's task must be made explicit in order to understand user goals and choose adaptation according to these goals. In mobility, the assessment of the context of use of information output can vary, depending on the user's task. If the user is driving a vehicle, auditory output is preferred over visual output, for example. Scenario 4 in section 3.3 describes such a situation. A smart mobility system should therefore consider *mobility tasks*, but *general tasks*, such as shopping for shoes (scenario 3.3) or packing bags (scenario 3.3) could also be considered and modeled for task context. In this work, I focused on mobility tasks.

Physical Context: Physical context comprises context facts that can be measured physically and context facts inferred from those. Schmidt et al. [1999] mention lighting conditions as a relevant physical context and Preuveneers et al. [2004] discuss physical context as "environmental conditions". In many cases, physical context is considered as part of the environment, for example by Schilit et al. [1994]; Dey and Abowd [2000] and Calvary et al. [2005]. However, the definitions of environmental context often also include social context, sometimes also location aspects or time [Preuveneers et al., 2004]. In this work, physical context is modeled separately from social context and is considered as an separate context dimension.

Socio-Technical Context: A socio-technical context dimension includes socio-technical and operational details of mobility systems that may influence user interaction. The dimension is split in *operational context* and *social context*. Context factors from both categories can influence each other, but also have an independent impact on usability. This context dimension therefore includes the social context that is sometimes regarded as part of the environment in other approaches. Interacting in public spaces is greatly affected by social context, with regard to the amount of people in the vicinity and the device used. For public displays, several influencing factors have been reported by Russell et al. [2002]; Brignull and Rogers [2003]; Peltonen et al. [2008], ranging from grouping patterns to the honeypot effect, observed by Brignull and Rogers [2003]. Social context is, however, hard to detect. Eagle and Pentland [2006] have described a sensing system to infer social situations and relationships as context, using sensor data from mobile phones. The amount of data necessary to detect those situations raises privacy concerns, though.

Operational context covers context facts that relate to the operation of the public system, which is in this case mobility. Operational context also contributes to the characterization of situations in mobility and is built on a mobility domain model. It is split up into *individual transport context* and *public transport context* since operational circumstances greatly differ for individual transport and public transport. The individual transport category covers all context facts regarding individual transport modes. It includes the current state of and on roads and parking spaces, for example. Context facts that describe the situation in public transport are compiled in the public transport context category.

Spatial Context: In mobility, location is very important. This context dimension comprises all location information relevant to the context of use. This includes not only absolute location information, such as given by GPS information, for example, but also relative location information or distances. Depending on the situation, the distance of a passenger from a public display can be highly relevant, for instance.

Temporal Context: Similarly, time is also relevant in mobility and can be expressed as an absolute time, a time of day or relative to a given point in time by a schedule, for example.

User Context: This context dimension aggregates facts about the user. User models have been researched in many different areas for different purposes. As a part of a context-aware pervasive or ubiquitous systems, personal or user information often includes the user's activities, location as well as information about their role, identity and authorizations [Henricksen et al., 2002; Chen et al., 2004]. Other works also contain the user's cognitive state or emotional state [Schmidt et al., 1999; Jameson, 2001]. Often, the user's preferences are included in user context, particularly for personalization [Cheverst et al., 2000; Chen et al., 2004; Strimpakou et al., 2006; Skillen et al., 2013]. Activities, location and social situation are context facts that our taxonomy includes in task context, spatial context and social-technical context. The user context model can refer to the ontologies that describe these dimensions to express a user's tasks or location. In our user context dimension, we capture additional information related to the user, such as *user preferences* and *user requirements*, including user abilities, for example. User specific *mobility context* and *journey context* are also part of the user context. Mobility context captures general mobility information about a user, whereas journey context is related to a specific journey. Our context dimension also details *user device context*, which can be used to model personal devices of a user, for example.

Interaction Context: Regarding the context of use, various interaction facts are highly relevant. The dimension of interaction context therefore covers context that shapes interaction between user and system. It includes *cognitive context*, *input context* and *output context*. Cognitive context covers the user's attention, as is sometimes described as part of user context [Schmidt et al., 1999; Garlan et al., 2002; Ho and Intille, 2005; Mathur et al., 2016]. It can also contain other cognitive factors that influence interaction [Zhou et al., 2007]. Input and output context cover input and output devices. This dimension also formalizes context facts to describe interaction abilities and preferences users can have that shape interaction. This part of the interaction context dimension is comparable to the platform context described by Calvary et al. [2005] and the device context, that is often defined as an extra context dimension [Schmidt et al., 1999; Chen et al., 2004; Preuveneers et al., 2004].

This context taxonomy was complemented using a bottom-up context analysis of context facts for public transport to detail each context dimension and provide a basis for the development of context ontologies modeling each dimension.

6.1.2 Analysis of Context Factors in Smart Ubiquitous Mobility Systems

Based on the identified context dimensions, a bottom-up analysis was used to understand each context dimension. In this bottom-up analysis, detailed context facts for each dimension were identified and an internal structure for each context dimension was derived. The bottom-up analysis was based on the analysis of scenarios described in section 3.3, but also scenarios that were developed in the research projects "IP-KOM-ÖV"[3] and "SmartMMI"[4] (cf. [Radermacher et al., 2012; Keller et al., 2018]). Context facts identified in these scenarios were gathered and then categorized in the developed context dimensions shown in figure 6.1.

Another component of this analysis was a hierarchical task analysis of the tasks identified in section 3.2 and a context analysis of the resulting detailed tasks. Goal of this task analysis was to support the identification of context factors for all context dimensions as well as to form a basis for the task model described in section 6.3.

Hierarchical Task Analysis for Public Transport Tasks

In section 3.2 situations and tasks in mobility were determined. For each of the identified user tasks, I then performed a task analysis, identifying subtasks, goals and plans that describe the relationship between subtasks.

The hierarchical task analysis was developed as an instrument for ergonomics by Annett and Duncan [1967] to describe and analyze tasks in industrial training, focusing on cognitive tasks. Since then, the method has been applied to many domains, including human-computer interaction and interaction design [Shepherd, 2001; Stanton, 2006; Diaper, 2004]. In a hierarchical task analysis, a task is associated with one or several goals. The analysis includes the identifcation of parameters or events that indicate that a goal is attained. They form conditions for goal attainment, also called feedback [Annett, 2003].

A top-level task is then decomposed into several sub-tasks, or operations, with each of them having one or several sub-goals. A sub-task or operation includes actions that a user must take to attain the goal. An hierarchical task analysis includes the development of plans that define the order of sub-tasks and allow

---

[3] "IP-KOM-ÖV: Internet Protokoll basierte Kommunikationsdienste im Öffentlichen Verkehr", `http://ip-kom.net`, last accessed October 8th, 2022

[4] "SmartMMI: Modell- und kontextbasierte Mobilitätsinformation auf Smart public displays und Mobilgeräten im Öffentlichen Verkehr", `http://smartmmi.de/`, last accessed October 8th, 2022

to describe parallel tasks or selection rules. The result of an hierarchical task analysis is displayed in a diagram or a decomposition table. When plans are displayed in a diagram, some characters are used to signal the order of sub-task execution:

- > indicates a sequence of operations
- / indicates either/or decisions
- + indicates parallel operations
- : indicates operations, where timing or order is not critical

The following paragraphs show only the diagrams of the hierarchical task analysis, the tables are listed in the appendix in section A.1.

Tasks and operations identified in a hierarchical task analysis can be tasks performed by humans or by machines [Stanton, 2006]. The so-called stop rules describe on what level of detail the analysis is concluded. This analysis stops at a level of device-independent tasks. Some subtasks that are only concerned with physical objects do not require any digital data or services. Such subtasks can not be supported by intelligent systems and have been omitted. An example for such a task is the adjustment of a rear-view mirror when boarding a car. Tasks that are not directly derived from the itinerary of the trip are also omitted in this work. Examples for such tasks are shopping or stowing and retrieving luggage in a vehicle.

This section only details the result of the hierarchical task analysis of tasks in public transport. The analysis results of tasks using different transport modes are listed in the appendix in section A.1. The numbering of the tasks was determined for all tasks, as shown in figure 3.4, which is why the public transport tasks are not consecutively numbered.

T1: Boarding a Public Transport Vehicle: Figure 6.2 shows all subtasks of the boarding task. The overall goal of boarding a public transport vehicle is that the user is in the right vehicle when it departs. Boarding a public transport vehicle begins with identifying the correct platform, if there are different options to choose from. If the station layout is not clear, the platform must first be located, before going there.

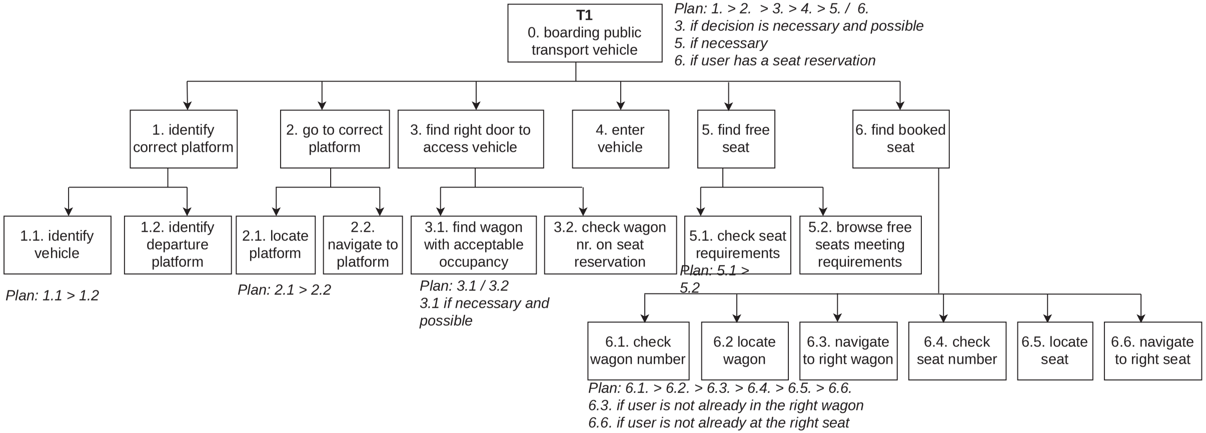


Figure 6.2 — Hierarchical task analysis for boarding a public transport vehicle.

On arrival of the vehicle, the user must find the right door to access it. In some vehicles, there is more than one option. The simplest strategy for the user is to choose the door next to them. In this case, the user skips subtask 3. Otherwise, subtask 3 details other options to choose a wagon and door to enter the vehicle. Once the user has entered the vehicle, they can keep standing or search for a seat. Passengers choose a seat based on their seat requirements, such as wheelchair spaces, for example. Some users have a seat reservation and search for their booked seat. This task has time constraints, since public transport vehicles depart according to the timetable.

T7: In Transit in a Public Transport Vehicle: Riding with a public transport vehicle means the user first has to identify the stop they need to alight at. They then have to monitor the next stops of the vehicle, in order to decide when to exit. Figure 6.3 shows the details of this task. This task has no special time constraints.

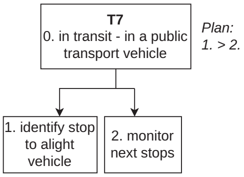


Figure 6.3 — Hierarchical task analysis for riding a public transport vehicle.

T11: Alighting a Public Transport Vehicle: Most public transport vehicles have several doors that can be used as exits. A user first locates the door they will exit, choosing the nearest door or, based on knowledge about the platform the vehicle will stop at, choosing a door close th the user's next destination, for example. The user then has to go to this exit and leave the vehicle when it has halted.

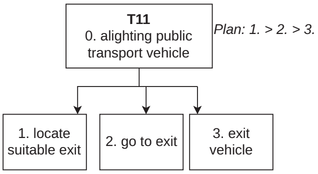


Figure 6.4 — Hierarchical task analysis for alighting a public transport vehicle.

This task is successfully executed if the user is outside of the vehicle and at the correct stop. Figure 6.4 shows details of this task. Alighting a vehicle is time critical, since the user needs to exit before the vehicle departs again.

T16: Wait at Station or Stop: If their public transport vehicle has not arrived yet, passengers wait. This task may occur after the user arrived at the platform and then is a subtask for "**T1:** boarding a public transport vehicle". The waiting task is detailed in figure A.15. The task is completed, if the vehicle arrived or if a delay or disruption occured and the user must deal with them.

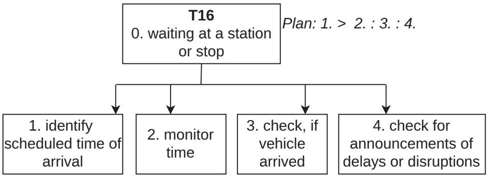


Figure 6.5 — Hierarchical task analysis for waiting in public transport.

In order to wait, a user has to know the scheduled time of arrival of the vehicle and to monitor, if the current time is near the scheduled time. If the vehicle did not arrive, but the scheduled time of arrival approached, the user checks for announcements of delays or disruptions as a cause. The waiting task's time constraints depend on the period of time until the vehicle arrives.

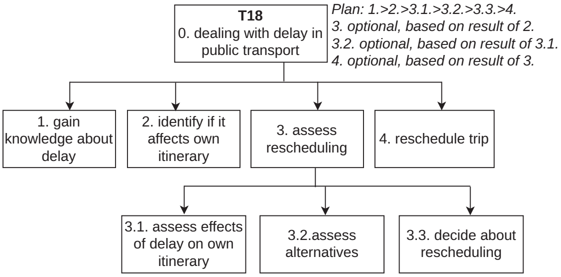


Figure 6.6 — Hierarchical task analysis for dealing with a delay in public transport.

T18: Dealing with a Delay in Public Transport: Dealing with a delay in public transport can have very different outcomes. There are delays that do not require the user to act at all, when no connection is affected, for example. However, if delays are longer or if connections can not be kept, the user can assess if a rescheduling is necessary. The task analysis, as shown in figure 6.6 details these steps. After becoming aware of a delay, the user identifies if their own itinerary is affected. If it is, the user evaluates rescheduling. They then decide if they need to reschedule. The evaluation of alternatives as well as the rescheduling require subtasks similar to ad hoc planning of a trip. These are not detailed further at this point. This task can have time constraints, based on the options for rescheduling.

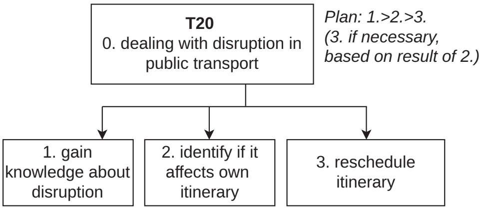


Figure 6.7 — Hierarchical task analysis for dealing with a disruption in public transport.

T20: Dealing with a Disruption in Public Transport: A disruption is an event that makes continuing with the current plan impossible. A train might get cancelled, for example. After gaining knowledge about a disruption, the user identifies if their itinerary or route is affected. If the itinerary of the user is affected, the user has to reschedule their itinerary. Figure 6.7 shows the details of this task. As for task **T18**, time constraints during this task depend on the options for rescheduling.

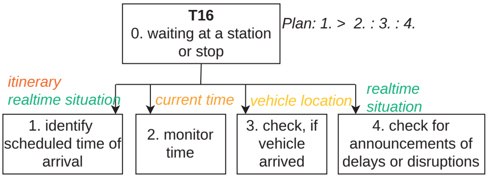


Figure 6.8 — Relevant context facts during the task "waiting in public transport".

Based on this task analysis, a systematic context analysis for each of the tasks followed. For each task and subtask, I analyzed, which context facts contribute to each task and subtask. The result of one of these is shown in figure 6.8. The rest is listed in the appendix in section A.1.22. All of the gathered context facts were categorized in the developed context dimensions for an informal context model shown in figure 6.9.

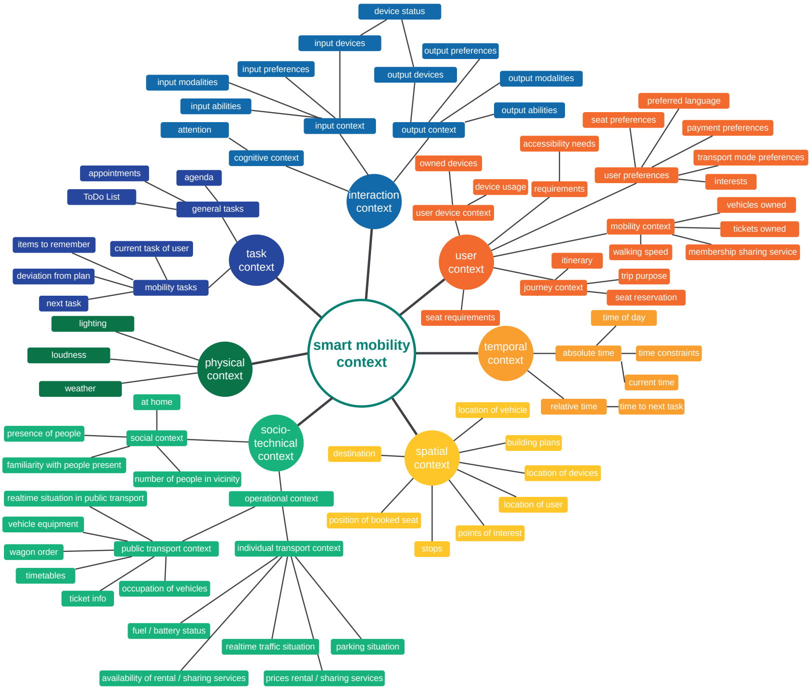


Figure 6.9 — Context factors in smart ubiquitous mobility.

The context facts were identified and categorized at this point without further analysis of context acquisition or reasoning possibilities. For several of these context facts, simple and accessible acquisition methods exist. The readings of the GPS sensor of the user's smartphone are a common indicator of the user's location, for example. Some context facts are subject of current research, for example the detection of the user's attention [Anderson et al., 2018; Pagliari et al., 2019]. For other context facts listed in figure 6.9, such sensing or deduction methods must still be developed, for example the detection of the user's familiarity with people that are nearby.

Based on this context analysis, the context ontologies that formalize the context facts and dimensions shown in figure 6.9 were modeled. I developed a number of distinct ontologies to facilitate their reuse and extension. Some of these ontologies cover one of the context dimensions and others merge two or more dimensions, depending on their application in the usability assessment framework. The level of detail of these ontologies is tailored to their usage in the usability assessment framework presented in chapter 7 and its application in the proof of concept prototype described in section 9.1. The level of detail was also guided by wether it would be possible to detect the represented context facts in an implementation. All ontologies are extendable or even replaceable, should the framework be applied in a different domain. The ontologies provide the vocabulary to describe context in a public transport setting and are meant to be applied with the adaptation rules described in section 7.3.

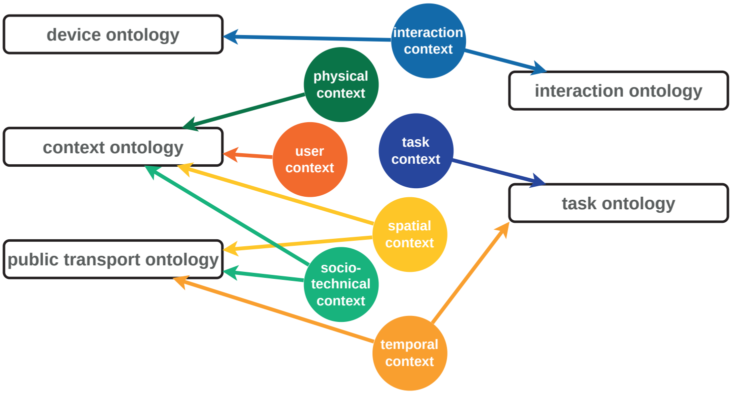


Figure 6.10 — Mapping of context dimensions to ontologies.

Figure 6.10 shows the mapping of context dimensions to ontologies. A *device ontology* models interaction-specific context facts about interaction devices, mapping a part of the interaction context dimension. The *interaction ontology* describes interaction context related to the user and facts about interaction that are needed to describe the context of use for usability assessment, such as input and output modalities. The *task ontology* models user tasks and uses concepts of temporal context to relate those tasks. The *public transport ontology* provides public transport terminology, using spatial and temporal concepts. Finally, a *context ontology* represents user context and uses spatial and physical context to detail the context of use. Social context as part of socio-technical context is also a part of this general context ontology. The context covered by this ontology is more volatile, while the other ontologies model more static vocabularies. Additionally, it integrates the other ontologies by providing properties that allow links between concepts of different ontologies. The ontology engineering and the resulting ontologies will be described in the remaining sections of this chapter.

6.2 Public Transport Ontology

The public transport ontology serves as a domain ontology, providing a vocabulary to describe mobility situations. I developed two ontologies that are used jointly for this purpose. The foundation for the domain ontology was developed in the the research and standardization project IP-KOM-ÖV[5], funded by the German Federal Ministry for Economic Affairs and Energy. I then extended this ontologies by additional concepts. Those additions are modeled in a second ontology.

One of the goals of IP-KOM-ÖV, was to develop a standard interface for passenger information resulting in the *Travellor's Realtime Information Advisory Standard* (TRIAS), an interface description that was subsequently standardized by the german association for public transport agencies (Verband deutscher Verkehrsunternehmen, VDV[6]) [VDV, 2014; Englert et al., 2019].

TRIAS describes, among other things, an interface and a data structure for passenger information requests. It is compatible with several european standards that are used in passenger information systems across Europe. Travellor's Realtime Information Advisory Standard (TRIAS) uses the communication structure of the SIRI standard, wich is the "Standard Interface for Real-Time Information" by CEN, the European Committee for Standardization[7] [CEN, 2015a,b,c, 2011, 2016]. Additionally, the Transmodel standard, which is a "Reference Data Model For Public Transport" was a basis for the development of TRIAS, [CEN, 2006]. TRIAS is also based on Network Timetable Exchange standard (NeTEx), [CEN, 2014a,b, 2015d] and Identification of Fixed Objects in Public Transport (IFOPT) [CEN, 2012].

During this project, I developed an ontology compatible with TRIAS and the standards TRIAS is based upon. This work inlcuded extracting a taxonomy of concepts from these interface standards as well as extracting and explicitly

---

[5] IP-KOM-ÖV: Internet Protokoll basierte Kommunikationsdienste im Öffentlichen Verkehr,`http://ip-kom.net`, last accessed October 8th, 2022

[6] Verband deutscher Verkehrsunternehmen, `https://www.vdv.de/`, last accessed October 12th, 2022

[7] `https://www.cen.eu`, last accessed October 12th, 2022

modeling the relations between these concepts [Keller et al., 2014a,c]. Alongside the core domain model, I also developed an ontology for points of interests, accessibility in public transport, for weather and an ontology that made it possible to model passenger context. These ontologies are named "Ontologien für Fahrgastinformation" (OFI) and were published by the VDV as a nonstandardized extension to the TRIAS standard for interested parties [Keller et al., 2014c].

I extended the OFI ontologies to a domain model for public transport, appplicable in an autonomous usability assessment. To maintain compatibility with TRIAS for the OFI ontologies, I modeled all extensions in a separate ontology, using a separate namespace. Some concepts of public transport are modeled differently from OFI for brevity and additional concepts are added. The resulting simplified public transport ontology that is used in this work will still be compatible with the commonly used features of passenger information systems that are based on TRIAS or are related to TRIAS.

The ontology a class or property belongs to is identifiable via its namespace. The namespaces of the public transport ontologies and the prefixes used in the following are defined in listing 6.1. Several concepts from these public transport ontologies will be presented in the following paragraphs.

```
PREFIX pt: <http://iums.eu/ontologies/publictransport#>
PREFIX ofi: <http://vdv.de/ofi/ontology#>
```

Listing 6.1 — Namespaces and prefixes of the public transport ontologies (SPARQL notation).

Geography and Location: Figure 6.11 shows the model for the spatial context dimension described in section 6.1.1. It models several classes and properties that can be used to describe locations in different forms and comprises spatial entities.

A class `Location` is an abstract description of a location and range for the property `ofi:hasLocation`. The domain of this property is not fixed, so that various entities can denote their location using this property. The original OFI ontology uses the GeoSPARQL standard to define locations via coordinates as a point or as polygon. However, for the purposes of this work, GPS coordinates can be defined for a location using the WSG84 RDF vocabulary that was proposed by the Semantic Web Interest Group of the W3C [Brickley, 2003]. A location can also be specified using an address, which is modeled by the `Address` class. It is

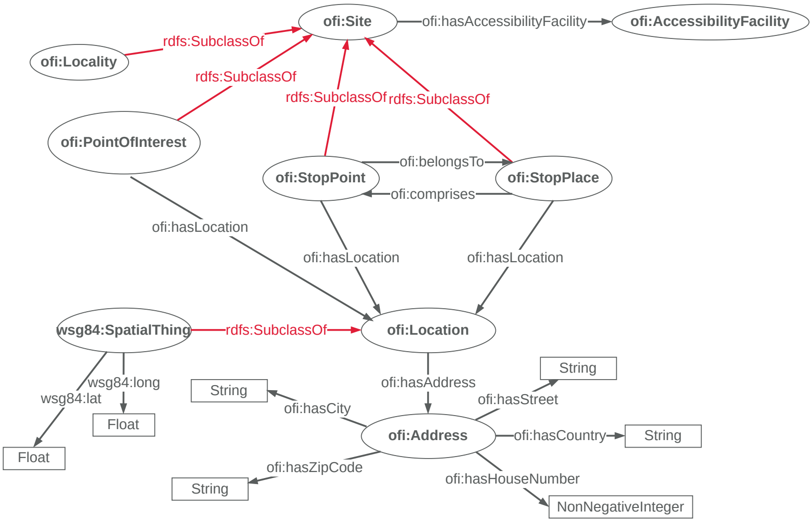


Figure 6.11 — Geographic and location concepts in the public transport ontology, own diagram after [Keller et al., 2014c].

possible that a location can have several addresses, several GPS coordinates or GPS coordinates and address combined.

The ontology also models several concepts for fixed locations. The `Sites` class describes abstract places. A site's location is specified by the property `hasLocation`. Since in public transport there are several types of stops, they are modeled as two sub-classes of `Site`. A `StopPlace` is a general stop or station. It can comprise several `StopPoints`, which is the case, for example, if there are a bus stop and a tram stop that belong to the same general stop. TRIAS implementations often only work with `StopPoints` and omit `StopPlaces`.

Trips: The trip-related part of the ontology is shown in figure 6.12. A trip in TRIAS and in the OFI ontology has a (planned) starting time and end time, modeled as a `DatatypeProperty` with an `xsd:dateTime` literal. Further characteristics that are directly modeled are the duration of a trip, the number of interchanges and the distance it covers. Destination and origin of a trip are, on an abstract level, modeled as `Sites`, since trips can start and end at every type of site or location. Trips can have `OperatingDays`. `OperatingDays` is a class that can model on which days and, if applicable, in what time period a trip is valid.

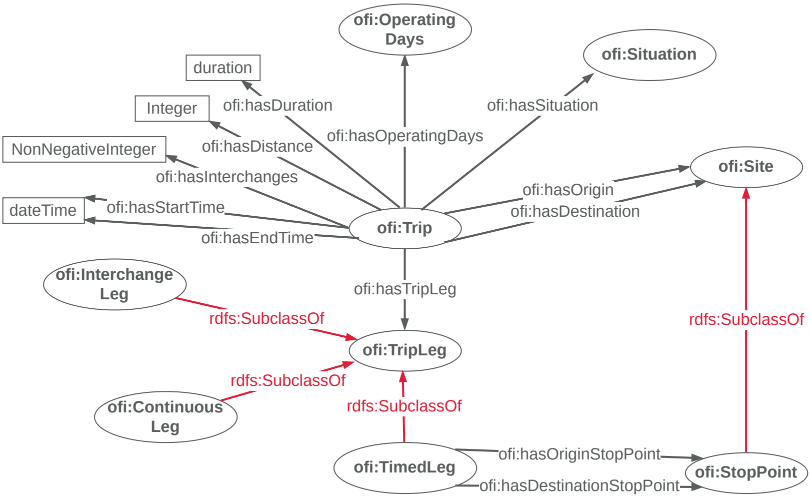


Figure 6.12 — Trip-related concepts in the public transport ontology, own diagram after [Keller et al., 2014c].

A `Trip` consists of one or more trip legs. The class `TripLeg` has several subclasses for different types of trip legs. The property `hasLegNumber` is used to specify a number for each trip leg that allows to construct their correct sequence in a trip.

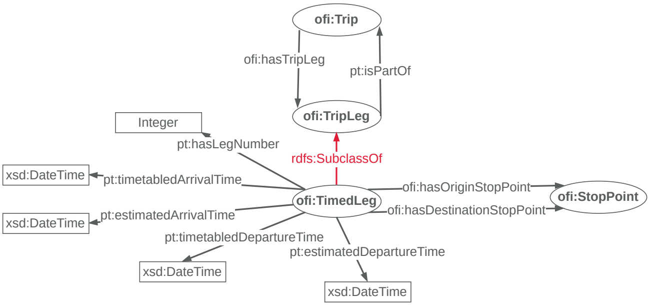


Figure 6.13 — Properties that are related to a timed trip leg.

`TimedLegs` are trip legs using a public transport vehicle. They therefore have a `StopPoint` as origin and as destination, as shown in figure 6.13. The departure time at the origin of a `TimedLeg` as well as the arrival time at the destination are modeled as `timetabledArrivalTime` and `timetabledDepartureTime`. In reality, a public transport service can deviate from these times. Realtime information is given in TRIAS as estimated times and therefore can be indicated as `estimatedArrivalTime` and `estimatedDepartureTime`.

`ContinuousLegs` describe trip legs on which another type of transport mode is used. Following the TRIAS specification, these can be walking, cycling, using a taxi, driving a car or getting a lift. The class `ContinuousLeg` is shown in figure 6.14. It has a duration (`hasDuration`) and a time window between the arrival at its origin and the departure at its destination. The properties `hasTimeWindowStart` and `hasTimeWindowEnd` can be used to specify this time window.

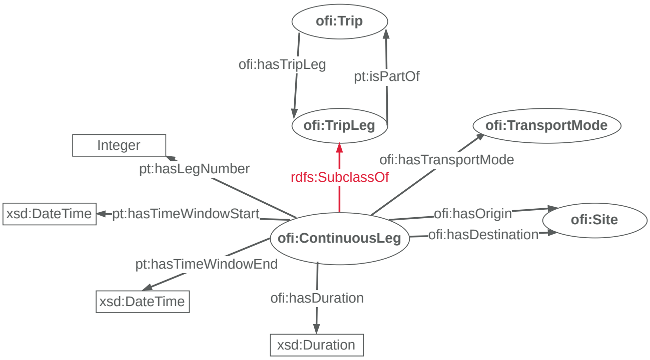


Figure 6.14 — Properties that are related to a continuous trip leg.

An `InterchangeLeg` models the interchange between two public transport trip legs. Its transport mode is walking. In addition to a duration, similar to a continuous leg, an interchange leg has a walking duration and a buffer time given. In passenger information systems that use TRIAS, the walking duration is computed using an average walking speed. The properties `hasWalkingDuration` and `hasBufferTime` are used to express these characteristics, see figure 6.15. Similar to a continuous leg, a time window is given for an interchange. The class `LegIntermediate` maps intermediate stops and legs in case a trip contains this additional information.

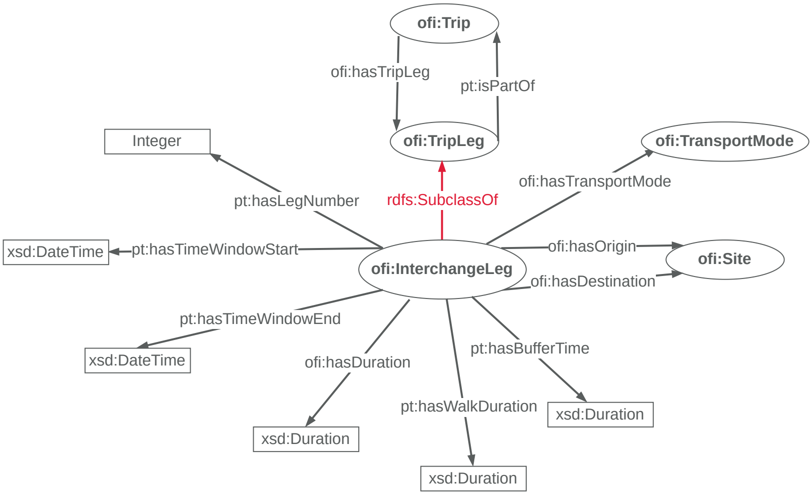


Figure 6.15 — Properties that are related to an interchange trip leg.

Stop Points and Stop Places: Stops are a central concept in public transport. The corresponding ontology classes are `StopPoint` and `StopPlace`, shown in detail in figures 6.17 and 6.16.

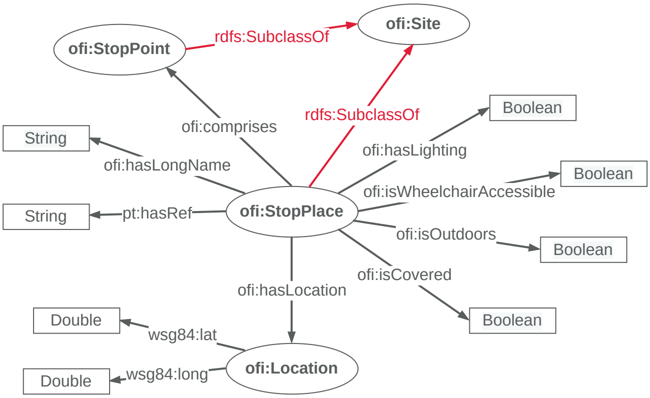


Figure 6.16 — The class `StopPlace` in the public transport ontology.

A stop place includes one or more stop points, indicated with the `comprises` and `belongsTo` properties. Besides these properties, individuals from both classes can have the same additional properties. Several datatype properties can be used to state wether a stop point or stop place are covered, outdoors, accessible for wheelchairs or have lighting.

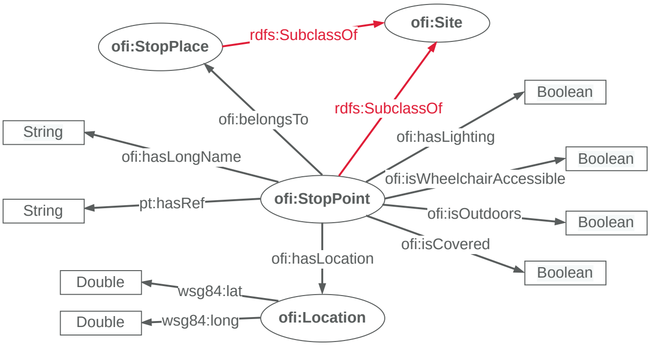


Figure 6.17 — The class `StopPoint` in the public transport ontology.

TRIAS uses a stop point name or stop place name that is human readable and can be specified using the `hasLongName` property. The unique reference is used for unambiguous identification of stops and can be specified as `hasRef`.

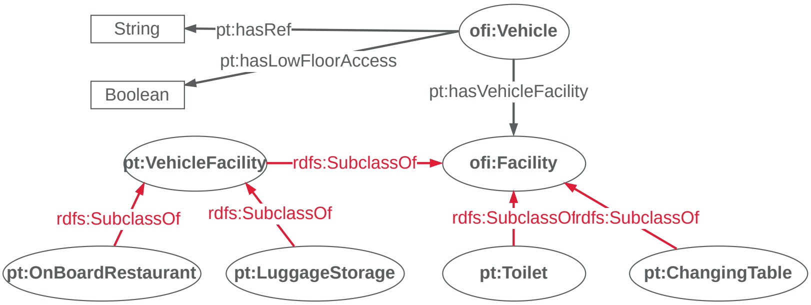


Figure 6.18 — The `Vehicle` class in the public transport ontologies.

Vehicle: The `Vehicle` class specifies vehicles of all kinds. The public transport ontologies allow the specification of several properties of a public transport vehicle, as shown in figure 6.18. A low floor vehicle provides easier access for wheelchair users or strollers. Information about a vehicle's equipment and facilities can be expressed using the `hasVehicleFacility` property. In figure 6.18, some facility classes relevant for vehicles are shown, more types of facilities are possible.

Transport Modes: Transport modes are modeled with a corresponding class. Two sub-classes structure transport modes: `IndividualTransportMode` and `PublicTransportMode`. For each, several specific sub-classes are defined. A specific transport mode has a fixed type of vehicle that is used in this transport mode. Vehicles are modeled in a class hierarchy of sub-classes of the `Vehicle` class. The possible vehicles for public transport and derived from these the corresponding transport modes are based on the modes known in TRIAS and include *buses*, *coaches*, *funiculars*, *metros*, *trains*, *telecabins* and *trams* [VDV, 2014; Englert et al., 2019]. The `ObjectProperty` `hasTransportMode` has the range `TransportMode` and can be used to model the transport mode of a trip leg, for example.

6.3 Task Ontology

The task ontology provides a structure to describe user tasks in mobility and specifically in public transport. The task ontology can be extended by additional tasks if the framework should be applied in another domain. It builds on the results of the task analysis in sections 3.2 and 6.1.2. Similarly to the hierarchical task analysis, this task ontology models mobility tasks on a device independent level. In contrast to task models that are used with model-based user interfaces, such as CTT models, this task ontology aims at modeling tasks for the characterization of a situation and the context of use in which the system needs to relay a message to the user and it does not aim at defining interactive tasks that are performed on a user interface [Paternò, 2003]. The level of detail therefore differs between the two modeling approaches. The namespace of this ontology is listed in listing 6.2.

```
PREFIX mt: <http://iums.eu/ontologies/mobilitytasks#>
```

Listing 6.2 — Namespace and prefix of the task ontology (SPARQL notation).

The basic taxonomy of tasks displayed in figure 6.19 is derived from user activities and tasks identified in the analysis in section 3.2. It models different types of tasks as sub-classes of a general `Task` class. As direct sub-classes of `Task`, two categorizing classes are defined, `RegularTask` and `DeviatingTask`.

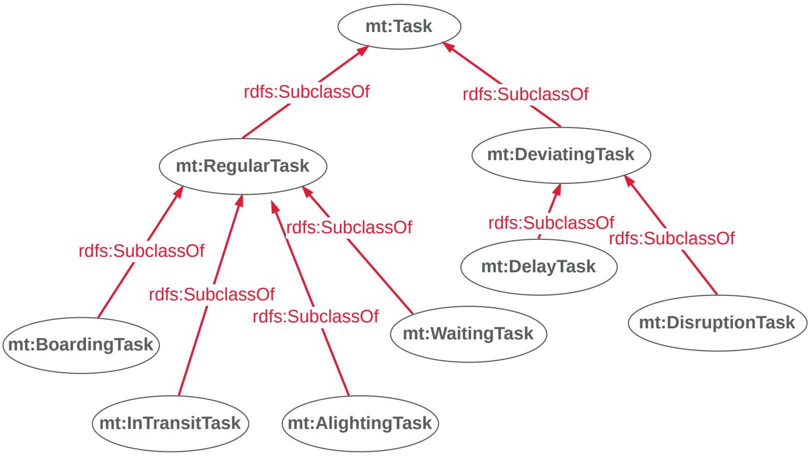


Figure 6.19 — The task taxonomy.

Regular tasks are tasks that can be part of a planned journey. The `DeviatingTask` class comprises tasks that deviate from a plan. Its sub-classes `DelayTask` and `DisruptionTask` therefore model delays and disruptions. Regular and Deviating tasks have different implications on the context of use and therefore can be handled differently.

Figure 6.20 shows how details of each type of task can be modeled. Tasks generally can have a planned start time and end time. They have a transport mode that is either used during the task or a mode that was used or will be used. In an application of the usability assessment framework that focuses not only on public transpot, this transport mode could be used to infer a level of attentiveness, for example by distinguishing individual transport modes from public transport modes.

The `BoardingTask`, `InTransitTask` and the `AlightingTask` tasks can reference a vehicle that is, was or will be involved in this task. These tasks also can reference a line, if public transport is used. They can also reference a `TripLeg` that this task belongs to. A `WaitingTask` references the `Site` or `Location` that the user waits at. An `AlightingTask` and a `BoardingTask` can both refer to the `Site` or, in public transport, the `StopPoint` at which the user either alights

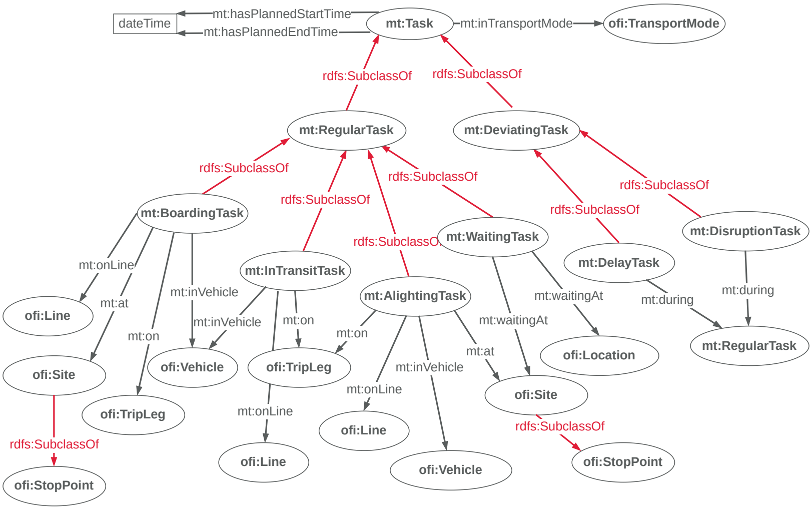


Figure 6.20 — Details of the task ontology.

or boards a vehicle. A `DelayTask` or `DisruptionTask` deviate from a specific regular task, which can be referenced using the `during` property.

6.4 Interaction Ontology

The interaction ontology describes characteristics of information output and provides the vocabulary for expressing and assessing characteristics of interaction modalities. It is closely linked with the device ontology described in section 6.5. The namespace of this interaction ontology is defined in listing 6.3.

```
PREFIX int: <http://iums.eu/ontologies/interaction#>
```

Listing 6.3 — Namespace and prefix of the interaction ontology (SPARQL notation).

The analysis described in section 6.1 identified input and output context. The ontology shown in this section will only detail output context since the framework building upon these ontologies will focus on information output. The cognitive context is not modeled at this moment, since attention recognition and management or the recognition of other cognitive states of the user are out of scope of this work. The ontology can be extended to incorporate cognitive context in future work. The user's abilities and preferences concerning output are modeled in the context ontology described in section 6.6, based on the vocabulary the interaction and device ontologies provide.

There are some approaches towards ontologies or models of interaction in the literature. Bachvarova et al. [2007] present an ontology for modality choice that models interaction modalities. The models they propose describe properties of modalities and the content that can be expressed in a modality. The authors distinguish between linguistic and analogous modalities, where linguistic modalities are concerned with textual representations and analogous modalities use images, for example. The authors structure these down to a very specific level, including classes of concrete information representation, such as scatterplots and lines as sub-classes of graphs, for example. These characteristics venture far into categories for information visualization and are too specific for the purpose of this work.

Clerckx et al. [2008] present an interaction environment ontology that is meant to be used with a task model in their model-based approach towards user interface distribution at runtime. It models modalities using a modality class with sub-classes for input modalities and output modalities. Concrete modalities are modeled as sub-classes to those two classes. These are differentiated based on modality such as direct manipulation or GUI, which the authors call the interaction language property of the modality. They also include types of devices referencing interaction resources. Examples are classes `MouseDirectManipulation` or `ProjectorGUI` [Clerckx et al., 2008]. The authors modeled interaction resources and interaction devices, where the former categorizes input or output resources, such as a mouse or projector and the latter models devices such as smartphones or desktop computers and are meant to aggregate interaction resources. For the application in this work, these concepts are too interleaved. Classes that merge device and modality properties are not necessary, but would even complicate a usability assessment, since characteristics of devices or modalities should be able to be evaluated independent of each other for portability of usability assessment rules.

In their work on unobtrusive self-adaptive interfaces, Gil and Pelechano [2017] present a taxonomy of interaction modalities focusing on mobile devices. They classify modalities in visual, auditory, haptic and radio. In these categories they included the type of content, for example text, sound or speech. The radio category comprises touching, pointing or scanning the environment with the device. The categories also include properties of these concepts, characterizing sounds as soft or loud, for example. Their model was used to specify obtrusiveness levels of modality combinations.

For my interaction ontology, I modeled classes and properties that describe interaction between users and a system aiming at supporting usability assessment. Modalities are modeled independently from devices, and these concepts are described in separate ontologies. Both ontologies are supposed to be used together to represent interactive devices and their capabilities.

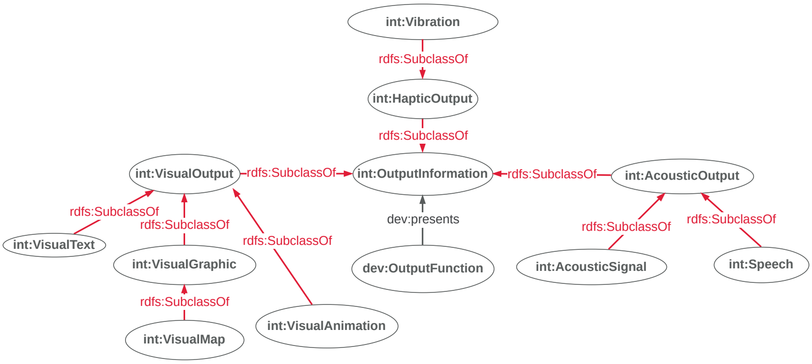


Figure 6.21 — Output classes for the description of modalities.

The core of this interaction ontology is the modality taxonomy shown in figure 6.21. The core concept is the class `OutputInformation`. Output information can be presented by devices and is categorized in several sub-classes differentiated primarily based on modality. Output modalities that are commonly in use are visual outputs, acoustic outputs and haptic outputs, mostly using vibration, for example in smartphones.

Specified are therefore `HapticOutput`, `AcousticOutput` and `VisualOutput` as main sub-classes of `OutputInformation`. The sub-classes of these basic output classes further specify the modalities comparable to the work of Gil and Pelechano [2017]. The more specific sub-classes are `Speech` and `AcousticSignal` as sub-classes of `AcousticOutput`, `Vibration` as sub-class of `HapticOutput` and `VisualText`, `VisualGraphic` and `VisualAnimation` as sub-classes of `VisualOutput`. Based on the level of detail that is needed for the rules applied in the usability assessment, these classes can be further detailed, for example following the work of Bachvarova et al. [2007], incorporating specific types of presentation. As an example, figure 6.21 includes the class `VisualMap` as a sub-class of `VisualGraphic`. A representation of visual graphic information on a map is a common use case in mobility systems and this class can be used to specify usability constraints for information displayed on a map.

6.5 Device Ontology

The device ontology describes interactive devices, focusing on output devices. It can be used to model device characteristics that influence the usability of utilizing these devices for passenger information output. The context ontology described in section 6.6 contains classes and properties to describe device context. In contrast to the device ontology, the context ontology models volatile facts about devices. The namespace of the device ontology is defined in listing 6.4.

```
PREFIX dev: <http://iums.eu/ontologies/device#>
```

Listing 6.4 — Namespace and prefix of the device ontology (SPARQL notation).

The device ontology consists of a rough device taxonomy, shown in figure 6.22. The taxonomy shown only maps the devices that were identified in the analysis of public transport systems and the usage scenarios in chapter 3. It can be extended by any additional device type that may be needed in a target system. The `Device` class is the basic class describing devices. Its sub-classes are differentiated based on properties that are relevant for a usability assessment. The class `PersonalDevice` contains devices that belong to a person and therefore are not public. Sub-classes of `PersonalDevice` are `Smartphone`, `Headphones` and `Smartwatch`. Personal devices can, in combination with a suitable modality, achieve higher privacy than public devices, for example.

The taxonomy also defines several types of displays, all sub-classes of the `Display` class. The `PublicDisplay` class models public displays, while the `VehicleDisplay` class specifies displays that are installed in vehicles. The taxonomy differentiates between `VehicleInformationDisplays` and `SmartWindows`. Vehicle information displays are not interactive and often mounted at the ceiling of a vehicle. A Smart Window is a display that is interactive and is built into the vehicle instead of a window.

Figure 6.23 shows several properties that express device characteristics. The boolean datatype property `isPublic` can be used to specify a device as a public or as a private device. The property `isInteractive` describes if a display is interactive and the `maxNumberUsers` property can specify if a device is a multi-user device.

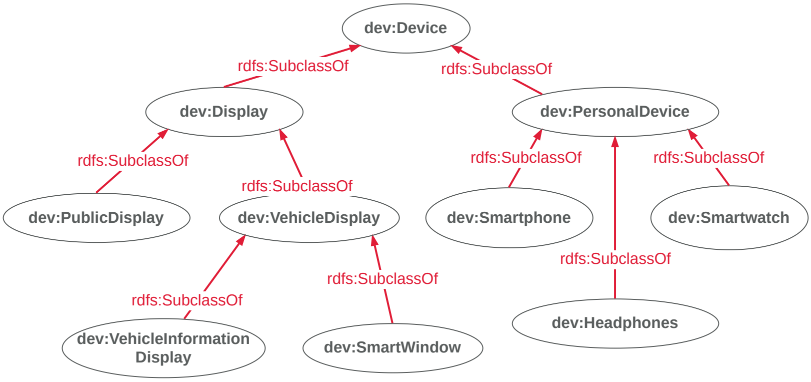


Figure 6.22 — The device taxonomy.

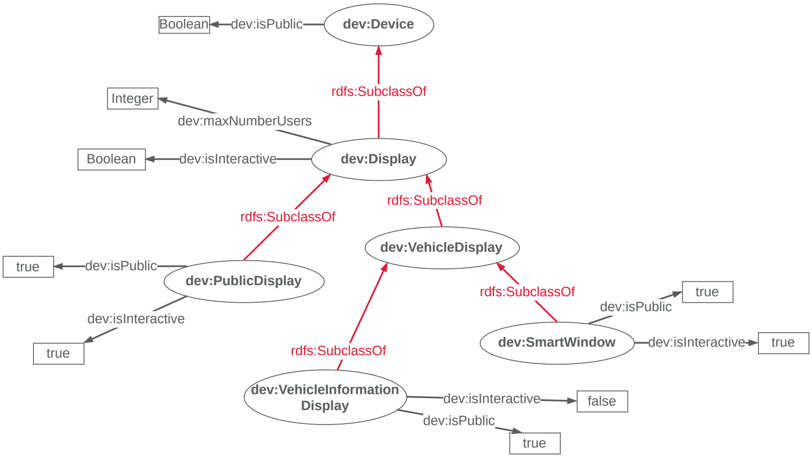


Figure 6.23 — The classes for displays and their properties.

In order to describe device features, the `providesFunction` property can be used. Figure 6.24 shows, how it can be used to model device functionality. A `Function` can have one or several `TechnicalPropertys`. One example is shown in figure 6.24. A technical property of a smartwatch can be that acoustic output only works with headphones, which could be expressed as an instantiation of the `TechnicalProperty` using the `onlyWorksWithHeadphones` property. Other

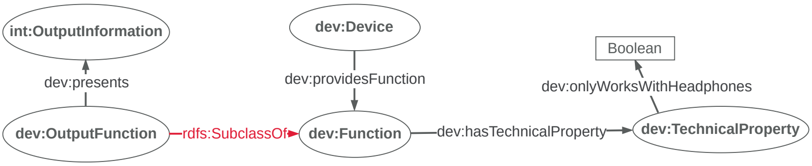


Figure 6.24 — Functions of devices.

examples for technical properties are screen sizes or microphone loudness. The class `OutputFunction` is a sub-class of `Function` and represents functions that present information to the user. Such a function can be specified using the `presents` property. Its range is the class `OutputInformation` of the interaction ontology. Using this model, it can be expressed that a device has the function to present visual graphic output, for example.

The device ontology also models converters that are able to convert output data from one format to the other, as shown in figure 6.25. If a converter is present in the system, output information can be presented in an another modality and a converter can therefore extend the output possibilities of a system. A converter is defined by specifying the input information and the output information, both as instantiations of the `OutputInformation` class or sub-classes. If needed, this model can be extended by file formats to further specify converters and to map their capabilities. This level of detail is not covered in this work and depends on the setup of the target system.

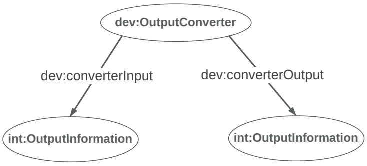


Figure 6.25 — An output converter that converts output information.

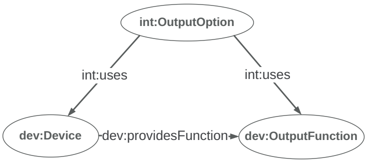


Figure 6.26 — An output option is one option for a system to present information to the user.

Using the concepts of the interaction and device ontologies combined, output properties of devices can be described so that a system can ascertain available output options. As defined in section 5.1, an output option is the combination of a device and a modality. An output option specifies one possibility for a system to present information to the user. It can be expressed using the `OutputOption` class shown in figure 6.26. An output option comprises a device and one or more of its output functions using the `uses` property. The output function specifies the modality or, using several output functions, modalities that is or are used with this output option. Using an option with several modalities is indicated when using the combination of vibration and a text message on a smartphone, for example. Other multimodal options can also be modeled. In principle, multiple devices are also conceivable, altough such options and possible applications should be discussed in more detail in future work. Given several of these options, a system then can proceed to assess their applicability in a given situation.

6.6 Context Ontology

The context ontology integrates the other ontologies. It is specifically modeled to express volatile context facts about entities defined in other ontologies, describing one context of use in a specific situation. Additionally, it specifies static and dynamic user context. The ontology is extendable to include all types of context facts that may be relevant in a specific implementation of the usability assessment framework. For the context ontology the namespace and prefix as defined in listing 6.5 is used.

```
PREFIX context: <http://iums.eu/ontologies/context#>
```

Listing 6.5 — Namespace and prefix of the context ontology (SPARQL notation).

Currently, the context ontology contains concepts for three main types of context, including devices, user and vehicles. These parts of the model are described in the following paragraphs.

Device Context: Figure 6.27 shows general device context. Devices can have a location, using the `hasLocation` property and the `Location` class. The location can be further determined using GPS coordinates or an address, as specified in the `OFI` ontology. The location of a device can also be specified relatively, by indicating the `Site` it is located at. Public displays, in the scope of this work, are located at public transport `StopPoints`, for example. `VehicleInformationDisplays` or `SmartWindows` are located in a `Vehicle` which can be indicated using the `isInVehicle` property. The `isInVehicle` property and the `atSite` property both have inverse properties. These allow to infer available devices for a given site or vehicle. Another context fact for devices is, if their display is switched on, indicated by the `displayOn` property.

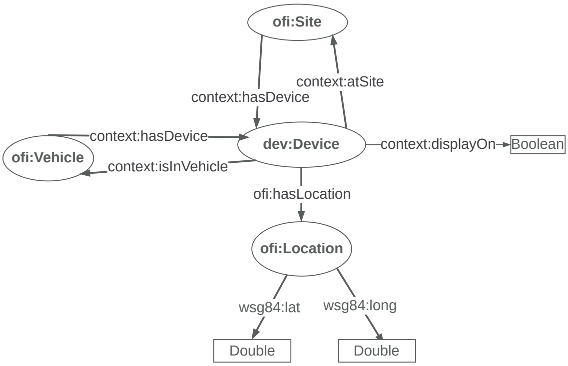


Figure 6.27 — Device context.

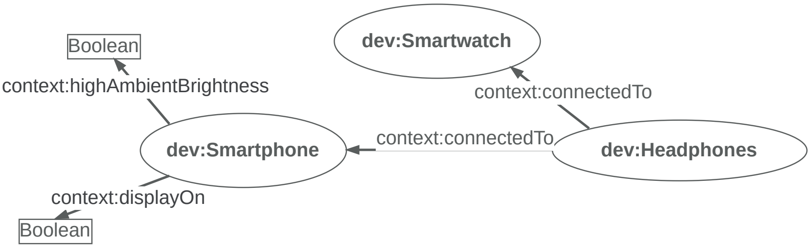


Figure 6.28 — Specific smartphone context.

Figure 6.28 shows more specific context facts for smartphones and smartwatches. The property `connectedTo` specifies if headphones are connected to a device, which can be a smartphone or a smartwatch. The display state of a smartphone can be stated using the `displayOn` property, as well. Values of smartphone sensors can be important to infer higher level context and can therefore be expressed using corresponding properties. For this work, the `highAmbientBrightness` property is used to indicate a categorization of high or low ambient brightness, as an example for the usage of sensor data from devices.

Vehicle Context: The context model also defines several properties and classes that describe the context of a vehicle, as shown in figure 6.29. The location of a vehicle is indicated using the `hasLocation` property of the `OFI` ontology. Additionally, a vehicle has an occupation that can be given as a percentage of occupied seats, using the `hasOccupation` property. The vehicle's next stop

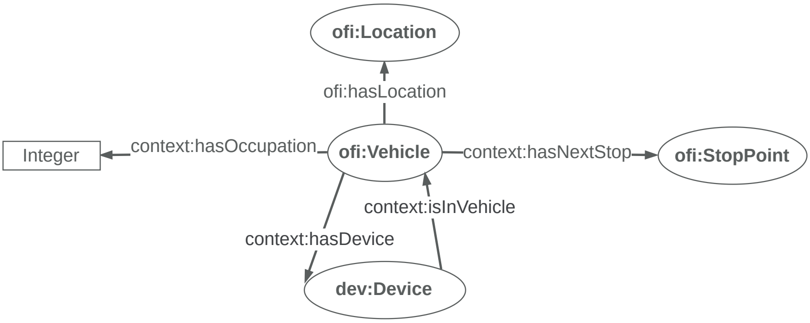


Figure 6.29 — Vehicle context.

can be specified using the `hasNextStop` property. As described above, the `hasDevice` property is used to refer to interactive devices that are built in a vehicle.

User Context: The user context comprises several context facts that change infrequently as well as context that is updated regularly. The static user context is shown in figure 6.30. A user's first and last name are specified as an example of personal information that might be useful in a smart mobility system. Several other details are conceivable but in this work, they are out of scope. Additionally, the user's interaction preferences and abilities are expressed as context data. Comparable to the work of Casas et al. [2008], the model specifies the interaction capabilities of a person rather than explicitly modeling disabilities. Using the property `isAbleToSense`, it can be specified which types out `OutputInformation` a user is able to perceive. The property `prefersToSense` specifies a user's preferences for output modalities.

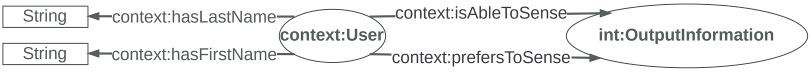


Figure 6.30 — Static user context.

The modeling of location and task context is shown in figure 6.31. The user's current location can be expressed using the `Location` class, but it can also be specified relatively, using the `isInVehicle` property. A piece of rather static context is the user's home that is specified as a `Location` object, assigned to the user with the `hasHome` property. This expresses a type of social context that can be used to assess the appropriateness of output options. Other social contexts can be complemented in this ontology in future work.

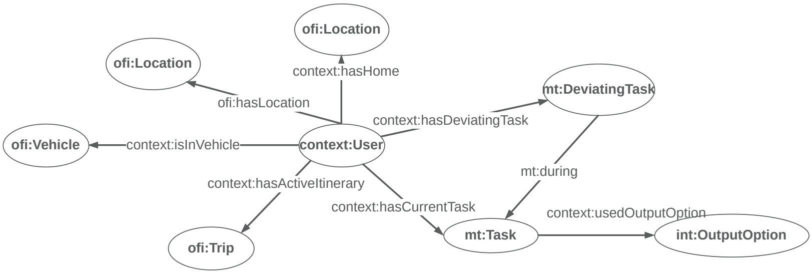


Figure 6.31 — User location and task context.

The user's active itinerary is given as a `Trip`, assigned with the `hasActiveItinerary` property. Additionally, using the `hasCurrentTask` property, the current task of the user can be specified. If a `CurrentTask` is given, an output option that was used during this task can be stored in context data, as well. A `DeviatingTask` can be assigned additionally, using `hasDeviatingTask`. The properties are modeled as specific properties to facilitate queries to the semantic data and to limit reasoning efforts.

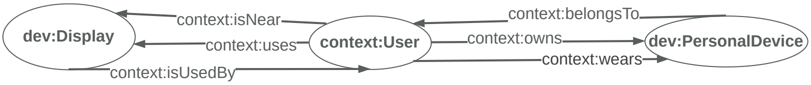


Figure 6.32 — Device user context.

Figure 6.32 shows device related context facts for a user. If a user is detected to be near a `Display`, the property `isNear` can express that fact. The inverse properties `uses` and `isUsedBy` describe the users that currently use a display. A display can, for example, identify a user if their smartphone is connected to the display. Anonymous users can be detected by the display if interaction events are being triggered, for instance. The personal devices a user owns can be specified by using the inverse properties `owns` and `belongsTo`. Smartwatches and headphones are personal devices that also can be worn by the user, expressed by the `wears` property. This further specification is needed, because device can be known to belong to a user, but if the user does not wear it, it should not be used for output, for example.

The ontologies provided in this chapter are the foundation for the usability assessment framework described in the following chapter. Since the ontologies are modeled as OWL ontologies, they can be easily extended by additional classes and properties, if further concepts are needed. They can be used to model the context of use in public transport situations and subsequently to assess the usability of output options in such a situation using the models and rules described in chapter 7.

CHAPTER 7

Autonomous Usability Assessment and Output Adaptation

This chapter introduces a conceptual framework for autonomous usability assessment and adaptation of information output in smart ubiquitous mobility systems. Implemented for a specific target system, this framework can analyze current context data and assess available output options for a given user and a given message with information that should be delivered to this user. It then returns a specified number of output options that were established as suitable and usable for the known context. The framework relies on context data expressed using the ontologies described in chapter 6. This context data is used to include knowledge about a user's situation - the context of use - in the usability assessment process. The context data also expresses knowledge about several parameters that influence the usability assessment, such as device characteristics and interaction modalities.

Apart from this knowledge, the framework needs an understanding of usability criteria and rules that shape the usability assessment. Therefore, a systematic analysis of usability attributes and criteria is a basis for the modeling of usability knowledge and is described in section 7.1. A message specification ontology is used to represent details about the message that are relevant to the usability assessment. It is described in section 7.2.1. The remainder of section 7.2 describes the usability ontology. It models usability criteria and assessment categories for a usability assessment of output options. Assessment rules are used to rate output options and the message based on the modeled usability criteria and context data. A set of such rules is described in section 7.3. Depending on the implementation of the framework, this set of rules can be extended or altered to map requirements of the target domain and system. Section 7.4 presents the assessment and decision process. In this process, the assessment rules, context data and message input are used to identify available output options and to rate them. Based on the rating results, the requested number of output options are chosen as a result.

This chapter is in part based on the following publications:

- C. Keller, M. Korzetz, R. Kühn, and T. Schlegel. Nutzerorientierte Visualisierung von Fahrplaninformationen auf mobilen Geräten im öffentlichen Verkehr. In M. Eibl, editor, *Mensch & Computer 2011: überMEDIEN | ÜBERmorgen*, page 59–68, Chemnitz, Germany, 2011. Oldenbourg-Verlag; [Keller et al., 2011]
- C. Keller, R. Pöhland, S. Brunk, and T. Schlegel. An Adaptive Semantic Mobile Application for Individual Touristic Exploration. In M. Kurosu, editor, *Human-Computer Interaction. Applications and Services*, page 434–443, Cham, 2014b. Springer International Publishing; [Keller et al., 2014b]
- C. Keller and T. Schlegel. Model Based and Service Oriented Interaction for Ubiquitous Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, page 429–434, New York, NY, USA, 2016. Association for Computing Machinery; [Keller and Schlegel, 2016]
- C. Keller and T. Schlegel. How to get in Touch with the Passenger: Context-Aware Choices of Output Modality in Smart Public Transport. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers*, UbiComp/ISWC '19 Adjunct, New York, NY, USA, 2019. ACM; [Keller and Schlegel, 2019]
- C. Keller, W. Titov, S. Sawilla, and T. Schlegel. Evaluation of a smart public display in public transport. In *Mensch und Computer 2019 - Workshopband*, Bonn, 2019b. Gesellschaft für Informatik e.V; [Keller et al., 2019b]
- C. Keller, S. Struwe, W. Titov, and T. Schlegel. Understanding the Usefulness and Acceptance of Adaptivity in Smart Public Transport. In H. Krömker, editor, *HCI in Mobility, Transport, and Automotive Systems*, page 307–326, Cham, 2019a. Springer International Publishing; [Keller et al., 2019a]

- W. Titov, H. Tran, C. Keller, and T. Schlegel. A Multi-device Evaluation Approach of Passenger Information Systems in Smart Public Transport. In H. Krömker, editor, *HCI in Mobility, Transport, and Automotive Systems. Driving Behavior, Urban and Smart Mobility*, pages 340–358, Cham, 2020. Springer International Publishing; [Titov et al., 2020]

7.1 Analysis of Usability Attributes and Criteria for Ubiquitous Mobility Systems

This section addresses usability attributes and criteria that form the basis of an assessment framework to autonomously assess the usability of output options. This analysis targets research question 1. Its goal is to identify usability attributes relevant to ubiquitous mobility systems and to derive criteria and rules that can be applied by a system autonomously and at runtime, taking available context information into account. They form a basis for the usability ontology.

Figure 7.1 — Analysis process for a usability ontology and assessment rules.

The process for this analysis is shown in figure 7.1. In a first step, I reviewed and analyzed usability attributes from several works on software quality of ubiquitous systems, usability of ubiquitous systems and evaluation frameworks for ubiquitous systems. In order to identify attributes that can be modeled in a usability model and criteria that can be expressed in usability rules for autonomous usability assessment, I filtered these attributes in a second step, based on requirements derived from their intended application in the framework. The first and second steps are presented in section 7.1.1. In the third step of the analysis, I identified synonyms and semantic overlap, merged and clustered the attributes. For each cluster, representative attributes were chosen. In a fourth step, characteristics of message content, user context as well as devices and modality were identified that influence these usability attributes and can be used to form criteria and rules to assess these aspects. These analysis steps are described in section 7.1.2. Based on the results of step 3 and 4, the usability ontology and assessment rules were modeled in a fifth step. Step 5 and its results will be described in sections 7.2 and 7.3.

7.1.1 Literature Review and Filtering

Usability attributes described and applied in literature often imply several criteria for their evaluation and most metrics are not defined precisely. Concrete measurement functions are rare, as, for example Carvalho et al. [2017] determined during their literature review of quality charactaristics for ubiquitous systems. In a first step, I therefore reviewed 14 papers on usability attributes and extracted the attributes mentioned in these works. Table B.1 in the appendix shows the result of this review. In table 7.1 the attributes that are left after applying the filter step are displayed. The reviewed works focus on ubiquitous systems and their usability or the evaluation of such systems, with adaptive interaction, calm computing as well as the usability of mobile systems. These works were chosen to cover the key characteristics of ubiquitous mobility systems that are the focus of this work. Some of these papers contain literature reviews and merge charactaristics and criteria from several sources, for example the work of Santos et al. [2013] or Carvalho et al. [2017].

As a basis and a reference, I included the general usability attributes of Nielsen [1993b]. In his book on usability engineering, Nielsen [1993b] discusses the meaning and extent of usability and discusses five core usability attributes: *learnability*, *memorability*, *satisfaction*, *efficiency* and *errors*. In their work on ubiquitous evaluation areas, Scholtz and Consolvo [2004] consider usability attributes specifically for ubiquitous systems. The authors present them as part of a framework to evaluate ubiquitous applications. From the five core usability attributes they include *efficiency*, *satisfaction* and *application robustness*, which correlates to the *errors* attribute of Nielsen [1993b]. They add several more attributes and discuss metrics to measure these attributes. Riekki et al. [2004] focus on usability attributes to evaluate applications that follow the principle of calm computing [Weiser and Brown, 1996]. Attributes that address the calmness of applications aim at limiting distraction. In the result tables, *courtesy of interaction* by Riekki et al. [2004] and *distraction* by Scholtz and Consolvo [2004] are aggregated as *cognitive load*.

Seffah et al. [2006] present the Quality in Use Integrated Measurement (QUIM) model as a consolidated model for measuring usability. They build on several existing standards, including Nielsen [1993b] and introduce attributes like *accessibility* and *safety*. The authors present several measurable criteria and discuss which attributes can be measured using which criteria. It is notable that many of these criteria can only be evaluated after the execution of a software, such as *time behavior*, *fault-tolerance* or *resource utilization*. These are therefore not applicable to an autonomous usability assessment before the execution of a feature.

Ryu et al. [2006] provide a very comprehensible list of usability attributes targeting ubiquitous software and middleware. Each attribute or characteristic is discussed with evaluation questions and metrics that can be used to evaluate this attribute. In table B.1, I included the attribute *accountability* of Ryu et al. [2006] as *understandability* as Mantoro [2009] defined it. Both attributes *fault tolerance* and *fault recovery* by Ryu et al. [2006] are aggregated as the *errors* attribute. Ryu et al. [2006] list several contexts in separate awareness attributes that are combined as *context-awareness* in tables B.1 and 7.1. The attribute *security* is included in the attribute *safety*.

Kim et al. [2008] presented a usability evaluation framework specifically for ubiquitous computing devices, detailing several evaluation factors, while Kemp et al. [2008] defined heuristics for the evaluation of ubiquitous e-learning systems. They introduce the *visibility of system status* as a heuristic that is included in the *feedback* attribute. The authors also emphasize *recognition rather than recall*, which matches *memorability* in the result tables. Their attribute *aesthetic and minimalistic design* is divided into two attributes in the review, *appeal* and *visual clarity*. Kemp et al. [2008] also list *focus*, which I included in the *cognitive load* attribute. Mantoro [2009] presents quality metrics specifically for context-aware systems. They mention the system's ability to fulfil a need or assist the user in an effort as *assistance*, which is included as *utility* in tables B.1 and 7.1. The attributes *fault tolerance* and *recoverability* are both included in the *errors* attribute. Mantoro [2009] also differentiate *maintainability* into four separate attributes, but these are aggregated in the review results. The authors list *compliance* twice, regarding application standards and portability standards. Both are aggregated in the *compliance* attribute. The metrics *installability* and *replaceability* by Mantoro [2009] are omitted, since these factors are not relevant for this work.

Quigley [2010] defined seven key usability metrics for ubiquitous user interfaces. The metric *conciseness* is included in the *efficiency* attribute in table 7.1 and *invisibility* is categorized as *unobtrusiveness*. Lee and Yun [2012] discuss usability attributes and metrics for interactive and ubiquitous services. They introduce the attributes *search-ableness* to categorize the quality of an information search or search interface and *responsiveness*. The *interconnectivity* attribute is expressed in their work as *two-way communication* and *connectivity*.

Table 7.1 — A compilation of usability attributes for ubiquitous systems

Attribute	accessibility	accuracy	attention	awareness	cognitive load	consistency	context-awareness	context-sensitve timing	ease of use	effectiveness	efficiency	expressiveness	familiarity	feedback	impact and side effects	mental model match	personalization	predictability	privacy	relevancy of interaction	suitability	timeliness	transparency	trust	understandability	unobtrusiveness
Nielsen [1993b]	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Scholtz and Consolvo [2004]	-	●	●	-	●	-	-	-	-	●	●	-	-	-	●	●	-	-	●	-	-	-	-	●	-	-
Riekki et al. [2004]	-	-	-	-	●	-	-	●	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-
Seffah et al. [2006]	●	-	-	-	-	-	-	-	-	●	●	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-
Ryu et al. [2006]	-	●	-	●	-	-	●	-	-	-	●	-	-	●	-	-	●	●	●	-	●	-	●	-	●	●
Kim et al. [2008]	●	-	-	-	-	●	-	-	-	-	-	-	-	-	-	●	-	●	-	-	-	-	●	-	-	-
Kemp et al. [2008]	-	●	-	●	●	●	-	-	●	●	●	-	●	●	-	-	-	-	-	-	-	●	-	●	-	-
Mantoro [2009]	-	●	-	-	-	●	-	-	-	-	●	-	-	-	-	-	-	-	-	-	●	-	●	-	●	-
Quigley [2010]	-	-	-	-	-	-	-	-	●	-	●	●	-	-	-	-	-	-	-	-	-	-	●	-	-	●
Lee and Yun [2012]	-	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	●	-	-	-	-	-	-	-	-	-
Harrison et al. [2013]	-	-	-	-	●	-	-	-	-	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Santos et al. [2013]	-	-	-	●	●	-	●	-	●	●	●	-	●	-	-	-	-	-	●	-	-	-	●	-	-	-
Evers et al. [2014]	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-
Carvalho et al. [2017]	-	-	●	-	●	-	●	-	●	●	●	-	●	-	-	-	-	●	●	-	-	-	●	●	-	-

Harrison et al. [2013] present the PACMAD usability model (People At the Centre of Mobile Application Development) that focuses on mobile applications, while Santos et al. [2013] developed a quality model specifically for the evaluation of ubiquitous systems. In their work on user participation for self-adaptive applications, Evers et al. [2014] discuss three usability principles focusing on adaptive interactive systems. Carvalho et al. [2017] present a comprehensive overview over quality measures for interaction in ubiquitous systems that summarizes 186 quality characteristics in a literature review. They analyzed and filtered those characteristics and condensed them to a final set of 27 quality characteristics. Their attribute *acceptability* is listed in table B.1 as *adoption*.

In a second step, I filtered these attributes, based on the definitions or comments given by the respective authors. I used the following requirements to identify the attributes and metrics that are applicable in an autonomous usability assessment:

- they should concern usability and be applicable to output devices and modalities
- they should be applicable in the situation itself, not only retrospectively or at design time
- they should not concern the aesthetics, design or visual makeup of a graphical user interface

After applying the requirements towards the list of usability attributes of table B.1, a list of 26 usability charactaristics was left. These are shown in table 7.1.

7.1.2 Clustering of Criteria and Identification of Characteristics

In the third step of the analysis, clusters of related concepts were identified, as shown in figure 7.2. For each cluster, a suitable representative attribute was found, which is shown as framed in figure 7.2. For each attribute cluster, characteristics influencing these attributes were extracted from literature. The goal is to identify parameters and relations to structure the usability ontology as well as assessment rules that can be implemented in the usability assessment framework. The clusters, representative attributes as well as relevant parameters and relations will be discussed in the following paragraphs.

Privacy: *Trust* is often mentioned as a relevant usability attribute. It is closely linked with privacy [Seffah et al., 2006; Kemp et al., 2008; Carvalho et al.,

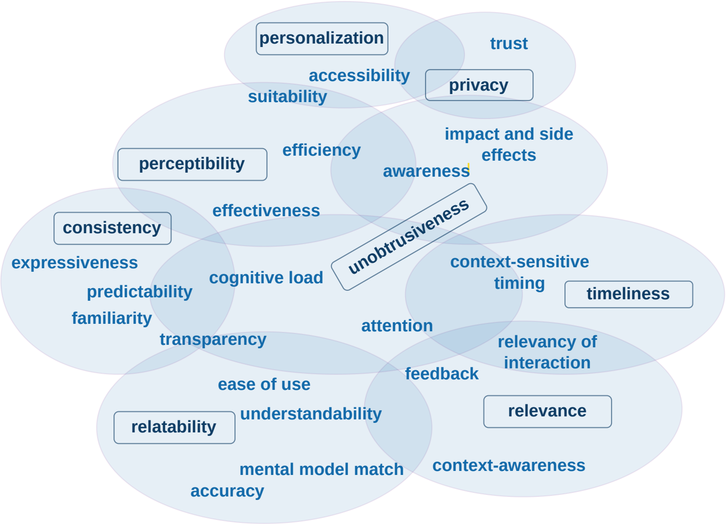


Figure 7.2 — Clustering of the resulting usability attributes.

2017]. A system should, for example, enable a user to see and change the data that is used by the system to foster the user's trust in it. Like in this example, trust often adresses issues that can not be decided in the situation directly and building trust takes time. It is a complex attribute, that is hard to implement in runtime decisions [Scholtz and Consolvo, 2004]. *Privacy* is often considered a relevant part of *trust*, Scholtz and Consolvo [2004] list *privacy* as a part of their ubiquitous evaluation area *trust*, for example. Unlike with trust, there are privacy concerns of output option choices that can be influenced in a context-aware usability assessment at runtime. As a privacy measure, they refer to the type and amount of information from a user that is revealed to the system, but also to other users. This is a concern in public systems, since messages to the user might contain private information and therefore should not be perceivable by third parties [Ryu et al., 2006]. Public output options would not be appropriate for such messages. This leads to the conclusion that the *sensitivity* of the message content is a relevant parameter to be considered in an assessment of privacy.

Personalization: Ryu et al. [2006] include *personalization* in their list of quality characteristics for ubiquitous systems. They write that a ubiquitous system should be able to provide personalized information to its users and should consider the user's *preferences*. Lee and Yun [2012] also include *personalization* and add that services should be customizable, which is implementable by user preferences, as well. The *suitability* attribute is part of this cluster. Ryu et al. [2006]; Mantoro [2009] discuss suitability as providing suitable functionality. In this cluster, this relates to personalization as a feature that can tailor functionality to the user. In a smart ubiquitous mobility system that delivers messages to users, suitability can mean that the system applies user prefrences to choose output options that are suitable for the user. *Accessibility* is also assigned to this cluster. It requires a ubiquitous system to provide services that are accessible to all customers, including persons with disabilities [Seffah et al., 2006; Kim et al., 2008]. Extensions to systems, such as braille interfaces or especially tailored information, for example using easy language, are not within the scope of this work. However, with regard to output options in smart ubiquitous mobility systems, certain aspects of accessibility can be seen as a type of personalization. The used models should provide means of defining user preferences but also *abilities* regarding output options. Abilities, in contrast to preferences, must be considered during adaptation, whereas preferences can be omitted, if necessary.

Perceptibility: *Effectiveness* and *efficiency* are core usability attributes and therefore referenced by almost all authors of the reviewed papers. Effectiveness is in most cases defined as the user being able to successfully complete their task. Many authors describe efficiency as enabling the user to complete their tasks in a short time [Ryu et al., 2006]. However, it can also be concerned with other types of resources [Seffah et al., 2006]. In the domain of ubiquitous mobility systems, the task of the user is one of the tasks described in the hierarchical task analysis in section 3.2. The effectiveness and efficiency of a system proactively informing passengers is therefore concerned with the delivery of messages to the user in a way that the user can perceive them at the right time and with adequate information to enable the user to fulfil the task at hand. This is why *perceptibility* was chosen as a representative attribute for this cluster. The messages should be noticeable by the user in order to be effective. A message that is perceptible at the right time, so that the user can still comfortably react to it, is efficient. This cluster also includes *suitability*, since a suitable output option is perceivable.

Consistency: A consistent presentation of information or consistent system actions given the same circumstances are frequently mentioned as important usability metrics, for example by Kim et al. [2008]; Kemp et al. [2008]. Related is also *familiarity*, described by Kemp et al. [2008]; Santos et al. [2013] and Carvalho et al. [2017]. Quigley [2010] described *expressiveness* as a key metric for ubiquitous user interfaces, meaning that the system should give consistent results even if a user acted in unexpected ways. Another attribute related to *consistency* is *predictability*. If a system's actions or outputs are consistent, they are predictable by the user and the user's mental model of the system therefore matches the system behavior [Ryu et al., 2006; Scholtz and Consolvo, 2004; Carvalho et al., 2017]. A predictable user interface is more comprehensible and less confusing. *Transparency* is also a partly related attribute, referring to the user understanding system decisions and actions Kim et al. [2008]; Quigley [2010]. Especially adaptive interfaces can be confusing, therefore Evers et al. [2014] discuss predictability as an important attribute for the usability of adaptive interfaces. Consistency in an adaptive environment can, for example, be achieved if the system does not adapt too frequently and adaptations are comprehensible. It can mean that an already chosen device configuration should not be changed unless it is necessary, for example if the device used before is not available any more.

Unobtrusiveness: Several attributes are related to the obtrusiveness of an interface. There are attributes that are concerned with the social implications and those that concern the user's processing of an interface. Scholtz and Consolvo [2004] discussed *impact and side effects* of ubiquitous systems. Specifically the social acceptance of the usage of such a system is a usability factor that is relevant for smart ubiquitous mobility systems, since public transport requires being in the public. Public and private spaces are regarded differently considering unobtrusiveness. Certain output modalities are less obtrusive than others. Acoustic output from a personal device might be considered obtrusive in a public setting with many strangers present. The usability attribute *awareness* was described by Kemp et al. [2008]; Ryu et al. [2006] and Santos et al. [2013] and refers to possible conflicts in multi-user settings. An aware system can prevent conflicts between the interactions or actions of several users. An unobtrusive system both is socially acceptable and prevents collisions between its user's interactions. Especially regarding calm computing, but also for ubiquitous systems in general, the *cognitive load* of the user is an important usability attribute [Scholtz and Consolvo, 2004; Riekki et al., 2004; Kemp et al., 2008; Harrison et al., 2013; Santos et al., 2013; Carvalho et al., 2017]. Ubiquitous and mobile systems are often used parallel to another, primary task. The user's performance of their primary tasks should not decrease because of the ubiquitous system. Linked to the user's cognitive load is the usability attribute *attention* by Scholtz and Consolvo [2004]; Carvalho et al. [2017]. The user should be able to focus on their primary task and should not be required to shift their focus due to system actions. Ryu et al. [2006] write that the output of a transparent system adds minimal cognitive load to the user, referring to *transparency* as relevant attribute.

Timeliness: Another factor that Riekki et al. [2004] describe as important for calm computing is the *context-sensitive timing* of interaction. Proactive communication of a system should take place at the right time and suitable to the situation [Riekki et al., 2004]. Another aspect of time is described by Kemp et al. [2008] using the attribute *timeliness*. According to the authors, a user should be able to complete their tasks in a short time and a ubiquitous system should not waste the user's time. I combine these aspects in the attribute *timeliness* that can express the correct timing of interaction as well as the short duration of needed interaction.

Relevancy: The *relevancy of interaction* is described by Riekki et al. [2004] as an attribute for calm computing. The system should present an appropriate amount of information that suits the user and their situation - not too much information and no lack of information. Similarly, Ryu et al. [2006] described *feedback*, where the system presents "just enough" information and information that is relevant to the user. Santos et al. [2013] listed *context-awareness* in their quality model and wrote that a system should present relevant services to its user and determine the relevancy of a service using context information. The attribute *relevance* therefore describes the relevance of information and services as well as the right amount of information or services a system offers. Related to the relevance of information is the attribute *suitability* that is also part of the *personalization* and the *perceptibility* cluster. If messages delivered to a user are highly relevant for them, the functionality was suitable.

Relatability: The *understandability* of a system and of information provided by the system is a usability attribute described by Mantoro [2009]. The *ease of use* concerns the usage of the system and how much a user needs to remember about the system in order to use it [Quigley, 2010; Kemp et al., 2008; Santos et al., 2013; Carvalho et al., 2017]. Scholtz and Consolvo [2004]; Kim et al. [2008] mention the *mental model match*, the compatibility of the interface and system's behavior with the mental model of the user. The *accuracy* attribute expresses the match between the situation represented in context and the actual situation and influences the understandability of system decisions [Scholtz and Consolvo, 2004; Kemp et al., 2008; Mantoro, 2009]. It is therefore also influencing the system's *transparency*. In a ubiquitous mobility system and concerning the messages from the system, these attributes influence the understandability and relatability of messages by the user. A user needs to recognize a message, relate to the messages and identify messages that are intended for them. This aspect is summarized by the attribute *relatability*.

The attributes and their parameters that were identified in this analysis are the basis for modeling a usability ontology and for usability assessment rules described in the next sections.

7.2 Ontologies for Autonomous Usability Assessment

Many works on ontology engineering state that the purpose or scope of an ontology should be in focus while modeling and that the structure of an ontology depends on its usage [Uschold and King, 1995; Grüninger and Fox, 1995; Noy, 2001]. The ontologies described in this section are not intended as a general knowledge representation of all concepts related with usability. Their purpose is to provide concepts and properties that can be applied in a usability assessment and utilized in rules for the assessment of output options for a specific message. For this purpose, a message ontology and a usability ontology were developed.

The ontologies presented in chapter 6 provide data about the context of use in which the output options should be evaluated as well as about device and interaction charactaristics. The usability ontology presented in section 7.2.2 models usability qualities and enables their application together with the context ontologies. The elements that usability properties will be applied to are the message itself and output options. A message ontology allows the representation of a message and is introduced in section 7.2.1. These ontologies are also published in the ontology and rules dataset [Keller, 2023]. The application of these ontologies in assessment rules is discussed in section 7.3.

Output options: As modeled in section 6.4, an output option is the combination of one or more specific devices and one or more specific modalities on these devices. For the remainder of this work, I will work with one device and modality per option with the exception of the combination of vibration and text message on smartphones or smartwatches. I will leave other multimodal or multi-device solutions to future work. Both device and modality need to be assessed for their usability, separately and in combination. Several characteristics of a device and a modality affect the usability of an output option. These charactaristics can be referenced using the interaction ontology described in section 6.4.

7.2.1 Message Specification Ontology

This message ontology models several charactaristics of messages in smart ubiquitous mobility systems that can be relevant during a usability assessment. If personal data is part of the message's content, this can affect the relatability of the message as well as the privacy of message delivery, for example. The message therefore is one element that usability rules can be applied to, in order to determine which output options are suitable to deliver it.

A message generated by a passenger information system can either be directly represented using this ontology, or in any other format. In the latter case, the assessment framework either works without further knowledge about the message and some usability attributes can not be assessed, or a component could be added to the framework, converting a given message to a representation using this ontology. Such a component could, for example, extract message properties from the original message, using natural language processing or other approaches. However, this step is not in the scope of this work.

```
PREFIX msg: <http://iums.eu/ontologies/message#>
```

Listing 7.1 — The namespace and prefix of the message ontology in SPARQL notation.

Listing 7.1 defines the namespace of this message ontology. Figure 7.3 shows the modeled classes and properties. The properties `references` and `contains` can be used to describe the message context. The `references` property is an `ObjectProperty` that can express that a message refers to some object. At this point, another ontology can be referenced, for example a `Vehicle` from the public transport ontology in section 6.2. The `DatatypeProperty` `contains` can be used to represent the raw format of the message, for example plain text.

A passenger information message in public transport has a period in which it is valid. The validity period is specified by a start and an end time. The content type of a message can be specified using the `OutputInformation` classes and therefore is compatible with device's modalities and converter definitions. Three boolean properties can be used to classify the type of the message. It can be a call to action that requires the user to act upon receiving this message. It also can be informative, meaning the user needs to know this information, but does not need to act upon it. And the message can be just a signal that has not

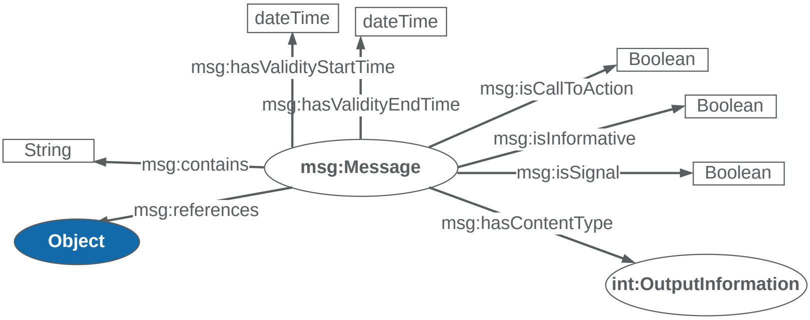


Figure 7.3 — Message specification.

much content but signals, for example, that a passenger should exit the vehicle at the next stop.

7.2.2 Representing Usability Qualities in an Ontology

The usability ontology will be described in the following. It can be used for a usability assessment based on reasoning, when rules are expressed using either OWL Restrictions or, for example, SWRL rules. The usability ontology can, however, also just serve as a documentation of the usability concepts and their relations that are the basis of the conceptual usability assessment framework and the usability rules can be expressed and implemented using other methods. The namespace of the usability ontology is defined in listing 7.2.

```
PREFIX us: <http://iums.eu/ontologies/usability#>
```

Listing 7.2 — The namespace and prefix of the usability ontology in SPARQL notation.

Several design decisions informed the modeling of usability assessment results and rules. The ontology should be able to express usability qualities of elements. It therefore needs to be able to identify those elements as well as the qualities and then express the assignment of a degree of such a quality to an element. Rules can express the conditions under which an element gets assigned which degree of a quality.

The elements whose usability can be rated during a usability assessment are subsumed under the `AssessableElement` class that is shown in figure 7.4. For this work, output options and messages are assessable elements, as discussed above. There are usability attributes that concern the message, for example *relevance*, where other attributes concern output options, such as *perceptibility*.

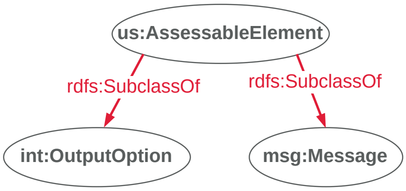


Figure 7.4 — Assessable elements.

A usability assessment assigns a degree of such a quality to the element that is subject to evaluation. In some cases, this degree can be expressed as a binary result, in others, the degree can be expressed on a scale. For scales, either a continuous or a discrete, categorical scale would be possible.

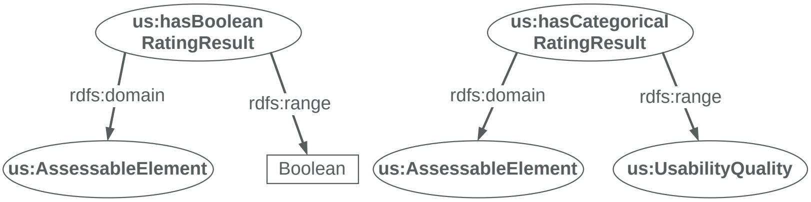


Figure 7.5 — Properties to express rating results.

Representing the degrees of usability qualities on a continuous and numerical scale would allow to calculate with scores, which is not possible using discrete and categorical values. However, the meaning of such a result would have to be interpreted by the framework. And while the extent of usability qualities can be continuous, people mostly talk in categories and comparisons about them. Since the goal of this approach is to use the usability assessment for making a decision about output options, the assessment should provide results that allow the system to come to a conclusion. Ratings based on categories can provide a good basis for decision making, since only a limited number of different values for each assessment is possible and their interpretation by the system is easier to implement. Rules that are used in the decision making process based on categorical assessment results are easier to express and to understand.

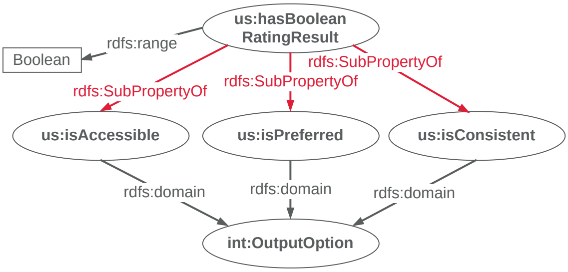


Figure 7.6 — The `hasBooleanRatingResult` property and its subproperties.

Additionally, the ratings and the decision can be more understandable and readable, which makes customizing the framework easier.

Using categories, each valuation category for a quality can have a distinct and predefined meaning, which simplifies their handling. Lastly, ontologies are better suited to model categories than continua, as are rules expressed in SWRL. Considering these arguments and the intended use of the framework, I decided to model valuation categories for usability attributes in addition to simple binary results. The categories represent different degrees of a usability quality and in the ontology, they are modeled as instances of the respective quality class.

The properties displayed in figure 7.5 allow to assign these different types of rating results to assessable elements. A simple usability assessment that results in a binary rating can be modeled using the `hasBooleanRatingResult` property. It is a `DatatypeProperty` that has `Boolean` as its range. The property `hasCategoricalRatingResult` can assign a quality category as a result. For both properties, specific subproperties were modeled for each usability quality that can be assigned.

The usability ontology builds on the usability attributes identified in the analysis described in section 7.1. The result of this section were clusters with representative usability attributes for each cluster, being *personalization*, *privacy*, *perceptibility*, *unobtrusiveness*, *consistency*, *relatability*, *timeliness* and *relevance*.

As discussed before, the structure of an ontology depends on its usage. These usability attributes were therefore incorporated into the ontology in alignment with their applicability in a usability assessment. The modeling of those attributes is described in the following.

As discussed in paragraph 7.1.2, *preference* and *accessibility* are both part of personalization, but they are not equally important. While an output option must be accessible, it can be acceptable if it is not aligned with the user's preferences. Therefore, the attribute *personalization* is divided into two qualities.

Personalization: For a usable interface, the user's preferences should be considered [Ryu et al., 2006; Lee and Yun, 2012]. A personalization assessment can match output options with user preferences and determine if an option corresponds to those preferences. In this ontology, it is modeled as having a boolean result, either matching the user's preferences or not. The property `isPreferred` can be used to denote for an `OutputOption` if it corresponds to the user's preferences. It is a subproperty of the `hasBooleanRatingResult` property and is shown in figure 7.6. It is, of course, possible to see personalization in a more gradual way and to model it using several categories for degrees of personalization, for example if users can express several preferences and a part of them is met. However, this introduces more complexity to the assessment and decision processes that would, in my opinion, not be beneficial for the result and would have a negative impact on understandability and maintainability as well. The number of available output options in a given mobility situation is limited. A highly complex assessment and decision process using a multitude of categories for each usability attribute can lead to empty results, because of overspecification. Additionally, a multitude of preference options for users to choose from can be overwhelming and may lead to users not specifying any preferences at all, which is why I chose only to incorporate user preferences regarding output modality. Other aspects of personalization are not modeled at this point, but can be added in an extension of the ontology, if deemed necessary.

Accessibility: As discussed above, accessibility is expressed separately. The accessibility assessment is also resulting in a boolean value. Figure 7.6 shows the property `isAccessible` that can describe this fact for an `OutputOption`.

Consistency: The consistency of an output option is either given or not. The result of a consistency assessment is therefore binary and can be expressed using the `isConsistent` property, as shown in figure 7.6, as well.

The usability ontology provides a `UsabilityQuality` class and several properties that can express a categorical rating result based on such a usability quality. The classes modeling usability qualities in this ontology are subclasses of the `UsabilityQuality` class, as shown in figure 7.8. As figure 7.8 shows, not all of the remaining usability qualities identified in section 7.1 are represented directly. The modeling of usability qualities again is targeted at the intended use of this ontology. Therefore, the attribute *privacy* is distinguished into *privacy*,

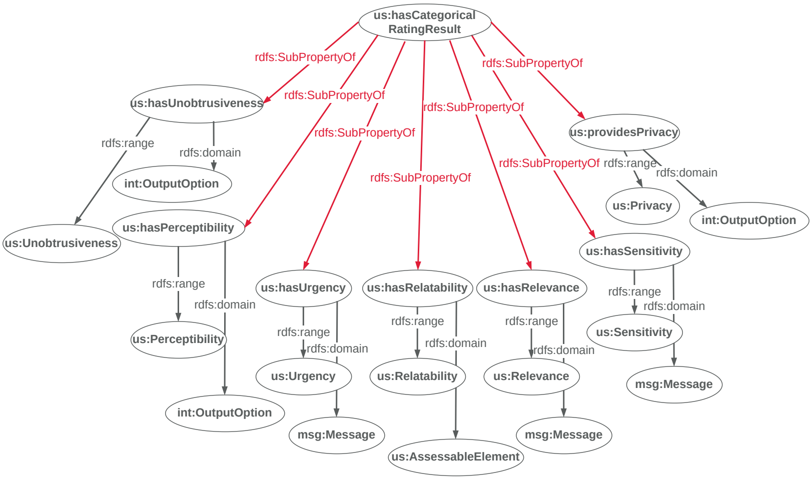


Figure 7.7 — The `hasCategoricalRatingResult` property and its subproperties.

describing qualities of output options, and *sensitivity*, describing qualities of the message. These qualities are assessed separately and are then evaluated together in the decision process. The *timeliness* attribute is in turn modeled as the *urgency* quality of a message. This modeling can be extended using further qualities, if needed.

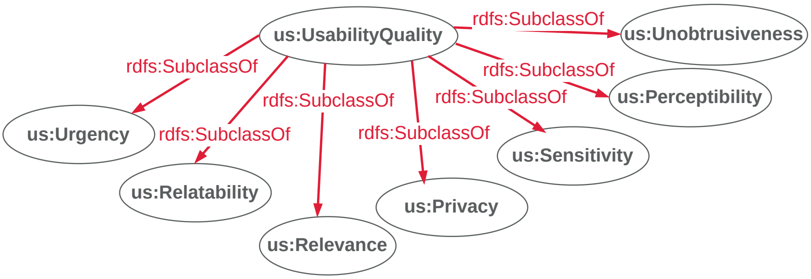


Figure 7.8 — Usability qualities.

The `hasCategoricalRatingResult` property and its subproperties are shown in figure 7.7. Some usability attributes are only applicable to the message or to output options and others can be applied to both. The subproperties therefore further specify their domains, if needed.

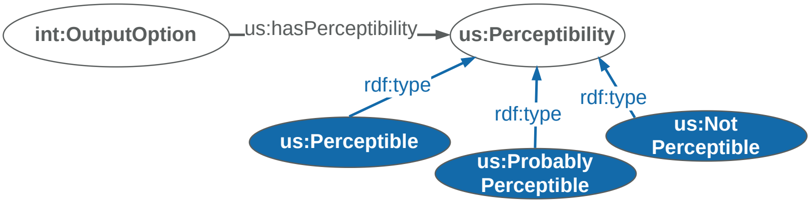


Figure 7.9 — The `Perceptibility` class.

Perceptibility: Perceptibility is a binary concept that is applicable to an output option, but not a message. Either an output option is perceptible or it is not. However, I chose to extend the modeling of perceptibility to include a probability. Context data always represents only facets of reality and, additionally, it is not always complete. Uncertainty of reasoning results, context data quality and ambiguity could be modeled more precisely using probability logic or fuzzy logic, as, for example, applied by Ranganathan et al. [2004] or discussed by Bolchini et al. [2007]. However, the utility of this approach lies in deciding for an output option based on assessment results. To extend assessment results with a differentiated model for quality and uncertainty of data and reasoning would greatly increase the verbosity of the model and also increase the amount of possible results to process during decision making. It would make the assessment process more opaque. It would also have little benefit since despite potentially missing or incomplete context data, the assessment process must focus on producing at least one result. I therefore decided against expressing uncertainty in greater detail and incorporated only three nuances of perceptibility.

In the case of perceptibility, it is very much possible that context data is incomplete and an assessment cannot decide with certainty, if an output option will be perceived. A situation where a passenger riding a train put their smartphone on the table and is reading a newspaper is, for example, very hard to capture using context data. The smartphone can detect that it is lying on the table, but using simple sensors it is very hard to detect where the passenger's attention lies. Anderson et al. [2018] surveyed recent approaches towards attention management that could improve context data for the assessment of such a situation. These can be included in future work and the context and usability ontologies can be extended to model results of attention management and more detailed categories of perceptibility, if their application is considered to improve the results of the usability assessment. However, for the scope of this work, my approach would determine in this situation that a visual message is probably perceptible by the user. I therefore determined the perceptibility categories as follows:

- `Perceptible`: This category is used if it can be reasoned with a high certainty that the user will perceive a message using this output option.
- `ProbablyPerceptible`: If there are indicators that the user would perceive a message using this output option and available context data do *not* allow the conclusion that they will not perceive the message.
- `NotPerceptible`: This category should be used if there are either no indicators that a user would perceive a message or the context data allows the conclusion that they will not.

The classification of perceptibility into these three categories, as shown in figure 7.9 allows a distinction based on probability but at the same time is simple enough to yield transparent results and allow for comprehensible rules. Further details can be added in extensions to this ontology, if needed.

Unobtrusiveness: Unobtrusiveness is rated in categories as well, as shown in figure 7.10. Output options can be obtrusive or unobtrusive, but this attribute is not applicable to a message. In this specific model, there are only the categories `us:Unobtrusive` and `us:Obtrusive`, but the model can be extended by as many intermediate categories as necessary for a specific application. An unobtrusive output option does not distract the user and does not add to their cognitive load [Scholtz and Consolvo, 2004; Cao et al., 2009; Quigley, 2010; Harrison et al., 2013].

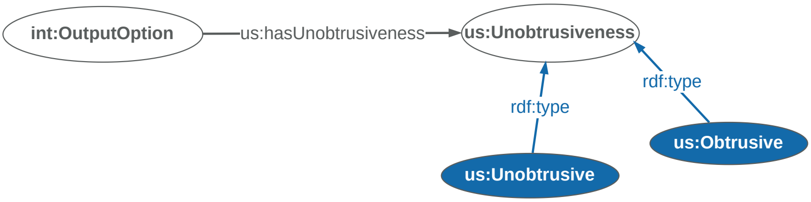


Figure 7.10 — The `Unobtrusiveness` class.

Intermediate categories are only usable if the user's cognitive load and attention can be assessed in greater detail and then obtrusiveness can be rated gradually. The developements in attention management can contribute to an unobtrusiveness assessment [Anderson et al., 2018]. Okoshi et al. [2017] use activity recognition to decide, when to interrupt the user with a notification, for example. However, activity recognition as well as attention management are not in the scope of this work. For this work, I decided to focus on the interruptibility of the user's task.The unobtrusiveness categories defined in this instance of the usability ontology are therefore defined as follows:

- `Unobtrusive`: This output option does not interrupt the user's primary task.
- `Obtrusive`: An obtrusive output option does interrupt the user's primary task and adds significant cognitive load.

In order to inform a passenger in public transport, sometimes an obtrusive output option must be chosen, if no other option is available and the message is urgent, for example. A usability assessment therefore needs to lead to a tradeoff between unobtrusiveness, perceptibility and the message's urgency.

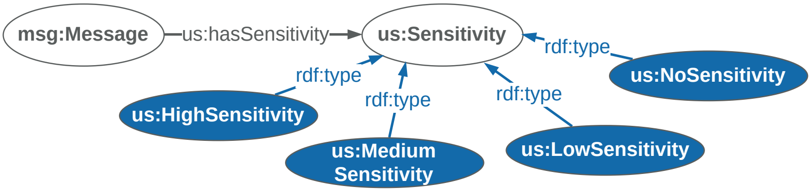


Figure 7.11 — The `Sensitivity` class.

Privacy and Sensitivity: The privacy of an output option should not just be determined by an assessment of the device and modality of this options and the given situation. The sensitivity of the message plays an important role. If the passenger's name is contained in the message, sensitivity is high, for example. If no personal data is included at all, the message is not sensitive. In our research project SmartMMI[1] we identified three main sensitivity categories for data in public transport messages, describing data of no sensitivity, personal data and highly sensitive data [Titov et al., 2020]. For this ontology, I detailed four categories of sensitivity, since there is data about the user's journey that allows

---

[1] 'SmartMMI: Modell- und kontextbasierte Mobilitätsinformation auf Smart Public Displays und Mobilgeräten im Öffentlichen Verkehr", `http://smartmmi.de/`, last accessed October 8th, 2022

conclusions about the user, which should be avoided. Some data, however, is only loosely linked to the user and is therefore less sensitive. This data with low sensitivity can be made public in some cases, especially in combination with a low privacy level of output. Since the goal of the system is to provide information to the user, situations may arise that require a compromise with regard to privacy and message sensitivity. These four levels of sensitivity allow a decision process to compromise, if necessary:

- `HighSensitivity`: A message has a high sensitivity level if it contains data that identifies the user or it contains sensitive data about the user, such as their exact location, address, health data, cultural background or similar data.
- `MediumSensitivity`: A message has a medium sensitivity if it contains data that can identify the user's itinerary or tasks. This is data that does not expose facts about the user but allows conclusions about the user.
- `LowSensitivity`: A low sensitivity rating is data that is loosely linked to the user or their journey.
- `NoSensitivity`: A message has no sensitivity if it only contains data without reference to the user.

The ontology provides a `Sensitivity` class and the corresponding sensitivity categories that allow a rating of the message, see figure 7.11.

An output option provides a certain privacy level that can depend on the current context. The `providesPrivacy` property can be used to annotate this privacy level, as shown in figure and 7.12. The privacy categories and their definitions are:

- `HighPrivacy`: A high privacy level means that no one other than the user will know that there is a message for the user.
- `MediumPrivacy`: At a medium privacy level, others may realize that there is a message for the user, but they can not perceive its content.
- `LowPrivacy`: An output option provides a low privacy level, if others may realize that there is a message and some can, under certain conditions, perceive it. If they can perceive it under all circumstances, privacy level is low if they can not identify the recipient.
- `NoPrivacy`: An output option provides no privacy, if the message can be perceived by all bystanders and it can be associated with the user.

In an assessment of the privacy of an output option for a specific message in a specific context, the privacy level of the output option and the sensitivity

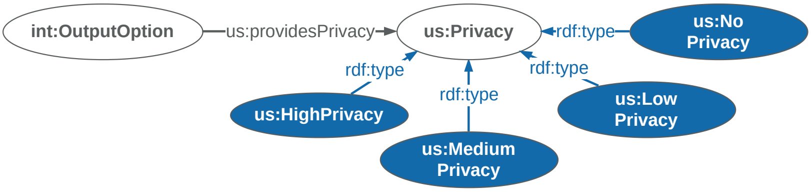


Figure 7.12 — The `Privacy` class.

level of the message are rated independently from each other and both levels then are matched. The assessment rules determine which level of sensitivity is acceptable for which level of privacy provided by a certain output option.

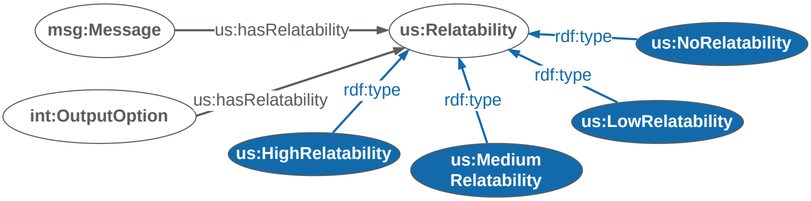


Figure 7.13 — The `Relatability` class.

Relatability: Relatability is a quality that can be attributed to a message as well as to an output option. It describes how well a user can relate to a message based on its content or based on the output option that delivers it. The `Relatability` class and its individuals express a rating for relatability in four categories and can be attached to a message and an output option, shown in figure 7.13.

The relatability levels are defined as:

- `HighRelatability`: A message or output option have a high relatability if it is certain that the user will recognize that the message concerns them. This is the case for messages on a private device, for example.
- `MediumRelatability`: The medium relatability level means that the user will probably realize that the message concerns them, for example if it contains data of their itinerary, such as their next stop.
- `LowRelatability`: A low relatability is given, if there is a connection to the user or their itinerary, but it must be concluded.

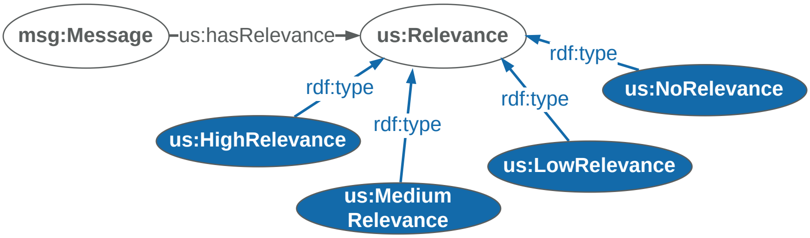


Figure 7.14 — The `Relevance` class.

- `NoRelatability`: A message has no relatability if there is no connection to the user or their itinerary. An output option would be not relatable, if it is not perceivable by the user, meaning this category is covered by the perceptibility rating.

Relevance: A message has a certain relevance to the user and this can influence the usability of a message delievery, while an output option can not have a relevance. The `Relevance` class and `hasRelevance` property are shown in figure 7.14. The definition of the relevance categories are as follows:

- `HighRelevance`: The message is highly relevant to the user if they need to act on the information contained in this message.
- `MediumRelevance`: The message's relevance level is medium if the user needs to know the information but does not need to act on it.
- `LowRelevance`: The low relevance category describes messages whose content relates to the user's goal or activity but does not affect it.
- `NoRelevance`: A message has no relevance, if it does not relate to the user's goals or activities.

The relevance of a message may influence its relatability. It is possible to use a relevance rating for the assessment of relatability. However, separating both assessments yields a higher modularity and better understandability of rules.

Urgency: The urgency of a message is an important measure for the assessment of timeliness. Figures 7.7 and 7.15 show the `Urgency` class and the `hasUrgency` property. The ontology defines three categories for urgency. The definition of these categories is as follows:

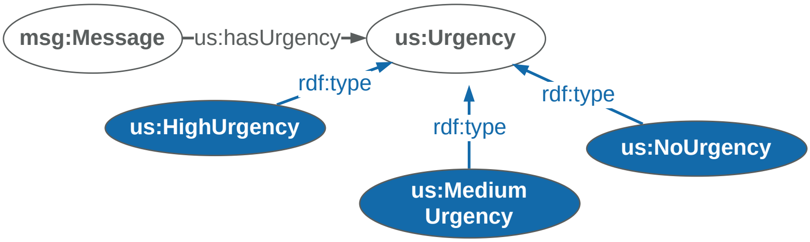


Figure 7.15 — The `Urgency` class.

- `HighUrgency`: The message relates to an event or point in time that is already happening or will begin immediately.
- `MediumUrgency`: The message relates to an event or point in time that will begin in the very near future.
- `NoUrgency`: The content of the message concerns events or points in time that are either completed and in the past or in a more distant future.

The terms "near future" and "distant future" are not very precise. Their extent must be discussed and determined for a potential target system, since it depends on the time periods that are covered by the system. Appropriate definitions can also depend on the context data that is available to assess urgency and the rules that use the urgency assessment for the determination of a usable output option.

7.3 Rules for Autonomous Usability Assessment

The ontologies presented in chapter 6 build a basis to express knowledge about user context in mobility and about output options, characterizing devices and modalities. The ontologies in sections 7.2.1 and 7.2.2 can be used to express knowledge about messages to the user and about usability attributes and usability ratings that can be assigned to messages and output options.

In order to achieve such a usability rating of a message and output options, a usability assessment must be performed. This section presents rules that can be used to implement such an assessment. They use the discussed ontologies to express metrics. When these rules are applied, they provide a usability rating for a message or output option.

The rules discussed in the following are not exhaustive and their intent is not to be exhaustive. The framework for usability assessment is designed to provide models and a conceptual structure for the implementation of systems that are able to autonomously assess usability for output options based on context data. In each target system, available context data, available output devices and applications utilizing the usability assessment differ. Also, stakeholders can focus on certain aspects of usability.

For each usability attribute, a multitude of message parameters, context facts and output option charactaristics matter in an usability assessment. Which of those are considered during usability assessment depends on available data, design choices and subsequently implemented rules. The selection and expression of rules determines a prioritization and design choice for such a system. The framework therefore supports the extension not only of its models but also of the assessment rules in the design process of a target system. The collection of rules presented in the following paragraphs represents a starting point for the development of a suitable set of rules and illustrates the applicability of the framework.

Implementing a usability assessment and executing assessment rules can be achieved in different ways. Rule engines are one conceivable solution as are other reasoning methods, discussed, for example, in section 2.4.1. The framework for usability assessment developed in the course of this work does not enforce one of these methods for its application. The rules presented in the following are formalized as SWRL rules, written in its human readable syntax [Horrocks et al., 2004]. Nevertheless, an implementation with an SWRL capable rule engine is not necessary in order to implement the usability assessment and the application of these rules, other approaches are possible, as well. The rules are published as a part of the ontology and rules dataset [Keller, 2023].

Usability qualities can interact and influence each other, sometimes also contradict each other and tradeoffs are often necessary in the design process of a user interface [Nielsen, 1993a]. Such tradeoffs and balancings are also part of an autonomous usability assessment. The framework can consider these tradeoffs in different ways. It can either focus on the interdependencies of usability attributes and implement assessment rules that directly reflect these interdependencies in a flat approach or implement a modular and hierarchical approach that evaluates usability attributes separately from each other first and balances them afterwards.

Implementing a flat approach would mean to model fine-grained rules that are closely linked. The rules would be difficult to view separately and the adaptation of rules to a target system, its available knowledge and devices would have many side effects. While the interdependencies between assessment parameters and between usability criteria would be easier to express, the set of rules would also be difficult to structure and to manage.

Therefore I chose to implement a modular and hierarchical approach. It realizes the assessment in different modules that each consider one usability attribute. After the assessment of each usability attribute, a holistic usability assessment brings those assessments together and works out tradeoffs representing the interdependencies between the attributes. With a modularized approach, an implementation of the usability assessment can weight each attribute with regard to the target system and can add or remove rules in order to customize the assessment's outputs. It also becomes possible to remove the assessment of an attribute from the process altogether. The assessment rules for each usability attribute are discussed in the paragraphs below. The holistic usability assessment is then discussed in section 7.4.

Assessing Personalization

For personalization the user's preferences must be checked and matched to the option that is assessed. In the models presented in chapter 6, preferences can be expressed for modalities. Those could similarly be extended for devices or properties of a message or devices. Rule 7.1 shows how preferences can be evaluated for modalities. The user for whom this assessment is performed is identified using their `UserID`. The first part of the rule determines the output modality the output option is using, which is of the type `OutputInformation`, represented in the variable $?output$. If this ist the same as the user prefers to sense, the output option is a preferred option.

$$\begin{aligned} &uses(?option, ?device) \wedge providesFunction(?device, ?function) \\ &\wedge presents(?function, ?output) \wedge prefersToSense(UserID, ?output) \\ &\qquad\qquad \Longrightarrow isPreferred(?option, true) \end{aligned} \tag{7.1}$$

Assessing Accessibility

Accessibility can be determined by matching the modality of the output option with the explicitly stated abilities of the user, as expressed in rule number 7.2.

$$\begin{aligned} &uses(?option, ?device) \wedge providesFunction(?device, ?function) \\ &\quad \wedge presents(?function, ?output) \wedge isAbleToSense(UserID, ?output) \\ &\qquad \implies isAccessible(?option, true) \end{aligned} \tag{7.2}$$

The model allows for different representations of a user's ability. A change in representation would require an adaptation of this rule, but it remains a simple matching.

Assessing Consistency

Regarding the usability of an output option for a specific message in a specific situation, an option is consistent if it was used for a message to this user before. A heavily consistent interface for information provision in an ubiquitous public transport system would always use the same device and modality. However, one reason for an adaptive output in ubiquitous mobility systems is that the same option is not suitable in all situations. Consistency should therefore be considered in a more differentiated way.

$$\begin{aligned} &hasCurrentTask(UserID, ?task) \wedge usedOutputOption(?task, ?option) \\ &\qquad \implies isConsistent(?option, true) \end{aligned} \tag{7.3}$$

The option I chose here is to differentiate for each of the user's tasks. In mobility, a new task is likely linked to a new situation and therefore an output option that was used for the prior task may not be suitable during the current task. Rule 7.3 shows that a consistent option would be the option that was used before during the same task. If no option was used before during this task, there is no consistent option available.

Assessing Perceptibility

A perceptibility assessment of an output option depends on the type of device and the modality that comprise this output option. The following rules are therefore device-dependent. For each device, there are rules for each modality. This modular approach allows to add new devices or modalities to a system withouth the need to revise all perceptibility rules for their applicability.

$$\begin{aligned}(7.4)\quad & uses(?option, ?device1) \wedge rdf:type(?device1, Smartphone)\\ & \wedge uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput)\\ & \wedge wears(UserID, ?device2) \wedge rdf:type(?device2, Headphones)\\ & \wedge connectedTo(?device2, ?device3) \wedge differentFrom(?device2, ?device3)\\ & \implies hasPerceptibility(?option, NotPerceptible)\end{aligned}$$

Rule 7.4 establishes that if the output option is comprised of a smartphone and acoustic output, the user wears headphones but they are connected to a device other than the smartphone, this option is not perceptible. Rule B.6, listed in the appendix, decides that in the same situation, if the user's headphones are connected to the smartphone, this option is perceptible. Rule 7.5 determines if the smartphone's display is on and its sensors detect high ambient brightness. In this case, acoustic output is judged to be perceptible. These context facts are used to decide if the smartphone is not in a pocket or bag.

$$\begin{aligned}(7.5)\quad & uses(?option, ?device) \wedge rdf:type(?device, Smartphone)\\ & \wedge uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput)\\ & \wedge highAmbientBrightness(?device, true) \wedge displayOn(?device, true)\\ & \implies hasPerceptibility(?option, Perceptible)\end{aligned}$$

This interpretation of context data can be improved based on available context data and context reasoning on the device itself, for example. Rule 7.5 exemplarily shows how a situation can be described, conclusions drawn and utilized using assessment rules. Approaches toward a reliable detection algorithm that decides if a smartphone is in a pocket, backpack or on a table were developed by Yang et al. [2013] and Darbar and Samanta [2015], for example. Visual perceptibility for smartphones can be assessed in many ways. As an example, rule B.8, listed in the appendix, determines if the smartphone's display is on and then concludes that visual output is perceptible. A more detailed assessment could involve more sophisticated methods to detect if a user is looking at their smartphone, for example described by Pagliari et al. [2019].

$$\begin{aligned}(7.6)\quad & uses(?option, ?device) \wedge rdf:type(?device, Smartphone)\\ & \wedge uses(?option, ?modality) \wedge rdf:type(?modality, HapticOutput)\\ & \implies hasPerceptibility(?option, ProbablyPerceptible)\end{aligned}$$

The perceptibility of vibration or generic haptic output also depends on the placement of the smartphone. Vibration is probably mostly used in combination with text messages on smartphones, since a single vibration output can transport only little information. In this case, the rules take effect in combination and the order of the application of rules is relevant. This order is determined by the implementation of the framework. If the check of the visual text option results in an assessment as perceptible, rule 7.9 should not overwrite this perceptibility assessment. However, if a visual text message should be assessed as not perceptible, added vibration can increase perceptibility. Generally, vibration is determined to be probably perceptible. As long as a user keeps their smartphone in their vicinity, which is likely, they are likely to perceive the smartphone vibrating, which is why rule 7.6 sets Perceptibility to `ProbablyPerceptible`. Rule 7.6 can be refined for better precision if more context data is available.

Rule B.10 in the appendix determines that vibration is perceptible if ambient brightness is high and the display is on. Here, it is also important to consider the execution sequence of the rules since rule 7.6 is more generic than rule B.10 and the wrong order could produce unwanted results, overwriting a previous assessment with a less precise assessment.

For smartwatches, perceptibility of acoustic output is determined analogously to smartphones. The device model allows the restriction for smartwatches that acoustic output only works with headphones. The rules utilizes this property.

$$
\begin{aligned}
(7.7)\quad & uses(?option, ?device1) \wedge rdf:type(?device1, Smartwatch) \\
& \wedge uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput) \\
& \wedge providesFunction(?device1, ?function) \\
& \wedge hasTechnicalProperty(?function, ?property) \\
& \wedge onlyWorksWithHeadphones(?property, false) \\
& \implies hasPerceptibility(?option, ProbablyPerceptible)
\end{aligned}
$$

If the smartwatch is able to play acoustic output withouth headphones, perceptibility is set to `ProbablyPerceptible` with rule 7.7, but if it only works with headphones, general perceptibility is set to `NotPerceptible` with rule B.12. Rule B.13 as a more specific rule determines if the user wears headphones connected to the smartwatch and sets perceptibility of acoustic output to `Perceptible`.

The rules 7.7, B.12 and B.13 also depend on a specific execution order, where 7.7 must be executed first and B.13 last. More differentiated rules that are not execution order sensitive are possible, but also are more complex and less modular.

$$
\begin{aligned}
(7.8)\quad & uses(?option, ?device1) \wedge rdf:type(?device1, Smartwatch) \\
& \wedge uses(?option, ?modality) \wedge rdf:type(?modality, VisualOutput) \\
& \wedge wears(UserID, ?device2) \wedge sameAs(?device1, ?device2) \\
& \implies hasPerceptibility(?option, Perceptible)
\end{aligned}
$$

For other modalities, rules 7.8 and 7.9 decide that if the user wears the smartwatch, the output options using this smartwatch and visual or haptic output are perceptible. Haptic output can also be combined with visual output, which is why the order of the rules should be considered, as discussed above in the case of smartphones.

$$
\begin{aligned}
(7.9)\quad & uses(?option, ?device1) \wedge rdf:type(?device1, Smartwatch) \\
& \wedge uses(?option, ?modality) \wedge rdf:type(?modality, HapticOutput) \\
& \wedge wears(UserID, ?device2) \wedge sameAs(?device1, ?device2) \\
& \implies hasPerceptibility(?option, Perceptible)
\end{aligned}
$$

A public display can have acoustic output and the rule 7.10 determines that if the user is near the display, acoustic output is `Perceptible`. The user's proximity to a display can, for example, be detected using Bluetooth on their smartphone.

$$
\begin{aligned}
(7.10)\quad & uses(?option, ?device) \wedge rdf:type(?device, PublicDisplay) \\
& \wedge uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput) \\
& \wedge isNear(UserID, ?device) \\
& \implies hasPerceptibility(?option, Perceptible)
\end{aligned}
$$

However, if the user wears headphones, this result is altered to `NotPerceptible` with rule B.17. For visual output, a public display is assessed as perceptible, if the user is near it, using rule B.18. Rule B.19 restricts this assessment if other users use this display. In this case, the display is assessed as not perceptible and the assessment of rule B.18 is overwritten. Rules B.17, B.18 and B.19 are listed in the appendix.

The assessment of visual perceptibility of a vehicle display is also based on the user's proximity and this rule is therefore anaologous to rule B.18. The situation is similar for Smart Windows. However, in contrast to public displays, smart windows are often directly next to seats and therefore their visual perceptibility can be assessed differently. Basically, visual output for a Smart Window is assessed as `ProbablyPerceptible` if the user is near the display, using a rule similar to B.18. If the user uses the display, perceptibility is set to `Perceptible`. The usage of the Smart Window can be determined if the user actively connected their smartphone to the Smart Window. Another differentiation of these rules could be devised based on a more precise distance measurement using Bluetooth. The perceptibility rules for vehicle displays and Smart Windows are listed in the appendix in sections B.2.2 and B.2.2.

Assessing Privacy

The privacy level of output options can be assessed modality-wise. The following rules determining the privacy of acoustic output must be executed in the given order, to ensure a correct outcome. As a baseline, acoustic output in general is rated as providing `NoPrivacy` using rule 7.11.

$$
\begin{aligned}
(7.11)\quad & uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput) \\
& \qquad \Longrightarrow providesPrivacy(?option, NoPrivacy)
\end{aligned}
$$

Exceptions from this rule are determined using additional rules. For public devices, privacy is set to low using rule 7.12.

$$
\begin{aligned}
(7.12)\quad & uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput) \\
& \quad \wedge uses(?option, ?device1) \wedge isPublic(device1, true) \\
& \qquad \Longrightarrow providesPrivacy(?option, LowPrivacy)
\end{aligned}
$$

If the output modality is given more precisely and it is only an acoustic signal, in contrast to output of speech, rule B.26 in the appendix sets privacy to medium. If the user is at home, privacy for acoustic output is set to high, described by rule 7.13. In this rule, the calculation if the user's current location is their home location is simplified for readability. In an implementation of this rule, the `sameAs` relation must be replaced by a calculation or algorithm that can match locations, for example based on a comparison of GPS coordinates.

$$(7.13) \quad hasHome(UserID, ?location1) \wedge hasLocation(UserID, ?location2) \wedge sameAs(?location1, ?location2) \Longrightarrow providesPrivacy(?option, HighPrivacy)$$

Privacy is also high, if the user wears headphones connected to the device of the output option, described by rule B.28, listed in the appendix. In the case of visual output, if the device is not public, then the output option is providing high privacy, expressed with rule 7.14.

$$(7.14) \quad uses(?option, ?modality) \wedge rdf:type(?modality, VisualOutput) \wedge uses(?option, ?device) \wedge isPublic(?device, false) \Longrightarrow providesPrivacy(?option, HighPrivacy)$$

For public devices, rule B.30 states that privacy is medium if the user uses this device. In contrast, if someone else uses the device, rule B.31 determines that this option provides a low privacy level. Both rules are listed in the appendix, section B.2.3.

The assessment of privacy for vibration determines first that haptic output provides generally a medium level of privacy, expressed in rule 7.15. A vibration signal of a smartphone can be detected by other passengers and therefore, they can realize that the user received a message, which is a medium level of privacy.

$$(7.15) \quad uses(?option, ?modality) \wedge rdf:type(?modality, HapticOutput) \Longrightarrow providesPrivacy(?option, MediumPrivacy)$$

In case the user is at home, a similar rule to 7.13 can be used to determine a high privacy level. This assessment can be detailed further if the necessary data is available, for example to determine if there are other people nearby while the user is at home.

Assessing Message Sensitivity

The sensitivity of a message is determined by its content. Therefore, an analysis of the message content must be performed before executing the respective rules. If certain facts about the content of the message can not be derived, its sensitivity can not be assessed in detail. If the message contains the user's overall destination or origin, sensitivity is set as medium by rules 7.16 and 7.17.

$$\begin{aligned} &hasActiveItinerary(UserID, ?trip) \wedge hasOrigin(?trip, ?location) \\ &\quad \wedge references(?message, ?location) \\ &\quad\quad \implies hasSensitivity(?message, MediumSensitivity) \end{aligned} \tag{7.16}$$

$$\begin{aligned} &hasActiveItinerary(UserID, ?trip) \wedge hasDestination(?trip, ?location) \\ &\quad \wedge references(?message, ?location) \\ &\quad\quad \implies hasSensitivity(?message, MediumSensitivity) \end{aligned} \tag{7.17}$$

The rules B.33 and B.34 set the sensitivity of the message to `LowSensitivity` if it contains one of the stops of the user's itinerary. If only a stop during the journey of a user is named in the message, third parties can not derive the origin or destination of the user's journey or other sensitive information about the user. Information about one stop along the way is therefore less sensitive than the origin or final destination of a user.

A high sensitivity is determined by rules B.37 and B.38 when the message contains the first or last name of the user. If any address is contained in the message, then the sensitivity of the message is also ruled as high, see rule 7.18. The identification of an address from text or speech must be implemented for the applicability of this rule. The rule uses a `isAddress` property that represents the result of such an analysis.

$$\begin{aligned} &isAddress(?content, true) \wedge contains(?message, ?content) \\ &\quad\quad \implies hasSensitivity(?message, HighSensitivity) \end{aligned} \tag{7.18}$$

Rule B.40 sets the sensitivity of the message to high if the message references the location of the user, which could be the case on a map, for example.

Assessing Unobtrusiveness

The unobtrusiveness attribute has two main aspects. The first focuses on the social situation and possible side effects of interaction in such situations.

The second aspect deals with the cognitive load of the user from various perspectives.

As discussed before, how obtrusive an interaction is depends on the social situation. If there are other people present, it can include the relation of the user to these people, for example. An interaction can be less obtrusive if the user is familiar with the people that are present than if the people in the vicinity are not familiar. It is very hard to detect and correctly classify such situations. Eagle and Pentland [2006] used extensive monitoring of bluetooth on the mobile phones of users to detect social situations. In public situations such as public transport, such extensive monitoring is not available. It can be assumed that most people in the vicinity of the user are not familiar to them. However, at home, it can be concluded that the social situation is more familiar. Since a detailed detection of social contexts is out of the scope of this work, the following rules use approximations to assess the obtrusiveness of an output option. If more sophisticated methods for detecting social context can be added in future work, these rules can be extended or replaced.

Rule 7.19 shows the assessment of an output option as unobtrusive if the user is currently at home as an approximation to consider the social situation.

$$\begin{aligned} &hasHome(UserID, ?location1) \wedge hasLocation(UserID, ?location2) \\ &\quad \wedge sameAs(?location1, ?location2) \\ &\qquad \Longrightarrow hasUnobtrusiveness(?option, Unobtrusive) \end{aligned} \tag{7.19}$$

If the user is not at home, a useful approximation towards unobtrusiveness is based on the discussion of conflicts in multi-user settings by Ryu et al. [2006] or Kemp et al. [2008]. An unobtrusive user interface prevents collisions between interactions of different users. Rule 7.20 expresses that an output option on a device that is already used by another user is obtrusive.

$$\begin{aligned} &uses(?option, ?device) \wedge isUsedBy(?device, ?user) \\ &\quad \wedge differentFrom(?user, UserID) \\ &\qquad \Longrightarrow hasUnobtrusiveness(?option, Obtrusive) \end{aligned} \tag{7.20}$$

As discussed above, it is difficult to detect social situations and to automatically assess if an interaction would be appropriate in a social setting. A possibility to express a high sensitivity towards social situations is to not use public devices if people are nearby. If the proximity that is used to express the nearby relation is chosen quite high, this rule counteracts the purpose of interactive public devices. The correct proximity threshold is probably different for different users and before implementing such a rule, there should be further investigation into the perception of social situations of users in public spaces and maybe in public transport in particular. The type of interaction with an interactive public display related to the proximity of one or more persons has been discussed by Prante et al. [2003] and Vogel and Balakrishnan [2004], for example, though not with regard to unobtrusiveness of interactions. Rule 7.21 assesses the usage of a public display as obtrusive if a person other than the user is in its proximity. The proxmity expressed as `nearBy` in this rule can be adjusted to any suitable proximity threshold. The rule can be adjusted to work for Smart Windows or other public devices as well.

$$\begin{aligned}(7.21)\quad & uses(?option, ?device) \wedge rdf : type(?device, PublicDisplay) \\ & \wedge isNear(?user, ?device) \wedge differentFrom(?user, UserID) \\ & \implies hasUnobtrusiveness(?option, Obtrusive)\end{aligned}$$

What is socially acceptable is changing over time, is influenced by the cultural background of people involved and depends heavily on various factors. It is hard to explicitly model but the following rules are examples of rules that aim at socially acceptable interactions.

$$\begin{aligned}(7.22)\quad & uses(?option, ?modality) \wedge rdf : type(?modality, AcousticOutput) \\ & \implies hasUnobtrusiveness(?option, Obtrusive)\end{aligned}$$

Rule 7.22 expresses that acoustic output is obtrusive. This rule can be extended or combined with other rules to restrict this assessment to situations in public, for example. Rule B.45, listed in the appendix, determines that acoustic output is unobtrusive, if the user wears headphones. The application of these rules should be discussed for the implementation of a target system, since acoustic output is not uncommon at stops or in vehicle. The impact of potentially obtrusive messages must at this point be weighted against the diversity of available output options.

The second aspect of unobtrusiveness is the cognitive load of the user. Unobtrusive interactions do not interrupt their primary task. However, using public transport, people engage in various different activities. It is not easy to detect the primary task of the user to ensure that the user is not interrupted. There is research about interruptibility and how to detect it using smartphones [Pejovic and Musolesi, 2014; Okoshi et al., 2017; Cha et al., 2020]. Such features can be incorporated in future systems, if necessary. Shirazi et al. [2014] showed, for example, that users judge interruptions by notifications differently based on the content of these messages. If a content analysis of the message is available, it would be possible to introduce rules into the framework that apply such findings. However, for the scope of this work, rules about interruptibility were omitted because of a lack of suitable context data in current public transport systems. The task ontology defined in section 6.3 focuses on mobility tasks.

$$\begin{aligned} (7.23) \quad & uses(?option, ?modality) \land rdf: type(?modality, VisualOutput) \\ & \land hasCurrentTask(UserID, ?task) \land rdf: type(?task, BoardingTask) \\ & \implies hasUnobtrusiveness(?option, Unobtrusive) \end{aligned}$$

The tasks waiting or being in transit are no primary tasks that could be interrupted by a message but boarding or alighting a vehicle are tasks the user is likely focused on as a primary task. Therefore, any message that is only informative and not valid for the exact moment of alighting and boarding, should not distract the user during these tasks.

$$\begin{aligned} (7.24) \quad & uses(?option, ?modality) \land rdf: type(?modality, AcousticOutput) \\ & \land hasCurrentTask(UserID, ?task) \land rdf: type(?task, BoardingTask) \\ & \land hasValidityStartTime(?message, ?start) \land hasValidityEndTime(?message, ?end) \\ & \land greaterThan(Now, ?end) \\ & \implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned}$$

For the rules 7.23, 7.24 and 7.25, visual output is rated as unobtrusive during a boarding task, while vibration or acoustic output is rated as obtrusive. The rules for alighting tasks are similar and are listed in the appendix, section B.2.

$$\begin{aligned} (7.25) \quad & uses(?option, ?modality) \land rdf: type(?modality, HapticOutput) \\ & \land hasCurrentTask(UserID, ?task) \land rdf: type(?task, BoardingTask) \\ & \land hasValidityStartTime(?message, ?start) \land hasValidityEndTime(?message, ?end) \\ & \land greaterThan(Now, ?end) \\ & \implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned}$$

Unobtrusive interfaces should not add to the cognitive load of the user. To express this fact, rule B.52 states that visual output on a device the user is already using, is unobtrusive. To differentiate this assessment, it would be necessary to evaluate the presentation of the message on the device and that would involve evaluating the GUI, which is out of the scope of this work. Rule B.45 can also be interpreted with regard to cognitive load. If the user uses headphones, acoustic output on these headphones can be considered more unobtrusive than output on another device or modality would be. Rule B.53 states that if a user uses a device, a message on another device is considered obtrusive.

As the discussion above shows, evaluating unobtrusiveness is difficult. There is active research that can give insights into aspects of unobtrusiveness, such as interruptibility and that can be incorporated into rules. However, the detection and reasoning of such complex context facts is out of scope of this work. The rules presented above are expressed relatively coarsely and could therefore lead to overly strict filters, leading to empty result lists. They should therefore be considered carefully before being applied.

Assessing Relatability

Relatability can be assessed for a message and for output options. For output options it is possible to rate personal devices with high relatability as expressed by rule 7.26.

$$uses(?option, ?device) \wedge isPublic(?device, false) \implies hasRelatability(?option, HighRelatability) \tag{7.26}$$

Other factors influencing the relatability of output options are if the user uses a certain device, for example. The following rules assess the relatability of a message based on an analysis of the content of the message.

$$\begin{aligned} &hasValidityStartTime(?message, ?start) \wedge hasValidityEndTime(?message, ?end) \\ &\wedge greaterThanOrEqual(Now, ?start) \\ &\wedge lessThanOrEqual(Now, ?end) \\ &\implies hasRelatability(?message, LowRelatability) \end{aligned} \tag{7.27}$$

Rule 7.27 determines the relatability of a message, if it is valid at the current time, `Now`. If the message refers to the current line and direction of the user, rule

B.56, listed in the appendix, sets relatability to `MediumRelatability`. Medium relatability is also assigned if the message references the next stop of the user or their location, as expressed by rules B.57 and B.58, both also listed in the appendix. Rules 7.28 and 7.29 set relatability for a message to `HighRelatability`, if it references the origin or destination of the user's active itinerary.

$$\begin{aligned}(7.28)\quad & hasActiveItinerary(UserID, ?trip) \land hasOrigin(?trip, ?stop)\\ & \land references(?message, ?stop)\\ & \implies hasRelatability(?message, HighRelatability)\end{aligned}$$

$$\begin{aligned}(7.29)\quad & hasActiveItinerary(UserID, ?trip) \land hasDestination(?trip, ?stop)\\ & \land references(?message, ?stop)\\ & \implies hasRelatability(?message, HighRelatability)\end{aligned}$$

And finally, the relatability of the message is also set to `HighRelatability` if it contains the user's first name or last name, expressed in rules B.61 and B.62, both listed in the appendix, section B.2.6.

Assessing Relevance

The relevance of a message to the user depends on whether they need to act based on the information in the message or if the information relates to the user and their journey. In order to determine this, an analysis of the message content is necessary or the provision of certain parameters by the client that requests the usability assessment.

$$\begin{aligned}(7.30)\quad & hasActiveItinerary(UserID, ?trip) \land hasDestination(?trip, ?stop)\\ & \land references(?message, ?stop)\\ & \implies hasRelevance(?message, LowRelevance)\end{aligned}$$

Rule 7.30 infers a low relevance if the destination of the user is referenced by a message. Rules B.64 and B.65 do the same if one of the stops of the user is referenced. A low relevance is also inferred if the current line and direction of the user are referenced by the message, expressed by rule 7.31.

$$(7.31)\quad hasActiveItinerary(UserID, ?trip) \wedge hasTripLeg(?trip, ?leg) \wedge hasDestination(?leg, ?location) \wedge references(?message, ?location) \implies hasRelevance(?message, LowRelevance)$$

Several other parameters of the active itinerary of the user can be similarly used to infer a low relevance. It is also possible to refine this inference using parameters that identify the journey of the user more precisely.

A medium relevance is determined by rule 7.32 using the `isInformative` property that can either be inferred by a more sophisticated message analysis algorithm or be set by the client in its request for assessment.

$$(7.32)\quad isInformative(?message, true) \implies hasRelevance(?message, MediumRelevance)$$

If the message contains a call to action, it is highly relevant to the user. This can be expressed by the property `isCallToAction`, which also can be set by a separate message analysis algorithm or by the initial request for assessment. Rule B.68 determines the relevance level.

Assessing Urgency

The assessment of urgency depends on the current time, expressed as `Now` in the rules. As discussed before, a message is highly urgent if its start time is in the near future, which is expressed as a duration `VerySoon` in rule 7.33.

$$(7.33)\quad hasValidityStartTime(?message, ?start) \wedge lessThanOrEqual(?start, Now) \wedge subtractTimes(?distance, ?start, Now) \wedge lessThan(?distance, VerySoon) \implies hasUrgency(?message, HighUrgency)$$

$$(7.34)\quad hasValidityEndTime(?message, ?end) \wedge greaterThanOrEqual(?end, Now) \implies hasUrgency(?message, HighUrgency)$$

Rule 7.34 determines the urgency of the message as high, if it is still valid. The following rules are listed in the appendix in section B.2.8. If the message will be valid in a near future, `Soon`, rule B.71 sets urgency to `MediumUrgency`, as well as rule B.72, setting urgency to `MediumUrgency` if the end time of the message's validity is later than a given period of time from now. If the message is no call to action or if it starts or ends in a distant future, it is rated as having no urgency, expressed in rules B.73, B.74 and B.75. The sequence of execution of these rules is important when implementing them, to prevent overwriting an already set assessment.

7.4 The Usability Assessment and Decision Making Process

The ontology and assessment rules presented in section 7.2 and 7.3 enable a rule-based assessment of the usability of output options and messages, considering context data. However, the output options that can be assessed must be identified first and after an assessment based on usability rules, a decision for one or more output options must be made.

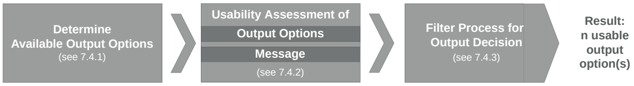


Figure 7.16 — Assessment and decision process for autonmous usability assessment.

This section presents an architecture for the autonomous usability assessment and decision processes. It needs access to up-to-date context data that includes user data as well as public transport data and facts about the situation modeled using the ontologies presented in chapter 6. The framework is intended to provide a service for external clients that request a specific number of output options for a specific message to a specific user. For this request, the client must specify the message and as many message parameters as possible as well as the intended recipient of the message. This information is then used in the assessment process.

Figure 7.16 shows a coarse-grained overview over the phases necessary to reach a decision for an output option. First, available output options are identified and listed, as described in section 7.4.1. Afterwards, the usability assessments are performed, as discussed in section 7.4.2. The assessment of the output options and the message can be performed in parallel or in either order. They are based on the rules discussed in section 7.3. Finally, the rated message and output options are used as input and a decision process is implemented to reach a decision for a usable output option, as described in section 7.4.3.

7.4.1 Determining Available Output Options

As a first phase, all available output options must be identified. Depending on the system architecture, there are various ways to achieve that. In a service-oriented ubiquitous system, this task could be implemented as service discovery, as in the MUSIC system, for example [Rouvoy et al., 2009]. However, as described in section 3.1, most public transport systems do not implement a service-oriented architecture and the available clients are not likely to support service discovery. In the following, I will describe a rule-based context-oriented approach, that relies on available context data to decide which output options are available. Other implementations of this step are possible due to the modular approach. The following assessment step, described in section 7.4.2 needs a list of available output options to build on.

The output options known to the system are represented in the context data modeled after using ontologies from chapter 6. Which of the devices and modalities known in context data is available for the message for a certain user depends on which devices and modalities are with the user or in their vicinity at a given time. However, it also depends on the message and the message's requirement for output. A speech message can not be played back on a device that only has visual output options and therefore excludes these options. The third source of restrictions concerns the accessibility of output options. Options that the user would not be able to perceive must be excluded, as well. These determinants are checked in three steps to get a list of available output options.

Figure 7.17 shows the steps necessary to determine which output options are available. First, the situation match component retrieves possible options using context data. Based on the user's location and on context data about their personal devices and device usage, this component creates a list of output options. It may be useful to classify these options based on the certainty with which their availability can be determined. Later steps of the process can then focus on options that are available with a high certainty, but more options whose availability is known with uncertainty can be considered as fallback options. In the discussion about the certainty of perceptibility ratings on page 163, I discussed modeling uncertainty explicitly, for example using fuzzy logic but decided to focus on a categorical representation to support an easier decision making process.

The message match component in the second step takes the list of output

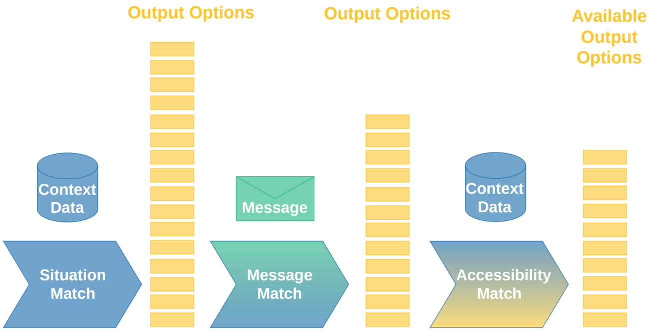


Figure 7.17 — Determining available output options

options that was generated by the situation match component and excludes all options that do not match the requirements of the message. This component matches output options to the message content type. Rule 7.35 includes options with a modality that matches the content type of the message, for example.

$$\begin{aligned}(7.35)\quad & hasContentType(?message, ?content) \wedge uses(?option, ?modality) \\ & \wedge rdf:type(?content, ?contentType) \wedge rdf:type(?modality, ?modalityType) \\ & \wedge sameAs(?contentType, ?modalityType) \\ & \implies include(?option, true)\end{aligned}$$

The message match component can also check if converters are listed that could convert the message's content type so that additional options are possible. Rule B.77, listed in the appendix, includes options that can be used if the content type is converted using an available converter. These rule are illustrative and operate on a abstract level of content types. In a target system, the interaction ontology can be extended for specific content types on file type level and more specific descriptions of converters to map capabilities of output devices, converters and message types. If there is no heterogeneity in message content and file types or if there is an established environment of suitable converters that ensure compatibility, this step can be omitted.

In a third step, the accessibility match component looks up the user's abilities from user context data and matches the available output options to them. Only options that the user is able to sense are included for further analysis. This component applies the accessibility rule discussed in section 7.3 and excludes all inaccessible options.

In this phase, all matching components can exclude output options and therefore it is possible that no output option remains at the end. Since each excluded option in this step is removed because it can not be sensed by the user, is not compatible with the message or is simply not available, because a device is turned off, for example, in this case, the usability assessment comes to a halt and returns no result. This means that no valid output options are available.

7.4.2 Usability Assessment

In the second step of the assessment and decision process shown in figure 7.16, the output options and the message are assessed. The assessment returns rated output options, as shown in figure 7.18 and a rated message, shown in figure 7.19. The output options are assessed regarding their relatability, perceptibility, privacy and unobtrusiveness. The assessment implements the respective rules discussed in section 7.3. These assessments use context data and available output options as their input.

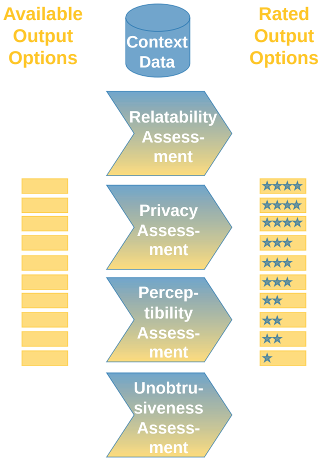


Figure 7.18 — Usability assessment of output options.

Similarly, the message is rated regarding its sensitivity, relatability, relevance and urgency. Figure 7.19 shows this process. The assessment operations also use context data and the message itself, including all properties that are known about the message.

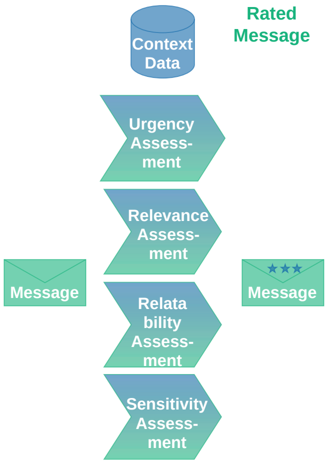


Figure 7.19 — Usability assessment of the message.

If necessary, the message can be analyzed further before its usability assessment, in order to specify parameters that can be used in the assessments. A text analysis can determine, for example, if the name of the user or an address is part of the message. As described before, these message parameters can also be specified by the client requesting an output option. Both assessments can be extended to assess additional usability attributes.

7.4.3 Decision Process

Finally, a decision for one or, if requested, several output options has to be made. At this point, the usability attributes need to be balanced. The tradeoffs can be necessary between conflicting usability attributes, such as a tradeoff between a high urgency of the message and no available output option with a high privacy rating. Most of the time, there also will be a limited amount of available output options. Applying all assessment rules can then lead to an empty result list. To avoid this, the attributes need to bei weighted and it needs to be decided which quality can be omitted, in order to get results and avoid an empty result list. In this decision process, a number of filters use the rated output options and rated message to select one or more options that are as suitable as possible. Figure 7.20 shows this process.

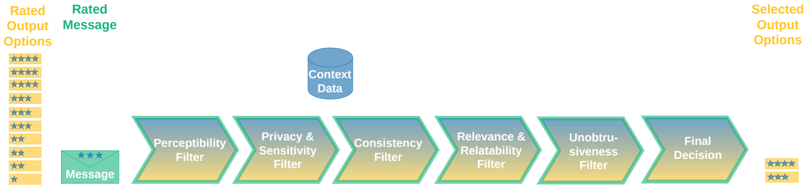


Figure 7.20 — Process for output decision.

The decision process consists of several modular filters, where each filter applies filter rules based on the message and output option ratings. The filters each take the list of rated output options and optionally the rated message as input and return a list of output options that is smaller or the same size as the input list. One filter can consider several attributes interdependently for its filter rules or consider one attribute separately. The filters can be modified, changed or new filters can be defined in the implementation for one target system. In figure 7.20, an exemplary decision process is shown. In this example, relatability and relevance are, for example, evaluated in relation to each other, while consistency is evaluated in a separate filter.

The purpose of these filters is to use decision rules for the exclusion of output options. These rules express a prioritization of usability attributes and trade-offs between usability attributes. How each rating is applied in the decision process can be influenced by context data or, for example, the urgency rating of the message. This enables a differentiated valuation of usability attributes depending on certain parameters of a situation. These details can be adjusted to fit the available context data and usability design decisions of a potential target system.

The first filter in this example process is a perceptibility filter. The rules B.78 and B.79 are listed in the appendix as two examples for such a filter. A more rigorous filter uses only rule B.78 that only includes output options that were rated as perceptible. It discards options that were assessed as probably perceptible. A more relaxed version of this filter uses both rules B.78 and B.79 and includes options that are probably perceptible.

The second filter is a privacy and sensitivity filter that chooses output options whose privacy level fits the sensitivity of the message. A high sensitivity rating requires output options that provide a high level of privacy, which is expressed with rule 7.36.

(7.36)
$$\begin{aligned} &hasSensitivity(?message, HighSensitivity)\\ &\quad\wedge providesPrivacy(?option, HighPrivacy)\\ &\qquad\implies include(?option, true)\end{aligned}$$

A message with a medium sensitivity rating requires a privacy level of high or medium, which rules B.81 and B.82 express. If the sensitivity level of the message is low, then only output options with a low, medium or high privacy are kept. The rules are analogous to the previous rules. As a third filter, a consistency filter uses the consistency assessment described in section 7.3 and keeps all options that were rated as consistent.

Regarding filtering output options, relatability and relevance can be considered interdepentently. If, for example, relatability of a message is medium and the message is highly relevant or medium relevant, then only highly relevant options are considered, as expressed by rules 7.37 and 7.38. If the relatability of the message is low, but the message is highly relevant or medium relevant, the rules are analogous to these.

(7.37)
$$\begin{aligned} &hasRelatability(?message, MediumRelatability)\\ &\quad\wedge hasRelevance(?message, HighRelevance)\\ &\quad\wedge hasRelatability(?option, HighRelatability)\\ &\qquad\implies include(?option, true)\end{aligned}$$

(7.38)
$$\begin{aligned} &hasRelatability(?message, MediumRelatability)\\ &\quad\wedge hasRelevance(?message, MediumRelevance)\\ &\quad\wedge hasRelatability(?option, HighRelatability)\\ &\qquad\implies include(?option, true)\end{aligned}$$

If the message has any relevance, but no relatability, the filter also includes only options with high relatability. The rules analogous to rules 7.37 and 7.38 for this case are listed in the appendix, see section B.2. The unobtrusiveness filter keeps all options that are rated as unobtrusive and discards obtrusive options.

In this decision process, several filters are executed in a row, each possibly removing output options from the list of possible results. It is very well possible that the process terminates before all filters are applied. If one result is requested by the client and after two filter steps, only one option remains, further filtering is not possible. The order in which filters are applied therefore represents a prioritization. If a filter results in an empty list or if following filters are considered important and the process should not terminate without applying these as well, a backtracking mechanism is needed. Filters can be either relaxed, including options that were rejected before, or a filter can be discarded if it yielded no results. In order to evaluate this step and to decide, for example, if a filter can be relaxed, context data or assessment results can be helpful. The urgency of the message can be a good indicator if the privacy and sensitivity filter should be relaxed, for example. Rules for a privacy and sensitivity filter that include urgency as a quality are listed in the appendix, see section B.2.

In such a decision process, the order of the filters is relevant. A different sequence of filters can yield a different result. The filter sequence displayed in figure 7.20 is a suggestion based on the following considerations.

Perceptibility of the output options is highly relevant. Options that are ruled as not perceptible with a high certainty, should not be considered further. If some of the output options are rated as `Perceptible`, these should be favoured over options with a `ProbablyPerceptible` rating, implementing only rule B.78 first. The differentiation of perceptibility can then be used in a backtracking process. If the result list is empty at one point further in the process, options that were rated as `ProbablyPerceptible` can be considered additionally, applying a filter that implements both rules B.78 and B.79. Overall, perceptibility in this example is weighted higher than privacy and consistency, since a consistent or privacy preserving output option that is not perceptible by the user is not usable at all.

Executing the privacy and sensitivity filter before the consistency filter follows a simliar reasoning. A consistent output option that is not privacy preserving any more because the situation changed, would not be a sensitive choice. If a user was receiving a speech message on their smartphone while using their headphones with the smartphone before, but had disconnected the headphones by now, a speech output on the smartphone would be consistent, but might now violate their privacy.

The consistency filter is then prioritized over the relevance and relatability filter and the unobtrusiveness filter, motivated among others by the consideration that a consistent output option will be relatable because of its familiarity. If the number of intended results is obtained after the consistency filter, the relatability and relevance filter can therefore be omitted. The unobtrusiveness filter is the last filter. As discussed in section 7.3, the unobtrusiveness rules are expressed coarsely because of a lack of precise context data to assess unobtrusiveness in greater detail. These rules can cut the number of available output options significantly, which is why the filter is used last. This way, the filtering process does not terminate too often and too early. These prioritizations can change, depending on the target system. The framework's modular structure for the assessment and filter components allows easy rearrangement of filters, if needed. Similarly, a modular implementation of each filter and assessment component is encouraged, separating the execution of each rule. This structure enables a system's engineer or usability expert to modify, replace, remove or complement the rules in each component. Additionally, separated rules and rule execution make results more transparent and increase comprehensibility.

In case the process does not result in the required number of results, a last component enforces the selection of the correct number of options. In figure 7.20 this component is called a final decision component. It can be implemented using any number of additional rules, or just by picking as many entries of the option list as needed. Afterwards, the process can return these options as result to the requesting client and is then terminated.

CHAPTER 8

Implementation

This chapter outlines the prototype implementation of the framework presented in chapters 5, 6 and 7. The framework was developed as a proof of concept implementation using the Java language, the RDF4J library[1], the JAX-RS reference implementation Jersey[2] and the Apache Jena Fuseki server[3].

Figure 8.1 shows the architecture of the prototype. Two components were implemented. One component implements the usability assessment framework. It has a SPARQL interface to access context data and can be invoked by the public transport system using a REST Application Programming Interface (API). For the sake of brevity and transparency, no SWRL engine was used. The adaptation rules were implemented in code, based on SPARQL queries. Since the rules are and can be expressed in SWRL, it is possible to use a SWRL engine for rule execution, if required.

The prototype also implements context generation to generate context data and to write this context data into the triple store using the SPARQL interface. The generated context data is used for testing and evaluation purposes.

The proof of concept implementation does not directly access passenger information systems. However, it is able to process data from passenger information systems based on the TRIAS standard [Englert et al., 2019]. To enable pars-

---

[1] `https://rdf4j.org/`, last accessed October 12th, 2022

[2] `https://eclipse-ee4j.github.io/jersey/download.html`, last accessed October 12th, 2022

[3] `https://jena.apache.org/documentation/fuseki2/`, last accessed October 12th, 2022

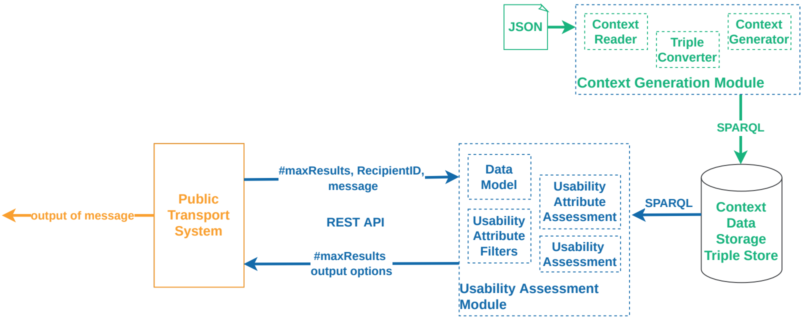


Figure 8.1 — Architecture of the proof of concept prototype.

ing TRIAS data, a library was used that was developed at the Institute for Ubiquitous Mobility Systems[4] and converts TRIAS XML files into Java classes. Additionally, no user interfaces were implemented or used. Available user interfaces can be modeled in context data and are then used in the usability assessment. The result of this assessment is a list of the specified number of output options that are chosen by the assessment component for the context of use that is represented in the context storage.

The implementation of context generation and of the usability assessment will be described in the following sections in greater detail.

8.1 Context Generation

Context data is generated as triples and written into the triple store for the usability assessment module to use. In an implementation of the framework for an existing public transport system, context generation must be replaced by context acquisition, preprocessing and reasoning modules. Approaches for context-awareness in public transport and the application of the context model described in this work were published in Keller et al. [2014b] and Keller et al. [2020].

Context can be provided in two different ways. Specific scenarios can be defined in JSON files or context can be generated randomly by generating a defined number of entities. Context triples are then written and stored in the triple store for the usability assessment module to access.

---

[4] `http://iums.eu`, last accessed October 22nd, 2022

Context Data Storage: Context data is written and stored in a Fuseki triple store. SPARQL `INSERT` queries are used to write triples into the triple store. The usability assessment accesses and retrieves context data using SPARQL queries, therefore relying on a standardized interface. The triple store implementation can be replaced by any other triple store or database that provides a SPARQL interface.

Triple Converter: For each entity that can be represented in context data, a converter exists to create the corresponding triples that express this entity and its properties as a list of triples. Triples are represented as strings, expressing a triple in the Turtle notation. These converters are used by the context generation as well as for loading context data from files.

Context Reader: Context data can be specified in JSON files. This way, context data for specific scenarios can be loaded. Additionally, two types of CSV files contain locations and stops from the Karlsruhe area that can be referenced by their `StopPointRef`, which is the same reference that is used in the TRIAS standard. These are a basis for specific scenarios taking place in Karlsruhe. Scenarios are specified in two JSON files. One file describes static context and a second file describes dynamic context. Static context includes the specification of devices and users and the static context facts for these entities. Static context for devices includes their ids and output functions. Static context about users specifies their id, devices they own, the location of their home, their name and their output abilities and preferences. Dynamic context includes context for vehicles, devices, messages and users that can change dynamically. Examples are locations of entities, if a device is used by a user, the current task of a user or a vehicle's next stop. A context reader is able to parse such JSON files and generate the triples to express the context facts stated in the files. A scenario loader provides a method to load a static and a dynamic context file that belong to a scenario, parse them and then write the resulting triples into the triple store using SPARQL `INSERT` queries.

Context Data Generation: The ability to generate randomized entities as context triples is provided by specialized implementations for each supported type of entity. The device generator, for example, provides methods to generate smartphones, smartwatches, public displays and smart windows. A vehicle generator allows the generation of vehicles at a randomly chosen stop. This stop is chosen from the list of stops in the Karlsruhe area. The user generator provides a method that generates a randomized user. This includes randomized user context such as a random location, personal devices, abilities and preferences.

The generated entities are expressed as triples that can either be written into the context storage or in files to be used at a later date.

8.2 Usability Assessment

The usability assessment implements the framework and the application of adaptation rules. The proof of concept implements the usability attribute categories and the assessment of the message and of output options for each usability attribute. Based on the rated message and output options, the filter rules are applied to filter out the resulting output options. In this case, assessments and filters were implemented regarding timeliness, availability, accessibility, perceptibility, consistency, personalization, privacy and message sensitivity, relatability and relevance. Factors considered in unobtrusiveness assessment as described in section 7.3 are in part covered by privacy and perceptibility rules and rules regarding the user's distraction and mental load often require advanced context data that is not yet available in current systems. Therefore, unobtrusiveness rules were not implemented in this proof of concept. The components that implement the usability assessment are described in the following sections.

8.2.1 Data Model

The data model maps modalities as specified in the interaction ontology described in section 6.4. The usability attribute rating categories are represented as enumerations. An example is the `Perceptibility` enumeration containing the categories `PERCEPTIBLE, PROBABLY_PERCEPTIBLE, NOT_PERCEPTIBLE, NO_RATING`. The rating categories were extended by a `NO_RATING` category to identify output options or messages that have not been rated yet. The data model also implements the device taxonomy described in section 6.5. Each device contains an availablity rating. One device with one or more modalities is implemented as an output option. An output option also contains an availablity rating, a perceptibility rating and a privacy rating. In the message model, it is possible to make references the message contains explicit, using message attributes. A message also contains a list of possible modalities that can be used to present the message. Additionally, a message contains a sensitivity rating, a relatability rating, relevance rating and an urgency rating.

8.2.2 Message Content Analysis

The proof of concept contains a short message content analysis implementation to extract information from a text message. The information is used in the usability assessment of the message, for example in the sensitivity assessment. In the prototype, the message analysis determines if sensitive information is contained in the text message, such as the user's name or an address, for example. SPARQL queries are used to determine context information of the user and it is then analyzed if this information is expressed in the message. The analysis is done using simple string comparisons.

8.2.3 Usability Assessments and Filters

The actual usability assessment consists of a sequence of assessments and filters that are applied successively. The assessments and filters implement a subset of the adaptation rules expressed in sections 7.3 and 7.4. The following paragraphs list the corresponding rules and discuss their implementation. The sequence of paragraphs corresponds to their sequence of execution.

Message Urgency: The urgency assessment of the message is an assessment that contributes to the usability attribute *timeliness*. The urgency assessment implements all urgency rules that are listed in the appendix section B.2.8. Rule B.73 is executed first, so that all urgency ratings only take effect if the message is a call to action, otherwise it is rated as having no urgency.

Availability: This component implements the situation match described in section 7.4.1. It retrieves available output options from the context storage. In this implementation, output options are rated as `AVAILABLE` and `PROBABLY_AVAILABLE` or are rejected as not available options. First, available devices are found, which is implemented differently based on device type. A smartwatch is, for example, rated as probably available if the user does own a smartwatch and as available if context data determines that they wear it. A smartphone is available if the user owns one. For displays in vehicles or on stops, the component determines if the user is in a vehicle or at a stop equipped with such a display. Further tests determine if there is context data available indicating that the user is near such a display or if they are using it. Each device stores its own availability rating. After that, the modalities of each device are requested from the context storage. For each combination of device and modality an output option is created. If less than or exactly the number of requested options are available after this step, these options are returned. If no available option is found, the process terminates without results.

Message Requirements: After finding available output options, the message match component is executed, as described in section 7.4.1. It implements rules

B.76 and B.77, matching the message format to the modalities of the available output options. It also checks if a converter is available that can transform the message to another format, enabling the usage of additional modalities. All output options that can not be used with the message format are then removed from the list of possible options. If no option remains, the process halts without results and if the amount of requested options or less remain at this point, those are returned.

Accessibility: The accessibility filter removes all options whose modalities do not match with the abilities of the user. If this removes all options and no option is left, the process stops without results. If the amount of requested options or less are left then the remaining options are returned as a result.

Perceptibility: The next step is the perceptibility assessment of all options. This assessment is implemented differently for each device type. Possible categories for perceptibility are `PERCEPTIBLE`, `PROBABLY_PERCEPTIBLE`, `NOT_PERCEPTIBLE`, `NO_RATING`. This component implements rules B.6 through B.10 for smartphones and adds that acoustic output is not perceptible if none of the rules for acoustic output apply. The rules for smartwatch listed in section B.2.2 are implemented, complemented by a fallback option that if none of these rules apply, an option using a smartwatch is rated as `PROBABLY_PERCEPTIBLE`. The rules for peceptibility of public displays as listed in the appendix section B.2.2 are implemented for public displays. For vehicle displays, the rule B.20 is implemented, meaning that if the user is near the display, it is rated as perceptible. If this rule does not apply, the vehicle display option is ruled as probably perceptible. Section B.2.2 lists the implemented rules for smart windows. In addition, if a user is not near the display, it is rated as not perceptible for visual output. After perceptibility assessment, a perceptibility filter removes all options that were evaluated as not perceptible, represented by the rules in section B.2.10 in appendix B. If no options remain, the process halts. Since this would mean that no perceptible output options are found, the usability assessment would return no result. If after this perceptibility assessment the number of options left are equal or less than the number of requested results, these are returned, otherwise the process continues.

Consistency: The consistency assessment implements consistency rules B.3 and B.4. All options that are not rated as consistent are then discarded. If the resulting options are matching the number of requested results, these options are returned as a result. If no option is left after the consistency filter, the process is continued with all options from before the consistency assessment. If no consistent option can be found, all available options should be considered further.

Personalization: The personalization filter removes all options that are not matching the user's preferred modalities. If the requested amount of options or less remain after this step, they are returned as a result. If no options remain, the process reverts back to the list of options that was the result of the consistency filter step and the personalization step is omitted. This follows the implication that passenger information messages should reach the user, because they are important for the user's journey. Therefore, if none of the options available are preferred by the user, the message should still be delivered.

Privacy and Sensitivity: In a next step, privacy and sensitivity are evaluated. First, a privacy assessment of all options and a sensitivity assessment of the message are performed. The privacy assessment implements all rules found in the appendix in section B.2.3. The message sensitivity assessment implements the rules listed in section B.2.4. In the implementation, the sensitivity level of rules B.35 and B.36 was set to `MEDIUM_SENSITIVITY` instead of `LOW_SENSITIVITY`. These rules determine if the user's destination or origin are identified in the message. After the privacy and sensitivity assessment, a privacy and sensitivity filter applies the filter rules B.80 to B.86, listed in section B.2.11. If, after this filter, the remaining number of options matches the requested number of options, they are returned as a result. If there are more options available, the process continues. However, if no options remain, a relaxed privacy and sensitivity filter is applied that takes into account the message's urgency. This filter implements rules B.87 to B.95. If this filter also returns no options, the process is halted without results. If the requested amount of results remains, these options are returned as results. Otherwise the process continues.

Relatability and Relevance: In this step, a relatability assessment and a relevance assessment of the message is performed. The relatability assessment implements rules B.55 through B.62 listed in section B.2.6 while the relevance assessment executes rules B.63 through B.68 listed in section B.2.7. After these assessments, the relatability and relevance filter evaluates both attributes together. This filter keeps all options with high relatability and additionally implements all filter rules listed in section B.2.12. If after this step no options remain, the results of this step are discarded and the next step will be performed with the result of the privacy and sensitivity filter. If the number of options left matches the number of requested results, these are returned. In all other cases, the next and last step is performed.

Results and Final Assessment: The final assessment step is performed in case that after all of the previous assessment and filter steps still more options are left over than are requested as results. All filters are only applied if options are left afterwards, otherwise the filters are skipped. In this proof of concept, the filters include a removal of all options that are only probably perceptible, the removal of all options that have a lower privacy rating than high privacy and after these steps the choice is the user's smartphone. If several modalities are available for the smartphone, the visual output is chosen.

The proof of concept implementation uses log output to allow an understanding and engineering of rules. This implementation was used for the performance test and usability evaluation that are described in the next chapter.

Chapter 9

Evaluation

This chapter reports on the evaluation of the approach for adaptive information provision presented in this work. The requirements described in section 3.4 provide the basis for this evaluation. All requirements are listed in table 3.1 for reference. The evaluation uses several scenarios. These evaluation scenarios represent the user tasks discussed in section 3.2 and use the persona described in section 3.3. Each scenario consists of a description, a picture illustrating the situation, the relevant context data as well as the message that is to be delivered to the persona in the scenario.The scenarios are listed in C.1 in the appendix. The evaluation was done in three parts.

Section 9.1 discusses the proof of concept implementation, its application to the scenarios and the fulfilment of requirements 3 to 7 and 9 to 16. Section 9.2 presents a performance evaluation and examines requirement 2.

In section 9.3, the usability evaluation and its results are presented. The evaluation scenarios were presented to participants during an online questionnaire. The proof of concept implementation was applied to the scenario context data and participants of the survey rated the framework's output decision and a baseline output. In section 9.3.3, the fulfilment of requirement 1 and the Main Research Question is discussed.

Section 9.4 summarizes the evaluation and discusses the requirements concerning the ontologies, including requirements 7, 8 as well as 13 and 14. It also gives an overview over all requirements and concludes that they alle have been met by the approach presented in this work.

9.1 Proof of Concept Implementation

The proof of concept prototype described in chapter 8 implements the framework for adaptive information provision as a modular and loosely coupled component. The assessment module only uses a SPARQL interface to communicate with context storage and provides a REST API to request a usability assessment. More interfaces and components are not needed. Requirements 10, autonomous assessment, and 11, loose coupling, are therefore met.

The assessment is triggered by a passenger information system that requests a usability assessment for a given message and user. This fulfils requirement 6, system triggered adaptation. During an assessment, context data is requested and evaluated, making the assessment context-aware. It therefore also meets requirement 5, context-awareness.

The assessment is implemented in a modular way, with separate assessment and filter modules. The prototype implements a subset of the usability assessment and filter rules that are presented in chapter 7. The rule set can be extended or rules can be changed by changing the respective module implementing the rules. A rule engine can be used in the usability assessment and can provide more flexibility, if needed. Requirement 14, extendability of adaptation rules, therefore is met.

The usability assessment decides which device and modality or modalities are best suited to deliver the message to the given user based on the given context data. The framework is able to choose devices and modalities, requirements 3 and 4 are therefore fulfilled.

The results of the assessment are one or more device and modality choices that a passenger information system can use to deliver the message. The message is then delivered using existing user interfaces of the chosen devices. All types of UI and also legacy systems can be integrated this way. The user interfaces need not to be extended beyond the ability to receive and render a message. The framework therefore meets requirement 9, support for legacy systems. New devices and modalities can easily be included, they just have to be represented in context data and be addressable for the system component that delivers the messages. Requirement 15, extendability of user interfaces is therefore met, too.

The prototype shows that implementing a usability assessment for a smart mobility systems requires a context management component that provides context data in a context storage with a SPARQL interface. Context acquisition, processing and reasoning can be implemented using existing system components and existing interfaces as context sources, which requires no or only little changes to existing system components. Context management provided, the approach can be used in addition to existing systems and it does not require a reimplementation of system components. It therefore has low prescriptiveness, meeting requirement 12.

9.2 Performance Evaluation

Requirement 2 demands adaptation at runtime. This means that the usability assessment service must respond to a request in a reasonable amount of time that allows its call at runtime, whenever a message needs to be delivered to a user. An important performance factor for a usability assessment service is the number of SPARQL queries it uses and the response time of these queries that can cause a latency for the responses of the assessment service. The response time of SPARQL queries varies based on the number of triples that are stored in the triple store. Therefore, the response time of the usability assessment service was measured for different numbers of triples in the triple store. This performance evaluation was executed on a laptop with 32 GB RAM memory and an Intel Core i7-8750H processor.

The usability assessment service was applied to the six evaluation scenarios described above. During this performance evaluation, entities were randomly generated and the triples defining these entities were written into the Fuseki triple store. The number of entities and triples was increased in one hundred steps and the response time of the service was measured at each step. At each step, 350 users, 350 vehicles and 350 devices were generated. Suitable properties were randomly added to each entity, which leads to a difference in the exact amount of generated triples. Roughly between 13,500 and 14,000 triples were added in each step. After 100 of these steps, on average, 1.3 million triples were generated in total. At each step, the usability assessment service was called ten times and the execution time of the service was measured each time.

For scenario 1, a number of 1,377,893 triples was generated at the last step, for example. These triples define, among other context data, 35,008 users, 6,916 vehicles and 35,011 devices. As a comparison, Karlsruher Verkehrsverbund, the local transport association for the city and region of Karlsruhe, had 1,098 vehicles in use in 2021 [Krauth, 2021]. In 2019, before the Covid-19 pandemic impacted the number of passengers, KVV had 166.7 million passengers, which translates to roughly 456,712 passengers per day [Krauth, 2021]. However, passengers are counted per vehicle, so individuals are counted multiple times. Of individual passengers, only a fraction use the mobile app KVV.regiomove[1].

---

[1] https://www.kvv.de/fahrkarten/verkauf/regiomove.html, last accessed September 25th,

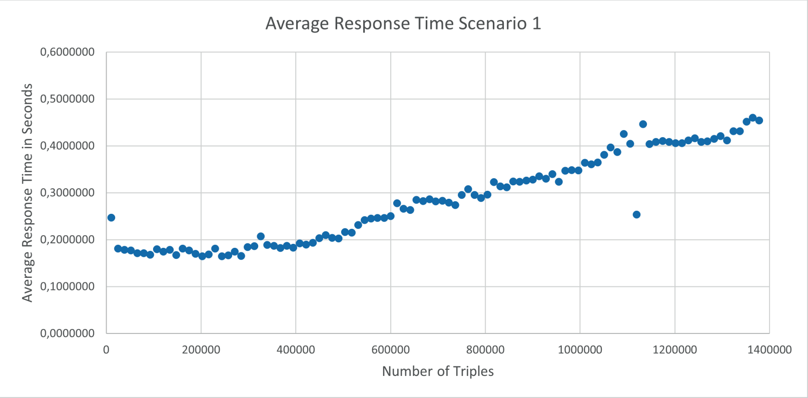


Figure 9.1 — Results of the performance evaluation using scenario 1: average response time of ten test runs of the usability assessment service per number of triples in the context triple store.

The manufacturer of this app states on their website that the app has around 60,000 users across all platforms (Android and iPhone)[2]. Since many users only use these apps to access information but do not use push information, the number of users that was used for this performance test was set to roughly half of these 60,000.

For scenario 1 the response time reached the maximum of all test runs and all scenarios, with a maximum response time of 0.69 seconds. Figure 9.1 shows the results for scenario 1. The response times for the 10 test runs at each amount of triples were averaged for this graph. All results for scenarios 1 to 6 are shown in the appendix in section C.2. The results show that for four scenarios the average response time is below 0.3 seconds for the maximum amount of triples. For scenario 3, it is below 0.4 seconds and scenario 1 has an average response time below 0.5 seconds at the maximum number of triples. For all scenarios, the results show a linear and relatively flat increase in response time.

Since the adaptations need not be executed during an existing interaction with the user, but are used for messages the system sends proactively, the response time of the service does not delay any user interactions. Limits that apply to bidirectional interactions do not have to be met in this case. For the

---

2022

[2] `https://www.raumobil.com/referenzen/regiomove`, last accessed September 25th, 2022

given scenarios and number of entities, the usability assessment service can be executed at runtime and reaches an adaptation decision in a time acceptable for its purpose. It is therefore possible to use this approach and this prototype implementation or comparable implementations in real circumstances and scenarios.

If the service is expected to meet higher expectations in terms of response time or number of users, there are triple stores that are explicitly built for scalability, such as Virtuoso[3] and AllegroGraph[4], for example. Faster servers that use more working memory and more powerful processors can also be used. Scalable and fast servers with scalable triple stores can be expected to significantly reduce the response time of the usability assessment service. Based on the results of this performance evaluation, requirement 2, adaptation at runtime, is met.

9.3 Usability Evaluation

This part of the evaluation approach targets requirement 1 and the Main Research Question, both focusing on usability. The Main Research Question can be answered positively if a system for device and modality adaptation is able to ensure adequate usability. Section 1.2 discussed adequate usability as a better usability than a static approach can achieve. The goal of this usability evaluation is therefore to determine if the framework presented in this work can achieve comparable or even better usability than a static approach such as current solutions to passenger information in public transport.

This evaluation focuses on the usability of the device and modality choice. It is important to note that the usability of the framework's choices depend on the implemented rules. As the set of rules used with the framework depends on the specific implementation and corresponding design choices, the usability achieved by any implementation of the usability assessment service can change. This evaluation is therefore only able to establish if it is *possible* to achieve adequate usability using the framework. In another implementation, the set of rules should be thoroughly tested to ensure a good fit and adequate usability.

The realized evaluation approach consists of an A/B test implemented in an online questionnaire. In this questionnaire, the evaluation scenarios are presented to participants and they are asked to rate an output option for each given scenario. This approach uses output options chosen by the proof of concept implementation of the framework presented in chapter 8 and compares

---

[3] `https://virtuoso.openlinksw.com/`, last accessed September 25th, 2022

[4] `https://allegrograph.com/products/allegrograph/`, last accessed September 25th, 2022

them to baseline output options in a between-subject test. The rationale for this evaluation approach is outlined in the following paragraphs.

Choice of Baseline: The baseline for comparison can either be an adaptive passenger information system that does not use usability metrics to reach an adaptation decision. As an alternative, the baseline system could be not adaptive at all. An adaptive system would introduce various side-effects and produce results hard to interpret. In this evaluation, the framework' output option choices are compared to the static output choices of passenger information systems mostly used and implemented at present. Since this evaluation focuses on device and modality choice, the baseline for comparison is the device and modality choice that is used most frequently in passenger information systems that are able to deliver personalized information at the present, which is text output on the passenger's smartphone.

Choice of Test Setup: In a field test, a multitude of factors would influence the user's experience of the system, for example other passengers and the situation in public transport, such as vehicle occupancy or delays. And while context factors are highly relevant to the adaptation of output by the framework, in a field test they can not be controlled. For a comparison of output choices, the situations in which participants rate these choices should be comparable. In a field test, achieving two comparable situations without confounding factors would be very difficult.

User tests in a lab could provide a more controlled setting. However, implementing two systems with several output options for the framework to choose would be very costly. With regard to comparisons of prototypes and real applications in usability tests, Beul-Leusmann et al. [2014] note that their prototype passenger information app has much less features and functionality compared to the existing app they used as a comparison (the app DB Navigator[5], widely used for long-distance rail transport in Germany). The authors state that the difference between a prototype and a real application can be a problem during evaluation.

Additionally, since two comparable applications would have to have several implemented user interfaces the user can interact with, the design of these user interfaces might interfere with the evaluation of output choices. It is highly likely that a test user would rate the user interface design of each output device rather than the output option choice or that the user interface design would at least influence the user's assessment.

---

[5] `https://www.bahn.de/service/mobile/db-navigator`, last accessed September 25th, 2022

An abstraction from specific user interfaces can help to separate the evaluation of the usability of an output choice from the specific user interface. Such an abstraction complicates task performance metrics, because of a lack of function and design. It can, however, be used with self-reported metrics. An online questionnaire can describe scenarios and context factors, device and modality choices and ask the participant's rating of these choices. While a lab or field test would allow for much more realistic situations, an online questionnaire allows to test a greater variety of situations that would be difficult or costly to recreate in a lab or field test. Since the subject of this evaluation is a context-aware framework for the adaptation of output choices due to changes in the context of use, testing a variety of situations is appropriate.

Choice of Metrics: Generally, several types of metrics can be used to assess usability, for example, task success, time on task, errors or self-reported metrics (cf. [Tullis and Albert, 2013]). For the evaluation of the usability assessment framework, these metrics have major drawbacks. Performance metrics, such as task success, time on task or errors are depending on the performance of suitable, representative and comparable tasks. For the evaluation of the assessment framework, these would be public transport tasks. In a field test, the effects of information output and an adaptation of output options on these metrics would, however, be hard to differentiate from various other influences, such as passenger interaction, vehicle properties, the public transport situation itself and so forth. In a lab test, context factors that do play a role in task performance in real public transport scenarios could not be replicated adequately and task performance measured in the lab setting would presumably not be comparable to performance in reality. A usability score measured in the lab would therefore probably not represent the usability of the system in real world application.

Self-reported metrics are a better fit for this evaluation goal and the application domain. A questionnaire can illustrate a situation and inquire the user's attitude and perception of the information output in this situation. This means that the effectiveness and efficiency of a system using the assessment framework can not be measured adequately in this evaluation approach, but the user's satisfaction and their expectation of effectiveness and efficiency can be investigated.

9.3.1 Survey Setup

For this evaluation, two online questionnaires were implemented using SoSci Survey[6]. The questionnaires were implemented in German, targeting a German-

---

[6] `https://www.soscisurvey.de/`, last accessed September 25th, 2022

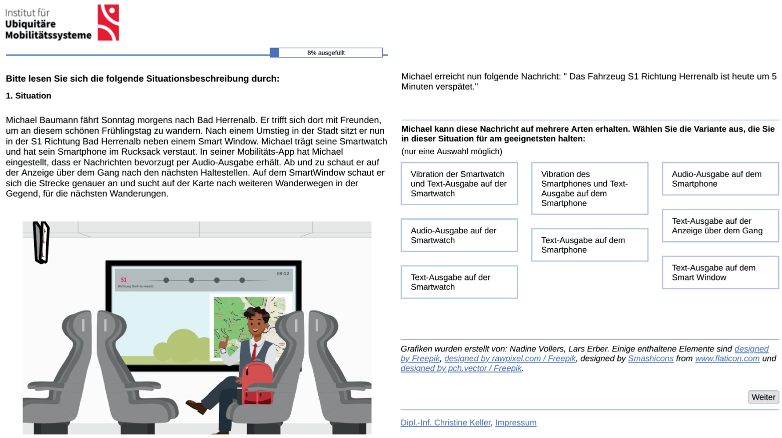
speaking audience. One questionnaire presented only output options chosen by the proof of concept implementation, the other questionnaire presented text output on the smartphone as baseline option. A starting page introduced the purpose of the questionnaire and explained that there were no prerequisites for participating in the survey. Six vouchers for a german online bookstore were advertised as incentives to fill out the questionnaire and were drawn randomly from all participants who voluntarily provided an e-mail address. The prizes were advertised on the starting page of the questionnaire to attract participants.

SoSci Survey allows to randomize questionnaire selection in A/B test settings, which was used to randomly choose one of the two questionnaires for each participant. After the welcoming starting page, all participants were shown a second page with further explanations about the questionnaire, the type of questions to expect and explanations of terminology.

In total, six scenarios with each one framework choice of output device and modality and one baseline choice were prepared. The scenarios used the personas Michael Baumann, Martina Grundler and Maria Ziegler that are described in section 3.3. All scenario texts and figures are listed in the appendix, section C.1. The scenarios were designed to represent each public transport task at least once and to contain messages of different levels of urgency. Scenario 6 resulted in the same output choice from the framework as in the baseline, namely text output on the smartphone. The scenario was used nevertheless as a validation scenario. If there would be a difference in ratings for the framework and baseline output choices, in scenario 6 there should in turn be no difference.

Each participant was shown three out of six scenarios. The first page for each scenario displayed a scenario description in text and a corresponding picture. Then the message the persona in the scenario receives was stated, without specification of how the message would reach the persona. Below, the participant was shown a list of all possible combinations of output devices and modalities that are available the scenario. The participant was then asked to select the variant of output they found the most suitable. This page of the questionnaire can be seen in figure 9.2.

On the next page, the participants were again shown the same scenario text and picture as before as well as the message the persona should receive. Below, an explanation was shown presenting the output option the system had chosen. The displayed choice depended on the type of questionnaire and was either the choice of the framework or the baseline choice. On this same page, the participant was then asked how they would assess the situation for the persona.



Bitte lesen Sie sich die folgende Situationsbeschreibung durch:

1. Situation

Michael Baumann fährt Sonntag morgens nach Bad Herrenalb. Er trifft sich dort mit Freunden, um an diesem schönen Frühlingstag zu wandern. Nach einem Umstieg in der Stadt sitzt er nun in der S1 Richtung Bad Herrenalb neben einem Smart Window. Michael trägt seine Smartwatch und hat sein Smartphone im Rucksack verstaut. In seiner Mobilitäts-App hat Michael eingestellt, dass er Nachrichten bevorzugt per Audio-Ausgabe erhält. Ab und zu schaut er auf der Anzeige über dem Gang nach den nächsten Haltestellen. Auf dem SmartWindow schaut er sich die Strecke genauer an und sucht auf der Karte nach weiteren Wanderwegen in der Gegend, für die nächsten Wanderungen.

Michael erreicht nun folgende Nachricht: " Das Fahrzeug S1 Richtung Herrenalb ist heute um 5 Minuten verspätet."

Michael kann diese Nachricht auf mehrere Arten erhalten. Wählen Sie die Variante aus, die Sie in dieser Situation für am geeignetsten halten:

(nur eine Auswahl möglich)

Vibration der Smartwatch und Text-Ausgabe auf der Smartwatch
Audio-Ausgabe auf der Smartwatch
Text-Ausgabe auf der Smartwatch
Vibration des Smartphones und Text-Ausgabe auf dem Smartphone
Text-Ausgabe auf dem Smartphone
Audio-Ausgabe auf dem Smartphone
Text-Ausgabe auf der Anzeige über dem Gang
Text-Ausgabe auf dem Smart Window

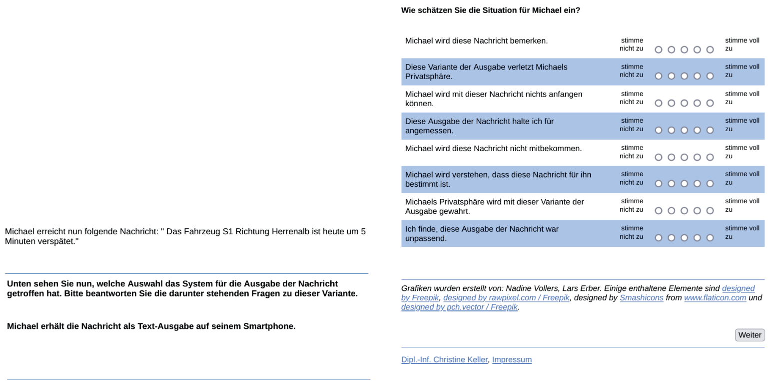
Grafiken wurden erstellt von: Nadine Vollers, Lars Erber. Einige enthaltene Elemente sind designed by Freepik, designed by rawpixel.com / Freepik, designed by Smashicons from www.flaticon.com und designed by pch.vector / Freepik.

Dipl.-Inf. Christine Keller, Impressum

Figure 9.2 — An example of the first page for a scenario in the online questionnaire.

Eight statements about the output choice were given, which were the same for each scenario. There were four pairs of statements and each pair addressed one usability attribute, one question in a positive manner and one in a negative manner. These pairs also served as control questions to see, if participants answer consistently. The participants were asked to answer on a likert scale of five steps from "do not agree" to "fully agree". The scales were aligned the same for each statement. The statements are shown in figure 9.3 and are listed in the following. They use the name of the persona in scenarios 1, 3 and 5. The name in the statements was adapted for the other scenarios.

1. "Michael will notice this message." - ("Michael wird diese Nachricht bemerken"), *perceptibility*
2. "This variant of output violates Michael's privacy." - ("Diese Variante der Ausgabe verletzt Michaels Privatsphäre."), *privacy*
3. "Michael will not be able to relate to this message." - ("Michael wird mit dieser Nachricht nichts anfangen können."), *relevance and relatability*
4. "I consider this output appropriate." - ("Diese Ausgabe halte ich für angemessen."), *overall usability*
5. "Michael will not be aware of this message." - ("Michael wird diese Nachricht nicht mitbekommen"), *perceptibility*



Michael erreicht nun folgende Nachricht: " Das Fahrzeug S1 Richtung Herrenalb ist heute um 5 Minuten verspätet."

Unten sehen Sie nun, welche Auswahl das System für die Ausgabe der Nachricht getroffen hat. Bitte beantworten Sie die darunter stehenden Fragen zu dieser Variante.

Michael erhält die Nachricht als Text-Ausgabe auf seinem Smartphone.

Wie schätzen Sie die Situation für Michael ein?

Michael wird diese Nachricht bemerken.	stimme nicht zu	○	○	○	○	○	stimme voll zu
Diese Variante der Ausgabe verletzt Michaels Privatsphäre.	stimme nicht zu	○	○	○	○	○	stimme voll zu
Michael wird mit dieser Nachricht nichts anfangen können.	stimme nicht zu	○	○	○	○	○	stimme voll zu
Diese Ausgabe der Nachricht halte ich für angemessen.	stimme nicht zu	○	○	○	○	○	stimme voll zu
Michael wird diese Nachricht nicht mitbekommen.	stimme nicht zu	○	○	○	○	○	stimme voll zu
Michael wird verstehen, dass diese Nachricht für ihn bestimmt ist.	stimme nicht zu	○	○	○	○	○	stimme voll zu
Michaels Privatsphäre wird mit dieser Variante der Ausgabe gewahrt.	stimme nicht zu	○	○	○	○	○	stimme voll zu
Ich finde, diese Ausgabe der Nachricht war unpassend.	stimme nicht zu	○	○	○	○	○	stimme voll zu

Grafiken wurden erstellt von: Nadine Vollers, Lars Erber. Einige enthaltene Elemente sind designed by Freepik, designed by rawpixel.com / Freepik, designed by Smashicons from www.flaticon.com und designed by pch.vector / Freepik.

Dipl.-Inf. Christine Keller, Impressum

Figure 9.3 — An example of the second page for a scenario in the online questionnaire, including the statements about the output choice.

6. "Michael will understand that this message is meant for him." - ("Michael wird verstehen, dass diese Nachricht für ihn bestimmt ist.", *relevance and relatability*
7. "Michael's privacy is preserved with this variant of output." - ("Michaels Privatsphäre wird mit dieser Variante der Ausgabe gewahrt."), *privacy*
8. "I think this output of the message was inappropriate." - ("Ich finde, diese Ausgabe der Nachricht war unpassend."), *overall usability*

Each participant was presented with three scenarios to keep the questionnaire short enough for participants to not abort. The three scenarios were chosen randomly from the six available scenarios and their order of presentation was randomized to avert learning effects. After the presentation and questions about three scenarios, the participants were asked about demographic data, their public transport usage and their smartphone and smartwatch usage.

The questionnaire was distributed among students of a bachelor and master degree in mobility management, but also on social media and among personal contacts.

9.3.2 Results

In total, 200 participants finished the questionnaire. Each questionnaire was completed by 100 participants. After checking for repetative and inconsistent

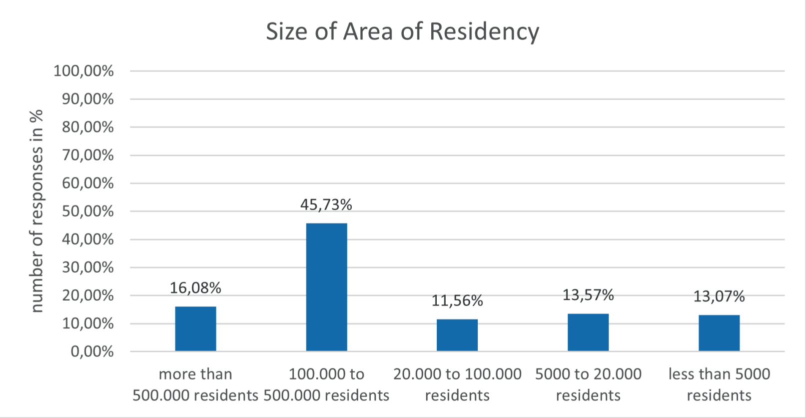


Figure 9.4 — The answers of participants to questions about the size of their area of residency.

answers in the questionnaire, using the control questions and dwell times, the answers of one participant were excluded. The answers were predominantly inconsistent with answers to the control questions and dwell times were notably short compared to the times of other participants. This resulted in 100 questionnaires being analyzed for the baseline and 99 questionnaires analyzed for the framework and in the following numbers of datasets for each scenario:

- **Scenario 1 (A Delay on the Commute):** 42 datasets for the baseline and 55 for the framework.
- **Scenario 2 (Alight on the Next Stop going Shopping with the Kids):** 43 datasets for the baseline and 54 for the framework.
- **Scenario 3 (A Detour on the Commute):** 54 datasets for the baseline and 48 for the framework.
- **Scenario 4 (Platform Changes at the Station):** 58 datasets for the baseline and 43 for the framework.
- **Scenario 5 (A Delay during a Leisure Trip):** 57 datasets for the baseline and 41 for the framework.
- **Scenario 6 (Alight on the nex Stop while Being Busy):** 46 datasets for the baseline and 56 for the framework.

The participants were aged 18 years to 71 years (average age: 34.22 years).

One participant reported their age as ′3′, which was replaced by the average age, 34 years. One participant stated to not use a smartphone, all other participants indicated that they used a smartphone. The majority of participants stated that they feel very confident or confident in using a smartphone (184 participants). 164 of all participants also felt comfortable or very comfortable using public transport apps on their smartphone. 19 out of 24 participants that use a smartwatch reported that they feel very confident or confident using a smartwatch.

The participants were also asked about the size of their area of residency, since the structure of public transport systems differs between rural and urban areas, for example. Figure 9.4 shows that most of the participants live in a city with 100,000 to 500,000 residents. This result is likely due to the fact that the questionnaire was mainly advertised to students and other people living in Karlsruhe, a city in this size range. In another question they were asked about their knowledge of the area they use public transport. Most participants have good or very good knowledge about the area they use public transport in, as can be seen in the appendix, in figure C.14.

In the questions about public transport, participants were specifically asked about their usage of public transport outside the Covid-19 pandemic. The graph in figure C.15 in the appendix shows that many people only rarely use public transport, but a total of 40.2% of participants stated that they use public transport at least several times a week. In a representative study on mobility in Germany in 2017, infas Institut für angewandte Sozialwissenschaften GmbH [2018] reported that 26% of interviewees use public transport at least one to three times a week. In this study, 41% of interviewees stated that they never or nearly never use public transport, whereas only 2.5% of my questionnaire reported to use public transport not at all. This comparison shows an overemphasis of public transport users in the questionnaire. Since adaptive passenger information systems aim at public transport users, this overemphasis does not impair the results of the questionnaire in my opinion.

The participants were also asked about their knowledge of the public transport system they use most often and most responded that they have good or very good knowledge, also shown in figure C.16 in the appendix.

The participants then chose purposes they use public transport for from a list and they stated which media they use for passenger information. Most participants use public transport for leisure and mostly rely on digital passenger information in apps or on websites.

In summary, the participants are mostly living in cities, are familiar with public transport and comfortably use digital passenger information systems. They are also very confident in using smartphones. It can be assumed that most of the participants are not likely to be hostile to new technologies.

Participant's Output Choices

For each scenario, the participants were first asked, which of the available output options they would choose in the described situation. Figure 9.2 shows this question and the options to answer for scenario 1. The participants were only allowed to choose one option. The available options depended on the scenario.

Figures 9.5 and 9.6 show how many percent of the participants that were shown a specific scenario chose each of the available output options for scenario 1 and 5 as examples. In section C.3, the results are shown for each scenario. In some cases, available output options were not chosen by any participant and these options are therefore displayed in the results. In scenario 1 and scenario 3 the baseline option was not selected by any participant.

Overall, it is notable that the option using vibration of the smartphone combined with text output performed better than the baseline option withouth vibration in all but one scenario. However, since the vibration and text output on the smartphone is not very much different from the baseline option, it should be taken into account when interpreting the results.

In scenario 1 (A Delay on the Commute), as shown in figure 9.5, the majority of participants (67.01%) chose the same option as the framework. The baseline was not chosen at all and the vibration and text output option on smartphone was only chosen by 7% of participants. In scenario 2 (Alight on the Next Stop going Shopping with the Kids), the option chosen by the majority, 72.16% of the participants was also the framework's choice. The baseline option was chosen by 4.12% of participants.

For scenario 3 (A Detour on the Commute) the option the framework decided for was only chosen by 9.80% of participants. The baseline option was not chosen by participants at all and the option of vibration and text on the smartphone was chosen in 10.78% of cases. Both baseline and framework did not perform well in this scenario and while the framework option was chosen more often than the baseline, the baseline plus vibration variant was performing slightly better than the framework.

In scenario 4 (Platform Changes at the Station), the baseline was chosen by 4.95% of participants and the option of vibration and text on the smartphone was chosen by 44.55% of participants. The framework option was chosen by 46.53% of participants. Results for scenario 5 (A Delay during a Leisure Trip) are shown in figure 9.6. In this scenario, the most frequently selected option was audio

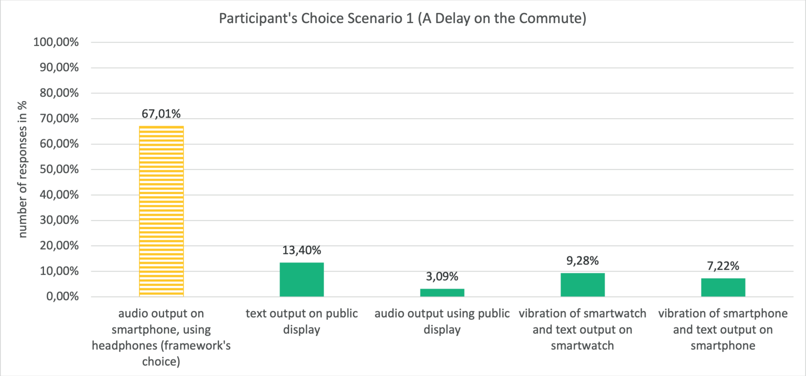


Figure 9.5 — The choices by participants for scenario 1: a delay on the commute. The framework's choice is shown in yellow stripes and the baseline is shown with blue zigzag lines.

output on the smartwatch with 26.53%, followed by "vibration and text output on smartwatch" with 22.45% and the framework's choice "text output on smart window", chosen by 22.45% of participants. The output on the vehicle display on the ceiling was chosen by 20.41% of participants. This distribution shows that participant's opinions differ widely on the suitable output choice in this scenario. The baseline was only chosen by 4.08% of respondents. In scenario 6, the baseline and framework option were the same, selected by 19.61% of participants. 48.04% chose vibration and text output on the smartphone for this scenario.

In all five scenarios where the framework and baseline options were different, the framework's choice was chosen more often than the baseline. Comparing to vibration and text on smartphone, the framework was selected more frequently in four scenarios. In three scenarios (scenario 1, 2 and 4), the majority of participants chose the same output option as the framework.

Usability Evaluation of Output Choices

The second set of questions participants were asked for each scenario assessed the usability of the baseline and framework output options. The participants answered the statements presented in section 9.3.1 on a Likert scale with five options. The extremes were marked as "do not agree" and "fully agree". There is an ongoing discussion about the question if such data can be treated and

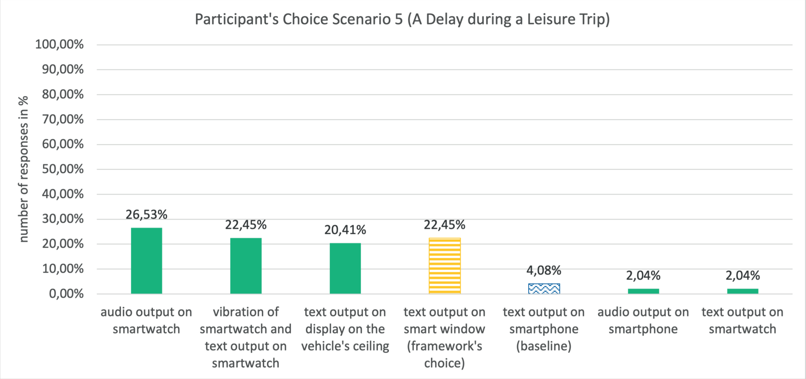


Figure 9.6 — The choices by participants for scenario 5: a delay during a leisure trip. The framework's choice is shown in yellow stripes and the baseline is shown with blue zigzag lines.

analyzed as interval data or if multipoint rating scales only result in ordinal data [Sauro and Lewis, 2016]. Following the arguments and recommendation of Sauro and Lewis [2016], I did use techniques such as a *t*-test to analyze the data of this survey and to decide if the results are statistically significant. However, I also used a χ^2 test which can also be applied to nominal data. For this test, the data was mapped to a nominal niveau. In a small number of cases, the results of the χ^2 test differed from the results of the *t*-test, cases that are discussed in greater detail in the appendix, section C.3. In the majority of cases, however, the results of both tests matched, indicating that the *t*-test results are suitable for the data obtained in the online survey.

The data for each statement listed in section 9.3.1, was analyzed for each scenario, comparing the participant's answers for the baseline output and the framework output. The participant's results are encoded as values ranging from 1 (do not agree) to 5 (fully agree) and if participants gave no answer, the value was -9. If parcticipants gave no answer, the value was excluded from the analysis. The analysis was performed using Excel and the data analysis plugin of Excel. The steps of the analysis will be presented in the following paragraphs.

χ^2 Test: For each statement in each scenario, the number of responses per value were counted. To perform a χ^2 test, the data was mapped to a nominal niveau by creating two categories, one for approval and one for rejection of the respective statement. The approval category contains the amount of answers from both approving values (4 and 5) while the rejection category contains the number of answers with both rejecting values (1 and 2). Answers with the neutral value (3) were not counted for this test. The χ^2 test was then applied to these categories for each statement in each scenario. Significance was decided using the critical value for $\alpha = 5\%$ and $df = 1$.

***t*-Test:** Since the questionnaire followed a between-subject design, I used a two-sample *t*-test for the data. First, an F-test was performed to determine if the variances of the two samples were equal, using a method provided by the data analysis plugin of Excel. Based on the results, the *t*-test variant suitable to the variances of the samples was used afterwards. Both types of *t*-tests one for equal and one non-equal variances, were also provided by the data analysis plugin.

Depending on the wording of the statement, a rejection of or agreement to a statment can be either negative or positive. Therefore, the wording of the statement must be taken into account when interpreting the results.

The answers for scenario 6 (Alight on the nex Stop while Being Busy) are not statistically significant. Since the output option of framework and baseline were the same, this result was expected. Graphs of all statistically significant results are presented in the appendix, section C.3.

The statistically significant mean values of agreement for statement 1 are shown in figure 9.7. Statement 1 targeted perceptibility. The statistically significant results for scenario 1 to 5 show a higher agreement for the framework's choice of output than for the baseline. Figure 9.7 also shows that the agreement to this statement for the baseline is much lower for most scenarios.

Statement 2 concerned privacy. Only the results of scenarios 2 and 3 are statistically relevant and both are better for the baseline. However, the rating for the framework's output choice is not very different from the baseline's rating. On average, the participants did not agree with the statement "This variant of output violates ... privacy.". Since the baseline option is a very privacy preserving option, it can be assumed that there is no option the framework could choose that would rate significantly higher for privacy.

For statement 3, the results of scenario 1 and 3 are statistically significant. The statement was worded as a negative statement regarding relevance and relatability. For both scenarios, agreement was not very high, which can be interpreted as a positive rating for the output options. Also in both cases, the framework's choice performed slightly better.

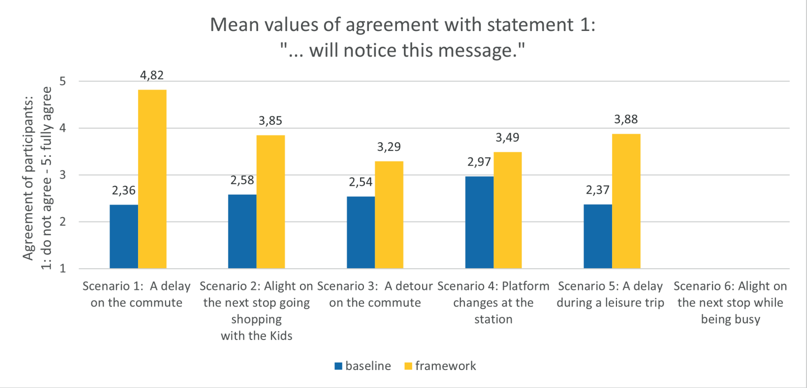


Figure 9.7 — Mean values of agreement with statement 1 for scenarios 1-6. Results for scenario 6 are not statistically significant and are therefore not shown.

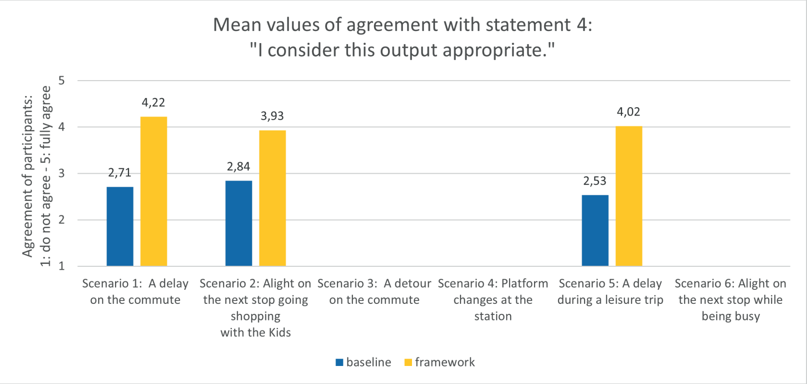


Figure 9.8 — Mean values of agreement with statement 4 for scenarios 1-6. Results for scenarios 3, 4 and 6 are not statistically significant and are therefore not shown.

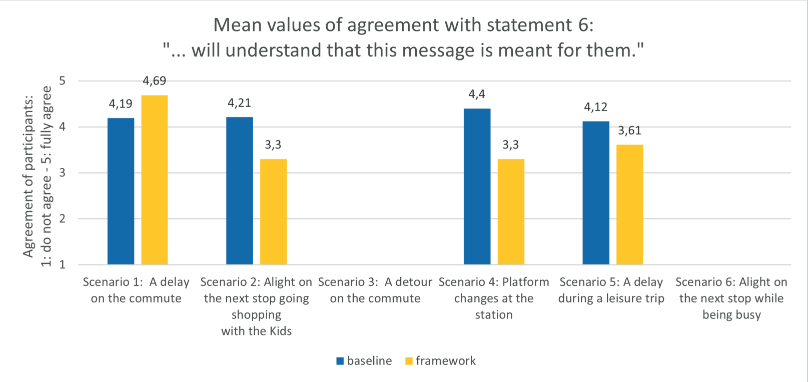


Figure 9.9 — Mean values of agreement with statement 6 for scenarios 1-6. Results for scenarios 3 and 6 are not statistically significant and are therefore not shown.

The results for statement 4 are shown in figure 9.8. They are statistically significant for scenario 1, 2 and 5. The statement was “I consider this output appropriate.” and for all three scenarios, the framework was rated better than the baseline. There is a noticeable difference between the agreement to this statement for the baseline and for the framework’s output choice.

Statement 5 again targeted perceptibility and was worded negatively as “... will not be aware of this message.”. The results for scenarios 1, 2, 4 and 5 are statistically significant. In all scenarios, the framework performed better.

Figure 9.9 shows the results for statement 6, which addressed relevance and relatability. The results are statistically significant for scenario 1, 2, 4 and 5. For scenario 1, the framework’s choice was rated slightly better than the baseline, where for scenarios 2, 4 and 5 the baseline performed better. However, the rating of the framework is still positive.

In figure 9.10, the results for statement 7 are shown. The statement was positively worded and aimed at privacy. The results for scenarios 2, 3 and 4 are statistically significant. For all scenarios, the baseline performed better than the framework. However, the results for the framework are still positive.

Statement 8 was “I think this output of the message was inappropriate.” and the results for scenarios 1, 2 and 5 are statistically significant. For all three scenarios, the framework performed better and the average value of agreeement is very low. There is a noticeable difference to the results of the baseline that

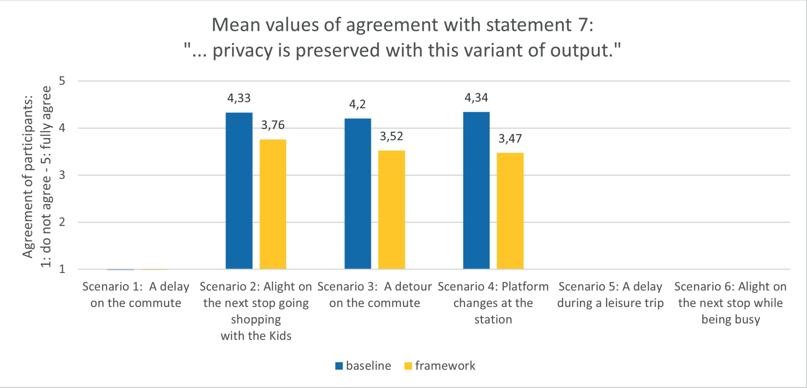


Figure 9.10 — Mean values of agreement with statement 7 for scenarios 1-6. Results for scenarios 5 and 6 are not statistically significant and are therefore not shown.

are not as positive.

In summary, in 18 of the statistically significant results, the framework performed better than the baseline, while in 8 cases, the baseline performed better. These cases were questions about privacy as well as relevance and relatability, factors that are high for an output of text on a smartphone.

9.3.3 Discussion

This usability evaluation was designed to answer the Main Research Question and addresses requirement 1. The goal was to determine if the framework for usability assessment for context-aware information provision in mobility systems can provide adequate usability adapting output devices and modalities. While interpreting the results of the online survey, the drawbacks of this evaluation approach should be kept in mind.

An important aspect of context-aware ubiquitous systems is that users use these systems in a multitude of different situation. These can not all be anticipated at design time. The goal for context-aware adaptation is for the system to autonomously adapt to these situations in order to provide its functionality to the user at all times, although designers and system engineers could not fully predefine the system's behavior. This fact has also implications for the evaluation of such systems.

Due to resource constraints, the system can only be evaluated in a limited amount of situations. An exhaustive evaluation of this type of systems is therefore not possible. This naturally reduces the informative value of such evaluations. Being able to cover several different situations in the evaluation was one of the reasons to evaluate the framework using an online questionnaire. The scenarios for the online questionnaire were chosen to represent diverse situations and contexts to allow a meaningful evaluation of the framework. It can not be ruled out that the framework performs differently in situations that have not been tested, though. The results can therefore not be seen as exhaustive. Additionally, it is clear that the performance of the framework depends on the usability rules that are implemented. The results of the usability evaluation therefore are valid for the rules implemented in the proof of concept. However, the evaluation results allow to conclude if the framework is able to achieve adequate usability and perform better or as good as the application of output options used in current passenger information systems.

The results of the usability evaluation do allow the conclusion, that the framework is able to achieve adequate usability by assessing the usability of output options and deciding autonomously based on context information. In some scenarios, the framework's choice performed similar or slightly poorer than the baseline option (scenarios 3 and 4), but in others, the framework was rated better than the baseline options. In three scenarios, the framework's choice of output was the same as a majority of participants chose themselves. When asked about the framework's output option choice and the baseline option, the results either show a clear preference for the framework or at least a comparable performance of both options, where the baseline was rated only slightly better. Overall, the results for 27 statements are statistically significant and of those, for 18 statements the framework was rated better than the baseline. The framework is therefore indeed able to achieve adequate usability and in some scenarios even perform better than a standard text output on the user's smartphone.

9.4 Summary

Table 9.1 shows an overview over all requirements and references the sections that discuss how the approach presented in this work meets each requirement. Some requirements have not been discussed yet and will be reviewed shortly in the following.

The context and domain models described in chapter 6 are ontologies modeled in OWL. Following the arguments of Strang and Linnhoff-Popien [2004] and Alegre et al. [2016] among others, ontologies are an extendable and reusable approach towards context modeling. Requirement 13 is therefore met with this approach.

The adaptation rules that shape the usability assessment as discussed in chapter 7 and implemented in a prototype described in chapter 8 are based on a usability ontology expressed in OWL that is extendable and reusable. The rules are expressed in SWRL and are also extendable. As shown in the proof of concept implementation, a subset of rules or additional rules can be implemented to fit the requirements of a target system, meaning that requirement 14 also is met. The abstraction level of adaptation rules is very high, not only using SWRL as formalization, but also the additional abstraction of usability attributes in an ontology, which fulfils requirement 7. Usability attributes and criteria are modeled explicitly using a usability ontology and SWRL rules based on these, meeting requirement 8. The usability assessment and decision making process using these rules and ontologies is implemented in the prototype.

The proof of concepts demonstrates the approach developed in this work, realizing an autonomous adaptation of output devices and modalities to the context of use while preserving adequate usability. The different parts of evaluation presented in this chapter showed that this goal was met. The following chapter summarizes this work, presents the contributions and discusses the answers to the research questions that can be given on the basis of this work.

Table 9.1 — Requirements overview for the approach based on the evaluation presented in this chapter.

	Why?	**When?**	**What?**	
Requirements:	Adaptation Goal Usability (1)	Adaptation at Runtime (2)	Adaptation of Device Choice (3)	Adaptation of Modality Choice (4)
Framework for adaptive information provision	✓ see section 9.3.3	✓ see section 9.2	✓ see section 9.1	✓ see section 9.1
	To What?	**Who?**	**How?**	
Requirements:	Context Awareness (5)	System Triggered Adaptation (6)	High Abstraction Level of Adaptation Rules (7)	Explicit Modeling of Usability Criteria (8)
Framework for adaptive information provision	✓ see section 9.1	✓ see section 9.1	✓ see section 9.4	✓ see section 9.4
	Where?			
Requirements:	Support for Legacy Systems(9)	Autonomous Usability Assessment (10)	Loose Coupling (11)	Low Prescriptiveness (12)
Framework for adaptive information provision	✓ see section 9.1	✓ see section 9.1	✓ see section 9.1	✓ see section 9.1
	Where?			
Requirements:	Extendability of Context(13)	Extendability of Adaptation Rules (14)	Extendability of UI (15)	Support for Device Mobility (16)
Framework for adaptive information provision	✓ see section 9.4	✓ see section 9.1	✓ see section 9.1	✓ see section 9.2

CHAPTER

10

Discussion and Outlook

This chapter presents the results of this work and discusses, if and how the research questions discussed in chapter 1 have been answered. Furthermore, this chapter presents a discussion of the extendability and generalizability of the approach presented in this thesis. The chapter concludes reviewing the limitations of the proposed approach towards a usability preserving adaptation of output devices and modality in smart ubiquitous mobility systems.

10.1 Summary

This thesis presented an ontology-based approach supporting adaptive information provision in smart ubiquitous mobility systems while preserving usability. It provided a usability assessment of output devices and modalities by taking the current context of use into account. Chapter 1 discusses the problem statement and the research questions that shaped this work, while chapter 2 explains the foundations and relevant fields of research. The work in this thesis consists of the following parts:

- **Requirements Analysis:** I have performed a requirements analysis developing persona and scenarios for smart ubiquitous mobility systems, as well as an analysis of user tasks in mobility systems. Additionally, I analyzed public transport systems for their characteristics and for requirements towards any approach that is intended to be used in public transport systems. Based on these analyses, I extracted 16 requirements towards an approach for adaptive user interfaces in mobility systems and specifically public transport systems. The analysis and requirements are documented in chapter 3. I used the requirements to review existing approaches in related work, described in chapter 4. In this chapter, I identified two gaps in current research that this work aims to fill.

- **Models and Usability Assessment Framework:** Based on the findings of the related work and requirements analysis, I developed a concept for a usability assessment service and adaptation framework that allows the autonomous and automated usability assessment of output options based on a given context of use and decides on the adaptation of output devices and modalities. The concept is described in chapter 5. It uses ontologies to model and represent the context of use that influences the usability of information output.

  I modeled a public transport domain ontology that is compatible with the TRIAS standard for passenger information on mobile devices [Englert et al., 2019; Keller et al., 2014c]. Based on this ontology, I developed a task ontology for public transport user tasks. Additionally, I modeled an interaction ontology, a device ontology and a context ontology. The approach of this thesis uses the information represented by these ontologies to assess the usability of output devices and modalities based on current context. The ontology engineering steps and the resulting ontologies are described in chapter 6.

  Chapter 7 presents the framework for autonomous usability assessment and output decision. It contains the results of an analysis of relevant usability attributes for public transport systems. Based on this analysis, I developed a usability ontology that represents these attributes as well as rating scales for usability qualities. Additionally, I developed a message ontology modeling the information to be passed on to the user. Both ontologies are defined in chapter 7. The framework also consists of a set of SWRL rules that references the aforementioned ontologies to guide a usability assessment and the decision process. On top of these, I defined an assessment process and a decision making process that apply the rules and are able to autonomously asses the usability of output options and decide which of the available options to use for a given message and context of use in the public transport domain. The framework therefore supports the implementation of adaptive systems that fit in the category *active configuration of a service*, as described in section 2.5.2.

- **Implementation and Evaluation:** I implemented the framework as a proof of concept which is described in chapter 8. It was realized as a REST service that receives a passenger information message and its intended recipient and returns one or more chosen output options to deliver this message. The service utilizes context stored in a triple store and accesses this context using a SPARQL interface. This prototype implementation is one part of the evaluation of the approach presented in this thesis. Additionally, I conducted a performance evaluation that showed that the prototype usability assessment service is able to decide for an output option in a reasonable amount of time. Finally, I performed a usability evaluation. Six different usage scenarios in public transport settings were tested in this evaluation. Using an online questionnaire, 200 participants rated the framework's choice of output device and modality compared to a baseline choice. The usability evaluation showed that the framework is able to ensure adequate usability for device and modality adaptation in public transport scenarios. The evaluation and its results are documented in chapter 9.

This work provides the following contributions:

- **Contribution 1: Ontologies modeling the context of use for smart ubiquitous mobility systems.** In the course of this work, several ontologies were developed that together allow the representation of the context of use in smart ubiquitous mobility systems. A public transport domain ontology provides a vocabulary for public transport entities and is compatible with the german standard for traveller information interfaces TRIAS. A task ontology models public transport tasks of passengers and a device and interaction ontology capture interaction context of passenger information systems. A context ontology represents other user context, vehicle context and device context, such as their location or environmental conditions, for example.
- **Contribution 2: A usability ontology modeling usability attributes for situations in smart ubiquitous mobility systems.** Based on an analysis of important usability attributes and criteria for smart ubiquitous mobility systems, an ontology modeling these attributes was developed. It can be used to express ratings and categorizations of usability attributes for devices or modalities.
- **Contribution 3: A usability rule framework to express rules for the assessment of usability criteria.** This framework allows to express assessment rules for the usability attributes in the aforementioned usability ontology. A set of assessment rules can guide a usability assessment of devices and modalities in smart ubiquitous mobility systems. Since in mo-

bility systems, situations and context often changes and therefore usability may change, too, these rules reference the context of use through the context ontologies and therefore allow for a situation specific assessment of usability.

- **Contribution 4: A process for a usability assessment of device and modality options in a given context of use.** The approach presented in this thesis also contains a process to perform such a usability assessment using a set of assessment rules. A proof of concept implementation shows that the process is able to assess the usability of output devices and modalities.
- **Contribution 5: A decision making process for the adaptation of output devices and modalities in smart ubiquitous mobility systems.** This approach finally includes a process that decides which output devices and modalities to use, based on their usability assessment. Additional rules guide the decision process that therefore can be shaped according to requirements of a target system and domain by replacing or extending the rule set. This process was also implemented in the proof of concept and the performance and usability evaluation of this prototype implementation showed that it is possible to successfully implement a usability assessment service. The evaluation found that this service can achieve adequate usability deciding for an output device and modality in a given situation with an acceptable reponse time.

10.2 Answers to the Research Questions

This section will present the answers to the research questions discussed in chapter 1 based on the approach developed in this work.

Research Question 1: Dimensions of the Context of Use

Which dimensions of the context of use are relevant as a basis for a usability assessment of device and modality adaptations?

Answer to Research Question 1: Dimensions of the Context of Use
An analysis of context dimensions identified the context dimensions for smart ubiquitous mobility systems. The results of this analysis are described in section 6.1.1. They comprise user tasks and user context facts such as user preferences. Further dimensions are physical context, spatial context, temporal context, socio-technical context and interaction context. These context dimensions are in part also presented in [Schlegel and Keller, 2011] and [Kühn et al., 2011]. The

identified context dimensions influence the usability of devices and modalities used for information output in smart ubiquitous mobility systems.

Research Question 2: Representation of the Context of Use

How can the context of use be represented in order to be processed autonomously by a smart ubiquitous mobility system?

Answer to Research Question 2: Representation of the Context of Use
For this work, the context of use was represented as context in a context-aware system. The foundations of context modeling for context-aware systems are detailed in section 2.4.1. For extendability, readability and expressivity reasons, a representation as an ontology was chosen. Chapter 6 presents the developed context ontologies that represent the dimensions of the context of use identified in section 6.1.1. Ontologies are use to provide machine-readable knowledge. Using the developed ontologies, knowledge about the context of use can be queried, interpreted and processed autonomously by a smart ubiquitous mobility system. A domain ontology for public transport was published in [Keller et al., 2014a], while part of the context model for passenger information was published in [Keller et al., 2020].

Research Question 3: Method for Autonomous Assessment

Which method is suitable for autonomously assessing the usability of an adaptation of device and modality in a given context of use?

Answer to Research Question 3: Method for Autonomous Assessment
In section 7.1, an analysis of usability attributes for smart ubiquitous mobility systems was presented. Relevant attributes are modeled in a usability ontology that is described in section 7.2. The ontology provides the vocabulary to express usability qualities of output devices and output modalities. The assessment rules described in section 7.3 reference the usability ontology. They also reference the context ontologies and therefore can express usability qualities with respect to the current context. A system that applies these rules, as shown in section 7.4.2 can autonomously assess the usability of output devices and modalities in relation to their context of use. Part of this assessment process has been published in [Keller and Schlegel, 2019].

Research Question 4: Decision Making

How can a system autonomously reach a decision to adapt device and modality while maintaining adequate usability?

Answer to Research Question 4: Decision Making
The decision about the adaptation of output device and modality in smart ubiquitous mobility systems is based on the usability assessment described in section 7.4.2. The process uses all available devices and modalities, determined by a lookup in the context storage, as described in section 7.4.1. After each output option is assessed, the decision making process uses decision rules to filter all options and to reach a decision. The decision making process is described in section 7.4.3 and was in part published in [Keller and Schlegel, 2019].

Main Research Question

Can a smart ubiquitous mobility system adapt output devices and modalities autonomously and at runtime to the context of use while preserving adequate usability?

Answer to the Main Research Question:
This thesis presents a framework for adaptation of output devices and modalities in smart ubiquitous mobility systems. Using several OWL ontologies, the context of use for smart ubiquitous mobility systems is modeled and can be expressed and processed as is common in ubiquitous context-aware systems. A usability ontology and SWRL assessment rules allow the system to autonomously assess the usability of output options, taking the context of use into account. A decision making process builds on that assessment to decide, which output device and modality should be used in the current situation. The solution works at runtime, as shown in section 9.2 and can decide to adapt output options with adequate usability, as shown in section 9.3. It answers the Main Research Question with yes, a smart ubiquitous mobility system can adapt output devices and modalities autonomously and at runtime to the context of use while preserving adequate usability.

10.3 Extendability and Generalizability of the Approach

The approach presented in this thesis is extendable and can be generalized to be used in different domains. This section discusses extendability and generalizability for different aspects of the approach.

Architecture: The approach is designed to be comaptible with ontology-based context-aware systems as discussed in section 2.5 and can be implemented to be used with such systems. The usability assessment is designed as a modular service component, that is loosely coupled and can easily be integrated into any type of system.

Context: The context models in this approach are modeled as OWL ontologies that are highly extendable, due to their expressivity and modularity, as described in section 2.4.1. Additional context factors can be added to the context ontologies and then be used in usability assessment rules or filter rules at any time. If the approach should be applied to different domains, the public transport specific ontologies can be replaced with ontologies of other domains, for example for smart home systems. Corresponding adaptation rules would have to be replaced as well, but the adaptation and decision making process remain the same.

Output Devices and Modalities: Devices and modalities that are used with the usability assessment need to be able to receive from the passenger information system. They then need to be able to output the message using the chosen modality. New types of devices or modalities that fulfil this requirement can be added to the device and interaction ontologies that are described in sections 6.4 and 6.5. As long as context data about these devices or modalities can be acquired and added to the context store, they are taken into account during the usability assessment process.

Usability Attributes: The usability attributes that are used in the usability assessment process are modeled in the usability ontology described in section 7.2. This ontology can also be extended with additional attributes and their ratings. An addition of usability attributes results in an addition of steps in the usability assessment and in the decision making process that consider these attributes. Since those processes are modular and can be implemented in any way necessary for the target system, this extension is also possible. If the approach should be implemented for a different domain, the attributes should be revisited, since they were identified with a smart ubiquitous mobility system in mind. However, replacing or adding attributes is entirely possible.

Usability Assessment and Filter Rules: The rules for usability assessment and decision filters are expressed in SWRL, referencing the ontologies, as described in section 7.3. SWRL provides an expressive and human-readable format for rules. The set of rules used for the implementation of a usability assessment can be changed, extended or replaced completely. The approach can therefore be extended by additional rules or, if needed, an entirely new set of adaptation rules can be designed and implemented.

Generalization to Include Input: The interaction and device ontologies as described in sections 6.4 and 6.5 are focused on output devices. However, it is possible to extend them by adding input modalities and devices and then using the approach to adapt input devices as well. It would be useful for such a scenario to revisit and extend the usability ontology, since different usability attributes are relevant for input interaction. The adaptation rules would also need to be extended by additional rules that specify usability for input devices and modalities based on the context of use. If input interactions are taken into account, an implementation of the usability assessment service would need to focus on efficiency, since a latency of the usability assessment would probably be more disruptive in scenarios adapting input and output devices, compared to the scenarios considered in this work.

10.4 Limitations

Although the contributions of this thesis are extensive, the approach has some limitations. First, it focuses only on the adaptation of modality and device choice and only output is considered. As described above, the approach can be generalized to include input and additional output modalities. The adaptation of other interaction features beyond device and modality choice, such as UI elements was not in the scope of this work. While extending this approach to adapt user interface elements may be possible, it would involve a large amount of modeling work that may be too time consuming. As discussed in section 4.2, many approaches exist that focus on the adaptation of user interface elements and may be better suited for this task.

This approach also does not include the generation of user interfaces. Generating user interfaces would require too much reimplementation of several types of user interface to be feasible for the integration into legacy systems such as mobility systems and specifically public transport systems. This approach focused on utilizing existing user interfaces. In comparison, many approaches towards adaptive user interfaces consider generating UIs, while not many adaptation approaches exist that can include existing user interfaces. However, since this approach is kept modular to be integrated into different types of systems, integration with a user interface generation approach might be possible.

Apart from the combination of vibration and text output, this approach does not support multimodality or multi-device interaction. Extending it to additionally assess multimodal or multi-device output options may be possible, but would require modeling not only the concepts of multimodality and multi-device interaction in the interaction ontology but also usability attributes that are relevant to multimodal and multi-device user interfaces. The CARE properties presented by Coutaz et al. [1995] can be a good starting point to review such usability attributes for multimodality.

The approach presented in this work is ontology-based. It would not work without several ontologies. If it were to be applied to a domain and system where no domain and context ontologies exist, modeling the necessary ontologies would require a great effort. The same limitation applies for the adaptation rule set. The SWRL rules are very expressive and readable, but modeling and testing a new rule set can be time-consuming.

A context management system is needed for this approach to work. This means that context acquisition, processing and reasoning need to be implemented. Context must be stored in a context storage that provides a SPARQL interface for access. While a lot of research exists that provides insight into context management, it still would need to be implemented and integrated in a public transport or mobility system. In our research project "SmartMMI"[1] we implemented such a context management system for public transport, an effort which is in part published in [Keller et al., 2020].

Finally, the evaluation of this work has its limitations. The implementation is a proof of concept implementation and can not easily be used as a template for future implementations. The evaluation that was done uses only one rule set and a limited amount of scenarios, due to ressource restrictions. While the scenarios were carefully chosen to represent a variety of important factors, not all contingencies can be covered. This is a conflict that generally affects context-aware adaptive systems, since real situations comprise such a multitude of factors that it is impossible to cover them all during evaluations. An online user test as conducted for the usability evaluation of this approach also has limitations, since participants do not experience the real context and do not interact with a real system. Also, long-term effects of users using such adaptive systems can not be examined. However, despite these limitations, the evaluation as described in chapter 9 allows the conclusion, that this approach does provide an answer to the Main Research Question.

---

[1] "SmartMMI: Modell- und kontextbasierte Mobilitätsinformation auf Smart Public Displays und Mobilgeräten im Öffentlichen Verkehr", `http://smartmmi.de/`, last accessed October 12th, 2022

10.5 Outlook

In public transport and mobility systems, personalization is becoming more important. New interaction technologies provide better passenger information and a more comfortable mobility experience. One example is the smart window prototype that we developed in the SmartMMI project. The semi-transparent display is built in one train of the Karlsruher Verkehrsverbund (KVV) and is currently still providing passenger information to passengers in Karlsruhe in October 2022[2]. The public transport ontology is compatible with the industry standard TRIAS, which is currently in widespread use and provides compatibility with passenger information systems currently in use. This work is a contribution to innovative passenger information in public transport, in research and in practice. The approach presented in this thesis also provides several potential starting points for future work.

For reusability and extendability, it will be beneficial to further evaluate the rule set and usability attributes to determine if an effect of a rule on the intended usability attribute can be verified. Further, an evaluation of a fully implemented system in the field can show how contextual factors influence the user's perception and rating of adaptations. Also, a long-term evaluation could show effects of familiarization and acceptance and it could reveal facts about the learnability of such an adaptive system.

To further improve the framework, it would be interesting to incorporate current research about interruptibility and attention management that aim to improve the reception of notifications by choosing the right moment to address the user and being as unobtrusive as possible [Turner et al., 2015; Anderson et al., 2018; Cha et al., 2020]. The findings of attention management research could be modeled as usability attributes and rules, provided that the necessary context facts can be obtained. Also, since social context can have a great impact in public systems, investigations of this impact and of possibilities to detect and model such social context factors can further improve the capabilities of the usability assessment.

As described in section 10.3, the approach presented in this thesis is extendable and can be implemented for other domains. It is compatible with context-aware systems that have been developed for several application domains, such as smart homes, smart factories or smart classrooms and more [Alam et al., 2012; Wilson et al., 2015; Beez et al., 2018; Rosenberger and Gerhard, 2018; Huang et al., 2019; Zhao and Peng, 2020]. The usability ontology developed in this work provides a machine-readable vocabulary and understanding of usability

---

[2] `https://smartmmi.de/category/feldtest/`, last accessed October 24th, 2022

that can benefit context-aware ubiquitous systems that implement adaptations affecting their user interfaces.

This work provides an approach to consider usability in adaptation processes of ontology-based systems, taking the current context of use into account. It is not intended to replace user interface designers and UX professionals, but to complement their efforts in systems that are deployed in changing environments and need to react to situations that can not all be forseen at design time. As such, it can further improve the usability of adaptation decisions the system needs to make based on current context facts and ultimately improve the acceptance of such systems.

APPENDIX A

Appendix Analysis Results

A.1 Hierarchical Task Analysis Results

This section documents the results of the hierarchical task analysis that was described in section 6.1.2 that are not related to public transport.

A.1.1 T1: Boarding a Public Transport Vehicle

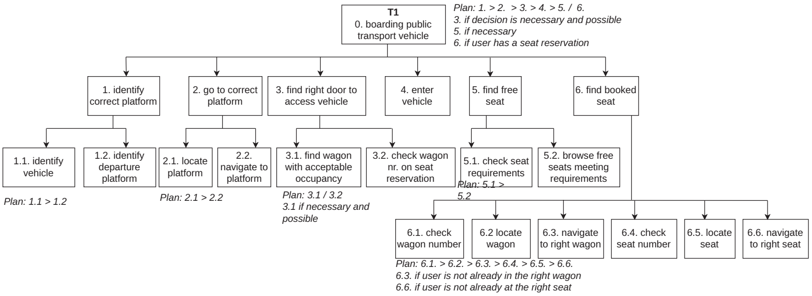


Figure A.1 — Hierarchical task analysis for boarding a public transport vehicle.

Table A.1 — **T1:** Boarding a public transport vehicle

T1: Boarding a public transport vehicle		
(Sub)Task and Operation	*Goal State*	*Plan*
0. boarding a public transport vehicle	user is in the right vehicle when it departs	-
1. identify correct platform	user knows on which platform their vehicle departs	-
1.1. identify vehicle	user knows vehicle identifcation	-
1.2. identify departure platform	user knows where the vehicle will depart	after 1.1.
2. go to correct platform	user is on correct platform	after 1.
2.1. locate platform	user knows location of correct platform	-
2.2. navigate to platform	user is on correct platform	after 2.1.
3. find right door to access vehilce	user knows where to enter the vehicle	after 2.
3.1. find wagon with acceptable occupancy	-	if necessary and possible
3.2. check wagon number on seat reservation	-	either 3.1. or 3.2.
4. enter vehicle	user is in correct vehicle	after 3.
5. find free seat	user is seated	after 4., if necessary
5.1. check seat requirements	-	-
5.2. browse free seats meeting requirements	-	after 5.1.
6. find booked seat	user is seated	either 5. or 6.
6.1. check wagon number	-	if the user is not already in the right wagon
6.2. locate wagon	-	after 6.2, if the user is not already in the right wagon
6.3. go to right wagon	user is in the right wagon	after 6.1., if the user is not already in the right wagon
6.4. check seat number	-	after 6.3.
6.5. locate seat	-	after 6.4.
6.6. navigate to the right seat	user is seated	after 6.5., if user is not already next to the seat

A.1.2 T2: Boarding a Motorized Vehicle

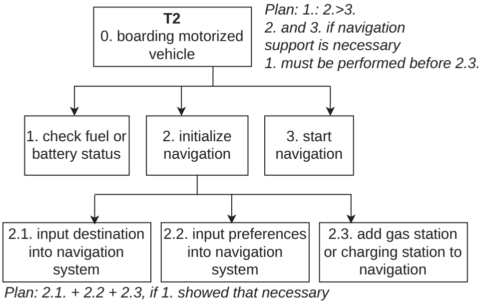


Figure A.2 — Hierarchical task analysis for boarding a motorized vehicle.

Table A.2 — **T2:** Boarding a motorized vehicle, tasks and operations.

T2: Boarding a motorized vehicle		
(Sub)Task and Operation	*Goal State*	*Plan*
0. boarding a motorized vehicle	user and vehicle are ready to drive, user knows where to drive.	-
1. check fuel or battery status	user knows, if refueling or recharching is necessary.	can be performed in any order before 2.3.
2. initialize navigation	-	Is only executed, if navigation support is necessary.
2.1. input destination into navigation system	navigation can be computed.	-
2.2. input preferences into navigation system	navigation can be personalized.	can be performed in any order.
2.3. add gas station or charging station to navigation	navigation is altered / computed to include refueling or recharching.	-
3. start navigation	user knows where to drive	-

A.1.3 T3: Boarding a Non-Motorized Vehicle

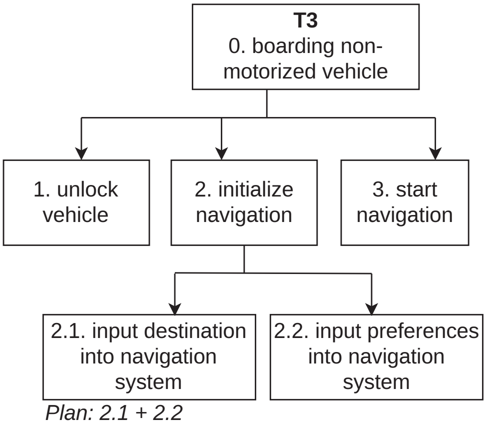


Figure A.3 — Hierarchical task analysis for boarding a non-motorized vehicle.

Table A.3 — **T3:** Boarding a non-motorized vehicle, tasks and operations.

T3: Boarding a non-motorized vehicle		
(Sub)Task and Operation	*Goal State*	*Plan*
0. boarding a non-motorized vehicle	user and vehicle are ready to drive, user knows where to drive.	-
1. unlock vehicle	vehicle is ready to be moved.	if vehicle is locked, in any order with 2. and 3.
2. initialize navigation	-	if navigation support is necessary.
2.1. input destination into navigation system	navigation can be computed.	-
2.2. input preferences into navigation system	navigation can be personalized.	can be performed in any order.
3. start navigation	user knows where to drive	-

A.1.4 T4: Boarding a Car Sharing or Rental Car

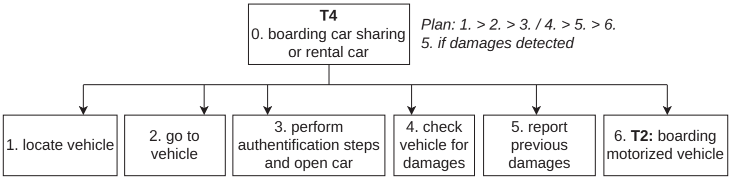


Figure A.4 — Hierarchical task analysis for boarding a shared or rental car.

Table A.4 — **T4:** Boarding a car sharing or rental car.

T4: Boarding a car sharing or rental car		
(Sub)Task and Operation	*Goal State*	*Plan*
0. boarding a car sharing or rental car	user and vehicle are ready to drive, user knows where to drive.	-
1. locate vehicle	user knows location of the vehicle	first step in any case
2. go to vehicle	user is next to the vehicle	after 1.
3. perform authentication steps and open car	car is open and user has the keys	-
4. check vehicle for damages	user knows of previous damages on the car	-
5. report any previous damages	damages are recorded.	if outcome of 4. was positive.
6. **T2:** boarding of a motorized vehicle.	user and vehicle are ready to drive, user knows where to drive.	-

A.1.5 T5: Boarding a Non-Motorized Sharing Vehicle

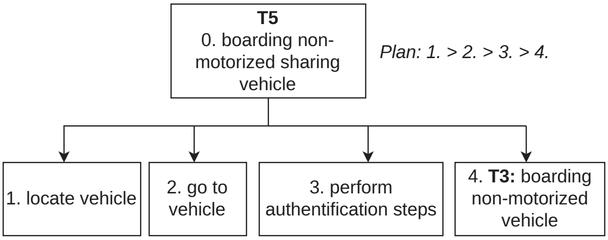


Figure A.5 — Hierarchical task analysis for boarding a non-motorized sharing vehicle.

Table A.5 — **T5:** Boarding a non-motorized sharing vehicle

T5: Boarding a non-motorized sharing vehicle		
(Sub)Task and Operation	*Goal State*	*Plan*
0. boarding a non-motorized sharing vehicle	user and vehicle are ready to drive, user knows where to go	-
1. locate vehicle	user knows location of vehicle	-
2. go to vehicle	user is next to vehicle	after 1.
3. perform authentication steps	user can unlock vehicle	after 2.
4. **T3:** board non-motorized vehicle	user and vehicle are ready to drive and user knows where to go	after 3.

A.1.6 T6: Boarding a Car as Passenger

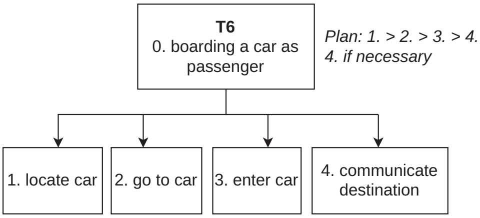


Figure A.6 — Hierarchical task analysis for boarding a car as a passenger.

Table A.6 — **T6:** Boarding a car as a passenger

T6: Boarding a car as a passenger		
(Sub)Task and Operation	*Goal State*	*Plan*
0. boarding a car as a passenger	user is in the car, driver knows where to go	-
1. locate vehicle	user knows location of vehicle	-
2. go to vehicle	user is next to vehicle	after 1.
3. enter vehicle	user is in the vehicle	after 2.
4. communicate destination	driver knows where to go	after 3., if necessary

A.1.7 T7: In Transit in a Public Transport Vehicle

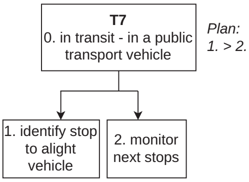


Figure A.7 — Hierarchical task analysis for being in transit in a public transport vehicle.

Table A.7 — **T7:** In transit in a public transport vehicle

T7: In transit in a public transport vehicle		
(Sub)Task and Operation	*Goal State*	*Plan*
0. in transit - in a public transport vehicle	user knows when they have to exit at the next stop	-
1. identify stop to alight vehicle	user knows at which stop they have to leave the vehicle	-
2. monitor next stops	user knows if the next stop of the vehicle is their last stop	after 1.

A.1.8 T8: In Transit - Driving a Vehicle

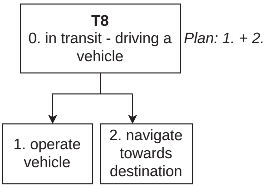


Figure A.8 — Hierarchical task analysis for driving a vehicle.

Table A.8 — **T8:** In transit - driving a vehicle

T8: In transit - driving a vehicle		
(Sub)Task and Operation	*Goal State*	*Plan*
0. in transit - driving a vehicle	user and vehicle reach destination	-
1. operate vehicle	-	-
2. navigate	-	-

A.1.9 T9: Getting a Lift

This task was not modeled, since it does not comprise any user tasks. Boarding and Alighting a car as a passenger are covered by tasks **T6** and **T13**.

A.1.10 T10: In Transit - Walking

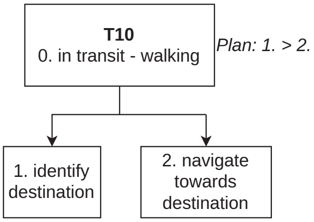


Figure A.9 — Hierarchical task analysis for walking.

A.1.11 T11: Alighting a Public Transport Vehicle

Figure A.10 — Hierarchical task analysis for alighting a public transport vehicle.

Table A.9 — **T11:** Alighting a public transport vehicle

T11: Alighting a public transport vehicle		
(Sub)Task and Operation	*Goal State*	*Plan*
0. alighting a public transport vehicle	user exited the vehicle at the correct stop	-
1. locate suitable exit	user knows where to exit	-
2. go to correct exit	user is at exit	after 1.
3. exit vehicle	user has exited vehicle at correct stop	after 2.

A.1.12 T12: Alighting an Individual Transport Vehicle (as Driver)

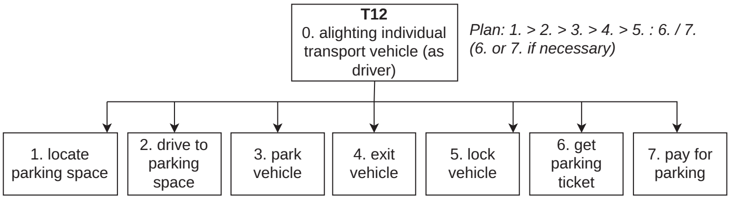


Figure A.11 — Hierarchical task analysis for alighting an individual transport vehicle as a driver.

Table A.10 — **T12:** Alighting an individual transport vehicle (as driver)

T12: Alighting an individual transport vehicle (as driver)		
(Sub)Task and Operation	*Goal State*	*Plan*
0. alighting an individual transport vehicle (driver)	vehicle is locked and legally parked, user exited vehicle	-
1. locate parking space	user knows where to park	-
2. drive to parking space	vehicle is at parking space	after 1.
3. park vehicle	vehicle is parked	after 2.
4. exit vehicle	user exited vehicle	after 3.
5. lock vehicle	vehicle is locked	after 4.
6. get parking ticket	user has parking ticket	if necessary
7. pay for parking	user has receipt,	if necessary, either 6. or 7.

A.1.13 T13: Alighting from a Car as a Passenger

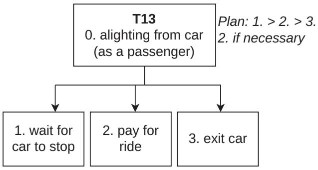


Figure A.12 — Hierarchical task analysis for alighting a car as a passenger.

Table A.11 — **T13:** Alighting from a car as a passenger

T13: Alighting from a car as a passenger		
(Sub)Task and Operation	*Goal State*	*Plan*
0. alighting from a car as a passenger	user exited the car	-
1. wait for car to stop	car stopped	-
2. pay for trip	trip is payed	after 1., if necessary
3. exit car	user exited the car	after 1.

A.1.14 T14: Alighting from a Car Sharing or Rental Car

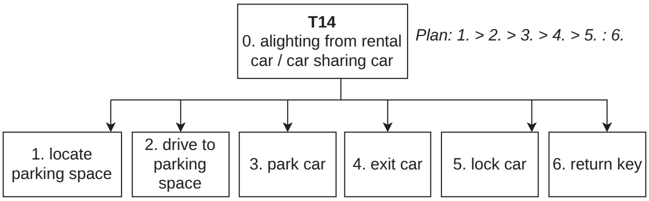


Figure A.13 — Hierarchical task analysis for alighting from a car sharing or rental car.

Table A.12 — **T14:** Alighting from a car sharing or rental car

T14: Alighting from a car sharing or rental car		
(Sub)Task and Operation	*Goal State*	*Plan*
0. alighting from a car sharing or rental car	user exited the car, car is locked and key returned	-
1. locate parking space	user knows where to park	-
2. drive to parking space	car is at parking space	after 1.

Table A.12 — **T14:** Alighting from a car sharing or rental car

T14: Alighting from a car sharing or rental car		
3. park car	car is parked	after 2.
4. exit car	user exited car	after 3.
5. lock car	car is locked	after 4.
6. return key	key returned or stored	after 4., in any order to 5.

A.1.15 T15: Alighting from a Non-Motorized Sharing Vehicle

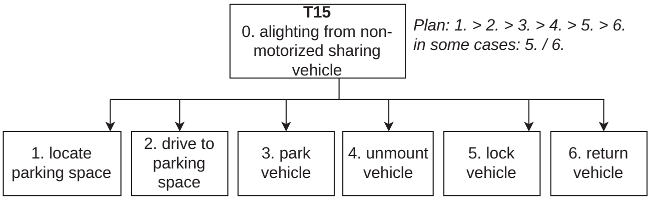


Figure A.14 — Hierarchical task analysis for alighting from a non-motorized sharing vehicle.

Table A.13 — **T15:** Alighting from a non-motorized sharing vehicle

T15: Alighting from a non-motorized sharing vehicle		
(Sub)Task and Operation	*Goal State*	*Plan*
0. from a non-motorized sharing vehicle	vehicle is locked and returned to the sharing service	-
1. locate parking space	user knows where to park	-
2. drive to parking space	vehicle is at parking space	after 1.
3. park vehicle	vehicle is parked	after 2.
4. unmount vehicle	user unmounted vehicle	after 3.
5. lock vehicle	vehicle is locked	after 4.
6. return vehicle	vehicle is returned to the sharing service	after 5.

A.1.16 T16: Wait at Station or Stop

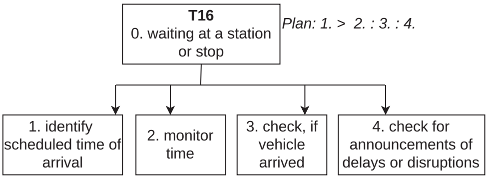


Figure A.15 — Hierarchical task analysis for waiting at a station or stop.

Table A.14 — **T16:** Waiting in public transport

T16: Waiting at a station or stop		
(Sub)Task and Operation	*Goal State*	*Plan*
0. wait at a station or stop	time passed until the vehicle arrived or delays or disruptions occurred	-
1. identify scheduled time of arrival	user knows when vehicle is scheduled	-
2. monitor time	user knows current time and if scheduled time of arrival has approached	after 1.
3. check if vehicle arrived	user knows if vehicle has arrived	-
4. check for delay or disruption announcements	time passed until the vehicle arrived or delays or disruptions occurred	after 1., 2. and if 3. is negative and time of arrival has approached (2.)

A.1.17 T17: Waiting for a Lift

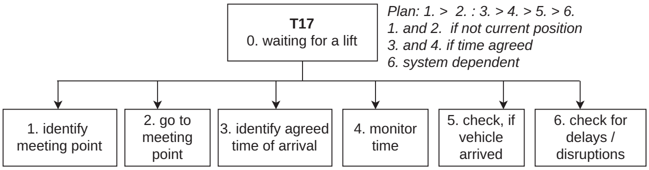


Figure A.16 — Hierarchical task analysis for waiting for a lift.

Table A.15 — **T17:** Waiting for a lift

T17: Waiting for a lift		
(Sub)Task and Operation	*Goal State*	*Plan*
0. wait for a lift	time passed until the vehicle arrived or delays or disruptions occurred	-
1. identify agreed meeting point	user knows where vehicle will arrive	if meeting point is not current position
2. go to meeting point	user is at meeting point	after 1.
3. identify agreed time of arrival	user knows when vehicle should arrive	if time of arrival was agreed upon
4. monitor time	user knows current time and if time of arrival has approached	after 3.
5. check if vehicle arrived	user knows if vehicle has arrived	-
6. check for delays or disruptions	time passed until the vehicle arrived or delays or disruptions occurred	after 1., 2., 3. and 4. and if 5. is negative and time of arrival has approached (4.)

A.1.18 T18: Dealing with a Delay in Public Transport

Table A.16 — **T18:** Dealing with a delay in public transport

T18: Dealing with a delay in public transport		
(Sub)Task and Operation	*Goal State*	*Plan*
0. dealing with delay in public transport	itinerary is up-to-date and user knows what to do next	-
1. gain knowledge about a delay	user knows about the delay	-
2. identify if delay affects own itinerary	user knows if, and how their itinerary is affected	after 1.
3. assess rescheduling	user has decided, if rescheduling is necessary	after 2., optional, based on result of 2.

Table A.16 — **T18:** Dealing with a delay in public transport

T18: Dealing with a delay in public transport		
3.1. assess effects of delay on own itinerary	user knows and evaluated the effects of the delay	after 2.
3.2. assess alternatives	user knows about and evaluated available alternative itineraries	after 3.1. and if result of 3.1. made it necessary
3.3. decide about rescheduling	user has decided to reschedule or not reschedule	after 3.2.
4. reschedule	itinerary is up-to-date and user knows what to do next	after 3.3 and based on the result of 3.3.

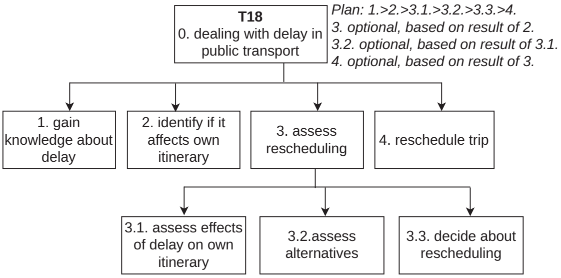


Figure A.17 — Hierarchical task analysis for dealing with a delay in public transport.

A.1.19 T19: Dealing with a Delay in Individual Transport

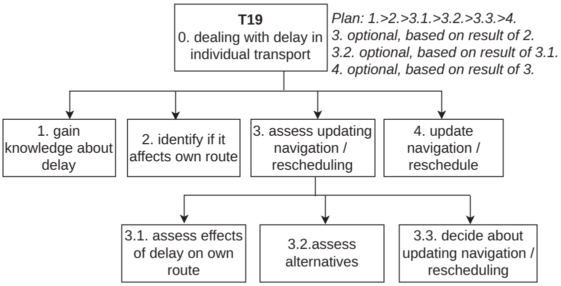


Figure A.18 — Hierarchical task analysis for dealing with a delay in individual transport.

Table A.17 — **T19:** Dealing with a delay in individual transport

T19: Dealing with a delay in individual transport		
(Sub)Task and Operation	*Goal State*	*Plan*
0. dealing with delay in individual transport	itinerary is up-to-date and user knows what to do next	-
1. gain knowledge about a delay	user knows about the delay	-
2. identify if delay affects own route	user knows if, and how their route is affected	after 1.
3. assess updating navigation / rescheduling	user has decided, if updating the navigation / rescheduling is necessary	after 2., optional, based on result of 2.
3.1. assess effects of delay on own itinerary	user knows and evaluated the effects of the delay	after 2.
3.2. assess alternatives	user knows about and evaluated available alternatives	after 3.1. and if result of 3.1. made it necessary
3.3. decide about updating navigation / rescheduling	user has decided anout alternatives	after 3.2.
4. replan route / reschedule	itinerary is up-to-date and user knows what to do next	after 3.3 and based on the result of 3.3.

A.1.20 T20: Dealing with a Disruption in Public Transport:

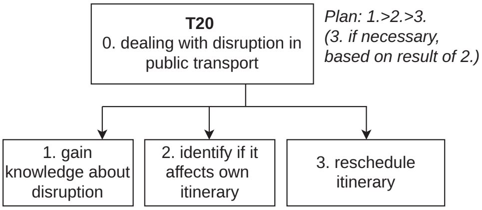


Figure A.19 — Hierarchical task analysis for dealing with a disruption in public transport.

Table A.18 — **T20:** Dealing with a disruption in public transport

T20: Dealing with a disruption in public transport		
(Sub)Task and Operation	*Goal State*	*Plan*
0. dealing with disruption in public transport	itinerary is up-to-date and user knows what to do next	-
1. gain knowledge about a disruption	user knows about the disruption	-
2. identify if disruption affects own itinerary	user knows if, and how their itinerary is affected	after 1.
3. reschedule itinerary	itinerary is up-to-date and user knows what to do next	after 2., optional, based on result of 2.

A.1.21 T21: Dealing with a Disruption in Individual Transport

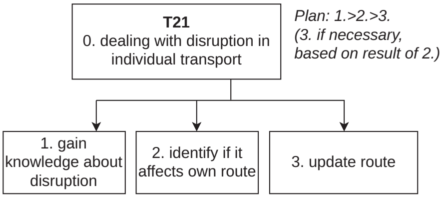


Figure A.20 — Hierarchical task analysis for dealing with a disruption in individual transport.

Table A.19 — **T21:** Dealing with a disruption in individual transport

T21: Dealing with a disruption in individual transport		
(Sub)Task and Operation	*Goal State*	*Plan*
0. dealing with disruption in individual transport	route is up-to-date and user knows what to do next	-
1. gain knowledge about a disruption	user knows about the disruption	-
2. identify if disruption affects own route	user knows if, and how their route is affected	after 1.
3. replan route	route is up-to-date and user knows what to do next	after 2., optional, based on result of 2.

A.1.22 Context Analysis for Task Analysis Results

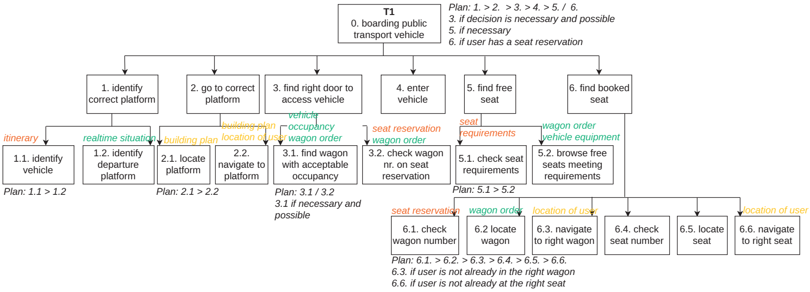


Figure A.21 — Context analysis for task T1.

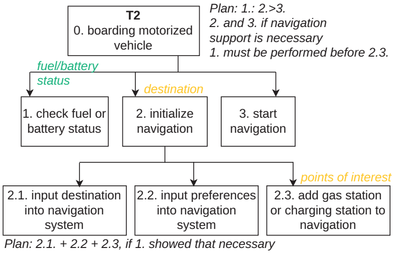


Figure A.22 — Context analysis for task T2.

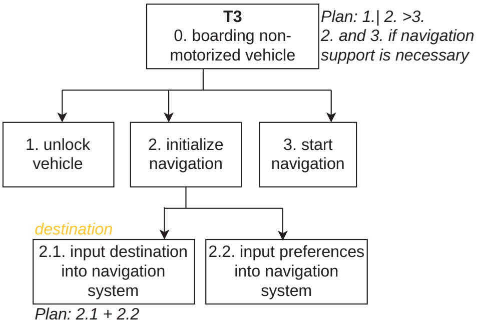


Figure A.23 — Context analysis for task T3.

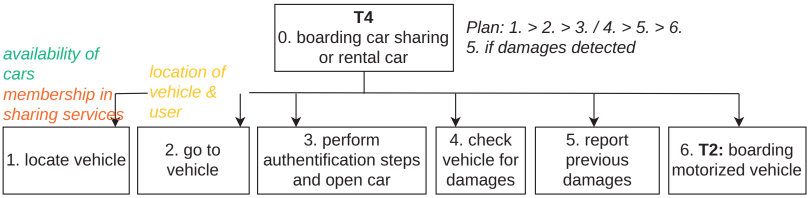


Figure A.24 — Context analysis for task T4.

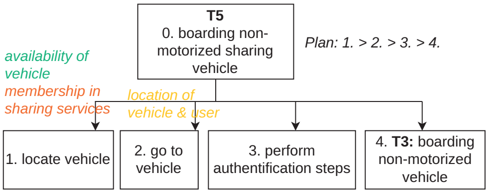


Figure A.25 — Context analysis for task T5.

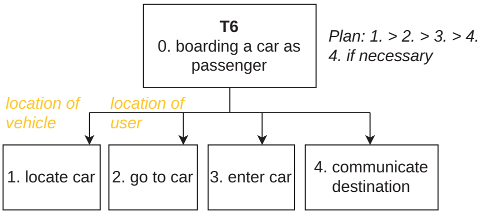


Figure A.26 — Context analysis for task T6.

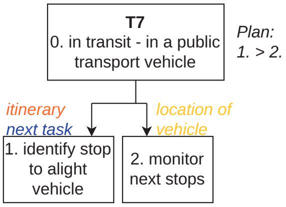


Figure A.27 — Context analysis for task T7.

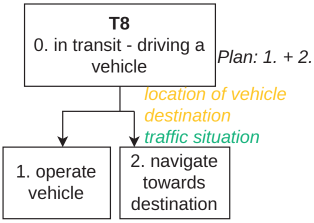


Figure A.28 — Context analysis for task T8.

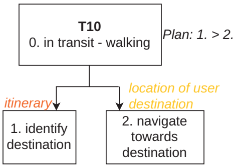


Figure A.29 — Context analysis for task T10.

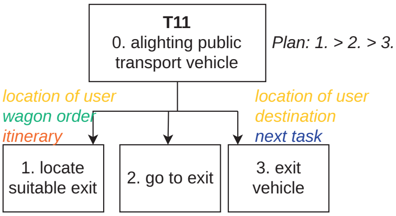


Figure A.30 — Context analysis for task T11.

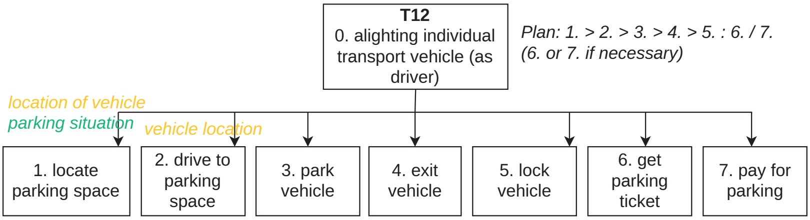


Figure A.31 — Context analysis for task T12.

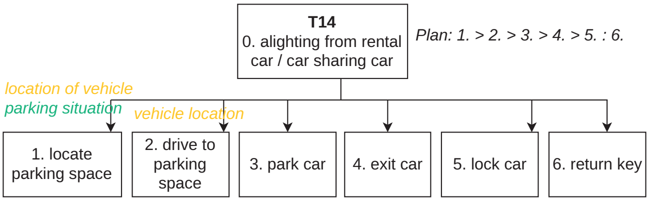


Figure A.32 — Context analysis for task T14.

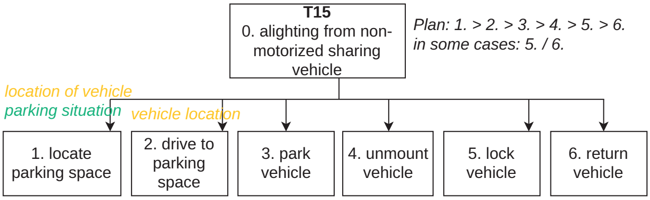


Figure A.33 — Context analysis for task T15.

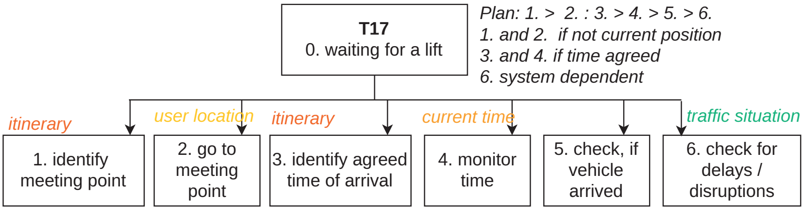


Figure A.34 — Context analysis for task T17.

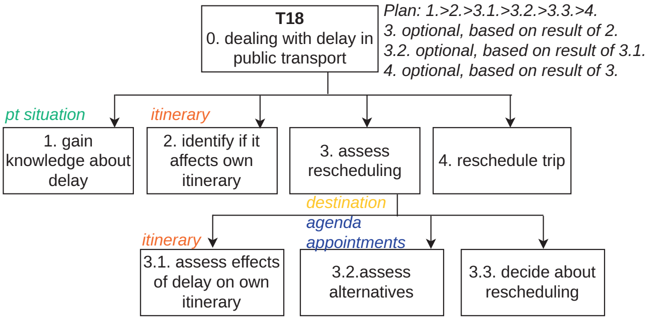


Figure A.35 — Context analysis for task T18.

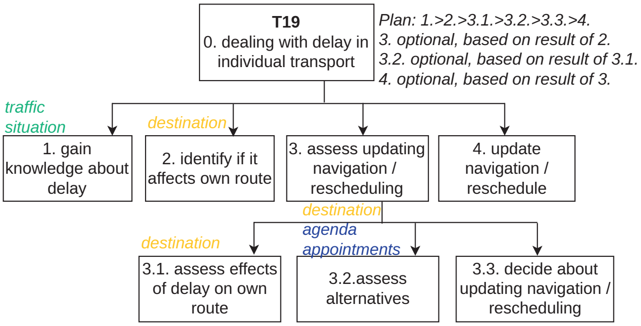


Figure A.36 — Context analysis for task T19.

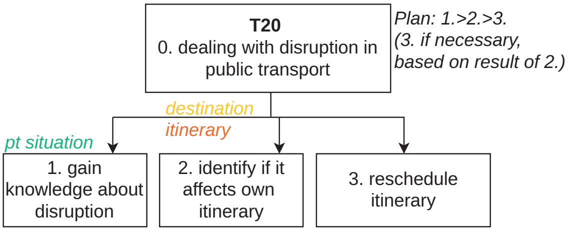


Figure A.37 — Context analysis for task T20.

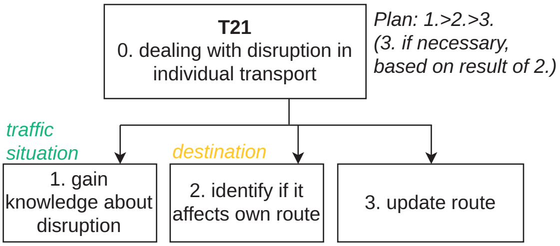


Figure A.38 — Context analysis for task T21.

APPENDIX B

Appendix: Usability Analysis and Rules

B.1 Usability Analysis

Table B.1 — A compilation of usability attributes for ubiquitous systems

Attribute	Nielsen [1993b]	Scholtz and Consolvo [2004]	Riekki et al. [2004]	Seffah et al. [2006]	Ryu et al. [2006]	Kim et al. [2008]	Kemp et al. [2008]	Mantoro [2009]	Quigley [2010]	Lee and Yun [2012]	Harrison et al. [2013]	Santos et al. [2013]	Evers et al. [2014]	Carvalho et al. [2017]
accessibility	-	-	-	●	-	●	-	-	-	-	-	-	-	-
accuracy	-	●	-	-	●	-	●	●	-	-	-	-	-	-
adaptability	-	-	-	-	●	●	-	●	-	-	-	-	-	-
adoption	-	●	-	-	-	-	-	-	-	-	-	-	-	●
appeal	-	●	-	-	-	-	●	-	-	-	-	-	-	-
attention	-	●	-	-	-	-	-	-	-	-	-	-	-	●
availability	-	-	●	-	-	-	-	-	-	-	-	●	-	●
awareness	-	-	-	-	●	-	●	-	-	-	-	●	-	-
cognitive load	-	●	●	-	-	-	●	-	-	-	●	●	-	●
compliance	-	-	-	-	●	-	●	●	-	-	-	-	-	-

Table B.1 — A compilation of usability attributes for ubiquitous systems

Attribute	Nielsen [1993b]	Scholtz and Consolvo [2004]	Riekki et al. [2004]	Seffah et al. [2006]	Ryu et al. [2006]	Kim et al. [2008]	Kemp et al. [2008]	Mantoro [2009]	Quigley [2010]	Lee and Yun [2012]	Harrison et al. [2013]	Santos et al. [2013]	Evers et al. [2014]	Carvalho et al. [2017]
consistency	-	-	-	-	-	●	●	●	-	-	-	-	●	-
context awareness	-	-	-	-	●	-	-	-	-	-	-	●	-	●
context-sensitve timing	-	-	●	-	-	-	-	-	-	-	-	-	-	-
controllability	-	●	-	-	●	●	●	●	-	●	-	●	●	-
device capability	-	-	-	-	-	-	-	-	-	-	-	●	-	●
discoverability	-	-	-	-	-	-	-	-	●	-	-	-	-	-
distributed systems support	-	-	-	-	●	-	-	-	-	-	-	-	-	-
ease of use	-	-	-	-	-	-	●	-	●	-	-	●	-	●
effectiveness	-	●	-	●	-	-	●	-	-	-	●	●	-	●
efficiency	●	●	-	●	●	-	●	●	●	●	●	●	-	●
errors	●	●	-	-	●	-	●	●	-	-	●	-	-	●
expressiveness	-	-	-	-	-	-	-	-	●	-	-	-	-	-
extensibility	-	-	-	-	●	-	-	-	-	-	-	-	-	-
familiarity	-	-	-	-	-	-	●	-	-	-	-	●	-	●
feedback	-	-	-	-	●	-	●	-	-	●	-	-	-	-
help and documentation	-	-	-	-	-	-	●	-	-	-	-	-	-	-
impact and side effects	-	●	-	-	-	-	-	-	-	-	-	-	-	-
interoperability	-	-	-	-	-	-	-	●	-	-	-	-	-	-
interconnectivity	-	-	-	-	●	●	-	-	-	●	-	-	-	●
learnability	●	-	-	●	-	-	-	●	-	-	●	-	-	-
maintainability	-	-	-	-	●	-	-	●	-	-	-	-	-	-
maturity	-	-	-	-	●	-	-	●	-	-	-	-	-	-
memorability	●	-	-	-	-	-	●	-	-	-	●	-	-	-
mental model match	-	●	-	-	-	●	-	-	-	-	-	-	-	-
mobility	-	-	-	-	●	●	-	-	-	-	-	-	-	●
network security	-	-	-	-	●	-	-	-	-	-	-	-	-	-
network capability or quality	-	-	-	-	●	-	-	-	-	-	-	●	-	●
personal-ization	-	-	-	-	●	-	-	-	-	●	-	-	-	-
predictability	-	-	-	-	●	●	-	-	-	-	-	-	-	●
privacy	-	●	-	-	●	-	-	-	-	-	-	●	-	●
productivity	-	-	-	●	-	-	-	-	-	-	-	-	-	-
programmability	-	-	-	-	-	-	-	-	●	-	-	-	-	-
relevancy of interaction	-	-	●	-	-	-	-	-	-	-	-	-	-	-
reliability	-	-	-	-	●	-	-	-	-	-	-	-	-	●
resource optimization	-	-	-	-	●	-	-	●	-	-	-	-	-	-
responsiveness	-	-	-	-	-	-	-	●	-	●	-	-	-	-
reversibility	-	-	-	-	●	-	-	-	-	-	-	-	-	●
safety & security	-	-	-	●	●	-	-	●	-	-	-	●	-	●

Table B.1 — A compilation of usability attributes for ubiquitous systems

Attribute	Nielsen [1993b]	Scholtz and Consolvo [2004]	Riekki et al. [2004]	Seffah et al. [2006]	Ryu et al. [2006]	Kim et al. [2008]	Kemp et al. [2008]	Mantoro [2009]	Quigley [2010]	Lee and Yun [2012]	Harrison et al. [2013]	Santos et al. [2013]	Evers et al. [2014]	Carvalho et al. [2017]
satisfaction	●	●	-	●	-	●	-	-	-	-	●	●	-	●
scalability	-	-	-	-	-	-	-	●	-	-	-	-	-	●
search-ableness	-	-	-	-	-	-	-	-	-	●	-	-	-	-
sensitivity	-	-	-	-	-	-	-	●	-	-	-	-	-	-
service resource availability	-	-	-	-	●	-	-	-	-	-	-	-	-	-
simplicity	-	-	-	-	●	●	-	-	-	-	-	-	-	●
suitability	-	-	-	-	●	-	-	●	-	-	-	-	-	-
timeliness	-	-	-	-	-	-	●	-	-	-	-	-	-	-
transparency	-	-	-	-	●	●	-	●	●	-	-	●	●	●
trust	-	●	-	●	-	-	●	-	-	-	-	-	-	●
understandability	-	-	-	-	●	-	-	●	-	-	-	-	-	-
universality	-	-	-	●	-	-	-	-	-	-	-	-	-	-
unobtrusiveness	-	-	-	-	●	-	-	-	●	-	-	-	-	-
updates	-	-	-	-	-	-	●	-	-	-	-	-	-	-
user recognition	-	-	-	-	●	-	-	-	-	-	-	-	-	-
utility	-	●	-	●	-	-	-	-	-	-	-	-	-	●
visual clarity	-	-	-	-	-	●	●	-	-	-	-	-	-	-

B.2 Usability Assessment and Filter Rules

$$\begin{aligned}(\text{B.1})\quad & uses(?option, ?device) \wedge providesFunction(?device, ?f)\\ & \wedge presents(?f, ?output) \wedge prefersToSense(?user, ?output)\\ & \implies isPreferred(?option, true)\end{aligned}$$

$$\begin{aligned}(\text{B.2})\quad & uses(?option, ?device) \wedge providesFunction(?device, ?f)\\ & \wedge presents(?f, ?output) \wedge isAbleToSense(?user, ?output)\\ & \implies isAccessible(?option, true)\end{aligned}$$

B.2.1 Consistency Rules

$$hasCurrentTask(?user, ?task) \land usedOutputOption(?task, ?option) \implies isConsistent(?option, true) \tag{B.3}$$

$$uses(?option, ?device) \land isUsedBy(?userID, ?device) \implies isConsistent(?option, true) \tag{B.4}$$

B.2.2 Perceptibility Rules

Perceptibility Rules for Smartphones

$$\begin{gathered} uses(?option, ?device1) \land rdf : type(?device1, Smartphone) \\ \land\, uses(?option, ?modality) \land rdf : type(?modality, AcousticOutput) \\ \land\, wears(?user, ?device2) \land rdf : type(?device2, Headphones) \\ \land\, connectedTo(?device2, ?device3) \land differentFrom(?device2, ?device3) \\ \implies hasPerceptibility(?option, NotPerceptible) \end{gathered} \tag{B.5}$$

$$\begin{gathered} uses(?option, ?device1) \land rdf : type(?device1, Smartphone) \\ \land\, uses(?option, ?modality) \land rdf : type(?modality, AcousticOutput) \\ \land\, wears(?user, ?device2) \land rdf : type(?device2, Headphones) \\ \land\, connectedTo(?device2, ?device3) \land sameAs(?device2, ?device3) \\ \implies hasPerceptibility(?option, Perceptible) \end{gathered} \tag{B.6}$$

$$\begin{gathered} uses(?option, ?device) \land rdf : type(?device, Smartphone) \\ \land\, uses(?option, ?modality) \land rdf : type(?modality, AcousticOutput) \\ \land\, highAmbientBrightness(?device, true) \land displayOn(?device, true) \\ \implies hasPerceptibility(?option, Perceptible) \end{gathered} \tag{B.7}$$

$$\begin{gathered} uses(?option, ?device) \land rdf : type(?device, Smartphone) \\ \land\, uses(?option, ?modality) \land rdf : type(?modality, VisualOutput) \\ \land\, displayOn(?device, true) \\ \implies hasPerceptibility(?option, Perceptible) \end{gathered} \tag{B.8}$$

$$uses(?option, ?device) \wedge rdf : type(?device, Smartphone) \\ \wedge uses(?option, ?modality) \wedge rdf : type(?modality, HapticOutput) \\ \implies hasPerceptibility(?option, ProbablyPerceptible) \tag{B.9}$$

$$uses(?option, ?device) \wedge rdf : type(?device, Smartphone) \\ \wedge uses(?option, ?modality) \wedge rdf : type(?modality, HapticOutput) \\ \wedge highAmbientBrightness(?device, true) \wedge displayOn(?device, true) \\ \implies hasPerceptibility(?option, Perceptible) \tag{B.10}$$

Percepibility Rules for Smartwatches

$$uses(?option, ?device1) \wedge rdf : type(?device1, Smartwatch) \\ \wedge uses(?option, ?modality) \wedge rdf : type(?modality, AcousticOutput) \\ \wedge providesFunction(?device1, ?function) \\ \wedge hasTechnicalProperty(?function, ?property) \\ \wedge onlyWorksWithHeadphones(?property, false) \\ \implies hasPerceptibility(?option, ProbablyPerceptible) \tag{B.11}$$

$$uses(?option, ?device1) \wedge rdf : type(?device1, Smartwatch) \\ \wedge uses(?option, ?modality) \wedge rdf : type(?modality, AcousticOutput) \\ \wedge providesFunction(?device1, ?function) \\ \wedge hasTechnicalProperty(?function, ?property) \\ \wedge onlyWorksWithHeadphones(?property, true) \\ \implies hasPerceptibility(?option, NotPerceptible) \tag{B.12}$$

$$uses(?option, ?device1) \wedge rdf : type(?device1, Smartwatch) \\ \wedge uses(?option, ?modality) \wedge rdf : type(?modality, AcousticOutput) \\ \wedge wears(?user, ?device2) \wedge rdf : type(?device2, Headphones) \\ \wedge connectedTo(?device2, ?device3) \wedge sameAs(?device2, ?device3) \\ \implies hasPerceptibility(?option, Perceptible) \tag{B.13}$$

$$\begin{aligned}&\mathit{uses}(?\mathit{option},?\mathit{device}1)\wedge \mathit{rdf}:\mathit{type}(?\mathit{device}1,\mathit{Smartwatch})\\&\wedge \mathit{uses}(?\mathit{option},?\mathit{modality})\wedge \mathit{rdf}:\mathit{type}(?\mathit{modality},\mathit{VisualOutput})\\&\wedge \mathit{wears}(?\mathit{user},?\mathit{device}2)\wedge \mathit{sameAs}(?\mathit{device}1,?\mathit{device}2)\\&\implies \mathit{hasPerceptibility}(?\mathit{option},\mathit{Perceptible})\end{aligned}\tag{B.14}$$

$$\begin{aligned}&\mathit{uses}(?\mathit{option},?\mathit{device}1)\wedge \mathit{rdf}:\mathit{type}(?\mathit{device}1,\mathit{Smartwatch})\\&\wedge \mathit{uses}(?\mathit{option},?\mathit{modality})\wedge \mathit{rdf}:\mathit{type}(?\mathit{modality},\mathit{HapticOutput})\\&\wedge \mathit{wears}(?\mathit{user},?\mathit{device}2)\wedge \mathit{sameAs}(?\mathit{device}1,?\mathit{device}2)\\&\implies \mathit{hasPerceptibility}(?\mathit{option},\mathit{Perceptible})\end{aligned}\tag{B.15}$$

Perceptibility Rules for Public Displays

$$\begin{aligned}&\mathit{uses}(?\mathit{option},?\mathit{device})\wedge \mathit{rdf}:\mathit{type}(?\mathit{device},\mathit{PublicDisplay})\\&\wedge \mathit{uses}(?\mathit{option},?\mathit{modality})\wedge \mathit{rdf}:\mathit{type}(?\mathit{modality},\mathit{AcousticOutput})\\&\wedge \mathit{isNear}(?\mathit{user},?\mathit{device})\\&\implies \mathit{hasPerceptibility}(?\mathit{option},\mathit{Perceptible})\end{aligned}\tag{B.16}$$

$$\begin{aligned}&\mathit{uses}(?\mathit{option},?\mathit{device}1)\wedge \mathit{rdf}:\mathit{type}(?\mathit{device}1,\mathit{PublicDisplay})\\&\wedge \mathit{uses}(?\mathit{option},?\mathit{modality})\wedge \mathit{rdf}:\mathit{type}(?\mathit{modality},\mathit{AcousticOutput})\\&\wedge \mathit{wears}(?\mathit{user},?\mathit{device}2)\wedge \mathit{rdf}:\mathit{type}(?\mathit{device}2,\mathit{Headphones})\\&\implies \mathit{hasPerceptibility}(?\mathit{option},\mathit{NotPerceptible})\end{aligned}\tag{B.17}$$

$$\begin{aligned}&\mathit{uses}(?\mathit{option},?\mathit{device})\wedge \mathit{rdf}:\mathit{type}(?\mathit{device},\mathit{PublicDisplay})\\&\wedge \mathit{uses}(?\mathit{option},?\mathit{modality})\wedge \mathit{rdf}:\mathit{type}(?\mathit{modality},\mathit{VisualOutput})\\&\wedge \mathit{isNear}(\mathit{userID},?\mathit{device})\\&\implies \mathit{hasPerceptibility}(?\mathit{option},\mathit{Perceptible})\end{aligned}\tag{B.18}$$

$$\begin{aligned}&\mathit{uses}(?\mathit{option},?\mathit{device})\wedge \mathit{rdf}:\mathit{type}(?\mathit{device},\mathit{PublicDisplay})\\&\wedge \mathit{uses}(?\mathit{option},?\mathit{modality})\wedge \mathit{rdf}:\mathit{type}(?\mathit{modality},\mathit{VisualOutput})\\&\wedge \mathit{isUsedBy}(?\mathit{user},?\mathit{device})\wedge \mathit{differentFrom}(?\mathit{user},\mathit{userID})\\&\implies \mathit{hasPerceptibility}(?\mathit{option},\mathit{NotPerceptible})\end{aligned}\tag{B.19}$$

Perceptibility Rules for Vehicle Displays

$$\begin{aligned}(B.20)\quad & uses(?option, ?device) \wedge rdf:type(?device, VehicleDisplay)\\ & \wedge uses(?option, ?modality) \wedge rdf:type(?modality, VisualOutput)\\ & \wedge isNear(userID, ?device)\\ & \implies hasPerceptibility(?option, Perceptible)\end{aligned}$$

Perceptibility Rules for Smart Windows

$$\begin{aligned}(B.21)\quad & uses(?option, ?device) \wedge rdf:type(?device, SmartWindow)\\ & \wedge uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput)\\ & \wedge isNear(userID, ?device)\\ & \implies hasPerceptibility(?option, Perceptible)\end{aligned}$$

$$\begin{aligned}(B.22)\quad & uses(?option, ?device) \wedge rdf:type(?device, SmartWindow)\\ & \wedge uses(?option, ?modality) \wedge rdf:type(?modality, VisualOutput)\\ & \wedge isNear(userID, ?device)\\ & \implies hasPerceptibility(?option, ProbablyPerceptible)\end{aligned}$$

$$\begin{aligned}(B.23)\quad & uses(?option, ?device) \wedge rdf:type(?device, SmartWindow)\\ & \wedge uses(?option, ?modality) \wedge rdf:type(?modality, VisualOutput)\\ & \wedge isNear(userID, ?device) \wedge isUsedBy(userID, ?device)\\ & \implies hasPerceptibility(?option, Perceptible)\end{aligned}$$

B.2.3 Privacy Rules

$$\begin{aligned}(B.24)\quad & uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput)\\ & \implies providesPrivacy(?option, NoPrivacy)\end{aligned}$$

$$\begin{aligned}(B.25)\quad & uses(?option, ?modality) \wedge rdf:type(?modality, AcousticOutput)\\ & \wedge uses(?option, ?device1) \wedge isPublic(device1, true)\\ & \implies providesPrivacy(?option, LowPrivacy)\end{aligned}$$

$$uses(?option, ?modality) \land rdf : type(?modality, AcousticSignal) \\ \implies providesPrivacy(?option, MediumPrivacy) \tag{B.26}$$

$$hasHome(?user, ?location1) \land hasLocation(?user, ?location2) \\ \land sameAs(?location1, ?location2) \\ \implies providesPrivacy(?option, HighPrivacy) \tag{B.27}$$

$$uses(?option, ?modality) \land rdf : type(?modality, AcousticOutput) \\ \land uses(?option, ?device1) \land wears(?user, ?device2) \\ \land rdf : type(?device2, Headphones) \land connectedTo(device2, device1) \\ \implies providesPrivacy(?option, HighPrivacy) \tag{B.28}$$

$$uses(?option, ?modality) \land rdf : type(?modality, VisualOutput) \\ \land uses(?option, ?device) \land isPublic(?device, false) \\ \implies providesPrivacy(?option, HighPrivacy) \tag{B.29}$$

$$uses(?option, ?modality) \land rdf : type(?modality, VisualOutput) \\ \land uses(?option, ?device) \land isPublic(?device, true) \\ \land isUsedBy(?device, ?user) \land sameAs(?user, UserID) \\ \implies providesPrivacy(?option, MediumPrivacy) \tag{B.30}$$

$$uses(?option, ?modality) \land rdf : type(?modality, VisualOutput) \\ \land uses(?option, ?device) \land isPublic(?device, true) \\ \land isUsedBy(?device, ?user) \land differentFrom(?user, UserID) \\ \implies providesPrivacy(?option, LowPrivacy) \tag{B.31}$$

$$uses(?option, ?modality) \land rdf : type(?modality, HapticOutput) \\ \implies providesPrivacy(?option, MediumPrivacy) \tag{B.32}$$

B.2.4 Message Sensitivity Rules

$$\begin{aligned} &hasActiveItinerary(?user, ?trip) \wedge hasTripLeg(?trip, ?leg) \\ &\quad \wedge hasOrigin(?leg, ?location) \wedge references(?message, ?location) \\ &\qquad \implies hasSensitivity(?message, LowSensitivity) \end{aligned} \tag{B.33}$$

$$\begin{aligned} &hasActiveItinerary(?user, ?trip) \wedge hasTripLeg(?trip, ?leg) \\ &\quad \wedge hasDestination(?leg, ?location) \wedge references(?message, ?location) \\ &\qquad \implies hasSensitivity(?message, LowSensitivity) \end{aligned} \tag{B.34}$$

$$\begin{aligned} &hasActiveItinerary(?user, ?trip) \wedge hasOrigin(?trip, ?location) \\ &\quad \wedge references(?message, ?location) \\ &\qquad \implies hasSensitivity(?message, LowSensitivity) \end{aligned} \tag{B.35}$$

$$\begin{aligned} &hasActiveItinerary(?user, ?trip) \wedge hasDestination(?trip, ?location) \\ &\quad \wedge references(?message, ?location) \\ &\qquad \implies hasSensitivity(?message, LowSensitivity) \end{aligned} \tag{B.36}$$

$$\begin{aligned} &contains(?message, ?name) \wedge hasFirstName(?user, ?name) \\ &\qquad \implies hasSensitivity(?message, HighSensitivity) \end{aligned} \tag{B.37}$$

$$\begin{aligned} &contains(?message, ?name) \wedge hasLastName(?user, ?name) \\ &\qquad \implies hasSensitivity(?message, HighSensitivity) \end{aligned} \tag{B.38}$$

$$\begin{aligned} &isAddress(?content, true) \wedge contains(?message, ?content) \\ &\qquad \implies hasSensitivity(?message, HighSensitivity) \end{aligned} \tag{B.39}$$

$$\begin{aligned} &hasLocation(?user, ?location) \wedge references(?message, ?location) \\ &\qquad \implies hasSensitivity(?message, HighSensitivity) \end{aligned} \tag{B.40}$$

B.2.5 Unobtrusiveness Rules

$$\begin{aligned} & hasHome(UserID, ?location1) \wedge hasLocation(UserID, ?location2) \\ & \wedge sameAs(?location1, ?location2) \\ & \implies hasUnobtrusiveness(?option, Unobtrusive) \end{aligned} \tag{B.41}$$

$$\begin{aligned} & uses(?option, ?device) \wedge isUsedBy(?device, ?user) \\ & \wedge differentFrom(?user, UserID) \\ & \implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned} \tag{B.42}$$

$$\begin{aligned} & uses(?option, ?device) \wedge rdf : type(?device, PublicDisplay) \\ & \wedge isNear(?user, ?device) \wedge differentFrom(?user, UserID) \\ & \implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned} \tag{B.43}$$

$$\begin{aligned} & uses(?option, ?modality) \wedge rdf : type(?modality, AcousticOutput) \\ & \implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned} \tag{B.44}$$

$$\begin{aligned} & uses(?option, ?modality) \wedge uses(?option, ?device1) \\ & \wedge rdf : type(?modality, AcousticOutput) \wedge wears(UserID, ?device2) \\ & \wedge rdf : type(?device2, Headphones) \wedge connectedTo(device2, device1) \\ & \implies hasUnobtrusiveness(?option, Unobtrusive) \end{aligned} \tag{B.45}$$

$$\begin{aligned} & uses(?option, ?modality) \wedge rdf : type(?modality, VisualOutput) \\ & \wedge hasCurrentTask(UserID, ?task) \wedge rdf : type(?task, BoardingTask) \\ & \implies hasUnobtrusiveness(?option, Unobtrusive) \end{aligned} \tag{B.46}$$

$$\begin{aligned} & uses(?option, ?modality) \wedge rdf : type(?modality, VisualOutput) \\ & \wedge hasCurrentTask(UserID, ?task) \wedge rdf : type(?task, AlightingTask) \\ & \implies hasUnobtrusiveness(?option, Unobtrusive) \end{aligned} \tag{B.47}$$

$$\begin{aligned} &uses(?option, ?modality) \wedge rdf : type(?modality, AcousticOutput) \\ &\wedge hasCurrentTask(UserID, ?task) \wedge rdf : type(?task, BoardingTask) \\ &\wedge hasValidityStartTime(?message, ?start) \wedge hasValidityEndTime(?message, ?end) \\ &\wedge greaterThan(Now, ?end) \\ &\implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned} \tag{B.48}$$

$$\begin{aligned} &uses(?option, ?modality) \wedge rdf : type(?modality, AcousticOutput) \\ &\wedge hasCurrentTask(UserID, ?task) \wedge rdf : type(?task, AlightingTask) \\ &\wedge hasValidityStartTime(?message, ?start) \wedge hasValidityEndTime(?message, ?end) \\ &\wedge greaterThan(Now, ?end) \\ &\implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned} \tag{B.49}$$

$$\begin{aligned} &uses(?option, ?modality) \wedge rdf : type(?modality, HapticOutput) \\ &\wedge hasCurrentTask(UserID, ?task) \wedge rdf : type(?task, BoardingTask) \\ &\wedge hasValidityStartTime(?message, ?start) \wedge hasValidityEndTime(?message, ?end) \\ &\wedge greaterThan(Now, ?end) \\ &\implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned} \tag{B.50}$$

$$\begin{aligned} &uses(?option, ?modality) \wedge rdf : type(?modality, HapticOutput) \\ &\wedge hasCurrentTask(UserID, ?task) \wedge rdf : type(?task, AlightingTask) \\ &\wedge hasValidityStartTime(?message, ?start) \wedge hasValidityEndTime(?message, ?end) \\ &\wedge greaterThan(Now, ?end) \\ &\implies hasUnobtrusiveness(?option, Obtrusive) \end{aligned} \tag{B.51}$$

$$\begin{aligned} &uses(?option, ?modality) \wedge uses(?option, ?device) \\ &\wedge isUsedBy(?device, userID) \wedge rdf : type(?modality, VisualOutput) \\ &\implies hasUnobtrusiveness(?option, Unobtrusive) \end{aligned} \tag{B.52}$$

$$uses(?option, ?device1) \land isUsedBy(?device2, UserID) \land differentFrom(device1, device2) \implies hasUnobtrusiveness(?option, Obtrusive) \tag{B.53}$$

B.2.6 Relatability Rules

$$uses(?option, ?device) \land isPublic(?device, false) \implies hasRelatability(?option, HighRelatability) \tag{B.54}$$

$$hasValidityStartTime(?message, ?start) \land hasValidityEndTime(?message, ?end) \land greaterThanOrEqual(Now, ?start) \land lessThanOrEqual(Now, ?end) \implies hasRelatability(?message, LowRelatability) \tag{B.55}$$

$$hasCurrentTask(?user, ?task) \land onLine(?task, ?line) \land hasDirection(?line, ?direction) \land references(?message, ?line) \land references(?message, ?direction) \implies hasRelatability(?message, MediumRelatability) \tag{B.56}$$

$$hasCurrentTask(?user, ?task) \land at(?task, ?stop) \land references(?message, ?stop) \implies hasRelatability(?message, MediumRelatability) \tag{B.57}$$

$$hasLocation(?user, ?location) \land references(?message, ?location) \implies hasRelatability(?message, MediumRelatability) \tag{B.58}$$

$$hasActiveItinerary(?user, ?trip) \land hasOrigin(?trip, ?stop) \\ \land references(?message, ?stop) \\ \implies hasRelatability(?message, HighRelatability) \tag{B.59}$$

$$hasActiveItinerary(?user, ?trip) \land hasDestination(?trip, ?stop) \\ \land references(?message, ?stop) \\ \implies hasRelatability(?message, HighRelatability) \tag{B.60}$$

$$hasFirstName(?user, ?name) \land contains(?message, ?name) \\ \implies hasRelatability(?message, HighRelatability) \tag{B.61}$$

$$hasLastName(?user, ?name) \land contains(?message, ?name) \\ \implies hasRelatability(?message, HighRelatability) \tag{B.62}$$

B.2.7 Relevance Rules

$$hasActiveItinerary(?user, ?trip) \land hasDestination(?trip, ?stop) \\ \land references(?message, ?stop) \\ \implies hasRelevance(?message, LowRelevance) \tag{B.63}$$

$$hasActiveItinerary(?user, ?trip) \land hasTripLeg(?trip, ?leg) \\ \land hasOrigin(?leg, ?location) \land references(?message, ?location) \\ \implies hasRelevance(?message, LowRelevance) \tag{B.64}$$

$$hasCurrentTask(?user, ?task) \land onLine(?task, ?line) \\ \land hasDirection(?line, ?direction) \\ \land references(?message, ?line) \land references(?message, ?direction) \\ \implies hasRelevance(?message, LowRelevance) \tag{B.65}$$

$$\begin{aligned} &\mathit{hasActiveItinerary}(?user, ?trip) \land \mathit{hasTripLeg}(?trip, ?leg) \\ &\land \mathit{hasDestination}(?leg, ?location) \land \mathit{references}(?message, ?location) \\ &\qquad\Longrightarrow \mathit{hasRelevance}(?message, \mathit{LowRelevance}) \end{aligned} \tag{B.66}$$

$$\begin{aligned} &\mathit{isInformative}(?message, \mathit{true}) \\ &\qquad\Longrightarrow \mathit{hasRelevance}(?message, \mathit{MediumRelevance}) \end{aligned} \tag{B.67}$$

$$\begin{aligned} &\mathit{isCallToAction}(?message, \mathit{true}) \\ &\qquad\Longrightarrow \mathit{hasRelevance}(?message, \mathit{HighRelevance}) \end{aligned} \tag{B.68}$$

B.2.8 Urgency Rules

$$\begin{aligned} &\mathit{hasValidityStartTime}(?message, ?start) \land \mathit{lessThanOrEqual}(?start, \mathit{Now}) \\ &\land \mathit{subtractTimes}(?distance, ?start, \mathit{Now}) \land \mathit{lessThan}(?distance, \mathit{VerySoon}) \\ &\qquad\Longrightarrow \mathit{hasUrgency}(?message, \mathit{HighUrgency}) \end{aligned} \tag{B.69}$$

$$\begin{aligned} &\mathit{hasValidityEndTime}(?message, ?end) \land \mathit{greaterThanOrEqual}(?end, \mathit{Now}) \\ &\qquad\Longrightarrow \mathit{hasUrgency}(?message, \mathit{HighUrgency}) \end{aligned} \tag{B.70}$$

$$\begin{aligned} &\mathit{hasValidityStartTime}(?message, ?start) \land \mathit{lessThanOrEqual}(?start, \mathit{Now}) \\ &\land \mathit{subtractTimes}(?distance, ?start, \mathit{Now}) \land \mathit{lessThan}(?distance, \mathit{Soon}) \\ &\qquad\Longrightarrow \mathit{hasUrgency}(?message, \mathit{MediumUrgency}) \end{aligned} \tag{B.71}$$

$$\begin{aligned} &\mathit{hasValidityEndTime}(?message, ?end) \land \mathit{greaterThanOrEqual}(?end, \mathit{Now}) \\ &\land \mathit{subtractTimes}(?distance, ?end, \mathit{Now}) \land \mathit{lessThan}(?distance, \mathit{Soon}) \\ &\qquad\Longrightarrow \mathit{hasUrgency}(?message, \mathit{MediumUrgency}) \end{aligned} \tag{B.72}$$

(B.73)

$$isCallToAction(?message, false) \implies hasUrgency(?message, NoUrgency)$$

(B.74)

$$hasValidityStartTime(?message, ?start) \wedge lessThanOrEqual(?start, Now) \wedge subtractTimes(?distance, ?start, Now) \wedge moreThan(?distance, Soon) \implies hasUrgency(?message, NoUrgency)$$

(B.75)

$$hasValidityEndTime(?message, ?end) \wedge greaterThanOrEqual(?end, Now) \wedge subtractTimes(?distance, ?end, Now) \wedge moreThan(?distance, Soon) \implies hasUrgency(?message, NoUrgency)$$

B.2.9 Message Requirement Rules

(B.76)

$$hasContentType(?message, ?content) \wedge uses(?option, ?modality) \wedge rdf : type(?content, ?contentType) \wedge rdf : type(?modality, ?modalityType) \wedge sameAs(?contentType, ?modalityType) \implies include(?option, true)$$

(B.77)

$$hasContentType(?message, ?content) \wedge uses(?option, ?modality) \wedge converterInput(?converter, ?input) \wedge converterOutput(?converter, ?output) \wedge rdf : type(?content, ?contentType) \wedge rdf : type(?modality, ?modalityType) \wedge rdf : type(?input, ?inputType) \wedge rdf : type(?output, ?outputType) \wedge sameAs(?contentType, ?inputType) \wedge sameAs(?modality, ?outputType) \implies include(?option, true)$$

B.2.10 Perceptibility Filter Rules

(B.78)

$$hasPerceptibility(?option, Perceptible) \implies include(?option, true)$$

$$\begin{aligned}&\mathit{hasPerceptibility}(?\mathit{option}, \mathit{ProbablyPerceptible})\\ &\qquad\qquad\qquad\qquad \Longrightarrow \mathit{include}(?\mathit{option}, \mathit{true})\end{aligned}\tag{B.79}$$

B.2.11 Privacy and Sensitivity Filter

$$\begin{aligned}&\mathit{hasSensitivity}(?\mathit{message}, \mathit{HighSensitivity})\\ &\qquad \wedge \mathit{providesPrivacy}(?\mathit{option}, \mathit{HighPrivacy})\\ &\qquad\qquad\qquad\qquad \Longrightarrow \mathit{include}(?\mathit{option}, \mathit{true})\end{aligned}\tag{B.80}$$

$$\begin{aligned}&\mathit{hasSensitivity}(?\mathit{message}, \mathit{MediumSensitivity})\\ &\qquad \wedge \mathit{providesPrivacy}(?\mathit{option}, \mathit{HighPrivacy})\\ &\qquad\qquad\qquad\qquad \Longrightarrow \mathit{include}(?\mathit{option}, \mathit{true})\end{aligned}\tag{B.81}$$

$$\begin{aligned}&\mathit{hasSensitivity}(?\mathit{message}, \mathit{MediumSensitivity})\\ &\qquad \wedge \mathit{providesPrivacy}(?\mathit{option}, \mathit{MediumPrivacy})\\ &\qquad\qquad\qquad\qquad \Longrightarrow \mathit{include}(?\mathit{option}, \mathit{true})\end{aligned}\tag{B.82}$$

$$\begin{aligned}&\mathit{hasSensitivity}(?\mathit{message}, \mathit{LowSensitivity})\\ &\qquad \wedge \mathit{providesPrivacy}(?\mathit{option}, \mathit{HighPrivacy})\\ &\qquad\qquad\qquad\qquad \Longrightarrow \mathit{include}(?\mathit{option}, \mathit{true})\end{aligned}\tag{B.83}$$

$$\begin{aligned}&\mathit{hasSensitivity}(?\mathit{message}, \mathit{LowSensitivity})\\ &\qquad \wedge \mathit{providesPrivacy}(?\mathit{option}, \mathit{MediumPrivacy})\\ &\qquad\qquad\qquad\qquad \Longrightarrow \mathit{include}(?\mathit{option}, \mathit{true})\end{aligned}\tag{B.84}$$

$$\begin{aligned}&\mathit{hasSensitivity}(?\mathit{message}, \mathit{LowSensitivity})\\ &\qquad \wedge \mathit{providesPrivacy}(?\mathit{option}, \mathit{LowPrivacy})\\ &\qquad\qquad\qquad\qquad \Longrightarrow \mathit{include}(?\mathit{option}, \mathit{true})\end{aligned}\tag{B.85}$$

$$hasSensitivity(?message, NoSensitivity) \wedge providesPrivacy(?option, ?anyPrivacy) \implies include(?option, true) \tag{B.86}$$

$$hasUrgency(?message, HighUrgency) \wedge hasSensitivity(?message, MediumSensitivity) \wedge providesPrivacy(?option, ?anyPrivacy) \implies include(?option, true) \tag{B.87}$$

$$hasUrgency(?message, HighUrgency) \wedge hasSensitivity(?message, LowSensitivity) \wedge providesPrivacy(?option, ?anyPrivacy) \implies include(?option, true) \tag{B.88}$$

$$hasUrgency(?message, HighUrgency) \wedge hasSensitivity(?message, NoSensitivity) \wedge providesPrivacy(?option, ?anyPrivacy) \implies include(?option, true) \tag{B.89}$$

$$hasUrgency(?message, MediumUrgency) \wedge hasSensitivity(?message, MediumSensitivity) \wedge providesPrivacy(?option, HighPrivacy) \implies include(?option, true) \tag{B.90}$$

$$hasUrgency(?message, MediumUrgency) \wedge hasSensitivity(?message, MediumSensitivity) \wedge providesPrivacy(?option, MediumPrivacy) \implies include(?option, true) \tag{B.91}$$

$$\begin{aligned}(B.92)\quad & hasUrgency(?message, MediumUrgency) \\ & \wedge hasSensitivity(?message, MediumSensitivity) \\ & \wedge providesPrivacy(?option, LowPrivacy) \\ & \implies include(?option, true)\end{aligned}$$

$$\begin{aligned}(B.93)\quad & hasUrgency(?message, NoUrgency) \\ & \wedge hasSensitivity(?message, MediumSensitivity) \\ & \wedge providesPrivacy(?option, HighPrivacy) \\ & \implies include(?option, true)\end{aligned}$$

$$\begin{aligned}(B.94)\quad & hasUrgency(?message, NoUrgency) \\ & \wedge hasSensitivity(?message, MediumSensitivity) \\ & \wedge providesPrivacy(?option, MediumPrivacy) \\ & \implies include(?option, true)\end{aligned}$$

$$\begin{aligned}(B.95)\quad & hasUrgency(?message, NoUrgency) \\ & \wedge hasSensitivity(?message, MediumSensitivity) \\ & \wedge providesPrivacy(?option, LowPrivacy) \\ & \implies include(?option, true)\end{aligned}$$

B.2.12 Relatability and Relevance Filter

$$\begin{aligned}(B.96)\quad & hasRelatability(?message, MediumRelatability) \\ & \wedge hasRelevance(?message, HighRelevance) \\ & \wedge hasRelatability(?option, HighRelatability) \\ & \implies include(?option, true)\end{aligned}$$

$$\begin{aligned} &hasRelatability(?message, MediumRelatability) \\ &\wedge hasRelevance(?message, MediumRelevance) \\ &\wedge hasRelatability(?option, HighRelatability) \\ &\implies include(?option, true) \end{aligned} \tag{B.97}$$

$$\begin{aligned} &hasRelatability(?message, LowRelatability) \\ &\wedge hasRelevance(?message, HighRelevance) \\ &\wedge hasRelatability(?option, HighRelatability) \\ &\implies include(?option, true) \end{aligned} \tag{B.98}$$

$$\begin{aligned} &hasRelatability(?message, LowRelatability) \\ &\wedge hasRelevance(?message, MediumRelevance) \\ &\wedge hasRelatability(?option, HighRelatability) \\ &\implies include(?option, true) \end{aligned} \tag{B.99}$$

$$\begin{aligned} &hasRelatability(?message, NoRelatability) \\ &\wedge hasRelevance(?message, HighRelevance) \\ &\wedge hasRelatability(?option, HighRelatability) \\ &\implies include(?option, true) \end{aligned} \tag{B.100}$$

$$\begin{aligned} &hasRelatability(?message, NoRelatability) \\ &\wedge hasRelevance(?message, MediumRelevance) \\ &\wedge hasRelatability(?option, HighRelatability) \\ &\implies include(?option, true) \end{aligned} \tag{B.101}$$

$$\begin{aligned} &hasRelatability(?message, NoRelatability) \\ &\wedge hasRelevance(?message, LowRelevance) \\ &\wedge hasRelatability(?option, HighRelatability) \\ &\implies include(?option, true) \end{aligned} \tag{B.102}$$

APPENDIX C

Appendix: Evaluation Results

C.1 Evaluation Scenarios

The following scenarios were used for the evaluation. The scenario descriptions were used in German, since the online questionnaire targeted a german audience.

C.1.1 Scenario 1: A Delay on the Commute

Text Scenario 1: Michael Baumann pendelt jeden Tag auf der Strecke zwischen Grötzingen Oberausstraße und ZKM. Meist hört er dabei mit Kopfhörern Musik oder Podcasts. Heute steht er um 06:55 Uhr ebenfalls auf dem Bahnsteig und wartet auf die nächste S4. Er trägt seine Smartwatch, sein Smartphone ist in seiner Westentasche. Über seine Kopfhörer hört er einen Nachrichtenpodcast auf seinem Smartphone. Er steht direkt neben dem interaktiven Public Display auf dem Bahnsteig. In seiner Mobilitäts-App hat Michael eingestellt, dass er Nachrichten bevorzugt per Audio-Ausgabe erhält.

Text Scenario 1 in English: Michael Baumann commutes everyday between the stops Grötzingen Oberausstraße and ZKM. Most of the time, he listens to music or podcasts on his headphones. Today, at 06:44am he is standing on the platform as always and is waiting for the next S4. He is wearing his smartwatch and his smartphone is in his vest pocket. Using his headphones he listens to a news podcast on his smartphone. He is standing right next to an interactive

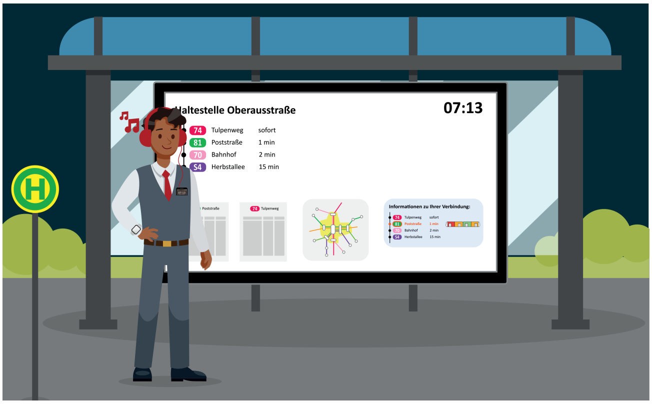


Figure C.1 — Picture illustrating scenario 1 in the online questionnaire. This graphic was created by Lars Erber based on my instructions.

public display on the platform. Michael configured his mobility app to express that he prefers to receive messages via audio output.

Message Scenario 1: Michael erreicht nun folgende Nachricht: "Die S4 in Richtung Karlsruhe Albtalbahnhof verspätet sich um 35 Minuten. Fahrgäste in Richtung Karlsruhe können die S5 um 07:20 Uhr nutzen."

Message Scenario 1 in English: Michael now recevies the following message: "The vehicle S4 in the direction Karlsruhe Albtalbahnhof is delayed for 35 minutes. Passengers traveling in the direction of Karlsruhe can use the S5 at 07:20am."

Available Output Options in Scenario 1:

- Vibration der Smartwatch und Text-Ausgabe auf der Smartwatch
- Audio-Ausgabe auf der Smartwatch
- Text-Ausgabe auf der Smartwatch
- Vibration des Smartphones und Text-Ausgabe auf dem Smartphone
- Audio-Ausgabe auf dem Smartphone (auf die Kopfhörer)
- Text-Ausgabe auf dem Smartphone
- Audio-Ausgabe über das Public Display

- Text-Ausgabe auf dem Public Display

C.1.2 Scenario 2: Alight on the Next Stop Going Shopping with the Kids

Text Scenario 2: Martina Grundler ist mit ihren Kindern auf dem Weg in die Stadt, da alle drei Kinder neue Schuhe für den kommenden Herbst brauchen. Nach einem Umstieg sitzt Martina mit ihren Kindern in einer Straßenbahn der Linie 1 in Richtung Stadtmitte. Sie haben sich auf vier Sitzplätze neben ein Smart Window gesetzt und die Kinder probieren das Smart Window begeistert aus. Nach einer Weile schaut auch Martina sich das Smart Window genauer an und lässt sich auf einer Karte die Geschäfte in der Nähe Ihrer Ausstiegshaltestelle Marktplatz anzeigen. Ihr Smartphone hat sie in ihrer Tasche verstaut. In ihrer Mobilitäts-App hat Martina eingestellt, dass sie Nachrichten bevorzugt per Text-Ausgabe erhält.

Text Scenario 2 in English: Martina Grundler is on her way to the city with her kids because all three kids need new shoes for next fall. After a change, Martina is sitting in a tram of the line 1 in the direction towards the city with her kids. They are sitting on four seats next to a Smart Window and the kids are testing the smart window enthusiastically. After a while, Martina is also looking at the Smart Window and explores a map to see the shops near their last stop. Her smartphone is in her bag. In her mobility app, Martina has set that she prefers to receive messages via text output.

Message Scenario 2: Martina erreicht nun folgende Nachricht: ”Nächste Haltestelle: Marktplatz.“

Message Scenario 2 in English: Martina now receives the following message: ”Next stop: Marktplatz“

Available Output Options in Scenario 2:

- Vibration des Smartphones und Text-Ausgabe auf dem Smartphone
- Audio-Ausgabe auf dem Smartphone
- Text-Ausgabe auf dem Smartphone
- Text-Ausgabe auf dem Smart Window

Figure C.2 — Picture illustrating scenario 2 in the online questionnaire. This graphic was created by Lars Erber based on my instructions.

C.1.3 Scenario 3: A Detour on the Commute

Text Scenario 3: Michael Baumann pendelt jeden Tag zwischen Grötzingen Oberausstraße und ZKM. Auch heute ist er auf dieser Strecke unterwegs ins Büro. Er sitzt in der S-Bahn S51 neben einem Smart Window. Sein Smartphone ist in seiner Manteltasche verstaut. Er trägt seine Smartwatch. In seiner Mobilitäts-App hat Michael eingestellt, dass er Nachrichten bevorzugt per Audio-Ausgabe erhält. Auf der Anzeige über dem Gang läuft Werbung, daneben werden die nächsten Haltestellen angezeigt. Die nächste Haltestelle ist Karlsruhe-Durlach.

Text Scenario 3 in English: Michael Baumann commutes everyday between the stops Grötzingen Oberausstraße and ZKM. Today he also is on this route on his way to his office. He is sitting in a train of the line S51 next to a Smart Window. His smartphone is in his coat pocket. He is wearing his smartwatch. Michael configured his mobility app to express that he prefers to receive messages via audio output. On a display above the aisle, advertisements are shown and next to them, the next stops are displayed. The next stop is Karlsruhe-Durlach.

Message Scenario 3: Michael erreicht nun folgende Nachricht: " Bitte beachten Sie: Aufgrund eines Unfalls auf der Strecke wird dieses Fahrzeug nach dem nächsten Halt umgeleitet. Bitte steigen Sie daher an der Haltestelle Karlsruhe-

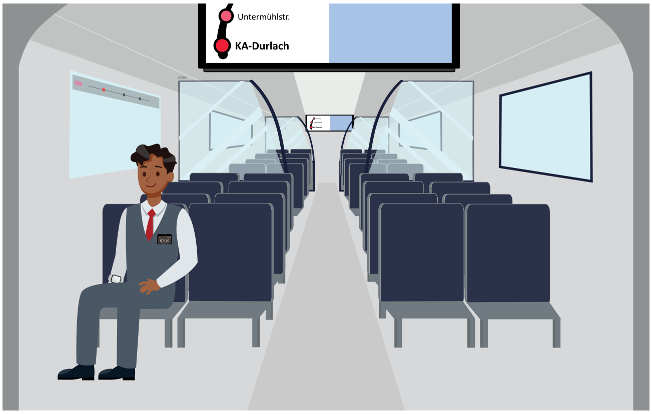


Figure C.3 — Picture illustrating scenario 3 in the online questionnaire. This graphic was created by Lars Erber based on my instructions.

Durlach aus und nutzen Sie die S3 Richtung Karlsruhe Hbf und von dort in die Straßenbahn 2 Richtung Rheinbergstraße, um die Haltestelle ZKM zu erreichen."

Message Scenario 3 in English: Michael now receives the following message: ”Please note: Due to an accident on the route, this vehicle will be diverted after the next stop. Please alight on the next stop Karlsruhe-Durlach and use the S3 in the direction Karlsruhe Hbf. From there, use the tram of the line 2 in the direction Rheinbergstraße to reach the stop ZKM.“

Available Output Options in Scenario 3:

- Vibration der Smartwatch und Text-Ausgabe auf der Smartwatch
- Audio-Ausgabe auf der Smartwatch
- Text-Ausgabe auf der Smartwatch
- Vibration des Smartphones und Text-Ausgabe auf dem Smartphone
- Audio-Ausgabe auf dem Smartphone
- Text-Ausgabe auf dem Smartphone
- Text-Ausgabe auf der Anzeige über dem Gang
- Text-Ausgabe auf dem Smart Window

Figure C.4 — Picture illustrating scenario 4 in the online questionnaire. This graphic was created by Lars Erber based on my instructions.

C.1.4 Scenario 4: Platform Changes at the Station

Text Scenario 4: Martina Grundler besucht mit ihren Kindern ihre Schwester und deren Familie in Rastatt. Sie hat sich zu Hause eine Verbindung mit dem öffentlichen Verkehr herausgesucht und auf ihr Smartphone übertragen. In ihrer Mobilitäts-App hat Martina eingestellt, dass sie Nachrichten bevorzugt per Text-Ausgabe erhält. Da sie über das Wochenende dort bleiben möchte, hat Martina Gepäck dabei und alle Hände voll zu tun. Am Hauptbahnhof muss sie nun in den Regionalexpress umsteigen. Martina und ihre Kinder stehen gerade an der Straßenbahnhaltestelle, wo sie soeben aus der Straßenbahn ausgestiegen sind. Martina versucht, sich anhand der Karte des Bahnhofs auf dem Public Display zu orientieren.

Text Scenario 4 in English: Martina Grundler wants to visit her sister's family in Rastatt with her children. At home, she already has picked a trip using public transport and has transmitted her itinerary to her Smartphone. In her mobility app, Martina has set that she prefers to receive messages via text output. Since they want to spend the weekend, Martina has some luggage and has her hands full. At the main station, they need to change into a regional train. Martina and her kids are standing at the tram stop, where they alighted a tram just now. Martina tries to orient herself using the station map on the public display.

Message Scenario 4: Martina erreicht nun folgende Nachricht: "Der Regionalexpress RE4723 in Richtung Radolfzell über Bahnhof Rastatt fährt heute um Uhr 10:09 Uhr ab Gleis 12."

Message Scenario 4 in English: Martina now receives the following message: ”The regional train RE4723 in the direction Radolfzell via Rastatt station leaves at 10:09am on platform 12 today.“

Available Output Options in Scenario 4:

- Vibration des Smartphones und Text-Ausgabe auf dem Smartphone
- Audio-Ausgabe auf dem Smartphone
- Text-Ausgabe auf dem Smartphone
- Text-Ausgabe auf dem Public Display

C.1.5 Scenario 5: A Delay During a Leisure Trip

Text Scenario 5: Michael Baumann fährt Sonntag morgens nach Bad Herrenalb. Er trifft sich dort mit Freunden, um an diesem schönen Frühlingstag zu wandern. Nach einem Umstieg in der Stadt sitzt er nun in der S1 Richtung Bad Herrenalb neben einem Smart Window. Michael trägt seine Smartwatch und hat sein Smartphone im Rucksack verstaut. In seiner Mobilitäts-App hat Michael eingestellt, dass er Nachrichten bevorzugt per Audio-Ausgabe erhält. Ab und zu schaut er auf der Anzeige über dem Gang nach den nächsten Haltestellen. Auf dem Smart Window schaut er sich die Strecke genauer an und sucht auf der Karte nach weiteren Wanderwegen in der Gegend, für die nächsten Wanderungen.

Text Scenario 5 in English: Michael Bauman is going to Bad Herrenalb on a sunday morning. He will meet some friends there, to go for a hike on this nice spring day. After a change in the city, he is now seated in the S1 in the direction Bad Herrenalb, next to a Smart Window. Michael is wearing his smartwatch and has put his smartphone in his backpack. Michael configured his mobility app to express that he prefers to receive messages via audio output. From time to time, he looks at the next stops shown on the display over the aisle. On the Smart Window, he inspects the route and searches for hiking trails in the vicinity for his next hikes.

Message Scenario 5: Michael erreicht nun folgende Nachricht: ”Das Fahrzeug S1 Richtung Herrenalb ist heute um 5 Minuten verspätet."

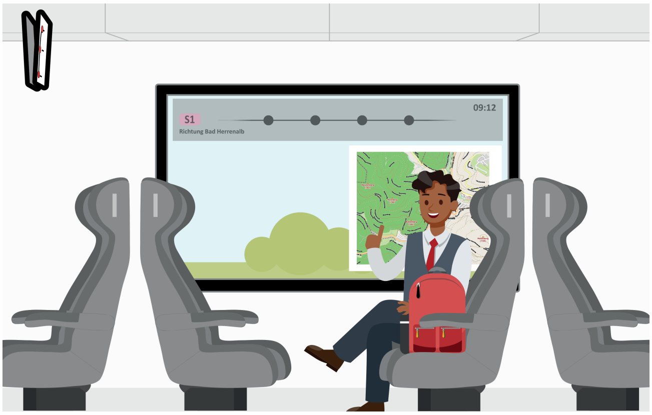


Figure C.5 — Picture illustrating scenario 5 in the online questionnaire. This graphic was created by Lars Erber based on my instructions.

Message Scenario 5 in English: Michael now receives the following message: ″The vehicle S1 in the direction Bad Herrenalb has a delay of 5 minutes today.“

Available Output Options in Scenario 5:

- Vibration der Smartwatch und Text-Ausgabe auf der Smartwatch
- Audio-Ausgabe auf der Smartwatch
- Text-Ausgabe auf der Smartwatch
- Vibration des Smartphones und Text-Ausgabe auf dem Smartphone
- Text-Ausgabe auf dem Smartphone
- Audio-Ausgabe auf dem Smartphone
- Text-Ausgabe auf der Anzeige über dem Gang
- Text-Ausgabe auf dem Smart Window

C.1.6 Scenario 6: Alight on the Next Stop While Being Busy

Text Scenario 6: Maria Ziegler ist Studentin und wohnt in Karlsruhe. Sie muss zusammen mit einer Kommilitonin eine Hausarbeit schreiben und fährt daher heute zu ihrer Kommilitonin nach Neureut, um gemeinsam am Thema

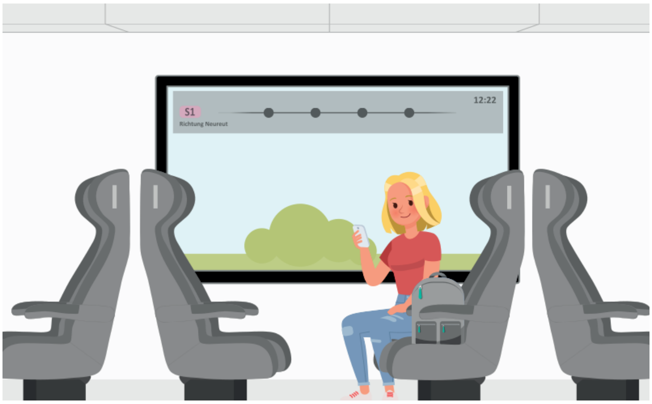


Figure C.6 — Picture illustrating scenario 6 in the online questionnaire. This graphic was created by Lars Erber based on my instructions.

zu arbeiten. Maria ist an der Haltestelle Werderstraße in die S1 eingestiegen und hat sich neben ein Smart Window gesetzt. Da sie zu ihrer Zielhaltestelle Bärenweg fast eine halbe Stunde unterwegs sein wird, liest sie sich auf ihrem Smartphone die Aufgabenstellung für die Hausarbeit und einige Grundlagen dazu durch.

Text Scenario 6 in English: Maria Ziegler is a student living in Karlsruhe. She has to write a term paper together with a fellow student.She is visiting her fellow student in Neuret today to work with her on the paper. She boarded the S1 on the stop Werderstraße and sat next to a Smart Window. Since she will have nearly half an hour until the vehicle reaches her final destination, she is the assignment for the term paper and some prerequisites on her smartphone.

Message Scenario 6: Maria erreicht nun folgende Nachricht: "Nächste Haltestelle: Bärenweg."

Message Scenario 6 in English: Maria now receives the following message: ”Next stop: Bärenweg.“

Available Output Options in Scenario 6:

- Vibration des Smartphones und Text-Ausgabe auf dem Smartphone

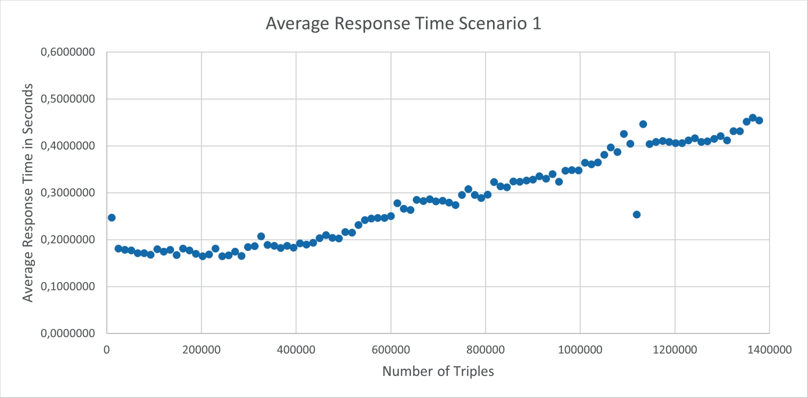


Figure C.7 — Results of the performance evaluation using scenario 1: average response time of ten test runs of the usability assessment service per number of triples in the context triple store.

- Text-Ausgabe auf dem Smartphone
- Audio-Ausgabe auf dem Smartphone
- Text-Ausgabe auf der Anzeige über dem Gang
- Text-Ausgabe auf dem Smart Window

C.2 Performance Evaluation Results

This section presents the results of the performance evaluation in figures C.7 to C.12.

C.3 Results Online Survey

The next section presents the survey results. First, the demographic data is presented, following the results of the participant's choice of output options for each scenario. After these, the results of the participant's assessment of given output options for each scenario are shown, in a comparison of baseline and framework choice.

Results for Scenario 1: For scenario 1, the results for statement 1, 3, 4, 5 and 8 are statistically relevant, both determined by χ^2 test and t-test with an α level $\leq$

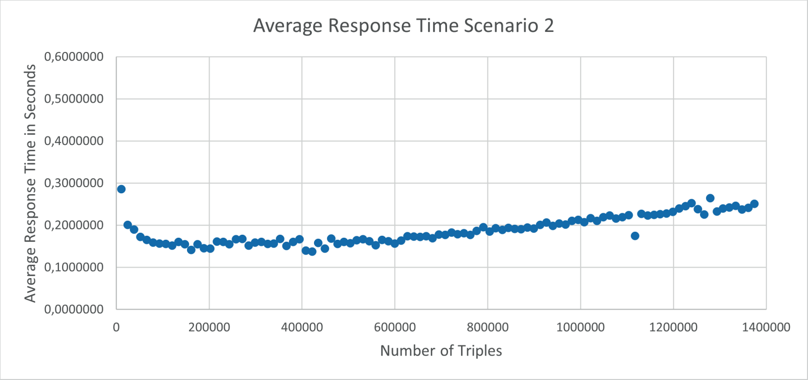


Figure C.8 — Results of the performance evaluation using scenario 2: average response time of ten test runs of the usability assessment service per number of triples in the context triple store.

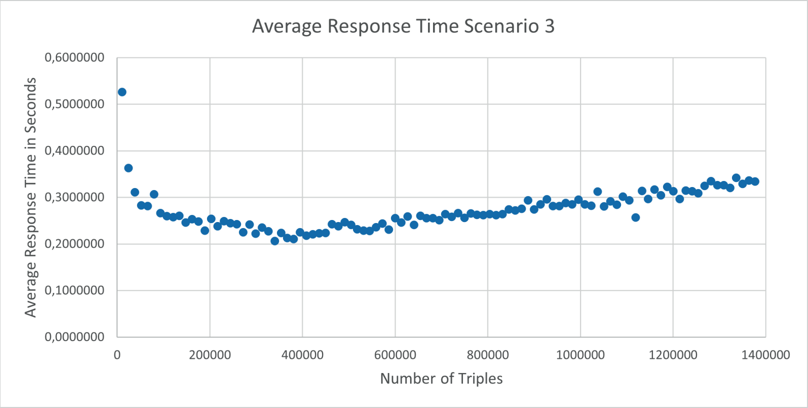


Figure C.9 — Results of the performance evaluation using scenario 3: average response time of ten test runs of the usability assessment service per number of triples in the context triple store.

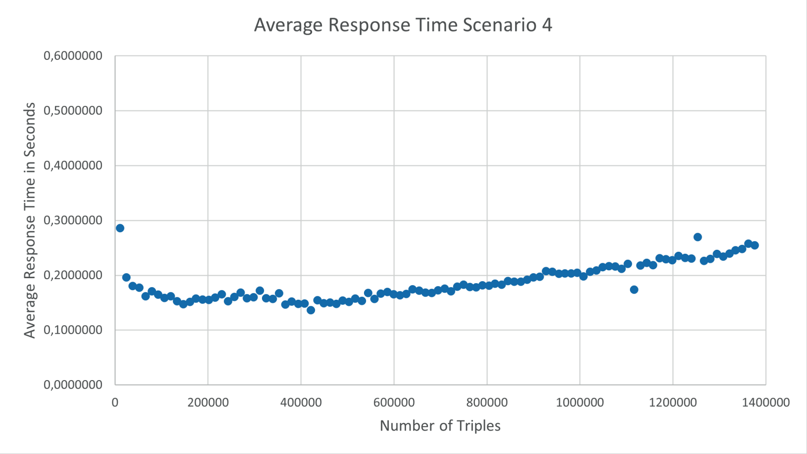


Figure C.10 — Results of the performance evaluation using scenario 4: average response time of ten test runs of the usability assessment service per number of triples in the context triple store.

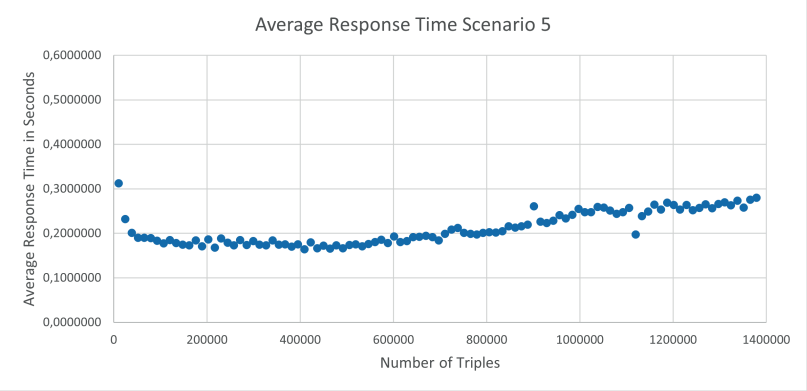


Figure C.11 — Results of the performance evaluation using scenario 5: average response time of ten test runs of the usability assessment service per number of triples in the context triple store.

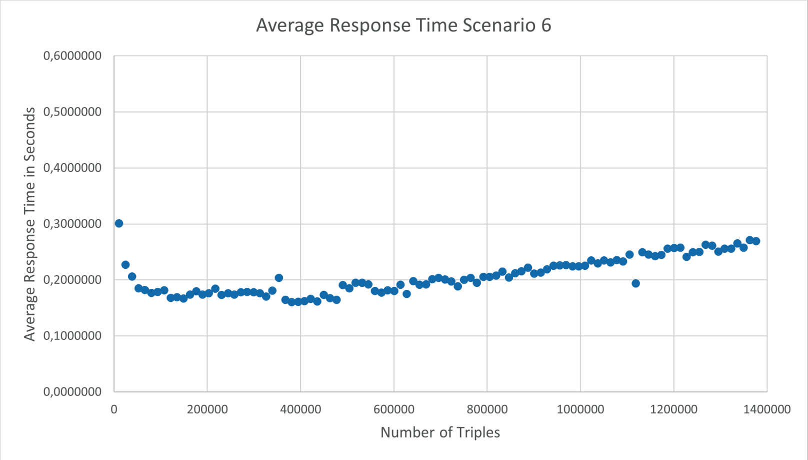


Figure C.12 — Results of the performance evaluation using scenario 6: average response time of ten test runs of the usability assessment service per number of triples in the context triple store.

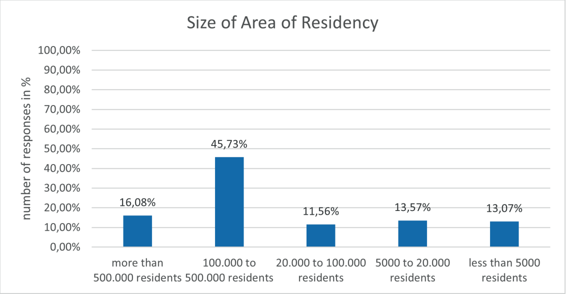


Figure C.13 — The answers of participants to questions about the size of their area of residency.

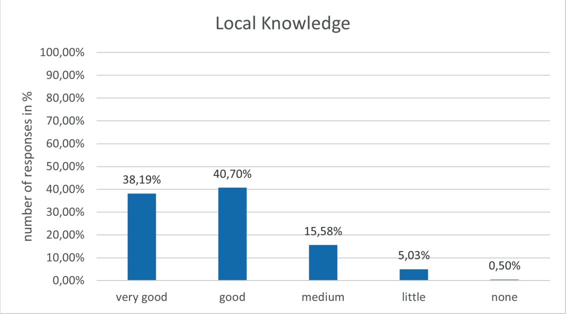


Figure C.14 — The answers of participants to questions about how well they know their way around the area they use public transport.

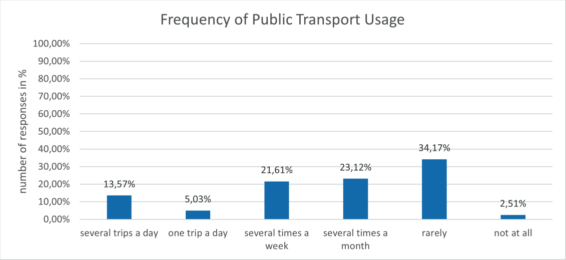


Figure C.15 — Frequency of public transport usage.

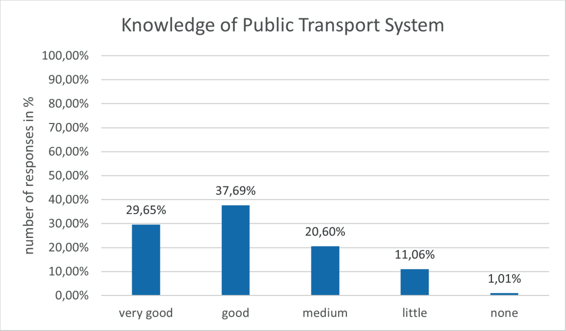


Figure C.16 — The answers of participants regarding their knowledge of public transport systems.

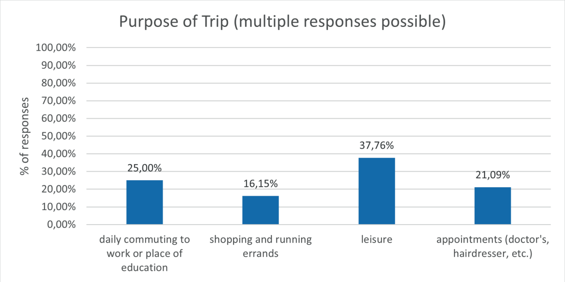


Figure C.17 — The answers of participants about the purposes of their public transport trips. Multiple answers were allowed.

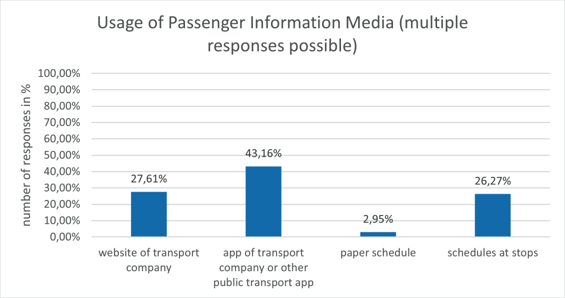


Figure C.18 — The answers of participants about the media they use for passenger information. Multiple answers were allowed.

0.05. Statement 6 was statistically significant using the χ^2 test with an α level of 0.1 and statistically significant using a t-test with a p value of 0.00726.

Results for Scenario 2: For scenario 2, the results for statements 1, 4, 5, 6 and 8 were statistically significant, determined both with a χ^2 test and t-test with an α level ≤ 0.05. The results for statement 2 were statistically significant with the χ^2 test at an α level of 0.5 and statsistically significant using the t-test with a p-value of 0.014. Similarly, the results for statement 7 was statistically significant with the χ^2 test at an α level of 0.1 and statistically significant with a t-test resulting in a p-value of 0.011.

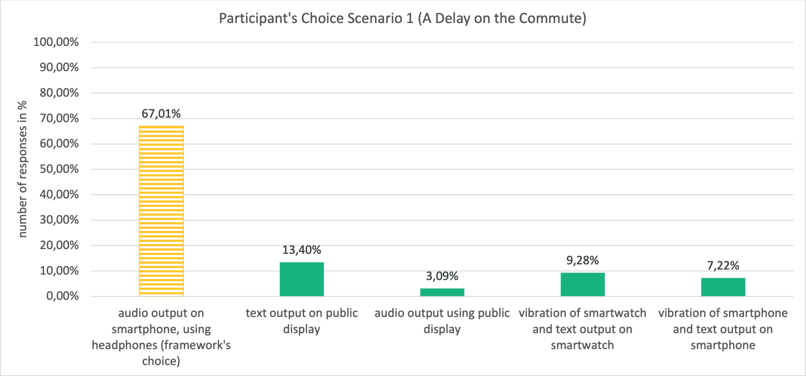


Figure C.19 — The number of choices of output options for scenario 1. In yellow and striped are the amount of participants that chose the same option as the framework. In this case, the baseline output option was not chosen by any participant. 97 participants saw scenario 1 and answered this question.

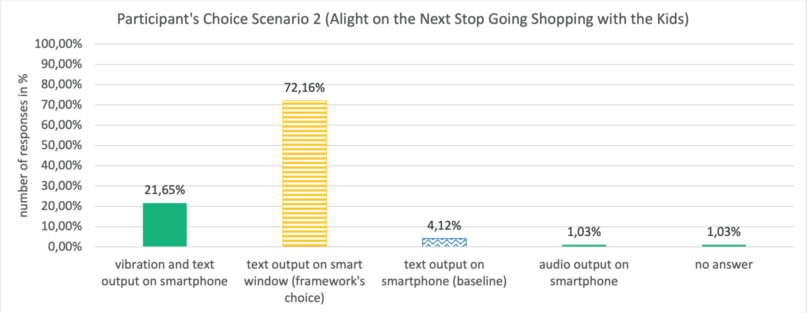


Figure C.20 — The number of choices of output options for scenario 2. In yellow and striped are the amount of participants that chose the same option as the framework. The blue bar with zigzag lines shows the number of participants choosing the baseline option, which is text output on the smartphone in all scenarios. 97 participants saw scenario 2 and answered this question.

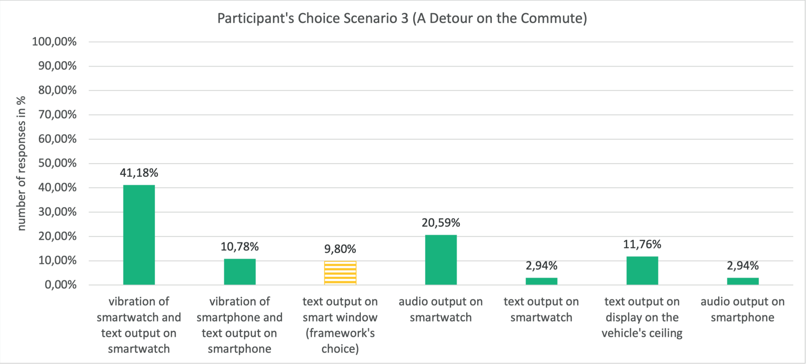


Figure C.21 — The number of choices of output options for scenario 3. In yellow and striped are the amount of participants that chose the same option as the framework. The blue bar with zigzag lines shows the number of participants choosing the baseline option, which is text output on the smartphone in all scenarios. 102 participants saw scenario 3 and answered this question.

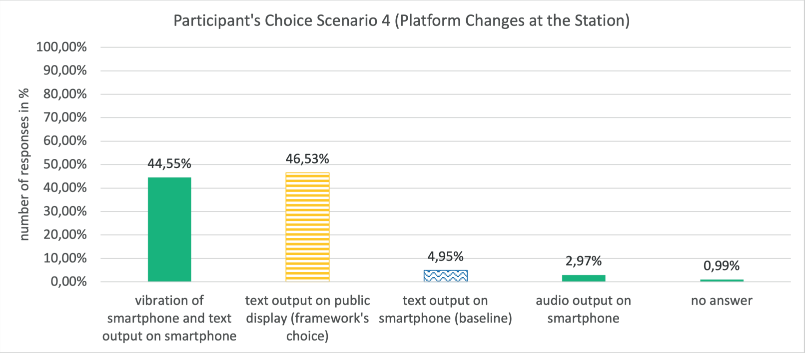


Figure C.22 — The number of choices of output options for scenario 4. In yellow and striped are the amount of participants that chose the same option as the framework. The blue bar with zigzag lines shows the number of participants choosing the baseline option, which is text output on the smartphone in all scenarios. 101 participants saw scenario 4 and answered this question.

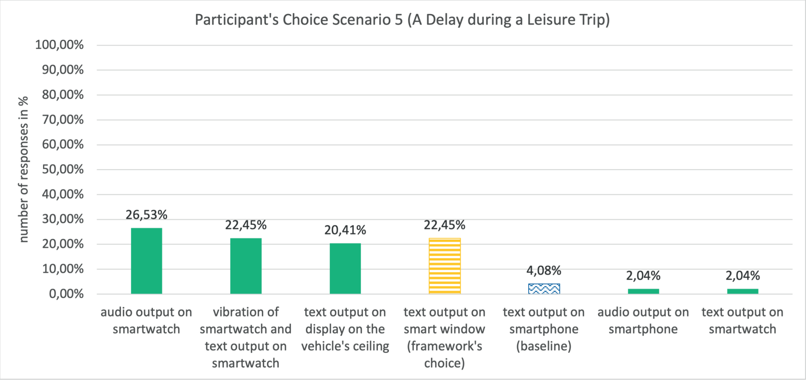


Figure C.23 — The number of choices of output options for scenario 5. In yellow and striped are the amount of participants that chose the same option as the framework. The blue bar with zigzag lines shows the number of participants choosing the baseline option, which is text output on the smartphone in all scenarios. 98 participants saw scenario 4 and answered this question.

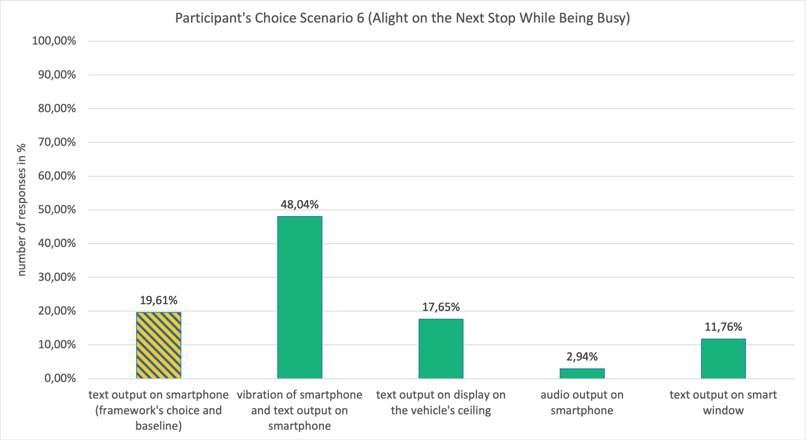


Figure C.24 — The number of choices of output options for scenario 6. In this scenario, the baseline and framework option are the same. The bar for this option is therefore diagonally striped in blue and yellow. 102 participants saw scenario 6 and answered this question.

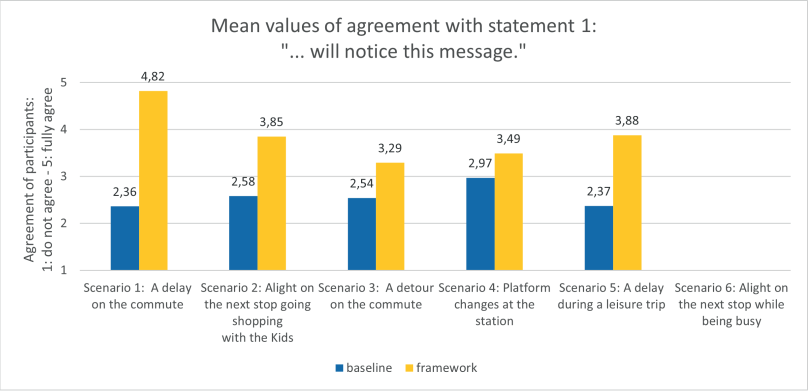


Figure C.25 — Mean values of agreement with statement 1 for scenarios 1-6. Results for scenario 6 are not statistically significant and are therefore not shown.

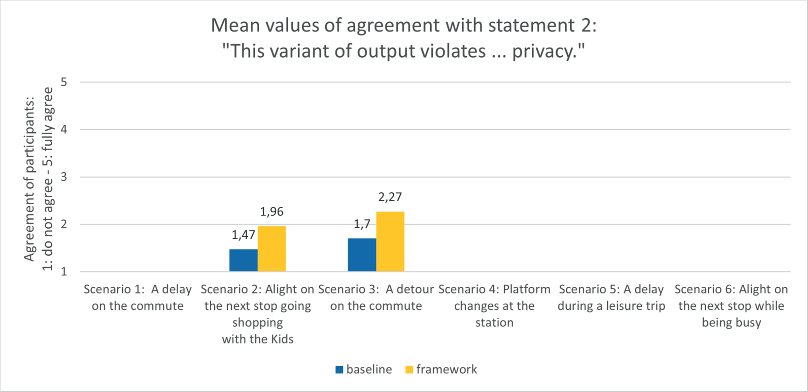


Figure C.26 — Mean values of agreement with statement 2 for scenarios 1-6. Results for scenarios 1, 4, 5 and 6 are not statistically significant and are therefore not shown.

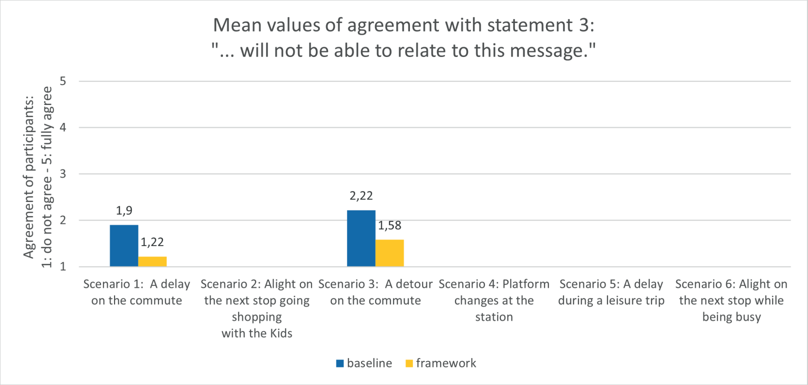


Figure C.27 — Mean values of agreement with statement 3 for scenarios 1-6. Results for scenarios 2, 4, 5 and 6 are not statistically significant and are therefore not shown.

Results for Scenario 3: For scenario 3, the results for statements 1, 3, 6 and 7 are statistically significant both with a χ^2 test and t-test with an α level ≤ 0.05. The results for statement 2 are statistically significant with the χ^2 test at an α level of 0.1 and using the t-test with a p-value of 0.003.

Results for Scenario 4: For scenario 4, the results for statements 1, 5, 6 and 7 are statsistically significant both with a χ^2 test and t-test with an α level ≤ 0.05. Statement 2 is statsistically significant based on the χ^2 test, but not significant using the t-test. Statement 2 will not be considered further.

Results for Scenario 5: For scenario 5, the results of statements 1, 4, 5 and 8 are statistically significant both with a χ^2 test and t-test with an α level ≤ 0.05. Statement 6 is statistically significant using the χ^2 test with an α level of 0.1 and statsistically significant using the t-test with a p value of 0.025. Statement 7 is statistically significant using the χ^2 test, but not significant with the t-test and will not be considered further.

Results for Scenario 6: The results for scenario 6 are not statistically significant. Since the baseline and framework option are the same option for this scenario, this is to be expected. However, the options were rated positively for all statements, allowing the conclusion that the selection of the framework was not a bad choice for this situation.

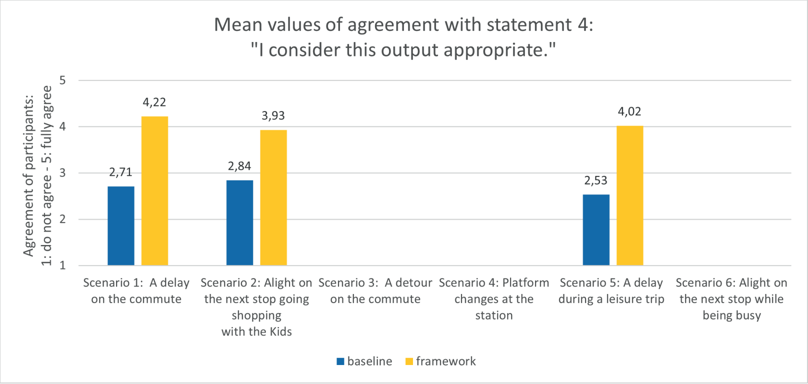


Figure C.28 — Mean values of agreement with statement 4 for scenarios 1-6. Results for scenarios 3, 4 and 6 are not statistically significant and are therefore not shown.

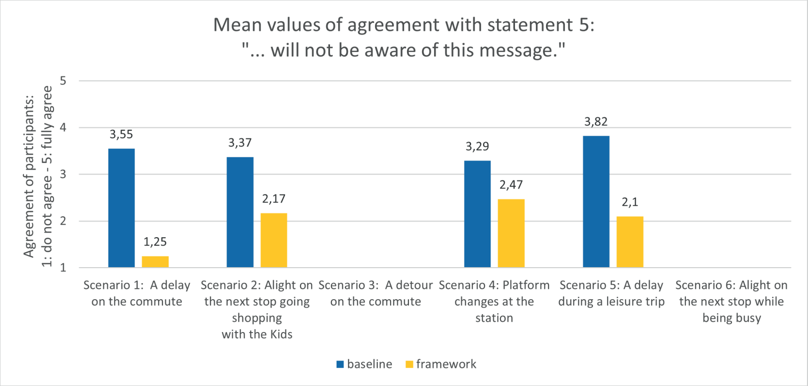


Figure C.29 — Mean values of agreement with statement 5 for scenarios 1-6. Results for scenario 3, 4 and 6 are not statistically significant and are therefore not shown.

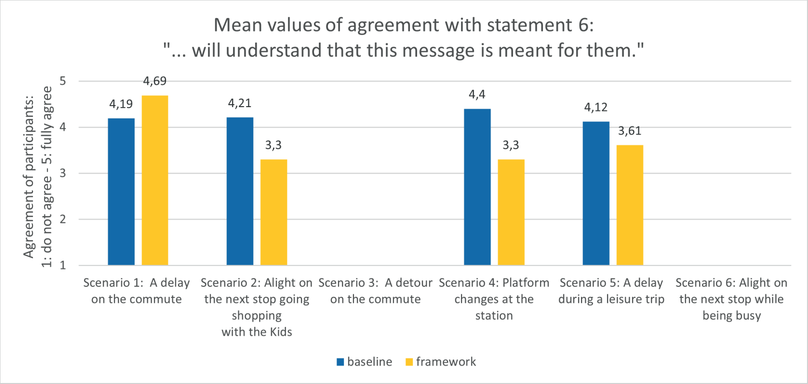


Figure C.30 — Mean values of agreement with statement 6 for scenarios 1-6. Results for scenarios 3 and 6 are not statistically significant and are therefore not shown.

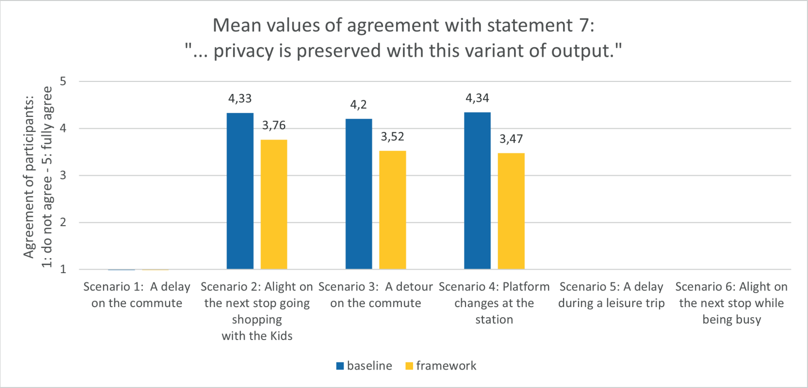


Figure C.31 — Mean values of agreement with statement 7 for scenarios 1-6. Results for scenario 1, 5 and 6 are not statistically significant and are therefore not shown.

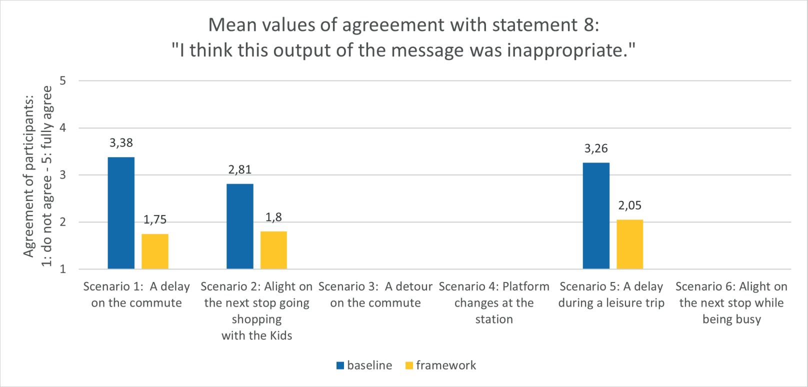


Figure C.32 — Mean values of agreement with statement 8 for scenarios 1-6. Results for scenario 3, 4 and 6 are not statistically significant and are therefore not shown.

Bibliography

Integriertes Bordinformationssystem (IBIS) - VDV 300. Technical report, June 1984. 56

Xml schema requirements. Technical report, February 1999, Last checked: June 2020. 29

Cen tc278. reference data model for public transport, en12896. Standard, European Committee for Standardization (CEN), 2006. 125

Namespaces in xml 1.0 (third edition). Technical report, December 2009, Last checked: June 2020. 28

Standard Interface for Real-time Information, Part 4. CEN TS 15531. Standard, European Committee for Standardization (CEN), 2011. 125

Identification of fixed objects in public transport en 28701. Standard, European Committee for Standardization (CEN), 2012. 125

Owl 2 web ontology language primer (second edition). Technical report, December 2012, Last checked: June 2020. 31

Owl 2 web ontology language xml serialization. Technical report, December 2012, Last checked: June 2020. 31

Owl 2 web ontology language. Technical report, December 2012, Last checked: June 2020. 26, 31

Sparql 1.1 query language. Technical report, March 2013, Last checked: June 2020. 29

Sparql query results xml format (second edition). Technical report, March 2013, Last checked: June 2020. 29

Sparql 1.1 query results json format. Technical report, March 2013. 29

RDF Vocabulary Description Language 1.0: RDF Schema. Technical report, Feb 2014, Last checked: June 2020. 29

Network Timetable Exchange Part 1 - CEN/TS 16614-1. Standard, European Committee for Standardization (CEN), 2014a. 125

Network Timetable Exchange Part 2 - CEN/TS 16614-2. Standard, European Committee for Standardization (CEN), 2014b. 125

Rdf 1.1 concepts and abstract syntax. Technical report, February 2014, Last checked: June 2020. 28

Rdf 1.1 xml syntax. Technical report, Februar 2014, Last checked: June 2020. 28

Rdf 1.1. primer. Technical report, 2014, Last checked: June 2020. 26, 27

Mobile Kundeninformation im ÖV. Standard, Cologne, Germany, January 2014. 125, 132

Standard Interface for Real-time Information, Part 1. CEN TS 15531. Standard, European Committee for Standardization (CEN), 2015a. 125

Standard Interface for Real-time Information, Part 2. CEN TS 15531. Standard, European Committee for Standardization (CEN), 2015b. 125

Standard Interface for Real-time Information, Part 3. CEN TS 15531. Standard, European Committee for Standardization (CEN), 2015c. 125

Network Timetable Exchange Part 3 - CEN/TS 16614-3. Standard, European Committee for Standardization (CEN), 2015d. 125

Standard Interface for Real-time Information, Part 5. CEN TS 15531. Standard, European Committee for Standardization (CEN), 2016. 125

Das Verkehrsmittel des 21. Jahrhunderts, January 2019. 55

Json-ld 1.1. Technical report, May 2020, Last checked: June 2020. 28

G. D. Abowd and E. D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1): 29–58, March 2000. 1, 14, 15, 19, 20, 22, 24

P. A. Akiki, A. K. Bandara, and Y. Yu. Using interpreted runtime models for devising adaptive user interfaces of enterprise applications. In *14th International Conference on Enterprise Information Systems (ICEIS 2012)*, 2012, Paper no.75. 91

P. A. Akiki, A. K. Bandara, and Y. Yu. Rbuis: Simplifying enterprise application user interfaces through engineering role-based adaptive behavior. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '13, page 3–12, New York, NY, USA, 2013. ACM. 91

P. A. Akiki, A. K. Bandara, and Y. Yu. Adaptive model-driven user interface development systems. *ACM Comput. Surv.*, 47(1), May 2014. 84

P. A. Akiki, A. K. Bandara, and Y. Yu. Engineering Adaptive Model-Driven User Interfaces. *IEEE Transactions on Software Engineering*, 42(12):1118–1147, 2016. 77, 86, 91, 95, 100

M. R. Alam, M. B. I. Reaz, and M. A. M. Ali. A Review of Smart Homes—Past, Present, and Future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1190–1203, Nov 2012. 234

U. Alegre, J. C. Augusto, and T. Clark. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software*, 117:55–83, 2016. 33, 39, 44, 45, 49, 222

C. Anderson, I. Hübener, A.-K. Seipp, S. Ohly, K. David, and V. Pejovic. A Survey of Attention Management Systems in Ubiquitous Computing Environments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(2):58:1–58:27, July 2018. 19, 123, 163, 165, 234

J. Annett. *Hierarchical Task Analysis*, page 17–35. Erlbaum Mahwah, 2003. 117

J. Annett and K. D. Duncan. Task analysis and training design. *Journal of Occupational Psychology*, 41:211–221, 1967. 117

N. Aquino, J. Vanderdonckt, and O. Pastor. Transformation Templates: Adding Flexibility to Model-driven Engineering of User Interfaces. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, page 1195–1202, New York, NY, USA, 2010. ACM. 3

C. Ardito, P. Buono, M. F. Costabile, and G. Desolda. Interaction with large displays: A survey. *ACM Comput. Surv.*, 47(3):46:1–46:38, Feb. 2015. 16

A. Attou and K. Moessner. Context-aware service adaptation management. In *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, page 1–5, Sept 2007. 49

J. Augusto. Increasing reliability in the development of intelligent environments. In *Proceedings of the 5th International Conference on Intelligent Environments*, volume 2, page 134. IOS Press, 2009. 13

J. C. Augusto and A. Muñoz. User Preferences in Intelligent Environments. *Applied Artificial Intelligence*, 33(12):1069–1091, 2019. 13

J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh. Intelligent environments: a manifesto. *Human-centric Computing and Information Sciences*, 3(1):12, Jun 2013. 1, 13

S. Azenkot, R. E. Ladner, and J. O. Wobbrock. Smartphone haptic feedback for nonvisual wayfinding. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '11, page 281–282, New York, NY, USA, 2011. Association for Computing Machinery. 15

Y. Bachvarova, B. van Dijk, and A. Nijholt. Towards a Unified Knowledge-Based Approach to Modality Choice. In Van der Sluis, Theune, Reiter, Krahmer, editor, *Proceedings Workshop on Multimodal Output Generation (MOG 2007)*, pages 5–15, 2007. 94, 98, 135, 136

L. Balme, A. Demeure, N. Barralon, J. Coutaz, and G. Calvary. CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces. In P. Markopoulos, B. Eggen, E. Aarts, and J. L. Crowley, editors, *Ambient Intelligence*, page 291–302, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 92, 93

V. Barbosa and M. T. Andrade. Multicao: A semantic approach to context-aware adaptation decision taking. In *2009 10th Workshop on Image Analysis for Multimedia Interactive Services*, page 133–136, May 2009. 47

L. Barkhuus and A. Dey. Is context-aware computing taking control away from the user? three levels of interactivity examined. In A. K. Dey, A. Schmidt, and J. F. McCarthy, editors, *UbiComp 2003: Ubiquitous Computing*, page 149–156, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 20

A. Barros. 150 fahrzeuge für karlsruher schienen: Vbk entwickeln eigene bahnen, auslieferung für fächerstadt ab 2028. https://www.ka-news.de/wirtschaft/karlsruhe-innovativ/150-Fahrzeuge-fuer-Karlsruher-Schienen-VBK-entwickeln-eigene-Bahnen-Auslieferung-fuer-Faecherstadt-ab-2028;art516981,2351790, March 2019, Last checked: September 2020. 56

C. Becker and D. Nicklas. Where do spatial context-models end and where do ontologies start? A proposal of a combined approach. In *Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management, in conjunction with UbiComp*, 2004. 38

D. Beckett, T. Berners-Lee, E. Prud'hommeaux, and G. Carothers. Rdf 1.1 turtle. Technical report, February 2014, Last checked: June 2020. 28

U. Beez, L. Kaupp, T. Deuschel, B. G. Humm, F. Schumann, J. Bock, and J. Hülsmann. *Context-Aware Documentation in the Smart Factory*, pages 163–180. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018. 234

P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini. A survey of context data distribution for mobile ubiquitous systems. *ACM Comput. Surv.*, 44(4), Sept. 2012. 42

V. Bellotti and K. Edwards. Intelligibility and accountability: Human considerations in context-aware systems. *Human–Computer Interaction*, 16(2-4):193–212, 2001. 20

Ben Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos. *Designing the User Interface - Strategies for Effective Human-Computer Interaction*. Pearson, global edition edition, 2018. 25

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001. 26

T. Berners-Lee, R. Fielding, and L. Masinter. Uniform resource identifier (uri): Generic syntax. Technical report, January 2005, Last checked: June 2020. 28

C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010, <ce:title>Context Modelling, Reasoning and Management</ce:title>. 37, 41, 47

S. Beul-Leusmann, C. Samsel, M. Wiederhold, K.-H. Krempels, E.-M. Jakobs, and M. Ziefle. Usability Evaluation of Mobile Passenger Information Systems. In A. Marcus, editor, *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*, page 217–228, Cham, 2014. Springer International Publishing. 208

M. C. Beutel, S. Gökay, W. Kluth, K.-H. Krempels, C. Samsel, C. Terwelp, and M. Wiederhold. Heterogeneous travel information exchange. In B. Mandler, J. Marquez-Barja, M. E. Mitre Campista, D. Cagáňová, H. Chaouchi, S. Zeadally, M. Badra, S. Giordano, M. Fazio, A. Somov, and R.-L. Vieriu, editors, *Internet of Things. IoT Infrastructures*, page 181–187, Cham, 2016. Springer International Publishing. 106

J. Bidot, C. Goumopoulos, and I. Calemis. Using ai planning and late binding for managing service workflows in intelligent environments. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, page 156–163, march 2011. 13

A. Bikakis, T. Patkos, G. Antoniou, and D. Plexousakis. A survey of semantics-based approaches for context reasoning in ambient intelligence. In M. Mühlhäuser, A. Ferscha, and E. Aitenbichler, editors, *Constructing Ambient*

Intelligence, page 14–23, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 47

M. Blumendorf and S. Albayrak. Towards a Framework for the Development of Adaptive Multimodal User Interfaces for Ambient Assisted Living Environments. In C. Stephanidis, editor, *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, pages 150–159, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. 89, 94

M. Blumendorf, G. Lehmann, S. Feuerstack, and S. Albayrak. Executable Models for Human-Computer Interaction. In T. C. N. Graham and P. Palanque, editors, *Interactive Systems. Design, Specification, and Verification*, pages 238–251, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 89, 90, 94

M. Blumendorf, G. Lehmann, and S. Albayrak. Bridging Models and Systems at Runtime to Build Adaptive User Interfaces. In *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '10, page 9–18, New York, NY, USA, 2010a. Association for Computing Machinery. 95

M. Blumendorf, D. Roscher, and S. Albayrak. Dynamic User Interface Distribution for Flexible Multimodal Interaction. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ICMI-MLMI '10, New York, NY, USA, 2010b. Association for Computing Machinery. 2, 90, 95

C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca. A data-oriented survey of context models. *SIGMOD Rec.*, 36:19–26, December 2007. 38, 163

H. Boley, S. Tabet, and G. Wagner. Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In *Proceedings of the First International Conference on Semantic Web Working*, SWWS'01, page 381–401, Aachen, DEU, 2001. CEUR-WS.org. 32

S. Bouzit, G. Calvary, J. Coutaz, D. Chêne, E. Petit, and J. Vanderdonckt. The PDA-LPA design space for user interface adaptation. In *2017 11th International Conference on Research Challenges in Information Science (RCIS)*, page 353–364, May 2017. 71

M. Brambilla and P. Fraternali. *Interaction flow modeling language : model-driven UI engineering of web and mobile apps with IFML*. The MK/OMG Press Series. Amsterdam : Elsevier, 2015. 97

D. Brickley. Basic Geo (WGS84 lat/long) Vocabulary. Technical report, World Wide Web Consortium (W3C) Semantic Web Interest Group, 2003. 126

H. Brignull and Y. Rogers. Enticing people to interact with large public displays in public spaces. In M. Rauterberg, M. Menozzi, and J. Wesson, editors, *Human-Computer Interaction INTERACT '03: IFIP TC13 InternationalConference on Human-Computer Interaction, 1st-5th September 2003,*. IOS Press, 2003. 16, 115

R. Buehler. Determinants of transport mode choice: a comparison of germany and the usa. *Journal of Transport Geography*, 19(4):644–657, 2011. 1

G. Calvary, J. Coutaz, and D. Thevenin. A Unifying Reference Framework for the Development of Plastic User Interfaces. In M. R. Little and L. Nigay, editors, *Engineering for Human-Computer Interaction*, page 173–192, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 18, 92, 93

G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt. A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15(3):289–308, 2003. 18, 92, 93

G. Calvary, J. Coutaz, O. Dâassi, L. Balme, and A. Demeure. Towards a new generation of widgets for supporting software plasticity: The "comet". In R. Bastide, P. Palanque, and J. Roth, editors, *Engineering Human Computer Interaction and Interactive Systems*, page 306–324, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 94, 96, 113, 114, 116

G. Calvary, A. Serna, C. Kolski, and J. Coutaz. *Transport: a fertile ground for the plasticity of user interfaces*, chapitres d'ouvrages 11, page 343–368. ISTE Ltd and John Wiley & Sons, Inc., 2011. 95, 97

T. D. Camacho, M. Foth, and A. Rakotonirainy. Pervasive technology and public transport: Opportunities beyond telematics. *IEEE Pervasive Computing*, 12(1): 18–25, Jan 2013. 57, 58

Y. Cao, M. Theune, and A. Nijholt. Modality Effects on Cognitive Load and Performance in High-Load Information Presentation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 335–344, New York, NY, USA, 2009. Association for Computing Machinery. 164

J. M. Carroll. *Making Use: Scenario-Based Design of Human-Computer Interactions*. MIT Press, Cambridge, MA, USA, 2000. 60

R. M. Carvalho, R. M. C. de Andrade, K. M. de Oliveira, I. S. de Santos, and C. I. M. Bezerra. Quality characteristics and measures for human–computer interaction evaluation in ubiquitous systems. *Software Quality Journal*, 25: 743–795, 2017. 3, 148, 150, 151, 154, 155, 259, 260, 261

R. Casas, R. Blasco Marín, A. Robinet, A. R. Delgado, A. R. Yarza, J. McGinn, R. Picking, and V. Grout. User Modelling in Ambient Intelligence for Elderly and Disabled People. In K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, editors, *Computers Helping People with Special Needs*, pages 114–122, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 142

E. Castillejo, A. Almeida, and D. López-de-Ipiña. Ontology-Based Model for Supporting Dynamic and Adaptive User Interfaces. *International Journal of Human–Computer Interaction*, 30(10):771–786, 2014. 95, 99

N. Cha, A. Kim, C. Y. Park, S. Kang, M. Park, J.-G. Lee, S. Lee, and U. Lee. Hello There! Is Now a Good Time to Talk? Opportune Moments for Proactive Interactions with Smart Speakers. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(3), sep 2020. 181, 234

T. Chaari, F. Laforest, and A. Celentano. Adaptation in context-aware pervasive information systems: the secas project. *International Journal of Pervasive Computing and Communications*, 3(4):400–425, 2008. 48

G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report, Department of Computer Science, Dartmouth College, 2000. 33, 113

H. Chen. An intelligent broker architecture for context-aware systems. Technical report, University of Maryland Baltimore County, 2003. 42

H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments. *Knowl. Eng. Rev.*, 18(3):197–207, 2003a. 26

H. Chen, T. Finin, and A. Joshi. Using owl in a pervasive computing broker. In *Proceedings of Workshop on Ontologies in Open Agent Systems (AAMAS 2003)*, July 2003b. 42

H. Chen, T. Finin, and A. Joshi. Semantic web in the context broker architecture. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, PERCOM '04, page 277, USA, 2004a. IEEE Computer Society. 49

H. Chen, F. Perich, D. Chakraborty, T. Finin, and A. Joshi. Intelligent agents meet semantic web in a smart meeting room. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, page 854–861, Washington, DC, USA, 2004b. IEEE Computer Society. 49

H. Chen, F. Perich, T. Finin, and A. Joshi. Soupa: standard ontology for ubiquitous and pervasive applications. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.*, page 258–267, 2004. 38, 116

K. Cheverst, N. Davies, K. Mitchell, and A. Friday. Experiences of developing and deploying a context-aware tourist guide: the guide project. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, page 20–31, New York, NY, USA, 2000. ACM. 33, 37, 48, 113, 116

J. Chmielewski, J. Flotyński, D. Rumiński, and A. Wójtowicz. Declarative GUI descriptions for device-independent applications. *Personal and Ubiquitous Computing*, 20(2):185–194, Apr 2016. 74

C. Choi, I. Park, S. J. Hyun, D. Lee, and D. H. Sim. Mire: A minimal rule engine for context-aware mobile devices. In *2008 Third International Conference on Digital Information Management*, page 172–177, 2008. 41

V. T. F. Chow, K. W. Sung, H. M. Meng, K. H. Wong, G. K. Leung, Y. Kuo, and K. K. F. Tsoi. Utilizing Real-Time Travel Information, Mobile Applications and Wearable Devices for Smart Public Transportation. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, page 138–144, Nov 2016. 16, 54, 106

T. Clerckx, K. Luyten, and K. Coninx. DynaMo-AID: A Design Process and a Runtime Architecture for Dynamic Model-Based User Interface Development. In R. Bastide, P. Palanque, and J. Roth, editors, *Engineering Human Computer Interaction and Interactive Systems*, volume 3425 of *Lecture Notes in Computer Science*, page 871–876. Springer Berlin / Heidelberg, 2005a, 10.1007/11431879_5. 87, 88, 93

T. Clerckx, F. Winters, and K. Coninx. Tool Support for Designing Context-Sensitive User Interfaces Using a Model-Based Approach. In *Proceedings of the 4th International Workshop on Task Models and Diagrams*, TAMODIA '05, page 11–18, New York, NY, USA, 2005b. Association for Computing Machinery. 17

T. Clerckx, C. Vandervelpen, and K. Coninx. *Engineering Interactive Systems: EIS 2007 Joint Working Conferences, EHCI 2007, DSV-IS 2007, HCSE 2007, Salamanca, Spain, March 22-24, 2007. Selected Papers*, chapter Task-Based Design and Runtime Support for Multimodal User Interface Distribution, pages 89–105. Springer-Verlag, Berlin, Heidelberg, 2008. 88, 94, 135

G. Cockton. From Quality in Use to Value in the World. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, page 1287–1290, New York, NY, USA, 2004. Association for Computing Machinery. 97

M. H. Coen. Design principles for intelligent environments. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, page 547–554, USA, 1998. American Association for Artificial Intelligence. 13

S. Consolvo, L. Arnstein, and B. R. Franza. User study techniques in the design and evaluation of a ubicomp environment. In G. Borriello and L. E. Holmquist, editors, *UbiComp 2002: Ubiquitous Computing*, page 73–90, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 22, 23

S. Consolvo, B. Harrison, I. Smith, M. Y. Chen, K. Everitt, J. Froehlich, and J. A. Landay. Conducting in situ evaluations for and with ubiquitous computing technologies. *International Journal of Human-Computer Interaction*, 22(1 & 2): 103 – 118, 2007. 23

D. J. Cook, J. C. Augusto, and V. R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, 2009. 12, 13

F. Corno. User expectations in intelligent environments. *Journal of Reliable Intelligent Environments*, 4(4):189–198, July 2018. 13

J. Coutaz, L. Nigay, D. Salber, A. Blandford, J. May, and R. M. Young. *Human-Computer Interaction: Interact '95*, chapter Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The Care Properties, pages 115–120. Springer US, Boston, MA, 1995. 88, 233

L. Daniele, P. Dockhorn Costa, and L. Ferreira Pires. Towards a rule-based approach for context-aware applications. In A. Pras and M. van Sinderen, editors, *Dependable and Adaptable Networks and Services*, page 33–43, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 41, 47

R. Darbar and D. Samanta. SurfaceSense: Smartphone Can Recognize Where It Is Kept. In *Proceedings of the 7th International Conference on HCI, IndiaHCI*

2015, IndiaHCI'15, page 39–46, New York, NY, USA, 2015. Association for Computing Machinery. 173

D. M. de Amorim, T. G. Dias, and M. C. Ferreira. Usability evaluation of a public transport mobile ticketing solution. In T. Ahram, W. Karwowski, and R. Taiar, editors, *Human Systems Engineering and Design*, page 345–351, Cham, 2019. Springer International Publishing. 23, 24

R. De Virgilio and R. Torlone. A general methodology for context-aware data access. In *Proceedings of the 4th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, MobiDE '05, page 9–15, New York, NY, USA, 2005. Association for Computing Machinery. 36

R. De Virgilio, R. Torlone, and G. Houben. A rule-based approach to content delivery adaptation in web information systems. In *7th International Conference on Mobile Data Management (MDM'06)*, page 21–21, 2006. 36

A. Demeure, G. Calvary, J. Coutaz, and J. Vanderdonckt. The Comets Inspector: Towards Run Time Plasticity Control Based on a Semantic Network. In K. Coninx, K. Luyten, and K. A. Schneider, editors, *Task Models and Diagrams for Users Interface Design*, page 324–338, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 94, 96

A. Demeure, G. Calvary, and K. Coninx. COMET(s), A Software Architecture Style and an Interactors Toolkit for Plastic User Interfaces. In T. C. N. Graham and P. Palanque, editors, *Interactive Systems. Design, Specification, and Verification*, page 225–237, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 94, 96

A. K. Dey and G. D. Abowd. Towards a better understanding of context and context-awareness. In *Computer Human Interaction 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness*, 2000. 12, 34, 43, 44, 45, 113, 114

A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human–Computer Interaction*, 16(2-4):97–166, 2001. 33, 34, 38, 39, 40, 42

A. Dhouib, A. Trabelsi, C. Kolski, and M. Neji. A classification and comparison of usability evaluation methods for interactive adaptive systems. In *2016 9th International Conference on Human System Interactions (HSI)*, page 246–251, July 2016. 23

L. Di Geronimo, A. Canonica, M. Husmann, and M. C. Norrie. Continuous tilting interaction techniques on mobile devices for controlling public displays. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '17, page 21–26, New York, NY, USA, 2017. ACM. 16

D. Di Mauro and F. Cutugno. A framework for interaction design in intelligent environments. In *Intelligent Environments (IE), 2016 12th International Conference on*, 2016. 83, 84

D. Di Mauro, J. C. Augusto, A. Origlia, and F. Cutugno. A Framework for Distributed Interaction in Intelligent Environments. In A. Braun, R. Wichert, and A. Maña, editors, *Ambient Intelligence*, page 136–151, Cham, 2017. Springer International Publishing. 83, 84

D. Diaper. *Understanding task analysis for human-computer interaction*, page 5–47. Lawrence Erlbaum Associates, Inc, 2004. 117

P. Dockhorn Costa, L. Ferreira Pires, and M. van Sinderen. Architectural Patterns for Context-Aware Services Platforms. 2nd International Workshop on Ubiquitous Computing, IWUC 2005, pages 3–18. INSTICC Press, Apr. 2005. 33

S. Dow, B. Macintyre, J. Lee, C. Oezbek, J. D. Bolter, and M. Gandy. Wizard of Oz Support throughout an Iterative Design Process. *Pervasive Computing, IEEE*, 4(4):18–26, 2005. 22

C. Duarte and L. Carriço. A Conceptual Framework for Developing Adaptive Multimodal Applications. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI '06, page 132–139, New York, NY, USA, 2006. ACM. 89, 94

B. DuCharme. *Learning SPARQL, Second Edition*. O'Reilly Media Inc., Sebastopol, CA, 2013. 29

M. Duerst and M. Suignard. Internationalized resource identifiers (iris). Technical report, January 2005, Last checked: June 2020. 28

N. Eagle and A. S. Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10:255–268, March 2006. 115, 179

J. Eisenstein and A. Puerta. Adaptation in Automated User-Interface Design. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, IUI '00, page 74–81, New York, NY, USA, 2000. Association for Computing Machinery. 17, 73, 86, 87, 93

J. Eisenstein, J. Vanderdonckt, and A. Puerta. Adapting to mobile contexts with user-interface modeling. In *Proceedings Third IEEE Workshop on Mobile Computing Systems and Applications*, pages 83–92, 2000. 86

J. Eisenstein, J. Vanderdonckt, and A. Puerta. Applying Model-Based Techniques to the Development of UIs for Mobile Computers. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, IUI '01, page 69–76, New York, NY, USA, 2001. Association for Computing Machinery. 86, 93

F. Englert, S. Großberndt, G. Gruber, P. von Grumbkow, M. Herlitze, S. Hörold, W. Isajkin, C. Keller, W. Kohl, R. Kühn, C. Mayas, W. Meier-Leu, B. Radermacher, A. Stelzer, K. Tietze, A. Wehrmann, and D. Weißer. Echtzeitkommunikations- und Auskunftsplattform EKAP Teil 2: EKAP-Schnittstellenbeschreibung V1.3. Standard 2, Verband Deutscher Verkehrsunternehmen, September 2019. 125, 132, 195, 226

R. Etter, P. Dockhorn Costa, and T. Broens. A rule-based approach towards context-aware user notification services. In *Proceedings of the 2006 ACS/IEEE International Conference on Pervasive Services*, PERSER '06, page 281–284, USA, 2006. IEEE Computer Society. 47

C. Evers, R. Kniewel, K. Geihs, and L. Schmidt. The user in the loop: Enabling user participation for self-adaptive applications. *Future Generation Computer Systems*, 34:110–123, 2014, Special Section: Distributed Solutions for Ubiquitous Computing and Ambient Intelligence. 83, 85, 99, 150, 151, 154, 259, 260, 261

P. Fahy and S. Clarke. Cass – a middleware for mobile context-aware applications. In *Workshop on Context Awareness, MobiSys*, 2004. 37

J. Favela, M. Rodríguez, A. Preciado, and V. M. González. Integrating context-aware public displays into a mobile hospital information system. *IEEE Transactions on Information Technology in Biomedicine*, 8(3):279–286, September 2004. 16

B. Ferris, K. Watkins, and A. Borning. Location-aware tools for improving public transit usability. *IEEE Pervasive Computing*, 9(1):13–19, 2010. 113

S. Feuerstack, M. Blumendorf, M. Kern, M. Kruppa, M. Quade, M. Runge, and S. Albayrak. Automated Usability Evaluation during Model-Based Interactive System Development. In P. Forbrig and F. Paternò, editors, *Engineering Interactive Systems*, pages 134–141, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 24

D. Fitton, K. Cheverst, C. Kray, A. J. Dix, M. Rouncefield, and G. Saslis-Lagoudakis. Rapid prototyping and user-centered design of interactive display-based systems. *IEEE Pervasive Computing*, 4(4):58–66, 2005. 14

J. Floch, C. Frà, R. Fricke, K. Geihs, M. Wagner, J. Lorenzo, E. Soladana, S. Mehlhase, N. Paspallis, H. Rahnama, P. Ruiz, and U. Scholz. Playing MUSIC — building context-aware and self-adaptive mobile applications. *Software: Practice and Experience*, 43(3):359–388, 2013. 85

J. D. Foley and S. N. Piyawadee. History, Results, and Bibliography of the User Interface Design Environment (UIDE), an Early Model-based System for User Interface Design and Implementation. In F. Paternó, editor, *Interactive Systems: Design, Specification, and Verification*, pages 3–14, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. 17

J. Forlizzi and K. Battarbee. Understanding experience in interactive systems. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '04, page 261–268, New York, NY, USA, 2004. Association for Computing Machinery. 18

R. Francese, I. Passero, and G. Tortora. Wiimote and kinect: Gestural user interfaces add a natural third dimension to hci. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, page 116–123, New York, NY, USA, 2012. ACM. 15

K. Gajos and D. S. Weld. SUPPLE: Automatically Generating User Interfaces. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, IUI '04, page 93–100, New York, NY, USA, 2004. Association for Computing Machinery. 88, 89, 93

K. Gajos, D. Christianson, R. Hoffmann, T. Shaked, K. Henning, J. J. Long, and D. S. Weld. Fast and Robust Interface Generation for Ubiquitous Applications. In M. Beigl, S. Intille, J. Rekimoto, and H. Tokuda, editors, *UbiComp 2005: Ubiquitous Computing*, pages 37–55, Berlin, Heidelberg, 2005a. Springer Berlin Heidelberg. 89, 93

K. Gajos, A. Wu, and D. S. Weld. Cross-Device Consistency in Automatically GeneratedUser Interfaces. In *Proceedings of Workshop onMulti-User and Ubiquitous User Interfaces*, 2005b. 89

K. Z. Gajos, J. O. Wobbrock, and D. S. Weld. Automatically Generating User Interfaces Adapted to Users' Motor and Vision Capabilities. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST

’07, page 231–240, New York, NY, USA, 2007. Association for Computing Machinery. 89

K. Z. Gajos, D. S. Weld, and J. O. Wobbrock. Automatically generating personalized user interfaces with Supple. *Artificial Intelligence*, 174(12):910–950, 2010. 17, 88, 89, 93

M. Gams, I. Y.-H. Gu, A. Härmä, A. Muñoz, and V. Tam. Artificial Intelligence and Ambient Intelligence. *Journal of Ambient Intelligence and Smart Environments*, 11(1):71–86, Jan. 2019. 12

D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Project aura: toward distraction-free pervasive computing. *Pervasive Computing, IEEE*, 1(2):22–31, Apr-Jun 2002. 82, 116

D. Garlan, V. Poladian, B. Schmerl, and J. a. P. Sousa. Task-based self-adaptation. In *Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems*, WOSS ’04, page 54–57. ACM, 2004. 114

K. Geihs and M. Wagner. Context-awareness for self-adaptive applications in ubiquitous computing environments. In P. C. Vinh, N. M. Hung, N. T. Tung, and J. Suzuki, editors, *Context-Aware Systems and Applications*, page 108–120, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 84

K. Geihs, P. Barone, F. Eliassen, J. Floch, R. Fricke, E. Gjorven, S. Hallsteinsen, G. Horn, M. U. Khan, A. Mamelli, G. A. Papadopoulos, N. Paspallis, R. Reichle, and E. Stav. A Comprehensive Solution for Application-level Adaptation. *Softw. Pract. Exper.*, 39(4):385–422, Mar. 2009a. 76, 85

K. Geihs, R. Reichle, M. W. 0001, and M. U. Khan. Modeling of context-aware self-adaptive applications in ubiquitous and service-oriented environments. In B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, editors, *Software Engineering for Self-Adaptive Systems*, volume 5525 of *Lecture Notes in Computer Science*, page 146–163. Springer, 2009b. 84, 85, 92

M. Gil and V. Pelechano. Self-adaptive unobtrusive interactions of mobile computing systems. *Journal of Ambient Intelligence and Smart Environments*, 9 (6):659–688, 2017. 26, 83, 85, 86, 100, 135, 136

S. A. Grandhi, G. Joue, and I. Mittelberg. Understanding naturalness and intuitiveness in gesture production: Insights for touchless gestural interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’11, page 821–824, New York, NY, USA, 2011. ACM. 15

P. Gray and D. Salber. Modelling and Using Sensed Context Information in the Design of Interactive Applications. In M. R. Little and L. Nigay, editors, *Engineering for Human-Computer Interaction*, page 317–335, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 37

M. Grossmann, M. Bauer, N. Honle, U.-P. Kappeler, D. Nicklas, and T. Schwarz. Efficiently Managing Context Information for Large-Scale Scenarios. In *Third IEEE International Conference on Pervasive Computing and Communications*, pages 331–340, 2005. 42

T. W. S. W. Group. Sparql 1.1 overview. Technical report, March 2013, Last checked: June 2020. 26

T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. 26

M. Grüninger and M. S. Fox. Methodology for the Design and Evaluation of Ontologies. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95*, 1995. 156

T. Gu, H. K. Pung, and D. Q. Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1): 1–18, 2005. 38

J. Guiry, P. van de Ven, and J. Nelson. Multi-sensor fusion for enhanced contextual awareness of everyday activities with ubiquitous devices. *Sensors*, 14(3):5687–5701, Mar 2014. 16

M. Halbrügge. *Automated Usability Evaluation (AUE)*, pages 23–33. Springer International Publishing, Cham, 2018. 24

S. Hallsteinsen, K. Geihs, N. Paspallis, F. Eliassen, G. Horn, J. Lorenzo, A. Mamelli, and G. Papadopoulos. A development framework and methodology for self-adapting applications in ubiquitous computing environments. *Journal of Systems and Software*, 85(12):2840–2859, 2012, Self-Adaptive Systems. 49, 83, 84, 99

T. Halpin. Morgan Kaufman, 2001. 37

M. Hammontree, P. Weiler, and N. Nayak. Remote Usability Testing. *Interactions*, 1(3):21–25, jul 1994. 22

R. Harrison, D. Flood, and D. Duce. Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science*, 1 (1), July 2013. 150, 151, 154, 164, 259, 260, 261

M. Hassenzahl and N. Tractinsky. User experience - a research agenda. *Behaviour & Information Technology*, 25(2):91–97, 2006. 18

S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: A programmable pervasive space. *IEEE Computer*, 38 (3):50–60, 2005. 82

K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(1): 37–64, 2006. 41

K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. In F. Mattern and M. Naghshineh, editors, *Pervasive Computing*, volume 2414 of *Lecture Notes in Computer Science*, page 79–117. Springer Berlin / Heidelberg, 2002, 10.1007/3-540-45866-2_14. 36, 116

K. Henricksen, J. Indulska, and A. Rakotonirainy. Generating context management infrastructure from high-level context models. In *In 4th International Conference on Mobile Data Management (MDM) - Industrial Track*, page 1–6, 2003. 37

P. Hitzler, M. Krötzsch, S. Rudolph, and Y. Sure. *Semantic Web Grundlagen*. Springer Verlag Berlin Heidelberg, 1 edition, 2008. 27, 29, 31

J. Ho and S. S. Intille. Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’05, page 909–918, New York, NY, USA, 2005. Association for Computing Machinery. 116

F. Hohl, U. Kubach, A. Leonhardi, K. Rothermel, and M. Schwehm. Next Century Challenges: Nexus - an Open Global Infrastructure for Spatial-Aware Applications. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom ’99, page 249–255, New York, NY, USA, 1999. Association for Computing Machinery. 33

J. I. Hong and J. A. Landay. An Infrastructure Approach to Context-Aware Computing. *Human–Computer Interaction*, 16(2-4):287–303, 2001. 33

T. Horak, S. K. Badam, N. Elmqvist, and R. Dachselt. When david meets goliath: Combining smartwatches with a large vertical display for visual data exploration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI ’18, page 19:1–19:13, New York, NY, USA, 2018. ACM. 16

I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, World Wide Web Consortium (W3C), 2004. 32, 170

L.-S. Huang, J.-Y. Su, and T.-L. Pao. A Context Aware Smart Classroom Architecture for Smart Campuses. *Applied Sciences*, 9(9), 2019. 234

J. Hunter. *Adding Multimedia to the SemanticWeb: Building and Applying an MPEG-7 Ontology*, chapter 3, pages 75–106. John Wiley & Sons, Ltd, 2005. 98

J. Hussain, A. Ul Hassan, H. S. Muhammad Bilal, R. Ali, M. Afzal, S. Hussain, J. Bang, O. Banos, and S. Lee. Model-based adaptive user interface based on context and user experience evaluation. *Journal on Multimodal User Interfaces*, 12(1):1–16, Mar 2018. 75, 77, 91, 95

S. Hörold, C. Mayas, and H. Krömker. Analyzing varying environmental contexts in public transport. In M. Kurosu, editor, *Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments*, page 85–94, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 57, 58

S. Hörold, C. Mayas, and H. Krömker. Guidelines for usability field tests in the dynamic contexts of public transport. In M. Kurosu, editor, *Human-Computer Interaction. Theories, Methods, and Tools*, page 489–499, Cham, 2014. Springer International Publishing. 23

S. Hörold, C. Mayas, and H. Krömker. Interactive displays in public transport – challenges and expectations. *Procedia Manufacturing*, 3:2808–2815, 2015, 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015. 16

J. Indulska and P. Sutton. Location management in pervasive systems. In *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003 - Volume 21*, ACSW Frontiers '03, page 143–151, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc. 39

infas Institut für angewandte Sozialwissenschaften GmbH. Mobilität in Deutschland - MiD 2017, Ergebnisbericht. resreport, Bundesministerium für Verkehr und digitale Infrastruktur, Dec. 2018. 214

International Organization for Standardization. ISO 9241-11:2018Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. Standard, Grenoble, CH, 2018. 17

International Organization for Standardization. ISO 9241-210:2019Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. Standard, Grenoble, CH, 2019. 18

H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, page 234–241, New York, NY, USA, 1997. ACM. 15

M. Y. Ivory and M. A. Hearst. The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Comput. Surv.*, 33(4):470–516, Dec. 2001. 24

G. Jacucci, A. Morrison, G. T. Richard, J. Kleimola, P. Peltonen, L. Parisi, and T. Laitinen. Worlds of Information: Designing for Engagement at a Public Multi-Touch Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 2267–2276, New York, NY, USA, 2010. Association for Computing Machinery. 16

A. Jameson. Modelling both the context and the user. *Personal and Ubiquitous Computing*, 5:29–33, 2001. 116

A. Jameson, A. Mahr, M. Kruppa, A. Rieger, and R. Schleicher. Looking for Unexpected Consequences of Interface Design Decisions: The MeMo Workbench. In M. Winckler, H. Johnson, and P. Palanque, editors, *Task Models and Diagrams for User Interface Design*, pages 279–286, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 24

S. Jones, S. Hara, and J. C. Augusto. efriend: an ethical framework for intelligent environments development. *Ethics and Information Technology*, 17(1):11–25, Mar 2015. 13

W. Ju and L. Leifer. The design of implicit interactions: Making interactive systems less obnoxious. *Design Issues*, 24(3):72–84, 2008. 14

B. H. Juang. Ubiquitous speech communication interface. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU '01.*, page 85–92, 2001. 15

D. K., B. M., S. F., L. J., and B. J. The european union report, scenarios for ambient intelligence in 2010. Technical report, European Commission, Information Society Directorate-General, 2001. 12, 13

C. Karagiannidis, A. Koumpis, and C. Stephanidis. Deciding ‘what’,‘when’, ‘why’, and ‘how’ to adapt in intelligent multimedia presentation systems. In *Workshop "Towards a Standard Reference Model for Intelligent Multimedia Presentation Systems", 12th European Conference on Artificial Intelligence*, Budapest, Hungary, Aug. 1996. 71, 72, 73, 74, 76

K. Katsuragawa, K. Pietroszek, J. R. Wallace, and E. Lank. Watchpoint: Freehand pointing with a smartwatch in a ubiquitous display environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI ’16, page 128–135, New York, NY, USA, 2016. Association for Computing Machinery. 14, 16

C. Keller. Ontologies and Rules for Context-Aware Assessment of the Usability of Output Devices and Modalities in Ubiquitous Mobility Systems, 2023. [Online]. Available: `https://doi.org/10.18419/darus-3385`. 112, 156, 170

C. Keller and T. Schlegel. Model Based and Service Oriented Interaction for Ubiquitous Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp ’16, page 429–434, New York, NY, USA, 2016. Association for Computing Machinery. 7, 8, 103, 112, 146

C. Keller and T. Schlegel. How to get in Touch with the Passenger: Context-Aware Choices of Output Modality in Smart Public Transport. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers*, UbiComp/ISWC ’19 Adjunct, New York, NY, USA, 2019. ACM. 6, 7, 8, 103, 112, 146, 229, 230

C. Keller, M. Korzetz, R. Kühn, and T. Schlegel. Nutzerorientierte Visualisierung von Fahrplaninformationen auf mobilen Geräten im öffentlichen Verkehr. In M. Eibl, editor, *Mensch & Computer 2011: überMEDIEN | ÜBERmorgen*, page 59–68, Chemnitz, Germany, 2011. Oldenbourg-Verlag. 7, 146

C. Keller, S. Brunk, and T. Schlegel. Introducing the Public Transport Domain to the Web of Data. In B. Benatallah, A. Bestavros, Y. Manolopoulos, A. Vakali, and Y. Zhang, editors, *Web Information Systems Engineering – WISE 2014*, pages 521–530, Cham, 2014a. Springer International Publishing. 6, 9, 112, 126, 229

C. Keller, R. Pöhland, S. Brunk, and T. Schlegel. An Adaptive Semantic Mobile Application for Individual Touristic Exploration. In M. Kurosu, editor, *Human-Computer Interaction. Applications and Services*, page 434–443, Cham, 2014b. Springer International Publishing. 7, 112, 146, 196

C. Keller, B. Radermacher, and A. Wehrmann. Kundeninformation für Smarte Informationsdienste - Ontologien. Technical report, 2014c. 126, 127, 128, 226

C. Keller, W. Titov, and T. Schlegel. SmartMMI Analyseergebnisse - Modell- und Kontextbasierte Mobilitätsinformationen auf smart Public Displays und Mobilgeräten im öffentlichen Verkehr. techreport, Hochschule Karlsruhe - Technik und Wirtschaft, 2018. 117

C. Keller, S. Struwe, W. Titov, and T. Schlegel. Understanding the Usefulness and Acceptance of Adaptivity in Smart Public Transport. In H. Krömker, editor, *HCI in Mobility, Transport, and Automotive Systems*, page 307–326, Cham, 2019a. Springer International Publishing. 7, 54, 106, 146

C. Keller, W. Titov, S. Sawilla, and T. Schlegel. Evaluation of a smart public display in public transport. In *Mensch und Computer 2019 - Workshopband*, Bonn, 2019b. Gesellschaft für Informatik e.V. 7, 54, 106, 146

C. Keller, W. Titov, and T. Schlegel. A Passenger Context Model for Adaptive Passenger Information in Public Transport. In H. Krömker, editor, *HCI in Mobility, Transport, and Automotive Systems. Driving Behavior, Urban and Smart Mobility*, pages 238–248, Cham, 2020. Springer International Publishing. 6, 9, 112, 113, 196, 229, 233

E. A. Kemp, A.-J. Thompson, and R. S. Johnson. Interface evaluation for invisibility and ubiquity: An example from e-learning. In *Proceedings of the 9th ACM SIGCHI New Zealand Chapter's International Conference on Human-Computer Interaction: Design Centered HCI*, CHINZ '08, page 31–38, New York, NY, USA, 2008. Association for Computing Machinery. 149, 150, 151, 153, 154, 155, 179, 259, 260, 261

J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, Jan 2003. 84

M. U. Khan. *Unanticipated Dynamic Adaptation of Mobile Applications*. PhD thesis, University of Kassel, Fachbereich 16: Elektrotechnik/Informatik, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany, March 2010. 73

D. Kim, P. Dunphy, P. Briggs, J. Hook, J. W. Nicholson, J. Nicholson, and P. Olivier. Multi-touch authentication on tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 1093–1102, New York, NY, USA, 2010. Association for Computing Machinery. 16

H. J. Kim, J. K. Choi, and Y. Ji. Usability evaluation framework for ubiquitous computing device. In *2008 Third International Conference on Convergence and Hybrid Information Technology*, volume 1, page 164–170, Nov 2008. 149, 150, 153, 154, 155, 259, 260, 261

J. Kjeldskov and J. Paay. A longitudinal review of mobile hci research methods. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, page 69–78, New York, NY, USA, 2012. Association for Computing Machinery. 23

J. Kjeldskov, M. B. Skov, B. S. Als, and R. T. Høegh. Is it worth the hassle? exploring the added value of evaluating the usability of context-aware mobile systems in the field. In S. Brewster and M. Dunlop, editors, *Mobile Human-Computer Interaction - MobileHCI 2004*, page 61–73, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 22

M. Knappmeyer, S. L. Kiani, C. Fra, B. Moltchanov, and N. Baker. Contextml: A light-weight context representation and context management schema. In *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*, page 367–372, May 2010. 36

M. Knappmeyer, S. L. Kiani, E. S. Reetz, N. Baker, and R. Tonjes. Survey of context provisioning middleware. *IEEE Communications Surveys Tutorials*, 15 (3):1492–1519, 2013. 46

A. Kofod-Petersen and M. Mikalsen. Context: Representation and reasoning: Representing and reasoning about context in a mobile environment. *Revue d'intelligence artificielle*, 2005. 47

M. Krauth. KVV Verbundbericht 2021. Technical report, Karlsruher Verkehrsverbund GmbH, 2021. 205

C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker. A survey on engineering approaches for self-adaptive systems. *Pervasive and Mobile Computing*, 17:184–206, 2015, 10 years of Pervasive Computing' In Honor of Chatschik Bisdikian. 45

R. Kühn, M. Korzetz, L. Büschel, F.-W. Schumann, and T. Schlegel. Device-Based Interactions for Anonymous Voting and Rating with Mobile Devices in Collaborative Scenarios. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, MUM '16, page 315–317, New York, NY, USA, 2016. Association for Computing Machinery. 19

T. Kuhnimhof, R. Buehler, M. Wirtz, and D. Kalinowska. Travel trends among young adults in germany: increasing multimodality and declining car use for men. *Journal of Transport Geography*, 24:443–450, 2012, Special Section on Theoretical Perspectives on Climate Change Mitigation in Transport. 1

O. Kwon, Y. Song, J. Kim, and K. Li. Sconstream: A spatial context stream processing system. In *2010 International Conference on Computational Science and Its Applications*, page 165–170, 2010. 39

R. Kühn, C. Keller, and T. Schlegel. A Context Taxonomy Supporting Public System Design. In *Proceedings of the 1st International Workshop on Model-based Interactive Ubiquitous Systems*, Pisa, Italy, 2011. vii, 6, 112, 113, 114, 228

R. Kühn, D. Lemme, and T. Schlegel. An Interaction Concept for Public Displays and Mobile Devices in Public Transport. In M. Kurosu, editor, *Human-Computer Interaction. Interaction Modalities and Techniques*, page 698–705, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 16

J. A. Landay and B. A. Myers. Sketching interfaces: Toward more human interface design. *Computer*, 34:56–64, March 2001. 22

M. Lapinski, M. Feldmeier, and J. A. Paradiso. Wearable wireless sensing for sports and ubiquitous interactivity. In *SENSORS, 2011 IEEE*, page 1425–1428, 2011. 16

J. Lazar, J. Feng, and H. Hochheiser. *Research Methods in Human-Computer Interaction*. Morgan Kaufmann Publishers Inc., Cambridge, MA, USA, 2017. 18

J. Lee and M. H. Yun. Usability assessment for ubiquitous services: Quantification of the interactivity in inter-personal services. In *2012 IEEE International Conference on Management of Innovation Technology (ICMIT)*, page 718–724, 2012. 149, 150, 153, 161, 259, 260, 261

J. Lee, J. Lim, H. Shin, and K. Kyung. Haptic interaction with user manipulation for smartphone. In *2013 IEEE International Conference on Consumer Electronics (ICCE)*, page 47–48, 2013. 15

O. Lehmann, M. Bauer, C. Becker, and D. Nicklas. From home to world - supporting context-aware applications through world models. In *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the*, pages 297–306, 2004. 33

D. Lemme, R. Kühn, A. Funke, and T. Schlegel. A feasibility study of context-adaptive visualizations in public transport information systems. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, MUM '14, page 262–263, New York, NY, USA, 2014. ACM. 16

L. H. Leong, S. Kobayashi, N. Koshizuka, and K. Sakamura. Casis: A context-aware speech interface system. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI '05, page 231–238, New York, NY, USA, 2005. Association for Computing Machinery. 15

U. Leonhardt, J. Magee, and P. Dias. Location service in mobile computing environments. *Computers & Graphics*, 20(5):627–632, 1996, <ce:title>Mobile Computing</ce:title>. 33

B. Y. Lim and A. K. Dey. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th International Conference on Ubiquitous Computing*, UbiComp '09, page 195–204, New York, NY, USA, 2009. ACM. 20

B. Y. Lim and A. K. Dey. Toolkit to support intelligibility in context-aware applications. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, page 13–22, New York, NY, USA, 2010. Association for Computing Machinery. 20, 48

V. López-Jaquero, J. Vanderdonckt, F. Montero, and P. González. Towards an Extended Model of User Interface Adaptation: The Isatine Framework. In J. Gulliksen, M. B. Harning, P. Palanque, G. C. van der Veer, and J. Wesson, editors, *Engineering Interactive Systems*, pages 374–392, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 90, 94

Y. Lu. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6:1–10, 2017. 13

K. Luyten and K. Coninx. Distributed user interface elements to support smart interaction spaces. In *Seventh IEEE International Symposium on Multimedia (ISM'05)*, page 8 pp., Dec 2005. 87, 88, 93

K. Luyten, C. Vandervelpen, and K. Coninx. Task Modeling for Ambient Intelligent Environments: Design Support for Situated Task Executions. In *Proceedings of the 4th International Workshop on Task Models and Diagrams*, TAMODIA '05, page 87–94, New York, NY, USA, 2005. Association for Computing Machinery. 88, 93

K. Lyytinen and Y. Yoo. Issues and challenges in ubiquitous computing. *Commun. ACM*, 45(12):62–65, Dec. 2002. 12

G. López, L. Quesada, and L. A. Guerrero. Alexa vs. siri vs. cortana vs. google assistant: A comparison of speech-based natural user interfaces. In I. L. Nunes, editor, *Advances in Human Factors and Systems Interaction*, page 241–250, Cham, 2018. Springer International Publishing. 15

R. S. Machado, R. B. Almeida, A. M. Pernas, and A. C. Yamin. State of the art in hybrid strategies for context reasoning: A systematic literature review. *Information and Software Technology*, 111:122–130, 2019. 41

M. W. Maier. Architecting principles for systems-of-systems. *Systems Engineering*, 1(4):267–284, 1998. 52

D. Malandrino, F. Mazzoni, D. Riboni, C. Bettini, M. Colajanni, and V. Scarano. MIMOSA: context-aware adaptation forubiquitous web access. *Personal and Ubiquitous Computing*, 14(4):301–320, 2010. 82, 83

J. Mankoff, A. K. Dey, G. Hsieh, J. Kientz, S. Lederer, and M. Ames. Heuristic evaluation of ambient displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, page 169–176, New York, NY, USA, 2003. Association for Computing Machinery. 21

T. Mantoro. Metrics evaluation for context-aware computing. In *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '09, page 574–578, New York, NY, USA, 2009. Association for Computing Machinery. 149, 150, 153, 155, 259, 260, 261

A. Mathur, N. D. Lane, and F. Kawsar. Engagement-Aware Computing: Modelling User Engagement from Mobile Contexts. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, page 622–633, New York, NY, USA, 2016. Association for Computing Machinery. 116

D. Maulsby, S. Greenberg, and R. Mander. Prototyping an intelligent agent through wizard of oz. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, page 277–284, New York, NY, USA, 1993. Association for Computing Machinery. 22

C. Mayas, S. Hörold, C. Rosenmöller, and H. Krömker. Evaluating methods and equipment for usability field tests in public transport. In M. Kurosu, editor, *Human-Computer Interaction. Theories, Methods, and Tools*, page 545–553, Cham, 2014. Springer International Publishing. 23

C. Mayas, T. Steinert, and H. Krömker. Interactive public displays for paperless mobility stations. In M. Kurosu, editor, *Human-Computer Interaction. Interaction* *in Context*, page 542–551, Cham, 2018. Springer International Publishing. 54, 106

J. McCarthy. Notes on formalizing context. In *Proceedings of the 13th International Joint Conference on Artifical Intelligence - Volume 1*, IJCAI'93, page 555–560, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. 37

J. F. McCarthy. Using public displays to create conversation opportunities. In *Workshop on Public, Community, and Situated Displays at the ACM press, 2002 Conference on Computer Supported Cooperative Work (CSCW)*, page 1–4, 2002. 16

P. K. McKinley, S. M. Sadjadi, E. P. Kasten, and B. H. C. Cheng. Composing adaptive software. *Computer*, 37(7):56–64, 2004. 71

G. Meixner, F. Paternò, and J. Vanderdonckt. Past, Present, and Future of Model-Based User Interface Development. *i-com*, 10(3):2–11, 2011. 17

J. F. Mejia Bernal, P. Falcarin, M. Morisio, and J. Dai. Dynamic context-aware business process: A rule-based approach supported by pattern identification. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, page 470–474, New York, NY, USA, 2010. Association for Computing Machinery. 47

J. Meskens, J. Vermeulen, K. Luyten, and K. Coninx. Gummy for Multi-platform User Interface Designs: Shape Me, Multiply Me, Fix Me, Use Me. *Avi '08*, page 233–240, 2008. 73

R. Miñón, F. Paternò, and M. Arrue. An Environment for Designing and Sharing Adaptation Rules for Accessible Applications. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '13, page 43–48, New York, NY, USA, 2013. Association for Computing Machinery. 90, 91, 95

R. Mizoguchi and M. Ikeda. Towards ontology engineering. Technical report, 1998. 27

R. Mizoguchi and K. Kozaki. *Ontology Engineering Environments*, page 315–336. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 27

T. Moran and P. Dourish. Introduction to this special issue on context- aware computing. *Human-Computer Interaction*, 16(2-4):87–97, 2001. 20

V. G. Motti. *TriPlet: A Conceptual Framework for Multidimensional Adaptation of User Interfaces to the Context of Use*. PhD thesis, École Polytechnique de Louvain, Université catholique de Louvain, Nov. 2013. 72

V. G. Motti and J. Vanderdonckt. A computational framework for context-aware adaptation of user interfaces. In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, page 1–12, 2013. 17, 71, 72, 73, 74, 75, 77, 91, 95

J. Müller, F. Alt, D. Michelis, and A. Schmidt. Requirements and Design Space for Interactive Public Displays. In *Proceedings of the international conference on Multimedia*, MM '10, page 1285–1294, New York, NY, USA, 2010. ACM. 16

C. Müller-Tomfelde and M. Fjeld. Tabletops: Interactive horizontal displays for ubiquitous computing. *Computer*, 45(02):78–81, feb 2012. 16

T. Naganuma and S. Kurakake. Task knowledge based retrieval for service relevant to mobile user's activity. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *The Semantic Web – ISWC 2005*, page 959–973, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 48

S. Neely, G. Stevenson, C. Kray, I. Mulder, K. Connelly, and K. A. Siek. Evaluating Pervasive and Ubiquitous Systems. *IEEE Pervasive Computing*, 7(3):85–88, July 2008. 23

D. Nicklas and B. Mitschang. On building location aware applications using an open platform based on the NEXUS Augmented World Model. *Software and Systems Modeling*, 3:301–313, 2004. 33

D. Nicklas, M. Großmann, T. Schwarz, S. Volz, and B. Mitschang. A Model-Based, Open Architecture for Mobile, Spatially Aware Applications. In C. S. Jensen, M. Schneider, B. Seeger, and V. J. Tsotras, editors, *Advances in Spatial and Temporal Databases*, pages 117–135, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 33, 37

J. Nielsen. Noncommand user interfaces. *Commun. ACM*, 36(4):83–99, Apr. 1993a. 14, 170

J. Nielsen. *Usability Engineering*. Academic Press, Inc., 1993b. 17, 18, 25, 26, 148, 150, 259, 260, 261

J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, page 249–256, New York, NY, USA, 1990. Association for Computing Machinery. 21

L. Nielsen. From user to character: An investigation into user-descriptions in scenarios. In *Proceedings of the 4th Conference on Designing Interactive Systems:*

Processes, Practices, Methods, and Techniques, DIS ’02, page 99–104, New York, NY, USA, 2002. Association for Computing Machinery. 60

L. Nielsen. *Personas - User Focused Design*, volume 2nd edition. Springer-Verlag, 2019. 60

A. F. Norcio and J. Stanley. Adaptive human-computer interfaces: a literature survey and perspective. *IEEE Transactions on Systems, Man, and Cybernetics*, 19 (2):399–408, 1989. 71

D. M. N. F. Noy. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford knowledge systems laboratory technical report ksl-01-05, Stanford Knowledge Systems Laboratory, 2001. 156

S. Nylander and M. Bylund. Providing Device Independence to Mobile Services. In N. Carbonell and C. Stephanidis, editors, *Universal Access Theoretical Perspectives, Practice, and Experience*, pages 465–473, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 17, 75, 86, 87, 93

S. Nylander, M. Bylund, and A. Waern. The Ubiquitous Interactor — Device Independent Access to Mobile Services. In R. J. Jacob, Q. Limbourg, and J. Vanderdonckt, editors, *Computer-Aided Design of User Interfaces IV*, pages 271–282, Dordrecht, 2005. Springer Netherlands. 87, 93

M. O’Connor, H. Knublauch, S. Tu, B. Grosof, M. Dean, W. Grosso, and M. Musen. Supporting Rule System Interoperability on the Semantic Web with SWRL. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *The Semantic Web – ISWC 2005*, pages 974–986, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 32

T. Okoshi, K. Tsubouchi, M. Taji, T. Ichikawa, and H. Tokuda. Attention and engagement-awareness in the wild: A large-scale study with adaptive notifications. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, page 100–110, March 2017. 19, 165, 181

E. O’Neill, M. Kaenampornpan, V. Kostakos, A. Warr, and D. Woodgate. Can we do without guis? gesture and speech interaction with a patient information system. *Personal and Ubiquitous Computing*, 10(5):269–283, Aug. 2006. 14

A. Padovitz, S. W. Loke, A. Zaslavsky, B. Burg, and C. Bartolini. An approach to data fusion for context awareness. In A. Dey, B. Kokinov, D. Leake, and R. Turner, editors, *Modeling and Using Context*, page 353–367, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 40

D. J. Pagliari, M. Ansaldi, E. Macii, and M. Poncino. CNN-Based Camera-less User Attention Detection for Smartphone Power Management. In *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6, 2019. 123, 173

C. Parker, J. Kay, and M. Tomitsch. Device-free: An implicit personalisation approach for public interactive displays. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '18, New York, NY, USA, 2018. Association for Computing Machinery. 16

J. Pascoe. Adding generic contextual capabilities to wearable computers. In *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*, page 92–99, 1998. 43, 44

N. Paspallis. *Middleware-based development of context-aware applications with reusable components*. PhD thesis, University of Cyprus, Faculty of Pure and Applied Sciences, , 2009. 85

F. Paternò. *Model-Based Design and Evaluation of Interactive Applications*. Springer-Verlag, 1999. 17

F. Paternò. *The handbook of task analysis in human-computer interaction*, chapter ConcurTaskTrees:An Engineered Notationfor Task Models, pages 483–501. Chemical Rubber Company Press, 2003. 132

F. Paternò, C. Santoro, and L. D. Spano. MARIA (Model-based lAnguage foR Interactive Applications). W3c working group submission, World Wide Web Consortium (W3C), 2012. 98

F. Paternò, C. Santoro, and L. D. Spano. MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *ACM Trans. Comput.-Hum. Interact.*, 16:19:1–19:30, November 2009. 17, 91, 94, 98, 99

M. Peissner, D. Häbe, D. Janssen, and T. Sellner. Myui: Generating accessible user interfaces from multimodal design patterns. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '12, page 81–90, New York, NY, USA, 2012. Association for Computing Machinery. 90, 95

V. Pejovic and M. Musolesi. InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, page 897–908, New York, NY, USA, 2014. Association for Computing Machinery. 181

P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko. It's Mine, Don't Touch! Interactions at a Large Multi-Touch Display in a City Centre. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 1285–1294, New York, NY, USA, 2008. Association for Computing Machinery. 115

C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, First 2014. 33, 34, 35, 39, 40, 41

M. Perttunen, J. Riekki, and O. Lassila. Context representation and reasoning in pervasive computing: a review. *International Journal of Multimedia and Ubiquitous Engineering*, page 1–28, 2009. 41

C. Petit-Rozé and E. Grislin-Le Strugeon. Mapis, a multi-agent system for information personalization. *Information and Software Technology*, 48(2):107–120, 2006. 49

D. Petrelli, E. Not, C. Strapparava, O. Stock, and M. Zancanaro. Modeling context is like taking pictures. In *Proc. of the Workshop "The What, Who, Where, When, Why and How of Context-Awareness" in CHI2000*, 2000. 36

T. Prante, C. Röcker, N. A. Streitz, R. Stenzel, C. Magerkurth, D. van Alphen, and D. A. Plewe. Hello.Wall – Beyond Ambient Displays. In P. Ljungstrand and J. Brotherton, editors, *Video Track and Adjunct Proceedingsof the 5th Intern. Conference on Ubiquitous Computing (UBICOMP'03)*, Seattle, Washington, USA, Oct. 2003. 180

D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. De Bosschere. Towards an extensible context ontology for ambient intelligence. In P. Markopoulos, B. Eggen, E. Aarts, and J. L. Crowley, editors, *Ambient Intelligence*, page 148–159, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 114, 115, 116

A. J. Quigley. *From GUI to UUI: Interfaces for Ubiquitous Computing*, page 237–284. Taylor & Francis Group, New York, USA, 2010. 14, 149, 150, 154, 155, 164, 259, 260, 261

B. Radermacher, A. Wehrmann, and AK2, Forschungsprojekt IP-KOM-ÖV. Kommunikation im ÖV (IP-KOM-ÖV) Szenarien & Personen sowie deren Anforderungen an die Kundeninformation. Technical report, Kamekestraße 37 - 39, 50672 Köln, Mai 2012. 60, 63, 117

A. Ranganathan and R. H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Comput.*, 7(6):353–364, Dec. 2003. 47

A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3 (2):62–70, 2004. 163

R. A. C. Reis, A. L. de Fontão, L. L. Gomes, and A. C. Dias-Neto. Usability evaluation approaches for (ubiquitous) mobile applications: A systematic mapping study. In *9th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2015. 23

J. Rekimoto and M. Saitoh. Augmented surfaces: A spatially continuous work space for hybrid computing environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, page 378–385, New York, NY, USA, 1999. Association for Computing Machinery. 15

M. L. Resnick. Ubiquitous computing: Ux when there is no ui. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 57(1):1007–1011, 2013. 14

B. J. Rhodes, N. Minar, and J. Weaver. Wearable computing meets ubiquitous computing: reaping the best of both worlds. In *Digest of Papers. Third International Symposium on Wearable Computers*, page 141–149, 1999. 16

V. Ricquebourg, D. Menga, B. Marhic, L. Delahoche, D. Durand, and C. Loge. Service oriented architecture for context perception based on heterogeneous sensors network. In *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, page 4557–4562, 2006. 41, 42, 46, 47

V. Ricquebourg, D. Durand, D. Menga, B. Marhic, L. Delahoche, C. Loge, and A. Jolly-Desodt. Context Inferring in the Smart Home: An SWRL Approach. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, volume 2, page 290–295, May 2007. 42

J. Riekki, P. Isomursu, and M. Isomursu. Evaluating the calmness of ubiquitous applications. In F. Bomarius and H. Iida, editors, *Product Focused Software Process Improvement*, page 105–119, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 148, 150, 154, 155, 259, 260, 261

E. Rigas, G. Meditskos, and N. Bassiliades. SWRL2COOL: Object-Oriented Transformation of SWRL in the CLIPS Production Rule Engine. In I. Maglogiannis, V. Plagianakos, and I. Vlahavas, editors, *Artificial Intelligence: Theories* *and Applications*, pages 49–56, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 32

L. C. Rocha, R. M. C. Andrade, A. L. Sampaio, and V. Lelli. Heuristics to evaluate the usability of ubiquitous systems. In N. Streitz and P. Markopoulos, editors, *Distributed, Ambient and Pervasive Interactions*, page 120–141, Cham, 2017. Springer International Publishing. 21, 25, 26

Y. Rogers and H. Brignull. Subtle ice-breaking: encouraging socializing and interaction around a large public display. In *Workshop on Public, Community, and Situated Displays at the ACM press, 2002 Conference on Computer Supported Cooperative Work (CSCW)*, 2002. 16

P. Rosenberger and D. Gerhard. Context-awareness in industrial applications: definition, classification and use case. *Procedia CIRP*, 72:1172–1177, 2018, 51st CIRP Conference on Manufacturing Systems. 234

M. B. Rosson and J. M. Carroll. Scenario-based design. In A. Sears and J. A. Jacko, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, page 1032–1050. Lawrence Erlbaum Associates, 2002. 60

R. Rouvoy, F. Eliassen, J. Floch, S. Hallsteinsen, and E. Stav. Composing components and services using a planning-based adaptation middleware. In C. Pautasso and E. Tanter, editors, *Software Composition*, volume 4954 of *Lecture Notes in Computer Science*, pages 52–67. Springer Berlin Heidelberg, 2008. 85

R. Rouvoy, P. Barone, Y. Ding, F. Eliassen, S. Hallsteinsen, J. Lorenzo, A. Mamelli, and U. Scholz. MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments. In B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, and J. Magee, editors, *Software Engineering for Self-Adaptive Systems*, volume 5525 of *Lecture Notes in Computer Science*, pages 164–182. Springer Berlin Heidelberg, 2009. 83, 84, 85, 186

D. E. Rowley. Usability Testing in the Field: Bringing the Laboratory to the User. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, page 252–257, New York, NY, USA, 1994. Association for Computing Machinery. 22

J. Rubin, D. Chisnell, and J. Spool. *Handbook of usability testing : how to plan, design, and conduct effective tests*. Wiley, May 2008. 18

D. M. Russell, C. Drews, and A. E. Sue. Social aspects of using large public interactive displays for collaboration. In G. Borriello and L. E. Holmquist, editors, *Ubicomp*, volume 2498 of *Lecture Notes in Computer Science*, page 229–236. Springer, 2002. 16, 115

H. Ryu, G. Hong, and H. James. Quality assessment technique for ubiquitous software and middleware. Technical report, 2006. 148, 149, 150, 152, 153, 154, 155, 161, 179, 259, 260, 261

D. Salber, A. K. Dey, and G. D. Abowd. Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm. Technical report, Graphics, Visualization and Usability Center at Georgia Institute of Technology, 1998. 12

D. Salber, A. K. Dey, and G. D. Abowd. The context toolkit: aiding the development of context-enabled applications. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, page 434–441, New York, NY, USA, 1999. ACM. 19

M. Salehie and L. Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.*, 4(2):14:1–14:42, May 2009. 84

L. Sanchez, J. Lanza, R. Olsen, M. Bauer, and M. Girod-Genet. A generic context management framework for personal networking environments. In *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking Services*, page 1–8, July 2006. 40

R. M. Santos, K. M. de Oliveira, R. M. C. Andrade, I. S. Santos, and E. R. Lima. A Quality Model for Human-Computer Interaction Evaluation in Ubiquitous Systems. In C. Collazos, A. Liborio, and C. Rusu, editors, *Human Computer Interaction*, page 63–70, Cham, 2013. Springer International Publishing. 148, 150, 151, 154, 155, 259, 260, 261

M. Satyanarayanan. Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4):10–17, Aug 2001. 12, 19, 99

J. Sauro and J. R. Lewis. *Quantifying the user experience : practical statistics for user research*. Amsterdam ; Boston ; Heidelberg : Elsevier, Morgan Kaufmann, 2nd edition edition, 2016. 217

B. Schilit and M. Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22–32, sep/oct 1994. 12, 33, 42, 43, 82, 113

B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Workshop on Mobile Computing Systems and Applications*, page 85–90, Dec 1994. 114

O.-A. Schipor, R.-D. Vatavu, and J. Vanderdonckt. Euphoria: A Scalable, event-driven architecture for designing interactions across heterogeneous devices in smart environments. *Information and Software Technology*, 109:43–59, 2019. 83, 84

T. Schlegel, editor. *Multi-Touch*. Xpert.press. Springer Vieweg, 2013. 15

T. Schlegel and C. Keller. Model-based Ubiquitous Interaction Concepts and Contexts in Public Systems. In *Proceedings of the 14th International Conference on Human-Computer Interaction*, 2011. vii, 6, 112, 113, 114, 228

A. Schmidt. Implicit human computer interaction through context. *Personal Technologies*, 4(2-3):191–199, June 2000. 14

A. Schmidt. Ubiquitous computing: Are we there yet? *IEEE Computer*, 43(2): 95–97, February 2010. 11, 12

A. Schmidt, M. Beigl, and H.-W. Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, 1999. 2, 16, 33, 40, 43, 44, 113, 114, 116

A. Schmidt, B. Pfleging, F. Alt, A. Sahami, and G. Fitzpatrick. Interacting with 21st-century computers. *Pervasive Computing, IEEE*, 11(11):22–31, January 2012. 11

J. Scholtz and S. Consolvo. Towards a discipline for evaluating ubiquitous computing applications. Technical report, Intel Corporation, January 2004. 21, 164

J. Scholtz and S. Consolvo. Toward a framework for evaluating ubiquitous computing applications. *IEEE Pervasive Computing*, 3(2):82–88, 2004. 25, 148, 150, 152, 154, 155, 259, 260, 261

V. Schwartze, M. Blumendorf, and S. Albayrak. Adjustable Context Adaptations for User Interfaces at Runtime. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI ’10, page 321–324, New York, NY, USA, 2010. Association for Computing Machinery. 2, 73, 90, 95

A. Sears. Layout appropriateness: a metric for evaluating user interface widget layout. *IEEE Transactions on Software Engineering*, 19(7):707–719, 1993. 24

A. Sears. AIDE: A Step toward Metric-Based Interface Development Tools. In *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*, UIST '95, page 101–110, New York, NY, USA, 1995. Association for Computing Machinery. 24

A. Seffah, M. Donyaee, R. Kline, and H. Padda. Usability measurement and metrics: a consolidated model. *Software Quality Journal*, 14(2):159–178, 2006. 25, 148, 150, 151, 153, 259, 260, 261

O. Shaer and E. Hornecker. Tangible user interfaces: Past, present, and future directions. *Found. Trends Hum.-Comput. Interact.*, 3(1–2):1–137, Jan. 2010. 15

S. A. N. Shafer, B. Brumitt, and J. Cadiz. Interaction Issues in Context-Aware Intelligent Environments. *Human–Computer Interaction*, 16(2-4):363–378, 2001. 2, 13

A. Shepherd. *Hierarchical Task Analysis*. Taylor & Francis, 2001. 117

A. S. Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt. Large-Scale Assessment of Mobile Notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 3055–3064, New York, NY, USA, 2014. Association for Computing Machinery. 181

C. Silpasuwanchai and X. Ren. Jump and shoot!: Prioritizing primary and alternative body gestures for intense gameplay. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, page 951–954, New York, NY, USA, 2014. ACM. 15

K.-L. Skillen, L. Chen, C. Nugent, M. Donnelly, W. Burns, and I. Solheim. Using SWRL and Ontological Reasoning for the Personalization of Context-Aware Assistive Services. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '13, New York, NY, USA, 2013. Association for Computing Machinery. 47, 82, 83, 99, 116

J.-S. Sottet, G. Calvary, and J.-M. Favre. Models at runtime for sustaining user interface plasticity. In *Models@run.time workshop (inconjunction with MoDELS/UML 2006 conference)*, 2006. 94, 96

J.-S. Sottet, V. Ganneau, G. Calvary, J. Coutaz, A. Demeure, J.-M. Favre, and R. Demumieux. Model-Driven Adaptation for Plastic User Interfaces. In C. Baranauskas, P. Palanque, J. Abascal, and S. D. J. Barbosa, editors, *Human-Computer Interaction – INTERACT 2007*, page 397–410, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 94, 96

N. A. Stanton. Hierarchical task analysis: Developments, applications, and extensions. *Applied Ergonomics*, 37(1):55–79, 2006, Special Issue: Fundamental Reviews. 117, 118

T. Strang and C. Linnhoff-Popien. A Context Modeling Survey. In *UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management*, page 31–41, Nottingham, September 2004. 35, 36, 37, 222

M. Strimpakou, I. Roussaki, and M. Anagnostou. A context ontology for pervasive service provision. In *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, page 5 pp., 2006. 116

R. Studer, R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data and Knowledge Engineering*, 25:161–197, 1998. 26

Y. Sure, S. Staab, and R. Studer. *Ontology Engineering Methodology*, page 135–152. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 27

A. G. Sutcliffe, N. A. M. Maiden, S. Minocha, and D. Manuel. Supporting scenario-based requirements engineering. *IEEE Trans. Softw. Eng.*, 24: 1072–1088, December 1998. 60

L. Takayama. The motivations of ubiquitous computing: revisiting the ideas behind and beyond the prototypes. *Personal and Ubiquitous Computing*, 21(3): 557–569, Jun 2017. 11

Y. Taniguchi. [invited paper] content scheduling and adaptation for networked and context-aware digital signage: A literature survey. *ITE Transactions on Media Technology and Applications*, 6(1):18–29, 2018. 16

D. Thevenin and J. Coutaz. Plasticity of User Interfaces: Framework and Research Agenda. In A. Sasse and C. Johnson, editors, *Human–Computer Interaction — INTERACT'99*. IOS Press, 1999. xxi, xxiii, 72, 92

P. Thomas and H. Thimbleby. The new usability: The challenge of designing for pervasive computing. In *Proceedings of the 15th International Conference on Computer Communication*, ICCC '02, page 382–388, USA, 2002. International Council for Computer Communication. 18

K. E. Thompson, E. P. Rozanski, and A. R. Haake. Here, There, Anywhere: Remote Usability Testing That Works. In *Proceedings of the 5th Conference on Information Technology Education*, CITC5 '04, page 132–137, New York, NY, USA, 2004. Association for Computing Machinery. 22

W. Titov, H. Tran, C. Keller, and T. Schlegel. A Multi-device Evaluation Approach of Passenger Information Systems in Smart Public Transport. In H. Krömker, editor, *HCI in Mobility, Transport, and Automotive Systems. Driving Behavior, Urban and Smart Mobility*, pages 340–358, Cham, 2020. Springer International Publishing. 7, 54, 147, 165

T. Tullis and B. Albert. *Measuring the User Experience. Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufman, second edition edition, 2013. 209

G. Tumas and F. Ricci. Personalized mobile city transport advisory system. In W. Höpken, U. Gretzel, and R. Law, editors, *Information and Communication Technologies in Tourism 2009*, page 173–183, Vienna, 2009. Springer Vienna. 113

L. D. Turner, S. M. Allen, and R. M. Whitaker. Interruptibility Prediction for Ubiquitous Systems: Conventions and New Directions from a Growing Field. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, page 801–812, New York, NY, USA, 2015. ACM. 234

T. Uemukai, T. Hara, M. Tsukamoto, and S. Nishio. A remote display environment: an integration of mobile and ubiquitous computing environments. In *2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002 (Cat. No.02TH8609)*, volume 2, page 618–624 vol.2, 2002. 16

J. Undercoffer, F. Perich, A. Cedilnik, L. Kagal, and A. Joshi. A secure infrastructure for service discovery and access in pervasive computing. *Mob. Netw. Appl.*, 8(2):113–125, Apr. 2003. 49

M. Uschold and M. King. Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995. 26, 156

C. Vandervelpen and K. Coninx. Towards Model-Based Design Support for Distributed User Interfaces. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction*, NordiCHI '04, page 61–70, New York, NY, USA, 2004. Association for Computing Machinery. 87, 93, 104

T. Vilarinho, B. Farshchian, D. G. Bajer, O. H. Dahl, I. Egge, S. S. Hegdal, A. Lønes, J. N. Slettevold, and S. M. Weggersen. A combined smartphone and smartwatch fall detection system. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, page 1443–1448, 2015. 16

D. Vogel and R. Balakrishnan. Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, page 137–146, New York, NY, USA, 2004. Association for Computing Machinery. 180

K. Väänänen-Vainio-Mattila, T. Olsson, and J. Häkkilä. Towards deeper understanding of user experience with ubiquitous computing systems: Systematic literature review and design framework. In J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, editors, *Human-Computer Interaction – INTERACT 2015*, page 384–401, Cham, 2015. Springer International Publishing. 18

X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung. Ontology based context modeling and reasoning using owl. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, PERCOMW '04, page 18–, Washington, DC, USA, 2004. IEEE Computer Society. 38

R. Want. *Ubiquitous Computing Fundamentals*, chapter An Introduction to Ubiquitous Computing, page 1–35. Chapman & Hall/CRC, 1st edition, 2009. 1

R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992. 33, 49, 82, 113

A. Wehrmann, D. Weißer, H. Bandelin, R. Kruppa, and T. Franke. Ip-kom-Öv entwickelt wirtschaftliche lösungen für verkehrsunternehmen und industrie: "ibis-ip", July 2012. 56

M. Weiser. The computer for the 21st century. *Scientific American*, 265:94–104, 1991. 1, 11, 12, 14, 16, 18, 82, 99

M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, July 1993. 1, 11, 14

M. Weiser and J. S. Brown. Designing Calm Technology. *Power Grid Journal*, 1996. 19, 85, 148

D. Weißer, T. Franke, H. Bandelin, B. Radermacher, A. Wehrmann, and W. Meier-Leu. Internetprotokoll basiertes integriertes Bordinformationssystem IBIS-IP - Teil 1: Systemarchitektur. Standard, Kamekestraße 37 - 39, 50672 Köln, January 2014. 54, 56

A. Whitmore, A. Agarwal, and L. Da Xu. The Internet of Things – A survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274, Apr 2015. 13

T. Wienken, H. Krömker, and S. Spundflasch. Agenda Planning - Design Guidelines for Holistic Mobility Planning. In M. Kurosu, editor, *Human-Computer Interaction. Interaction Contexts*, pages 713–720, Cham, 2017. Springer International Publishing. 106

C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin. Smart Homes and Their Users: A Systematic Analysis and Key Challenges. *Personal Ubiquitous Comput.*, 19(2):463–476, Feb. 2015. 234

F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng. From wireless sensor networks towards cyber physical systems. *Pervasive and Mobile Computing*, 7(4):397–413, 2011. 13

J. Yang, E. Munguia-Tapia, and S. Gibbs. Efficient In-Pocket Detection with Mobile Phones. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp '13 Adjunct, page 31–34, New York, NY, USA, 2013. Association for Computing Machinery. 173

E. Yiğitbaş, S. Grün, S. Sauer, and G. Engels. Model-Driven Context Management for Self-adaptive User Interfaces. In S. F. Ochoa, P. Singh, and J. Bravo, editors, *Ubiquitous Computing and Ambient Intelligence*, pages 624–635, Cham, 2017a. Springer International Publishing. 95, 97

E. Yiğitbaş, H. Stahl, S. Sauer, and G. Engels. Self-adaptive UIs: Integrated Model-Driven Development of UIs and Their Adaptations. In A. Anjorin and H. Espinoza, editors, *Modelling Foundations and Applications*, pages 126–141, Cham, 2017b. Springer International Publishing. 97

E. Yiğitbaş, I. Jovanovikj, K. Josifovska, S. Sauer, and G. Engels. On-the-Fly Usability Evaluation of Mobile Adaptive UIs Through Instant User Feedback. In D. Lamas, F. Loizides, L. Nacke, H. Petrie, M. Winckler, and P. Zaphiris, editors, *Human-Computer Interaction – INTERACT 2019*, pages 563–567, Cham, 2019. Springer International Publishing. 95, 97

E. Yiğitbaş, I. Jovanovikj, K. Biermeier, S. Sauer, and G. Engels. Integrated model-driven development of self-adaptive user interfaces. *Software and Systems Modeling*, (19):1057 – 1081, 2020. 95, 97

Y. Yun and I. Yu-Hua Gu. Chapter 15 - visual information-based activity recognition and fall detection for assisted living and ehealthcare. In C. Dobre, C. Mavromoustakis, N. Garcia, R. Goleva, and G. Mastorakis, editors, *Ambient Assisted Living and Enhanced Living Environments*, page 395–425. Butterworth-Heinemann, 2017. 12

N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan. Shoulder surfing defence for recall-based graphical passwords. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, SOUPS '11, New York, NY, USA, 2011. Association for Computing Machinery. 19

M. Zhang and A. A. Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 1036–1043, New York, NY, USA, 2012. Association for Computing Machinery. 16

L. Zhao and Z. Peng. Analysis and Design of a Context-Aware Smart Home System. In *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pages 716–719, 2020. 234

J. Zhou, C. Yu, J. Riekki, and E. Kärkkäinen. AmE Framework: a Model for Emotion-awareAmbient Intelligence. Technical report, University of Oulu, Department of Electrical and Information Engineering, Faculty of Humanities, Department of English VTT Technical Research Center of Finland, 2007. 116

A. Zimmermann. Context-Awareness in User Modelling: Requirements Analysis for a Case-Based Reasoning Application. In K. D. Ashley and D. G. Bridge, editors, *Case-Based Reasoning Research and Development*, page 718–732, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 41

A. Zimmermann, A. Lorenz, and R. Oppermann. An operational definition of context. In B. Kokinov, D. C. Richardson, T. R. Roth-Berghofer, and L. Vieu, editors, *Modeling and Using Context*, page 558–571, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 34