

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Problem Statement	6
1.3	Thesis Goals	7
1.4	Design Methodology	9
1.5	Thesis Context	10
1.6	Outline	11
2	Background and Technological Overview	13
2.1	Ubiquitous Computing vs Pervasive Computing vs Ambient Intelligence	14
2.2	Model Driven Engineering	15
2.2.1	Development Models, Executable Models & Runtime Models	15
2.2.2	Code Generation vs Model Interpretation	17
2.3	Ontology, Ontology Languages and Ontology Reasoners	20
2.3.1	Web Ontology Language (OWL)	22
2.3.2	Pellet: an OWL-DL Reasoner	22

2.3.3	SPARQL	23
2.4	Open Services Gateway initiative (OSGi)	24
2.5	Conclusions	28
3	State of the Art	29
3.1	Analysis Criteria	30
3.2	Machine Learning Approaches	34
3.2.1	Analysis and Discussion	44
3.3	Rule-based Context-aware Approaches	46
3.3.1	Analysis and Discussion	55
3.4	End-user Centred Approaches	56
3.4.1	Analysis and Discussion	65
3.5	Benefits of our Proposal	67
3.6	Discussion and Conclusions	68
4	Overview of the Proposal	71
4.1	Introduction	73
4.2	Process for Automating User Behaviour Patterns	74
4.2.1	SPEM notation	75
4.2.2	The Process Activities	76
4.3	Software Infrastructure	80
4.4	Validation	84
4.5	Conclusions	86
5	Modelling User Behaviour Patterns	87
5.1	Modelling Context	88
5.1.1	The Context Concept	89
5.1.2	Context Modelling in Pervasive Systems	91
5.1.3	An Ontology-based Context Model	92
5.1.4	Tool Support for Creating a Context Model	99

- 5.2 Modelling the Behaviour Patterns 101
 - 5.2.1 The Task Concept 101
 - 5.2.2 Task Modelling in Software Engineering 102
 - 5.2.3 A Context-adaptive Task Model 105
 - 5.2.4 Tool support 118
- 5.3 Conclusions 121

- 6 Automating User Behaviour Patterns 123**
 - 6.1 Requirements for Automating Behaviour Patterns 124
 - 6.2 Behaviour Patterns' Automation Process 125
 - 6.3 Software Infrastructure 128
 - 6.3.1 Components of the Software Infrastructure 128
 - 6.3.2 Implementation of the Software Infrastructure 131
 - 6.4 Conclusions 146

- 7 Addressing the Evolution of the User Behaviour Patterns 149**
 - 7.1 Evolution Characterization 151
 - 7.2 Mechanisms for Evolving the Behaviour Patterns 153
 - 7.3 Tool Support 159
 - 7.3.1 Interface Design Decisions 163
 - 7.3.2 Description of the Graphical User Interfaces 164
 - 7.3.3 Evolving the Behaviour Patterns 171
 - 7.4 Conclusions 174

- 8 Evaluation of the Approach 175**
 - 8.1 Smart Home Case Studies 176
 - 8.1.1 Design of the Smart Home Case Studies 177
 - 8.1.2 Results of the smart home case studies 183
 - 8.1.3 Conclusions of the Smart Home Case Studies' Validation 196

8.2	Nursing Home Case Study	197
8.2.1	Design of the Nursing Home Case Study	197
8.2.2	Results of the the Nursing Home Case Study	200
8.2.3	Conclusions of the Nursing Home Case Study Validation	207
8.3	Scalability of Using Models at Runtime	208
8.4	Conclusions	209
9	Conclusions	211
9.1	Contributions	212
9.2	Publications	213
9.2.1	Detail and Relevance of the publications	216
9.3	Future work	217
9.3.1	Combination with Machine-learning Algorithms	218
9.3.2	Providing Adaptive User Interfaces	219
9.3.3	Interactive and Iterative Tasks and Tasks with State	220
9.3.4	Facilitating the Routine Task Evolution by End- users	222
	Bibliography	224
A	Software Infrastructure	237
A.1	Model Management Mechanisms Implementation	237
A.1.1	Managing the Context Model: OSea	239
A.1.2	Managing the Task Model: MUTate	242
A.1.3	APIs' Testing	246
A.2	Pervasive Services	248
A.3	Context Monitor	250
A.4	MAtE	253
B	Case Study Requirements	257

- B.1 Smart Home Requirements 257
 - B.1.1 An Interview for Identifying the Behaviour Patterns 257
 - B.1.2 The Identified Behaviour Patterns 259
 - B.1.3 Required Services 261
- B.2 Nursing Home Requirements 263
 - B.2.1 ACube Requirement Elicitation Artefacts 263
 - B.2.2 The Identified Behaviour Patterns 269
 - B.2.3 Required Services 272