
Contents

List of Figures	viii
List of Tables	xi
1. Introduction	3
1.1. Motivation	3
1.2. Goals and Scope Delimitation	6
1.3. Thesis Contributions	8
1.4. Text Organization	10
2. Theoretical Background	11
2.1. Introduction	11
2.2. Distributed Embedded Real-Time Systems	11
2.2.1. Introduction	11
2.2.2. Real-Time Systems	11
2.2.3. Embedded Systems	13
2.2.4. Distributed Systems	13
2.3. Requirements in Embedded Systems Domain	14
2.4. Embedded Systems Design Approaches	17
2.4.1. Introduction	17
2.4.2. Object-Oriented Paradigm	18
2.4.3. Aspect-Oriented Paradigm	19
2.4.4. Evaluating the Design with Metrics	21
2.4.4.1. Introduction	21
2.4.4.2. C&K Metrics Suite	22
2.4.4.3. Assessment Framework for AO systems	23
2.5. Model-Driven Engineering	25
2.5.1. Overview	25
2.5.2. MARTE UML profile	26
3. State of the Art Analysis	31
3.1. Introduction	31
3.2. Design and Modeling Approaches	32
3.2.1. Overview of Related-Work	32
3.2.2. Discussion	36

3.3.	Separation of Concerns in Requirements Handling	37
3.3.1.	Introduction	37
3.3.2.	Separation of Concerns in General Systems Development	37
3.3.3.	The Use of AOD in the Design of DERTS	41
3.3.4.	Discussion	45
3.4.	Code Generation	46
3.4.1.	Introduction	46
3.4.2.	Code Generation from UML Models	47
3.4.3.	Commercial Tools	50
3.4.4.	Discussion	51
3.5.	Discussion on the Open Problems	52
4.	MDE process for DERTS design	55
4.1.	Introduction	55
4.2.	Aspect-Oriented Model-Driven Engineering for DERTS	56
4.3.	Adaptations in the SEEP design flow	60
5.	Specifying DERTS Using UML and Aspects	63
5.1.	Introduction	63
5.2.	Functional Requirements Handling Elements	64
5.2.1.	Introduction	64
5.2.2.	Specification of System Expected Functionalities	65
5.2.3.	Specification of System Structure	66
5.2.3.1.	Class Diagram	66
5.2.3.2.	Composite Structure Diagram	67
5.2.3.3.	Deployment Diagram	69
5.2.4.	System Behavior Specification	70
5.2.4.1.	Sequence Diagram	70
5.2.4.2.	Activity Diagram	75
5.2.4.3.	State Diagram	76
5.3.	Non-Functional Requirements Handling Elements	77
5.3.1.	Introduction	77
5.3.2.	Distributed Embedded Real-time Aspects Framework	79
5.3.2.1.	Overview	79
5.3.2.2.	Timing Package	80
5.3.2.3.	Precision Package	81
5.3.2.4.	Synchronization Package	82
5.3.2.5.	Communication Package	82
5.3.2.6.	TaskAllocation Package	83
5.3.2.7.	Embedded Package	83
5.3.2.8.	Discussion	84
5.3.3.	Aspects Crosscutting Overview Diagram	86
5.3.4.	Join Points: Selecting Model Elements Affected by Aspects	88
5.4.	Final Remarks	91

6. Tool Support for the Proposed Approach	93
6.1. Introduction	93
6.2. A Platform Independent Model for Code Generation	93
6.3. UML-to-DERCS Transformation	98
6.4. Mapping Rules	104
6.4.1. Overview	104
6.4.2. Application Code	105
6.4.3. Platform Configuration	112
6.5. Code Generation Process	113
6.6. Final Remarks	116
7. Validation	119
7.1. Introduction	119
7.2. Toolset Overview	119
7.2.1. RT-FemtoJava Platform	119
7.2.2. ORCOS Platform	120
7.2.3. Case Studies Assessment	122
7.3. Case Studies	123
7.3.1. Unmanned Aerial Vehicle	123
7.3.1.1. Object-Oriented Version	124
7.3.1.2. Aspect-Oriented Version	126
7.3.1.3. Results	129
7.3.2. Industrial Packing System	132
7.3.2.1. Object-Oriented Version	133
7.3.2.2. Aspect-Oriented Version	135
7.3.2.3. Results	136
7.3.3. Wheelchair Automation	138
7.4. Final Remarks	140
8. Conclusions and Future Work	145
A. DERAf Detailed Description	149
A.1. Timing Package	149
A.2. Precision Package	151
A.3. Synchronization Package	153
A.4. Communication Package	154
A.5. TaskAllocation Package	155
A.6. Embedded Package	157
B. UML Models for the UAV Case Study	159
C. Mapping Rules	175
C.1. Application	175
C.2. Platform Configuration	188
C.3. Source Code Generated by GenERTiCA	191

Bibliography	195
List of Abbreviations	205
List of Publications	207

List of Figures

1.1. Hardware and software design gap	4
1.2. Using higher abstraction levels in embedded system design	4
1.3. Most important tools in embedded system design	7
2.1. Non-Functional Requirements Classification	16
2.2. Scattering: same code in multiple places	20
2.3. Quality model proposed by Sant'Anna et al.	24
2.4. Overall MARTE architecture	28
2.5. Stereotypes of Time profile	29
2.6. Stereotypes of GRM profile	29
3.1. Methodology for Multimedia Systems available in Metropolis	34
3.2. SEEP design flow	36
3.3. Aspects and join points modeling in AODM	38
3.4. Examples of Theme/UML models	39
3.5. CAM model represented as a class diagram	40
3.6. (a) Aspects modeling; (b) Advice modeling; (c) Pointcut specification	41
3.7. Specification using a <i>time aspect</i>	42
3.8. AO-related concepts modeling	43
4.1. Overview of the AMoDE-RT design approach	55
4.2. Overview of RT-Frida	56
4.3. RT-FRIDA templates for requirements specification	57
4.4. Other tools provided by RT-FRIDA	58
4.5. Adaptations proposed to SEED design flow	61
5.1. Graphical representation of system requirements	66
5.2. Specification of the static structure	67
5.3. Specification of the dynamic structure	68
5.4. Specification of objects deployment	69
5.5. Specification of the behavior in terms of actions performed by objects	70
5.6. Invalid behavior specification using sequence diagram	71
5.7. System behavior overview specified using activity diagram	75
5.8. Behavior of classes specified using state diagrams	77
5.9. Conceptual AO model	78

5.10. All aspects provided by DERAf	80
5.11. Aspects specification using ACOD	86
5.12. JPDD for structural elements selection	90
5.13. JPDD for behavioral elements selection	90
6.1. GenERTiCA mains features overview	94
6.2. DERCS meta-model: structural elements	95
6.3. DERCS meta-model: behavioral elements	97
6.4. DERCS meta-model: AO-related elements	98
6.5. Mapping rules XML organization	106
6.6. Mapping rules: <SourceOptions> and <PrimaryElements> branches	107
6.7. Mapping rules: <Attributes> node	108
6.8. Mapping rules: <SendMessage> node	108
6.9. Mapping rules: <i>PeriodicTiming</i> aspect implementation	110
6.10. Source code fragment with modifications performed by aspect adaptations	111
6.11. Platform configuration XML structure	113
6.12. GenERTiCA: application code generation flowchart	115
6.13. GenERTiCA: platform configuration generation flowchart	116
7.1. Reusability quality model	122
7.2. UAV movement control use case diagram	124
7.3. UAV movement control class diagram	125
7.4. Fragments of UAV movement control sequence diagram	125
7.5. UAV non-functional requirements handling: (A) ACOD, and (B) JPDD	128
7.6. Calculated metrics for the UAV control system	129
7.7. Comparison of UAV's <code>MovementController</code> classes	130
7.8. Industrial packing system use case diagram	132
7.9. Industrial packing system class diagram	133
7.10. Industrial packing system sequence diagram	134
7.11. Industrial packing system: reused elements in (A) ACOD, and (B) JPDD	135
7.12. Calculated metrics for the industrial packing system	136
7.13. Calculated metrics for the wheelchair movement control system	138
A.1. <i>Timing Package</i> : handling time non-functional requirements	151
A.2. <i>Precision Package</i> : handling precision non-functional requirements	152
A.3. <i>Synchronization Package</i> : handling synchronization non-functional requirements	153
A.4. <i>Communication Package</i> : handling communication non-functional requirements	154
A.5. <i>TaskAllocation Package</i> : handling tasks allocation non-functional requirements	156
A.6. <i>Embedded package</i> : handling embedded non-functional requirements	158
B.1. UAV movement control use case diagram	159
B.2. UAV movement control class diagram	160
B.3. Environment sensing	161
B.4. Main and back rotors sensing	162
B.5. Helicopter movement control	163

B.6. Helicopter piloting	164
B.7. Environment sensing subsystem initialization	164
B.8. Movement sensing subsystem initialization	165
B.9. Control subsystem initialization	165
B.10. Other behavior: WindSensorDriver.getWindSpeed()	166
B.11. Other behavior: WindSensorDriver.getWindDirection()	166
B.12. Other behavior: MovementController.processInfo()	167
B.13. UAV movement control deployment diagram	168
B.14. Aspects Crosscutting Overview Diagram	169
B.15. JPDD: selection of active objects class	170
B.16. JPDD: selection of shared passive objects	170
B.17. JPDD: selection of passive class attributes	170
B.18. JPDD: selection of passive class constructor	171
B.19. JPDD: selection of sub systems classes	171
B.20. JPDD: selection of sub systems constructor	171
B.21. JPDD: selection of selection of active objects construction actions	171
B.22. JPDD: selection of active objects constructor behavior	172
B.23. JPDD: selection of active objects constructor	172
B.24. JPDD: selection of messages whose name starts with "get"	172
B.25. JPDD: selection of messages whose name starts with "set"	173
B.26. JPDD: selection of messages whose name starts with "get"	173
B.27. JPDD: selection of messages whose name starts with "set"	173
B.28. JPDD: selection of passive objects construction action	174
B.29. JPDD: selection of active objects periodic behavior	174
B.30. JPDD: selection of message sending action to remote objects	174
B.31. JPDD: selection of sub systems constructor behavior	174

List of Tables

2.1. Metrics influence in quality attributes	23
5.1. Reserved words for actions specification	73
5.2. Naming pattern for elements selection in JPDD	89
5.3. Summary of MARTE stereotypes used in AMoDE-RT	92
6.1. UML-to-DERCS mapping table	99
6.2. UML-to-DERCS behavior elements relationships	103
7.1. UAV: Statistics of the UML model of AO version	130
7.2. UAV: Statistics of the generated source code	131
7.3. Industrial packing system: Statistics of the UML model of AO version	137
7.4. Industrial packing system: Statistics of the generated source code	137
7.5. Wheelchair: Statistics of the UML model of AO version	139
7.6. Wheelchair: Statistics of the generated source code	140
7.7. AO elements reused in the different case studies	142