

CONTENTS

PART I. PREFACE	1
1. Introduction	3
1.1 Motivation	4
1.2 Problem Statement	6
1.3 Objectives.....	6
1.4 Thesis Context.....	7
1.5 Means of Achieving the Proposed Objectives.....	8
1.6 Thesis Outline.....	9
2. Research Methodology	13
2.1 Framework for the CoSTest Design Science Project	14
2.2 Statement of Research Goals and the Design Problem	15
2.3 Research Questions.....	18
2.4 Engineering, Design and Empirical Cycles	19
2.5 Summary	29
PART II. PROBLEM INVESTIGATION	31
3. Theoretical Framework	33
3.1 Concepts of Requirements Engineering.....	35
3.1.1 Modelling Requirements based on Communicational Analysis..	37
3.2 Concepts of the Conceptual Schema Quality.....	37
3.2.1 Model Quality for Conceptual Schemas.....	38
3.2.2 Practices to improve the Quality of Conceptual Schemas	40

3.3 Concepts of the Model-driven Environment	44
3.3.1 MDA Definitions and Assumptions	44
3.3.2 Overview of the Metamodeling Architecture	45
3.3.3 UML CD-based Conceptual Schemas	46
3.3.4 Executable UML Conceptual Schema Under Test	47
3.3.5 Defect Types in UML-based Conceptual Schemas	49
3.4 Summary and Conclusions	52
4. Related Work of Conceptual Schema Validation	55
4.1 Dimensions of the Related Work	56
4.1.1 Domain	57
4.1.2 Quality Goal.....	57
4.1.3 Method.....	58
4.2 Generation, Selection, Prioritization and Execution of Test Cases....	61
4.2.1 Test Case Generation	61
4.2.2 Test Case Selection and Prioritization.....	62
4.2.2 Test Case Execution.....	63
4.3 Comparison of Related Works	66
4.3.1 Dimension-Based Comparison	66
4.3.2 Testing feature Comparison.....	70
4.4 Summary and Conclusions.....	75
PART III. TREATMENT DESIGN	77
5. Validation Framework for Conceptual Schemas.....	79
5.1 Framework Overview.....	80
5.1.1 Phases of the Methodological Framework	81
5.2 Test Analysis.....	82

5.2.1	Requirements Specification based on Communicational Analysis	82
5.2.2	Modelling Requirements based on Communicational Analysis.	83
5.3	Test Design	89
5.3.1	Test Data	89
5.4	Test Generation	90
5.4.1	Test Case Selection.....	90
5.4.2	Addressed Quality Goals	91
5.4.3	Test Types.....	91
5.4.4	Test Generation Criteria.....	92
5.4.5	Deriving test goals.....	93
5.4.6	Concrete and Executable Test Cases.....	94
5.5	Test Prioritization.....	107
5.6	Test Execution.....	112
5.6.1	Executable Conceptual Schema based on UML Class Diagram	112
5.6.2	Architecture and Testing Environment	121
5.6.3	Execution Trace	124
5.7	Test Evaluation.....	125
5.8	Overview of the CoSTest Testing Process	126
5.9	Summary and Conclusions.....	128
6.	Transformation Rules	131
6.1	An Overview of the MDT Process	132
6.2	Metamodels	133
6.2.1	Test Metamodel	134
6.2.2	Test Scenario Metamodel	137
6.2.3	Test Data Metamodel.....	138

6.3 Transformations.....	139
6.3.1 Transformation from Requirements Model to Test Model	140
6.3.2 Transformation from Test Model to Test Scenario Model	146
6.3.3 Transformation from Test Model to Test Data Model.....	148
6.3.4 Transformation from Test Scenario Model to Test Scenario Model with Abstract Test Cases	148
6.3.5 Transformation from Test Data Model and Abstract Test Cases to Executable and Concrete Test Cases.....	150
6.3.6 Transformation from UML CD-based CS to Executable CS under test	151
6.4 Summary and Conclusions.....	155
PART IV. TREATMENT VALIDATION.....	157
7. CoSTest Tool Implementation.....	159
7.1 General Overview and Architecture.....	159
7.2 The Test Model Manager.....	161
7.2.1 Presentation Manager	163
7.2.2 Test Model Generator.....	165
7.2.3 Graph and Tree Builder	165
7.2.4 Element Report Generator.....	165
7.3 The Test Scenario Model Manager	166
7.3.1 Presentation Manager	166
7.3.2 Test Model Generator.....	167
7.3.3 Tree Builder	167
7.3.4 Element Report Generator.....	167
7.4 The Test-Data Manager	167
7.4.1 Presentation Manager	169

7.4.2	Web-based Generator.....	170
7.4.3	Requirement-based Generator	170
7.4.4	Database Manager	170
7.5	The CSUT Processor	170
7.5.1	Presentation Manager	172
7.5.2	CSUT Manager.....	174
7.6	The Test Processor	174
7.6.1	Presentation Manager	174
7.6.2	Test Manager	178
7.7	The Mutant Generator.....	179
7.7.1	Presentation Manager	180
7.7.2	Mutant Manager.....	181
7.8	The Batch Testing Processor.....	182
7.8.1	Presentation Manager	182
7.8.2	Batch Test Manager	184
7.9	Summary and Conclusions.....	184
8.	Validation and Evaluation of the CoSTest Tool.....	187
8.1	Validating the Effectiveness of CoSTest CSUT Processor.....	190
8.1.1	Experimental Design	190
8.1.2	Conclusions and Changes on the CoSTest CSUT Processor.....	193
8.2	Validating the CoSTest Transformation Rules	194
8.2.1	Definition of Basic and Derived Metrics with Rule Scope.....	196
8.2.2	Definition of Basic and Derived Metrics with Transformation scope	197
8.2.3	Experimental Design	201
8.2.4	Results and Discussion	202

8.3 Evaluating the CoSTest Mutant Generator.....	211
8.3.1 Experiment No 1: Evaluating the Mutation Operators Implemented in CoSTest	212
8.3.2 Experiment No 2: Validating the Effectiveness and Efficiency of Mutant Generator of CoSTest	215
8.4 Validating of the Effectiveness of CoSTest' Test Cases.....	218
8.4.1 Experiment Goal and Questions	218
8.4.2 Variables.....	219
8.4.3 Metrics	219
8.4.4 Hypotheses.....	220
8.4.5 Experimental Material.....	221
8.4.6 Procedure.....	222
8.4.7 Analysis of Results.....	226
8.4.8 Discussion.....	234
8.4.9 Analysis of the Threats to the Validity of the Results	235
8.4.10 Conclusions and Changes to the Tool	236
8.5 Evaluating CoSTest User Perceptions	237
8.5.1 Experiment Research Goal.....	237
8.5.2 Research Methodology	238
8.5.3 Experiment Context: The everis' Study Case	238
8.5.4 Experiment Reseach Questions.....	239
8.5.5 Case Selection	239
8.5.6 Methods of Data Collection	240
8.5.7 Experimental Subjects.....	241
8.5.8 Instrumentation	242
8.5.9 Experimental Procedure.....	242

8.5.10	Pilot Test.....	243
8.5.11	Analysis of the Threats to Validity.....	246
8.5.12	Answers to Experiment Research Questions	247
8.5.13	Discussion.....	249
8.6	Summary and Conclusions.....	251
PART V.	FINAL DISCUSSION	255
9.	Final Discussion	257
9.1	Summary of the Contributions of this Thesis	257
9.2	Thesis Impact	261
9.2.1	Publications	261
9.2.2	Academic Project Participation	263
9.2.3	Research Stay	263
9.3	A Work that Opens New Research Lines	264
9.3.1	Domain	264
9.3.2	Quality Goal.....	264
9.3.3	Method.....	265
REFERENCES	267
APPENDICES	283
Appendix A	285
Mutation Operators for UML CD-based Conceptual Schemas		285
Appendix B	287
Case Study: The Incident Management System.....		287
B.1	Test Analysis.....	287
B.1.1	Event Description Templates	287
B.1.2	Events Diagram	302

B.2 Test Design	303
B.2.1 Test Model	304
B.2.2 Test Scenario Model.....	304
B.2.3 Test Data	310
B.3 Test Case Generation	310
B.4 Mutant Generation	316
B.5 Test Execution	319
B.5.1 Generation of the Executable Conceptual Schema Under Test	319
B.5.2 Generation of the Execution Trace	323
B.6 Test Evaluation.....	324
B.7 Conclusions	325
Appendix C	327
Supplementary Material on the Evaluation Study	327
C1. Characterization Form.....	327
C2. CoSTest Tool Installation Guide	331
C3. Guideline with Task Template for VideoClub Case	332
C4. User Acceptance Form	332

LIST OF TABLES

Table 3.1 Quality Goals based on 6C quality model from Mohaghehi et al. [8]	39
Table 3.2. Validation relevant methods for Conceptual Schemas.....	42
Table 3.3. Defect types in a UML-based model (excerpt taken from [39])	52
Table 4.1. Related approach comparison	68
Table 4.2. Testing features comparison	72
Table 5.1. Example of an Event Specification Template.....	87
Table 5.2. Test generation criteria for UML CD-based Conceptual Schema	93
Table 5.3. Mutation operators for CS FOM taken from [111]	110
Table 5.4. Relationship between fault and reported defect	126
Table 6.1. Transformation rules for generation of the Test Model.....	141
Table 6.2. Transformation rules for generation of the Test Cases	142
Table 6.3. Transformation rules for generation of the Precedence relations.....	142
Table 6.4. Transformation rules for generation of the test items Assertions	142
Table 6.5. Transformation rules for generation of the test items Triggers.....	142
Table 6.6. Transformation rules for generation of the test items Services.....	143
Table 6.7. Transformation rules for generation of the test items Links	143
Table 6.8. Transformation rules for generation of the test items Parameters	144
Table 6.9. Requirements Metamodel constructs used in this transformation	145
Table 6.10. Transformation rules for generation of the Test Scenario Model	147
Table 6.11. Transformation rules for generation of the Test Scenario	147
Table 8.1. Elements of the Subject Conceptual Schemas.....	191
Table 8.2. Basic metrics for Semantic and Syntactic Correctness of a Rule	196
Table 8.3. Derived metrics for Semantic and Syntactic Correctness of a Rule	197
Table 8.4. GQM for M2M transformation validation	201
Table 8.5. Elements of the CSs	202
Table 8.6. Elements of the requirements model included in the five example	203
Table 8.7. Elements of the Test Model generated for the five example	204
Table 8.8. Results of SyC_T1 and SeC_T1 for the five cases	206
Table 8.9. Elements of the Test Scenario Model generated for the five examples	208
Table 8.10. Results of SyC_T2 and SeC_T2 for the five cases	209
Table 8.11. Specification of hypotheses	220
Table 8.12. Elements of the Subject Conceptual Schemas.....	221

Table 8.13. Faults and Fault Types detected by Mutant type	227
Table 8.14. Shapiro-Wilk Normality Tests	228
Table 8.15. Mann-Whitney U Test for Rate of Fault Detection by Mutant Type ...	229
Table 8.16. Tests of Normality of Shapiro-Wilk.....	229
Table 8.17. Mann-Whitney U Test for Rate of Fault Type Detection ^a	230
Table 8.18. Mutation Score by Mutant type	231
Table 8.19. Mutation Score of CoSTest Test Suites for First Order Mutants.....	232
Table 8.20. Mutation Score of CoSTest Test Suites for High Order Mutants	232
Table 8.21. Shapiro-Wilk Normality Tests	233
Table 8.22. Mann-Whitney U Test for Mutation Score by Mutant Type ^a	233
Table 8.23. Detail of the Activities.....	243
Table 8.24. Specification of hypotheses.....	248
Table A.1. Mutation Operators defined for a UML CD-based CS taken from [110]	285
Table B.1. Communication Structure for TECH1	288
Table B.2. Communication Structure for USR1	288
Table B.3. Communication Structure for PLAN1	289
Table B.4. Communication Structure for INC1	290
Table B.5. Communication Structure for INC2	290
Table B.6. Communication Structure for INC3	291
Table B.7. Communication Structure for INC4	292
Table B.8. Communication Structure for INC5	292
Table B.9. Communication Structure for INC6	293
Table B.10. Communication Structure for INC7	293
Table B.11. Communication Structure for INC8	294
Table B.12. Communication Structure for INC9	295
Table B.13. Communication Structure for INC10	295
Table B.14. Communication Structure for INC11	296
Table B.15. Communication Structure for INC12	296
Table B.16. Communication Structure for INC13	297
Table B.17. Communication Structure for INC14	298
Table B.18. Communication Structure for INC15	298
Table B.19. Communication Structure for INC16	299
Table B.20. Communication Structure for INC17	300
Table B.21. Communication Structure for INC18	300
Table B.22. Communication Structure for INC19	301
Table B.23. Communication Structure for INC20	301
Table B.24. Communication Structure for INC21	302

Table B.25. Values for variables of test model for Incident Management	311
Table B.26. List of First Order Mutants generated for the case study	318
Table B.27. Testing results for the mutants of Table B.26	324

LIST OF FIGURES

Figure 1.1 Context of research work	8
Figure 2.1. Framework for design science of the CosTest project	15
Figure 2.2 Goal Structure of the design science research project for CoSTest	17
Figure 2.3. Design cycle for the CoSTest project (part 1)	20
Figure 2.4. Design cycle for the CoSTest project (part 2)	21
Figure 2.5. Design cycle for the CoSTest project (part 3)	22
Figure 3.1. Research areas involved in this work	34
Figure 3.2. Excerpt of the Metamodel of an UML Class Diagram [36]	47
Figure 3.3. Excerpt of UML-CD-based CS for Video Club case	48
Figure 3.4. Example of constraints for the Video Club system	49
Figure 3.5. Relationships among conceptual entities	50
Figure 4.1. Related Work dimensions	56
Figure 5.1. Overview of the validation Framework	81
Figure 5.2. Excerpt of a CA model for the Video Club case.	85
Figure 5.3. Examples of test goals generated for Video Club CS	94
Figure 5.4. Test Case Structure	95
Figure 5.5. UML class diagram for Video Club CS	96
Figure 5.6. Example of Assertions code developed in ALF library	98
Figure 5.7. An example of a test case	99
Figure 5.8. An example of a test case	100
Figure 5.9. VideoClub CS with pre, post-conditions and invariants	103
Figure 5.10. Example for validating pre-, post-conditions and invariants	103
Figure 5.11. Example of an invariant	104
Figure 5.12. Example of test case for asserting the non-occurrence of events	105
Figure 5.13. Example of test case validating a derivation rule	107
Figure 5.14. Selection process of the mutation operators	109
Figure 5.15. Excerpt of the Metamodel of an UML Class Diagram [36]	112
Figure 5.16. Textual definition for the package VideoClub by using ALF language	115
Figure 5.17. ATL transformation rule for UML package	115
Figure 5.18. Partial definition for the class VideoClub by using ALF language	116
Figure 5.19. Association and Aggregation using ALF language	116
Figure 5.20. Example of an Inheritance Relationship	117

Figure 5.21. Partial view of the ALF including an inheritance relation	117
Figure 5.22. Example of a constraint translated to ALF code	118
Figure 5.23. Example of a derived association in ALF code	119
Figure 5.24. Example of a derived association in a UML CS	119
Figure 5.25. An example of a class association	120
Figure 5.26 Overview to generate an executable CSUT	120
Figure 5.27. Testing environment to test Conceptual Schemas	122
Figure 5.28. Example of an execution trace for Video Club CS	124
Figure 5.29. Example of the test case	125
Figure 5.30. Example of execution trace	125
Figure 5.31. Extended UML class diagram for Video Club CS	127
Figure 5.32. Overview of the testing process	127
Figure 6.1. An overview of our MDT approach	132
Figure 6.2. Overview of the sequence of proposed transformations	133
Figure 6.3. Metamodels for the first transformation adapted from [118]	135
Figure 6.4. OCL Constraints for Test Metamodel	136
Figure 6.5. Test Scenario Metamodel adapted from [118]	137
Figure 6.6. Test Data Metamodel	138
Figure 6.7. Structure of T1 Transformation	141
Figure 6.8. Examples using graphical concrete syntax of (a) RM, (b) TM, and (c) modified TM	146
Figure 6.9. Example of the first rule of the ATL transformation CA2TM	146
Figure 6.10. Structure of T2 Transformation	148
Figure 6.11. Structure of T3 transformation	148
Figure 6.12. Partial Acceleo code of transformation	149
Figure 6.13. Test Scenario with abstract test cases	150
Figure 6.14. Example of a concrete and executable test case for VideoClub CS	151
Figure 6.15. Acceleo transformation rule for UML package	151
Figure 6.16. Partial definition for the class VideoClub by using ALF language	152
Figure 6.17. Association and Aggregation of Order example using ALF language	152
Figure 6.18. Partial view of the ALF unit including an inheritance relation	153
Figure 6.19. Example of a constraint translated to ALF code	154
Figure 6.20. Example of a derived association using ALF code	154
Figure 6.21. An example of class association	155
Figure 7.1. Screenshot of the CoSTest tool support	162
Figure 7.2. The CoSTest tool architecture	162
Figure 7.3. Test Model Manager design	163

Figure 7.4. Screenshot with a test configuration example of the CoSTest tool	164
Figure 7.5. Screenshot with a test model example of the CoSTest tool.....	165
Figure 7.6. Test Scenario Model Manager design	166
Figure 7.7. Screenshot of a test scenario model example in the CoSTest tool	168
Figure 7.8. Test Data Manager design	168
Figure 7.9. Screenshot for the data concretization in the CoSTest tool.....	171
Figure 7.10. CSUT Processor design	171
Figure 7.11. Screenshot for editing an executable CSUT in the CoSTest tool	172
Figure 7.12. Screenshot for showing the parser results in the CoSTest tool.....	173
Figure 7.13. Screenshot for showing the CSUT elements in the CoSTest tool	173
Figure 7.14. Test Processor design	175
Figure 7.15. Screenshot of the test configuration in the CoSTest tool.....	175
Figure 7.16. Screenshot of a test suite management example in the CoSTest tool.....	176
Figure 7.17. Screenshot of a test execution report in the CoSTest tool.....	177
Figure 7.18. Screenshot of the Summary Generation tab of the CoSTest tool	177
Figure 7.19. Screenshot of a log and coverage report in the CoSTest tool	178
Figure 7.20. The Mutation UML tool architecture	180
Figure 7.21. Application of five mutation operators for our CS example.....	182
Figure 7.22. Batch Testing Processor design	183
Figure 7.23. Screenshot for Batch Testing of the CoSTest tool	183
Figure 8.1. i-th iteration of the experiment applying the CoSTest tool.....	193
Figure 8.2. Example of the calculation of the metrics SyC_T1 and SeC_T1.....	199
Figure 8.3. Structure of T1 Transformation with the identified problems	207
Figure 8.4. i-th iteration of the experiment applying the CoSTest tool.....	213
Figure 8.5. i-th iteration of the experiment applying the CoSTest tool.....	216
Figure 8.6. Steps taken in experimental process.....	222
Figure 8.7. Application of five mutation operators on Video Club CS.....	224
Figure 8.8. Excerpt of a Constraint mutated by WCO8 operator	225
Figure 8.9. Box-plot for Rate of Fault Detection by Mutant Type	228
Figure 8.10. Box-plot for Rate of Fault Type Detection by Mutant Type	230
Figure 8.11. Box-plot for Mutation Score by Mutant Type	231
Figure 8.12. Example of an assertion conditional	235
Figure 8.13. Experimental Procedure	242
Figure B.1. Partial view of the message structure in the GREAT tool [137]	302
Figure B.2. Event Diagram using Communication Analysis	303
Figure B.3. Test Model for IM case study	304
Figure B.4. Test cases of the test scenario #2	316

Figure B.5. Conceptual Schema for Incident Management System	317
Figure B.6. ALF unit for PMO class.....	319
Figure B.7. ALF unit for Incident_external_company association	319
Figure B.8. ALF unit for EXTERNAL_COMPANY_ANALYSYS class	320
Figure B.9. ALF unit for incident_resource_allocation association	320
Figure B.10. ALF unit for technician_resource_allocation association.....	320
Figure B.11. ALF unit for INCIDENT class	321
Figure B.12. ALF unit for RESOLUTION_PLAN class	322
Figure B.13. ALF Unit for RESOURCE_ALLOCATION class	322
Figure B.14. ALF unit for TECHNICIAN class.....	322
Figure B.15. ALF unit for USER class	323
Figure B.16. ALF unit for user_incident association	323
Figure B.17. Example of Execution Trace for the MAS_2 mutant	323
Figure B.18. Defect report obtained in the testing process for MAS_2 CS	325
Figure B.19. Coverage report obtained in the testing process for MAS_2 CS	326
Figure C.1. Characterization form: Demographic data	327
Figure C.2. Characterization form: Experience (1)	328
Figure C.3. Characterization form: Experience (2)	329
Figure C.4. Characterization form: Experience (3)	330
Figure C.5. Guideline for VideoClub case (1)	333
Figure C.6. Guideline for VideoClub case (2)	334
Figure C.7. User Acceptance Form: Perceived Usefulness	334
Figure C.8. User Acceptance Form: Perceived Ease-of-Use.....	335