
Contents

INTRODUCTION	xiii
CHAPTER 1. FORMAL DESCRIPTION AND MODELING OF RISKS	1
Jean-Louis BOULANGER	
1.1. Introduction.	1
1.2. Standard process	2
1.2.1. Risks, undesirable events and accidents.	2
1.2.2. Usual process	7
1.2.3. Formal software processes for safety-critical systems	8
1.2.4. Formal methods for safety-critical systems.	9
1.2.5. Safety kernel.	9
1.3. Methodology	10
1.3.1. Presentation	10
1.3.2. Risk mastery process.	10
1.4. Case study	13
1.4.1. Rail transport system.	13
1.4.2. Presentation	13
1.4.3. Description of the environment.	14
1.4.4. Definition of side-on collision	16
1.4.5. Risk analysis.	17
1.5. Implementation	18
1.5.1. The B method	18
1.5.2. Implementation	19
1.5.3. Specification of the rail transport system and side-on collision.	19
1.6. Conclusion	22
1.7. Glossary	23
1.8. Bibliography	23

CHAPTER 2. AN INNOVATIVE APPROACH AND AN ADVENTURE IN RAIL SAFETY	27
Sylvain FIORONI	
2.1. Introduction.	27
2.2. Open Control of Train Interchangeable and Integrated System	30
2.3. Computerized interlocking systems	32
2.4. Conclusion	34
2.5. Glossary.	35
2.6. Bibliography	36
CHAPTER 3. USE OF FORMAL PROOF FOR CBTC (OCTYS)	37
Christophe TREMBLIN, Pierre LESOILLE and Omar REZZOUG	
3.1. Introduction	37
3.2. Presentation of the Open Control of Train Interchangeable and Integrated System CBTC	38
3.2.1. Open Control of Train Interchangeable and Integrated System	38
3.2.2. Purpose of CBTC	39
3.2.3. CBTC architectures.	40
3.3. Zone control equipment	42
3.3.1. Presentation	42
3.3.2. SCADE model	43
3.4. Implementation of the solution	46
3.5. Technical solution and implementation	49
3.5.1. Property definition	49
3.5.2. Two basic principles of property definition	50
3.5.3. Test topologies	52
3.5.4. Initial analyses	53
3.5.5. The property treatment process.	57
3.5.6. Non-regression	63
3.6. Results.	65
3.7. Possible improvements	66
3.8. Conclusion	67
3.9. Glossary.	68
3.10. Bibliography	69
CHAPTER 4. SAFETY DEMONSTRATION FOR A RAIL SIGNALING APPLICATION IN NOMINAL AND DEGRADED MODES USING FORMAL PROOF.	71
Jean-Marc MOTA, Evguenia DMITRIEVA, Amel MAMMAR, Paul CASPI, Salimeh BEHNIA, Nicolas BRETON and Pascal RAYMOND	
4.1. Introduction.	71

4.1.1. Context	73
4.2. Case description	74
4.2.1. Operational architecture of the PMI system	75
4.2.2. CIM subsystem	76
4.2.3. CIM program verification with and without proof	78
4.2.4. Scope of verification	80
4.3. Modeling the whole system	82
4.3.1. Application model	82
4.3.2. Safety properties	83
4.3.3. Environment model.	86
4.4. Formal proof suite	97
4.4.1. Modeling the system	97
4.4.2. Non-certified analysis chain	98
4.4.3. The certified analysis chain	99
4.4.4. Assessment of the proof suite.	100
4.5. Application	101
4.6. Results of our experience.	105
4.6.1. Environment modeling.	105
4.6.2. Proof vs. testing	107
4.6.3. Limitations.	108
4.7. Conclusion and prospects	108
4.8. Glossary	110
4.9. Bibliography	111
CHAPTER 5. FORMAL VERIFICATION OF DATA FOR PARAMETERIZED SYSTEMS	115
Mathieu CLABAUT	
5.1. Introduction.	115
5.1.1. Systemel.	115
5.1.2. Data verification	116
5.1.3. Parameterized systems.	117
5.2. Data in the development cycle.	118
5.2.1. Data and property identification	119
5.2.2. Modeling.	119
5.2.3. Property validation	120
5.2.4. Data production	120
5.2.5. Property verification using data	120
5.2.6. Data integration	120
5.3. Data verification	122
5.3.1. Manual verification.	122
5.3.2. Algorithmic verification.	122
5.3.3. Formal verification	123
5.4. Example of implementation	130

5.4.1. Presentation	130
5.4.2. Property modeling	131
5.4.3. Data extraction	132
5.4.4. Tools	133
5.5. SSIL4 process	133
5.6. Conclusion	134
5.7. Glossary	134
5.8. Bibliography	134
CHAPTER 6. ERTMS MODELING USING EFS	137
Laurent FERIAER, Svitlana LUKICHEVA and Stanislas PINTE	
6.1. The context	137
6.2. EFS description	139
6.2.1. Characteristics	139
6.2.2. Modeling process	147
6.2.3. Interpretation or code generation	148
6.3. Braking curves modeling	149
6.3.1. Computing braking curves	149
6.3.2. Permitted speed and speed limitation curves	151
6.3.3. Deceleration factors	155
6.3.4. Deceleration curves	156
6.3.5. Target supervision limits	159
6.3.6. Symbolic computation	159
6.3.7. Braking curves verification	160
6.4. Conclusion	161
6.5. Further works	162
6.6. Bibliography	163
CHAPTER 7. THE USE OF A “MODEL-BASED DESIGN” APPROACH ON AN ERTMS LEVEL 2 GROUND SYSTEM	165
Stéphane CALLET, Saïd EL FASSI, Hervé FEDELER, Damien LEDOUX and Thierry NAVARRO	
7.1. Introduction	166
7.2. Modeling an ERTMS Level 2 RBC	168
7.2.1. Overall architecture of the model	170
7.2.2. Functional separation	171
7.3. Generation of the configuration	175
7.3.1. Development of a track plan	175
7.3.2. Writing the configuration	176
7.3.3. Translation of the configurations to the MATLAB/ Simulink format	177
7.4. Validating the model	177
7.4.1. Development of a language in which to write the scenarios	178

7.4.2. Writing the scenarios	178
7.4.3. Verification of the scenarios	179
7.4.4. Animation of the model	180
7.4.5. Addition of coherence properties for the scenarios	183
7.4.6. Coverage of the model	183
7.5. Proof of the model	184
7.5.1. Expressing the properties	184
7.5.2. Proof of the properties	186
7.6. Report generation	186
7.6.1. Documentation of the model	187
7.6.2. Automatic generation of the report	188
7.7. Principal modeling choices	189
7.8. Conclusion	190
CHAPTER 8. APPLYING ABSTRACT INTERPRETATION TO DEMONSTRATE FUNCTIONAL SAFETY	191
Daniel KASTNER	
8.1. Introduction	191
8.2. Abstract interpretation	193
8.3. Non-functional correctness	194
8.3.1. Stack usage	194
8.3.2. Worst-case execution time	195
8.3.3. Run-time errors	196
8.4. Why testing is not enough	197
8.5. Verifying non-functional program properties by abstract Interpretation	199
8.5.1. WCET and stack usage analysis	200
8.5.2. Run-time error analysis	206
8.6. The safety standards perspective	210
8.6.1. DO-178B	210
8.6.2. DO-178C / DO-333	211
8.6.3. ISO-26262	214
8.6.4. IEC-61508	216
8.6.5. CENELEC EN-50128	217
8.6.6. Regulations for medical software	218
8.7. Providing confidence – tool qualification and more	219
8.7.1. Tool qualification	220
8.8. Integration in the development process	222
8.9. Practical experience	223
8.10. Summary	224
8.11. Appendix A: Non-functional verification objectives of DO-178C	225
8.12. Appendix B: Non-functional requirements of ISO-26262	225
8.13. Bibliography	229

CHAPTER 9. BCARE: AUTOMATIC RULE CHECKING FOR USE WITH SIEMENS.	235
Karim BERKANI, Melanie JACQUEL and Eric LE LAY	
9.1. Overview	235
9.2. Introduction.	235
9.3. Description of the validation process for added rules	238
9.3.1. The proof activity.	238
9.3.2. Rules	238
9.3.3. Rule validation	241
9.4. The BCARE validation tool	243
9.4.1. BCARE: an environment for rule validation	243
9.4.2. Check_blvar	244
9.4.3. Chaîne_verif.	253
9.5. Proof of the BCARE validation lemmas	260
9.5.1. Automatic proof using L_{tac}	261
9.5.2. Evaluation and tests	269
9.6. Conclusion	271
9.7. Acknowledgments.	272
9.8. Bibliography	272
CHAPTER 10. VALIDATION OF RAILWAY SECURITY AUTOMATISMS BASED ON PETRI NETWORKS.	275
Marc ANTONI	
10.1. Introduction	275
10.1.1. Note to the reader	275
10.1.2. Summary	275
10.2. Issues involved	277
10.2.1. Introduction	277
10.2.2. An industry context: railways	278
10.2.3. Determinism versus probabilism for the safe management of critical computerized systems.	279
10.2.4. A key element: formal validation.	300
10.3. Railway safety: basic concepts.	301
10.3.1. Control of safety properties and postulates	302
10.3.2. Aspects that should be considered for carrying out a formal validation.	308
10.4. Formal validation method for critical computerized systems.	313
10.4.1. The interlocking module for modern signal boxes.	313
10.4.2. AEFD specification language	316
10.4.3. Method for proof by assertions	325
10.5. Application to a real signal box	337
10.5.1. Introduction	337

10.5.2. Presentation of the track plan and the signal box program	337
10.5.3. Safety properties and postulates	338
10.5.4. Exploration and formal validation of the application functional software of the signal box	339
10.6. Conclusion	340
10.6.1. From a general point of view	340
10.6.2. The use of the method	342
10.6.3. From a research point of view.	344
10.6.4. From the railway industry perspective.	344
10.6.5. The model and its implementation	346
10.7. Glossary	347
10.8. Bibliography	348
CHAPTER 11. COMBINATION OF FORMAL METHODS FOR CREATING A CRITICAL APPLICATION	353
Philippe COUPOUX	
11.1. Introduction	353
11.1.1. A history of the use of formal method in AREVA TA	354
11.2. Use of SCADE 6	355
11.2.1. Reasons for the choice of SCADE 6	355
11.2.2. SCADE 6 in the context of the lifecycle of a software package.	356
11.2.3. Organization and development rules of a SCADE 6 model	361
11.2.4. Usage summary SCADE 6.	363
11.3. Implementation of the B method	367
11.3.1. The reasons for choosing the B method for the ZC application.	367
11.3.2. Positioning the B method in the V cycle of the ZC software	368
11.3.3. B Method Usage Summary	372
11.4. Conclusion	375
11.5. Appendices	376
11.5.1. Appendix 1: SOFTWARE architecture on DRACK platform	376
11.5.2. Appendix 2: detailed description of the approach chosen for the B method	379
11.5.3. General design of the ZC security application	380
11.5.4. Detailed design ZC security application.	383
11.5.5. Proof of the formal model	384
11.5.6. Coding of the ZC security application.	386
11.5.7. Integration of the ZC security application.	387
11.5.8. Tests of the ZC security application	388

11.6 Glossary	388
11.7. Bibliography	389
CHAPTER 12. MATHEMATICAL PROOFS FOR THE NEW YORK SUBWAY	391
Denis SABATIER	
12.1. The CBTC of the New York subway Line 7 and the system proof	391
12.2. Formal proof of the system	392
12.2.1. Presentation	392
12.2.2. Benefits	393
12.2.3. Obtaining the first demonstration: organization and communication	397
12.2.4. A method based on exchange	398
12.3. An early insight into the obtained proof	400
12.3.1. The global proof.	400
12.3.2. Proof that localization has been correctly achieved	403
12.3.3. Proof of correct braking	404
12.4. Feedback based on experience	406
CONCLUSION	409
GLOSSARY	449
LIST OF AUTHORS	455
INDEX	457