

1. Record Nr.	UNINA9910784353403321
Autore	Drusinsky Doron
Titolo	Modeling and verification using UML statecharts [[electronic resource] ] : a working guide to reactive system design, runtime monitoring, and execution-based model checking // Doron Drusinsky
Pubbl/distr/stampa	Burlington, MA, : Newnes, c2006
ISBN	1-281-05218-3 9786611052188 0-08-048147-7
Edizione	[1st edition]
Descrizione fisica	1 online resource (321 p.)
Disciplina	005.1/17
Soggetti	UML (Computer science) Formal methods (Computer science) Computer software - Development
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references (p. 295-301) and index.
Nota di contenuto	Front matter; Dedication; Contents; Preface; Acknowledgments; What's on the CD-ROM?; Chapter 1: Formal Requirements and Finite Automata Overview; 1.1. Terms; 1.2. Finite Automata: The Basics; 1.3. Regular Expressions; 1.4. Deterministic Finite Automata and Finite State Diagrams; 1.5. Nondeterministic Finite Automata; 1.6. Other Forms of FA; 1.7. FA Conversions and Lower Bounds; 1.8. Operations on Regular Requirements; 1.9. Succinctness of FA; 1.10. Specifications as Zipped Requirements; 1.11. Finite State Machines; 1.12. Normal Form and Minimization of FA and FSMs; Chapter 2: Statecharts 2.1. Transformational vs. Reactive Components 2.2. Statecharts in Brief; 2.3. A Related Tool; 2.4. Basic Elements of Statecharts; 2.5. Code Generation and Scheduling; 2.6. Event-Driven Statecharts, Procedural Statecharts, and Mixed Flowcharts and Statecharts; 2.7. Flowcharts inside Statecharts: Workflow within Event-Driven Controllers; 2.8. Nonstandard Elements of Statecharts; 2.9. Passing Data to a Statechart Controller; 2.10. JUnit Testing of Statechart Objects; 2.11. Statecharts vs. Message Sequence Charts and Scenarios; 2.12. Probabilistic Statecharts

Chapter 3: Academic Specification Languages for Reactive Systems3.1. Natural Language Specifications; 3.2. Using Specification Languages for Runtime Monitoring; 3.3. Linear-time Temporal Logic (LTL); 3.4. Other Formal Specification Languages for Reactive Systems; Chapter 4: Using Statechart Assertions for Formal Specification; 4.1. Statechart Specification Assertions; 4.2. Nondeterministic Statechart Assertions; 4.3. Operations on Assertions; 4.4. Quantified Distributed Assertions; 4.5. Runtime Recovery for Assertion Violations 4.6. The Language Dog-Fight: Statechart Assertions vs. LTL and ERE4.7. Succinctness of Pure Statechart Assertions; 4.8. Temporal Assertions vs. JML and Java Assertions; 4.9. Commonly Used Assertions; Chapter 5: Creating and Using Temporal Statechart Assertions; 5.1. Motivation, or Why Use Temporal Assertions?; 5.2. Applying Assertions: Three Uses; 5.3. Writing Assertions; 5.4. Runtime Execution Monitoring- Runtime Verification; 5.5. Runtime Recovery from Requirement Violations; 5.6. Automatic Test Generation; 5.7. Execution-Based Model Checking Chapter 6: Application of Formal Specifications and Runtime Monitoring to the Ballistic Missile Defense ProjectIntroduction; 6.1. Abstract; 6.2. Context; 6.3. Formal Specification and Verification Approach; 6.4. Overall Value; 6.5. Challenges; Appendix: TLCharts: Syntax and Semantics; A.1. About TLCharts; A.2. Syntax; A.3. Semantics without Temporal Conditions; A.4. Semantics with Temporal Conditions; A.5. TLCharts with Overlapping States; Notes; About the Author; Index

---

## Sommario/riassunto

As systems being developed by industry and government grow larger and more complex, the need for superior specification and verification approaches and tools becomes increasingly vital. The developer and customer must have complete confidence that the design produced is correct, and that it meets forma development and verification standards. In this text, UML expert author Dr. Doron Drusinsky compiles all the latest information on the application of UML (Universal Modeling Language) statecharts, temporal logic, automata, and other advanced tools for run-time monitoring and verification. This i

---