1. Record Nr. UNINA9910138854403321

Titolo	Communicating embedded systems [[electronic resource]] : software and design : formal methods / / edited by Claude Jard, Olivier H. Roux
Pubbl/distr/stampa	London, : ISTE Hoboken, N.J., : Wiley, 2010
ISBN	1-118-55818-9 1-118-60009-6 1-118-60012-6 1-299-18745-5
Edizione	[1st edition]
Descrizione fisica	1 online resource (275 p.)
Collana	ISTE
Altri autori (Persone)	JardClaude RouxOlivier H
Disciplina	621.39/2
Soggetti	Embedded computer systems - Programming Embedded computer systems - Design and construction Computer software - Development Formal methods (Computer science)
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 and index.
Nota di contenuto	Cover; Communicating Embedded Systems; Title Page; Copyright Page; Table of Contents; Preface; Chapter 1. Models for Real-Time Embedded Systems; 1.1. Introduction; 1.1.1. Model-checking and control problems; 1.1.2. Timed models; 1.2. Notations, languages and timed transition systems; 1.3. Timed models; 1.3.1. Timed Automata; 1.3.2. Time Petri nets; 1.3.2.1. T-time Petri nets; 1.3.2.2. Timed-arc petri nets; 1.3.3. Compared expressiveness of several classes of timed models; 1.3.3.1. Bisimulation and expressiveness of timed models; 1.3.3.2. Compared expressiveness of different classes of TPN 1.3.3.3. Compared expressiveness of TA, TPN, and TAPN1.4. Models with stopwatches; 1.4.1. Formal models for scheduling aspects; 1.4.1.1. Automata and scheduling; 1.4.1.2. Time Petri nets and scheduling; 1.4.2. Stopwatch automata; 1.4.3. Scheduling time Petri nets; 1.4.4. Decidability results for stopwatch models; 1.5. Conclusion; 1.6. Bibliography; Chapter 2. Timed Model-Checking; 2.1. Introduction;

2.2. Timed models; 2.2.1. Timed transition system; 2.2.2. Timed automata; 2.2.3. Other models; 2.3. Timed logics; 2.3.1. Temporal logics CTL and LTL; 2.3.2. Timed extensions; 2.3.2.1. Timed CTL 2.3.2.2. Timed LTL2.4. Timed model-checking; 2.4.1. Model-checking LTL and CTL (untimed case); 2.4.2. Region automaton; 2.4.3. Model-checking TCTL; 2.4.4. Model-checking MTL; 2.4.5. Efficient model-checking; 2.4.6. Model-checking in practice; 2.5. Conclusion; 2.6. Bibliography; Chapter 3. Control of Timed Systems; 3.1. Introduction; 3.1.1. Verification of timed systems; 3.1.2. The controller synthesis problem; 3.1.3. From control to game; 3.1.4. Game objectives; 3.1.5. Varieties of untimed games; 3.2. Timed games; 3.2.1. Timed game automata; 3.2.2. Strategies and course of the game
3.2.2.1. The course of a timed game3.2.2.2. Strategies; 3.3. Computation of winning states and strategies; 3.3.1. Controllable predecessors; 3.3.2. Symbolic operators; 3.3.3. Symbolic computation of winning states; 3.3.4. Synthesis of winning strategies; 3.4. Zeno strategies; 3.5. Implementability; 3.5.1. Hybrid automata; 3.5.2. On the existence of non-implementable continuous controllers; 3.5.3. Recent results and open problems; 3.6. Specification of control objectives; 3.7. Optimal control; 3.7.1. TA with costs; 3.7.2. Optimal cost in timed games; 3.7.3. Computation of the optimal cost
3.7.4. Recent results and open problems3.8. Efficient algorithms for controller synthesis; 3.8.1. On-the-fly algorithms; 3.8.2. Recent results and open problems; 3.9. Partial observation; 3.10. Changing game rules...; 3.11. Bibliography; Chapter 4. Fault Diagnosis of Timed Systems; 4.1. Introduction; 4.2. Notations; 4.2.1. Timed words and timed languages; 4.2.2. Timed automata; 4.2.3. Region graph of a TA; 4.2.4. Product of TA; 4.2.5. Timed automata with faults; 4.3. Fault diagnosis problems; 4.3.1. Diagnoser; 4.3.2. The problems; 4.3.3. Necessary and sufficient condition for diagnosability
4.4. Fault diagnosis for discrete event systems

Sommario/riassunto

The increased complexity of embedded systems coupled with quick design cycles to accommodate faster time-to-market requires increased system design productivity that involves both model-based design and tool-supported methodologies. Formal methods are mathematically-based techniques and provide a clean framework in which to express requirements and models of the systems, taking into account discrete, stochastic and continuous (timed or hybrid) parameters with increasingly efficient tools. This book deals with these formal methods applied to communicating embedded systems by presenting the