1. Record Nr. UNISALENTO991004329238407536

Autore	Arnette, Marie-Lys
Titolo	Regressus ad uterum : La mort comme une nouvelle naissance dans les grands textes funÃ©raires de lâÃgypte pharaonique (5.-20. dynastie) / Marie-Lys Arnette
Pubbl/distr/stampa	Le Caire : Institut franÃ§ais d'archÃ©ologie orientale, 2020
Titolo uniforme	Regressus ad uterum 4165264
ISBN	9782724707434
Descrizione fisica	1XI, 451 p. : ill. ; 29 cm
Collana	BibliothÃ¨que d'Ã©tude ; 175
Disciplina	306.90932
Soggetti	Riti e cerimonie funebri egiziane - Egitto Morte - Aspetto religioso - Egitto
Lingua di pubblicazione	Francese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Bibliogr. p. 405-431

2. Record Nr. UNINA9910144336703321

Titolo	Formal Methods and Software Engineering : 6th International Conference on Formal Engineering Methods, ICFEM 2004, Seattle, WA, USA, November 8-12, 2004, Proceedings / / edited by Jim Davies, Wolfram Schulte, Mike Barnett
Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 2004
ISBN	3-540-30482-7
Edizione	[1st ed. 2004.]
Descrizione fisica	1 online resource (IX, 500 p.)
Collana	Lecture Notes in Computer Science, , 0302-9743 ; ; 3308
Disciplina	005.131
Soggetti	Software engineering Computer logic Programming languages (Electronic computers) Software Engineering/Programming and Operating Systems Software Engineering Logics and Meanings of Programs Programming Languages, Compilers, Interpreters
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di bibliografia	Includes bibliographical references at the end of each chapters and index.
Nota di contenuto	Tutorials -- Model-Based Development: Combining Engineering Approaches and Formal Techniques -- Tutorial on the RAISE Language, Method and Tools -- Model-Based Testing with Spec# -- Formal Engineering for Industrial Software Development – An Introduction to the SOFL Specification Language and Method -- Tutorial: Software Model Checking -- Invited Talks -- Engineering Quality Software -- When Can Formal Methods Make a Real Difference? -- On the Adoption of Formal Methods by Industry: The ACL2 Experience -- A CLP Approach to Modelling Systems -- Full Papers -- Multi-prover Verification of C Programs -- Memory-Model-Sensitive Data Race Analysis -- Formal Models for Web Navigations with Session Control and Browser Cache -- Managing Verification Activities Using SVM -- A General Model for Reachability Testing of Concurrent Programs -- A

Knowledge Based Analysis of Cache Coherence -- A Propositional Logic-Based Method for Verification of Feature Models -- Deriving Probabilistic Semantics Via the ‘Weakest Completion’ -- CSP Representation of Game Semantics for Second-Order Idealized Algol -- An Equational Calculus for Alloy -- Guiding Spin Simulation -- Linear Inequality LTL (iLTL): A Model Checker for Discrete Time Markov Chains -- Software Model Checking Using Linear Constraints -- Counterexample Guided Abstraction Refinement Via Program Execution -- Faster Analysis of Formal Specifications -- Bridging Refinement of Interface Automata to Forward Simulation of I/O Automata -- Learning to Verify Safety Properties -- Automatic Extraction of Object-Oriented Observer Abstractions from Unit-Test Executions -- A Specification-Based Approach to Testing Polymorphic Attributes -- From Circus to JCSP -- An Approach to Preserve Protocol Consistency and Executability Across Updates -- A Formal Monitoring-Based Framework for Software Development and Analysis -- Verifying a File System Implementation -- Verifying the On-line Help System of SIEMENS Magnetic Resonance Tomographs -- Implementing Dynamic Aggregations of Abstract Machines in the B Method -- Formal Proof from UML Models -- Interactive Verification of UML State Machines -- Refinement of Actions for Real-Time Concurrent Systems with Causal Ambiguity -- From Durational Specifications to TLA Designs of Timed Automata -- Timed Patterns: TCOZ to Timed Automata.

Sommario/riassunto

Formal engineering methods are changing the way that software systems are - veloped.Withlanguageandtoolsupport, theyarebeingusedforautomaticcode generation, and for the automatic abstraction and checking of implementations. In the future, they will be used at every stage of development: requirements, speci?cation, design, implementation, testing, and documentation. The ICFEM series of conferences aims to bring together those interested in the application of formal engineering methods to computer systems. Researchers and practitioners, from industry, academia, and government, are encouraged to attend, andtohelpadvancethestateoftheart.Authorsarestronglyencouraged to make their ideas as accessible as possible, and there is a clear emphasis upon work that promises to bring practical, tangible bene?t: reports of case studies should have a conceptual message, theory papers should have a clear link to application, and papers describing tools should have an account of results. ICFEM 2004 was the sixth conference in the series, and the ?rst to be held in North America. Previous conferences were held in Singapore, China, UK, A- tralia, and Japan. The Programme Committee received 110 papers and selected 30forpresentation.The?nalversionsofthosepapersareincludedhere, together with 2-page abstracts for the 5 accepted tutorials, and shorter abstracts for the 4 invited talks.