Igor Melatti's *Curriculum Vitae*

Univ. di Roma “La Sapienza”, Dip. di Informatica, via Salaria 113, 00198 Roma
melatti@di.uniroma1.it
Tel: +39 06 4991 8438

Working at the University

- From 12/30/2010 he is Assistant Professor (i.e., *Ricercatore Universitario*) at the Computer Science Department of the University of Rome “La Sapienza” (Rome, Italy)

- From 05/01/2010 to 12/29/2010 he held a Post-Doc position at the Department of Computer Science of the University of Rome “La Sapienza” (Rome, Italy), supervisor Prof. Enrico Tronci.

- From 02/01/2006 to 01/31/2010 he held a Post-Doc position at the Department of Computer Science of the University of Rome “La Sapienza” (Rome, Italy), supervisor Prof. Enrico Tronci.

- From 07/11/2005 to 12/31/2005 (and from 07/15/2006 to 09/15/2006) he had a “Post Doctoral Research Associate” position at the School of Computing of the University of Utah (Salt Lake City, UT, USA), supervisor Prof. Ganesh Gopalakrishnan.

    - The funding was provided by SRC (Semiconductor Research Corporation), a company consortium including also IBM, INTEL and Texas Instruments.

Education

- 06/06/2005: PhD in Informatica ed Applicazioni (Computer Science and Applications), at the University of L'Aquila. The PhD thesis has the following title: “Explicit Algorithms for Probabilistic Model Checking”, advisor Prof. Benedetto Intrigila. The PhD program lasted three years, with a scholarship of the Italian Ministry of the Instruction.

- 04/10/2001: degree in Informatics with the maximum score “cum laude”, at the University of L'Aquila. The thesis title was “Uso di SPIN in un approccio probabilistico alla verifica automatica di sistemi concorrenti” (“Using a probabilistic approach in the automatic verification of concurrent systems with SPIN”), and the relator was the Prof. Enrico Tronci

- August 1997: degree in Piano with the highest score.

- August 1995: *diploma di maturità classica* (degree of high school teaching also ancient Greek and Latin) with the highest score.

Publications

Journal

(j1). G. Della Penna, D. Magazzeni, B. Intrigila, I. Melatti, E. Tronci, M. Venturini Zilli, E. Ciancamerla, M. Minichino, A. Tofani. Automatic Verification of Hybrid System Controllers with the CMurphi Verifier. *IJDECS*, 1(1): 69-82, 2011, Serial Publications

(j2). I. Melatti, R. Palmer, G. Sawaya, Y. Yang, R. M. Kirby, and G. Gopalakrishnan. Parallel *and* Distributed Model Checking in Eddy. *STTT*, 11(1): 13–25, 2009, Springer

(j3). G. Della Penna, B. Intrigila, D. Magazzeni, I. Melatti, A. Tofani, E. Tronci. Automated Generation Of Optimal Controllers Through Model Checking Techniques. *Informatics in Control, Automation and Robotics: Selected Papers from the International Conference on Informatics in Control, Automation and Robotics 2006*, pp. 107–122, 2008, Springer Publishing Company, Incorporated (book chapter)

(j4). B. Intrigila, I. Melatti, A. Tofani, and G. Macchiarelli. Computational Models of Myocardial Endomysial Collagen Arrangement. *Computer Methods and Programs in Biomedicine*, 86(3):232–244, 2007, Elsevier North-Holland, Inc.

(j5). G. Della Penna, B. Intrigila, I. Melatti, E. Tronci, and M. Venturini Zilli. Finite Horizon Analysis of Markov Chains with the Murphi Verifier. *STTT*, 8(4):397 – 410, 2006, Springer

(j6). G. Della Penna, A. Di Marco, B. Intrigila, I. Melatti, A. Pierantonio Interoperability Mapping from XML Schemas to ER Diagrams. *Data & Knowledge Engineering*, 59:166 – 188, 2006, Elsevier

(j7). G. Della Penna, B. Intrigila, I. Melatti, E. Tronci, and M. Venturini Zilli. Exploiting Transition Locality in Automatic Verification of Finite State Concurrent Systems. *STTT*, 6(4):320–341, 2004, Springer

Proceedings of International Conferences

(c1). F. Mari, I. Melatti, I. Salvo, and E. Tronci, From Boolean Relations to Control Software. In Proc. of the Sixth International Conference on Software Engineering Advances *ICSEA '11*, ISBN: 978-1-61208-165-6

(c2). F. Cavaliere, F. Mari, I. Melatti, G. Minei, I. Salvo, E. Tronci, G. Verzino, and Y. Yushtein, Model Checking Satellite Operational Procedures. In Proc. of the Eurospace DAta Systems In Aerospace Conference *DASIA '11*, to appear

(c3). F. Mari, I. Melatti, I. Salvo, E. Tronci. Synthesis of Quantized Feedback Control Software for Discrete Time Linear Hybrid Systems. In Proc. of the 22nd International Conference on Computer Aided Verification *CAV '10*, volume 6174 of *Lecture Notes in Computer Science*, pages 180–195. Springer, 2010.

(c4). S. Mazzini, S. Puri, F. Mari, I. Melatti, E. Tronci, Formal Verification at System Level. In ESA SP-669, Proc. of the Eurospace DAta Systems In Aerospace Conference *DASIA '09*, ESA Proceedings ESA SP-669

(c5). A. Bobbio, E. Ciancamerla, S. Di Blasi, A. Iacomini, F. Mari, I. Melatti, M. Minichino, A. Scarlatti, E. Tronci, R. Terruggia, E. Zendri. Risk analysis of SCADA systems interconnecting Power Grids and Telco Networks via heterogeneous models and tools. In Proc. of the 4th International Conference on Risks and Security of Internet and Systems *CRISIS '09*, pages 90–97, IEEE Proceedings.

(c6). F. Mari, I. Melatti, I. Salvo, E. Tronci, L. Alvisi, A. Clement and H. Li. Model Checking Coalition Nash Equilibria in MAD Distributed Systems. In Rachid Guerraoui and Franck Petit, editors, *Stabilization, Safety, and Security of Distributed Systems, 11th International Symposium, SSS 2009, Lyon, France, November 3-6, 2009. Proceedings*, volume 5873 of *Lecture Notes in Computer Science*, pages 531–546. Springer, 2009.

(c7). F. Mari, I. Melatti, I. Salvo, E. Tronci, L. Alvisi, A. Clement and H. Li. Model Checking Nash Equilibria in MAD Distributed Systems. In Proc. of the 8th Conference on Formal Methods in Computer Aided Design *FMCAD '08* (IEEE Computer Society).

(c8). F. Brizzolari, G. Della Penna, I. Melatti, E. Tronci. Disk Based Software Verification via Bounded Model Checking. In Proc. of the 14th Asia-Pacific Software Engineering Conference *APSEC '07* (IEEE Computer Society).

(c9). G. Della Penna, D. Magazzeni, A. Tofani, B. Intrigila, I. Melatti, E. Tronci. Automatic Synthesis of Robust Numerical Controllers. In Proc. of the 3rd International Conference on Autonomic and Autonomous Systems *ICAS '07* (IEEE Computer Society).

(c10). G. Della Penna, B. Intrigila, D. Magazzeni, I. Melatti, A. Tofani, E. Tronci. Automatic Generation Of Optimal Controllers Through Model Checking Techniques. Proceedings of 3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2006)

(c11). G. Della Penna, B. Intrigila, I. Melatti, M. Pecorari, A. Tofani, E. Tronci. A Case Study on Automated Generation of Integration Tests. Proceedings of Forum on specification & Design Languages (FDL 2006)

(c12). I. Melatti, R. Palmer, G. Sawaya, Y. Yang, R. M. Kirby, and G. Gopalakrishnan. Parallel *and* Distributed Model Checking in Eddy. A. Valmari, editor, *Model Checking Software, 13th International SPIN Workshop, Vienna, Austria, March 30 – April 1, 2006, Proceedings*, volume 3925 of *Lecture Notes in Computer Science*. Springer, 2005.

(c13). B. Intrigila, D. Magazzeni, I. Melatti, A. Tofani, E. Tronci. A Model Checking Technique for the Verification of Fuzzy Control Systems. IEEE proceedings of the *International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA 2005).*

(c14). G. Della Penna, B. Intrigila, I. Melatti, and E. Tronci. Exploiting Hub States in Automatic Verification. D.A. Peled and Y.-K. Tsay, editors, *Automated Technology for Verification* *and Analysis: Third International Symposium, ATVA 2005, Taipei, Taiwan, October 4-7, 2005, Proceedings*, volume 3707 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2005.

(c15). B. Intrigila, G. Macchiarelli, I. Melatti, A. Tofani. Computational Models of the Micro Architecture of the Cardiac Endomysial Collagen. *IFMBE Proceedings EMBEC'05 "3rd European Medical & Biological Engineering Conference, IFMBE European Conference on Biomedical Engineering"*, Vol. 11, 2005, Prague, Czech Republic, CD

(c16). G. Della Penna, B. Intrigila, I. Melatti, E. Tronci, and M. Venturini Zilli. Bounded Probabilistic Model Checking with the Murphi Verifier. In Carlo Blundo and Cosimo Laneve, editors, *Formal Methods in Computer-Aided Design, 5th International Conference, FMCAD 2004, Austin, TX, USA, November 14-17, 2004, Proceedings*, volume 3312 of *Lecture Notes in Computer Science*, pages 214–229. Springer, 2004.

(c17). G. Della Penna, B. Intrigila, I. Melatti, E. Tronci, and M. Venturini Zilli. Finite Horizon Analysis of Markov Chains with the Murphi Verifier. In Daniel Geist and Enrico Tronci, editors, *Correct Hardware Design and Verification Methods, 12th IFIP WG 10.5 Advanced Research Working Conference, CHARME 2003, L'Aquila, Italy, October 21-24, 2003, Proceedings*, volume 2860 of *Lecture Notes in Computer Science*, pages 394–409. Springer, 2003.

(c18). G. Della Penna, B. Intrigila, I. Melatti, E. Tronci, and M. Venturini Zilli. Integrating Ram and Disk Based Verification within the Murphi Verifier. In Daniel Geist and Enrico Tronci, editors, *Correct Hardware Design and Verification Methods, 12th IFIP WG 10.5 Advanced Research Working Conference, CHARME 2003, L'Aquila, Italy, October 21-24, 2003, Proceedings*, volume 2860 of *Lecture Notes in Computer Science*, pages 277–282. Springer, 2003.

(c19). G. Della Penna, B. Intrigila, I. Melatti, E. Tronci, and M. Venturini Zilli. Finite Horizon Analysis of Stochastic Systems with the Murphi Verifier. In Carlo Blundo and Cosimo Laneve, editors, *Theoretical Computer Science, 8th Italian Conference, ICTCS 2003, Bertinoro, Italy, October 13-15, 2003, Proceedings*, volume 2841 of *Lecture Notes in Computer Science*, pages 58–71. Springer, 2003.

(c20). G. Della Penna, B. Intrigila, I. Melatti, M. Minichino, E. Ciancamerla, A. Parisse, E. Tronci, and M. Venturini Zilli. Automatic Verification of a Turbogas Control System with the Murphi Verifier. In Oded Maler and Amir Pnueli, editors, *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003 Prague, Czech Republic, April 3-5, 2003, Proceedings*, volume 2623 of *Lecture Notes in Computer Science*, pages 141–155. Springer, 2003.

(c21). G. Della Penna, A. Di Marco, B. Intrigila, I. Melatti, A. Pierantonio Xere: Towards a Natural Interoperability between XML and ER Diagrams. In Mauro Pezzè, editor, *Fundamental Approaches to Software Engineering, 6th International Conference, FASE 2003, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings*, volume 2621 of *Lecture Notes in Computer Science*, pages 356–371. Springer, 2003.

Proceedings of Italian Conferences

(aa1). G. Della Penna, B. Intrigila, I. Melatti, E. Tronci, and M. Venturini Zilli. Automatic Analysis of Hybrid Systems with the Murφ Verifier. *Atti Ufficiali del Congresso Annuale dell'Associazione Italiana per l'Informatica ed il Calcolo Automatico (AICA 2005)*

Research software

- QKS (*Quantized Kontrol Synthesizer*, a preliminary version is available at `http://mclab.di.uniroma1.it/software_qks.html`) implements the algorithms for automatic synthesis of control software described in (c3) and (c1). QKS takes in input:

    - the description of the system to be controlled (*plant*) as a Discrete Time Linear Hybrid System

    - the description of the AD/DA conversion to be used (i.e., the number of bits of AD/DA conversion)

    - the formal specifications of the closed loop system (desired controllable region and goal region).

  QKS outputs a software that implements the quantized controller, satisfies the formal specifications of the closed loop system and has a guaranteed and precomputed WCET (*Worst Case Execution Time*).

- NashMV (a preliminary version is available at `http://mclab.di.uniroma1.it/software.html#nashmv`). NashMV implements the algorithm described in (c7), by properly modifying the NuSMV model checker.

- Parallel Murphi (Eddy_Murphi, available at `http://www.cs.utah.edu/formal_verification/software/murphi` Eddy_Murphi is a parallel version (i.e., it runs at *computer clusters*) of the model checker Murphi. It implements the algorithm described in (c12) and (j2), by means of MPI (Message Passing Interface) and POSIX *threads* in Linux/Unix.

- 64-bits Murphi (CMurphi 5.4.6, available at `http://mclab.di.uniroma1.it/software.html#cmurphi`). Murphi port for 64-bits architectures.

- Finite Horizon Probabilistic Murphi (CMurphi 5.4.6, available at `http://mclab.di.uniroma1.it/software.html#cmurphi`). FHP-Murphi (Finite Horizon Probabilistic Murphi, see (j5), (c16), (c17), (c19)), is a model checker able to verify finite horizon probabilistic properties of discrete time stochastic processes. It has been used for the verification of probabilistic protocols and to evaluate complex systems reliability. FHP-Murphi may handle finite precision real numbers, thus it is also used to verify nonlinear stochastic hybrid systems.

- Caching & Disk Murphi (CMurphi 5.4.6, available at `http://mclab.di.uniroma1.it/software.html#cmurphi`). CMurphi (see (c14), (j7), (c20)), is an improved version of the Murphi verifier. CMurphi exploits transition locality in the system transition function in order to lower the RAM requirements and to speed up disk-based algorithms. CMurphi has been used by INTEL to verify Cache Coherence Protocols. Finally, CMurphi may handle finite precision real numbers, thus it is also used to verify nonlinear hybrid systems.

Teaching

As an Assistant Professor

- At the Sapienza University of Rome, I give lessons inside the classe of "Sistemi Operativi (II modulo)" (Operating Systems, academic years 2010/2011)

As a Post-Doc

- At the Sapienza University of Rome, I gave lessons inside the two classes (divided by students' names) of "Progettazione di Sistemi Digitali" (Digital Systems Design, academic years 2008/2009 and 2009/2010)
- At the University of Rome "Tor Vergata", he held the part of Informatics in the integrated course of "Matematica ed Informatica" for biologists (Mathematics and Informatics, academic year 2008/2009)
- At the University of L'Aquila, I was the lecturer of the class "Verifica dei Sistemi Complessi" (Verification of Complex Systems, academic year 2007/2008)
- At the Sapienza University of Rome, I gave lessons inside the class "Programmazione 1" (Programming 1, academic year 2006/2007 and 2007/2008)
- At the Sapienza University of Rome, I have been the teaching assistant for the class "Architettura degli Elaboratori 1" (Computers' Architecture 1, academic year 2007/2008)
- At the Sapienza University of Rome, I gave lessons inside the class "Programmazione ad Oggetto" (Object-Oriented Programming, academic year 2006/2007)
- At the School of Computing of the University of Utah, I gave lessons in the Model Checking class (CS 6964, Fall 2005) inside the AMPS seminars (Programming Languages and Systems Seminar, CS 7931, Fall 2005)
- At the University of Roma "Tor Vergata", I gave lessons inside the class "Metodi Formali per la Verifica dei Sistemi Complessi" (Formal Methods for the Verification of Complex Systems, academic year 2005/2006)
- In both these classes, I supported the students during the didactical projects
- At the Sapienza University of Rome, I support Prof. Enrico Tronci in supervising some theses.

As a PhD Student

- At the University of L'Aquila, I gave lessons inside the following classes:

  - Academic year 2004/2005: Metodi Formali per la Verifica dei Sistemi Complessi (Formal Methods for the Verification of Complex Systems)

  - Academic year 2003/2004: Metodi Formali per la Verifica dei Sistemi Complessi (Formal Methods for the Verification of Complex Systems)

  - Academic year 2003/2004: Architettura degli Elaboratori (Computers' Architecture)

  - Academic year 2002/2003: Architettura degli Elaboratori (Computers' Architecture)

  - Academic year 2001/2002: Laboratorio di Architettura degli Elaboratori (Lab of Computers' Architecture)

- In all these classes, I supported the students during the didactical projects

- At the University of L'Aquila, I supported Profs. Benedetto Intrigila, Enrico Tronci e Giuseppe Della Penna in supervising master theses.

Research Projects

As a member of MCLab (*Model Checking Laboratory*, research group of the Department of Computer Science at Sapienza University of Rome, coordinated by prof. Enrico Tronci) he takes part to many research projects funded by EC (European Community), ESA (European Space Agency), ENEA (Italian National agency for new technologies, Energy and sustainable economic development), CNR (National Research Council), MIUR (Italian Ministry for the Research and University), MSE (Ministry of Economin Development) and provate industries. This leads to a continuous hosmosis between research results and advanced industrial applications. Here is selected list of projects (please note that, for the projects from 2006 onwards, he also cooperated to the project proposal):

ESA ITI AO6067 - 2010 *Model Checker Validator for Satellite Operational Procedure.* The goal of this Innovation Triangle Initiative (ITI) of ESA is to design and implement a model checker for the Validation and Verification of satellites Operational Procedures (OP). MCLab contribution is focused on the design of the OP model checker and on its interface with ESA simulator SIMSAT.

WFR (MSE) - 2010 *Web Fitting Room* is a projected funded by MSE for the program "Industria 2015". The goal of WFR is to design and implement a web based system for the on-demand production of clothes, by connecting via Web a virtual fitting room (where clothes are virtually dressed) and a Decision Support System (DSS) which, in real time, organizes the production of the selected clothes. MCLab role focueses on the design and implementation of the DSS by using Model Checking based Planning to counteract the state explosion problem arising in DSSs of the type discussed above.

ULISSE (EC FP7) - 2009 *USOCs KnowLedge Integration and Dissemination for Space Science Experimentation* is a project funded by EC inside the program FP7. MCLab role in this project is to study model checking techniques for the automatic verification of procedures and plans related to on-board experiments on space stations (e.g. Columbus).

SAPP (FILAS) - 2008 *Advanced System for the Design and Planning of Wireless Networks* is a project funded by FILAS (local Italian agency). MCLab role in this project is to design and implement algorithms for the fault tolerant placement of relay nodes of a wireless network, given the models for the antennas and the positions of gateway e sensor nodes. The goal is to guarantee that, even if k relay nodes are faulty, the network satisfies the given specifications.

ESA 5459 SSFRT - 2008 *System and Software Functional Requirements Techniques* is a project funded by ESA, having INTECS as project main contractor. MCLab role in this project is to study if model checking techiniques for hybrid systems are applicable to the automatic verification of system requirements for satellites and generic space vehicles. Models for these requirements involves both software and systems (sensors and actuators) the software interacts with. This motivates the usage of hybrid systems to model, validate and verify such systems.

SINTESI - 2008 *Automatic Synthesis of Reaction Rules for Enterprise Processes Management* is a project funded by MIUR. MCLab role in this project is to design and implement new model checking based algorithms for the automatic synthesis of reaction rules in SaR

(Sense and Respond) systems for the enterprise processes management, focusing on automatic allocation of resources in multimedia enterprises.

CRESCO (MIUR) - 2007 *Computational Center for the Research on Complex Systems.* MCLab entered this project as advisor for University of Salento. MCLab role in this project is to design and implement algorithms for the automatic analysis of vulnerability indices for networks and telecommunication services.

International schools

- ISCL 2002: Second International Summer School in Computational Logic (Acquafredda di Maratea, 25-30 august 2002)

Peer-to-peer reviewing activity

He has served and serves as peer-to-peer reviewer for the following international journals:

- IEEE Distributed Systems Online
- International Journal of Business Data Communications and Networking.

He has served and serves as peer-to-peer reviewer for the following international conferences:

- CAV (Computer Aided Verification)
- FMCAD (Formal Methods for Computer-Aided Design)
- PSI (Ershov Informatics Conference)
- ICALP (International Colloquium on Automata, Languages and Programming)
- SAT (Theory and Applications of Satisfiability Testing)
- LICS (Logic In Computer Science)
- CHARME (Correct Hardware Design and Verification Methods).

Working outside university

- In May 2006, he gave a course about C++ Programming, organized by the Lazio Italian region
- Between 2002 and 2005 he gave courses about basic notions of informatics (programming, web design), organized by the Abruzzo Italian region
- In July 2002, he gave a course about XML organized by a small Italian company, BitMedia s.r.l.
- From July 2001 to March 2002 he worked in Nikesoft s.r.l., a small Italian company. There, he developed managing programs, working on Windows 2000 with Visual C++ and SQL Server 2000.

Research activity

My research activity focuses on *Model Checking*, i.e. on algorithm and tools (*model checkers*) which take in input the formal specification of a system $\mathcal{S}$ and of a property ϕ, and output *true* if ϕ is satisfied by $\mathcal{S}$, and *false* otherwise. Differently from other approaches, such as testing and simulation, if a model checker answers *true* then we have the *mathematical certification* that $\mathcal{S}$ satisfies ϕ. For *mission critical* as well as *safety critical* systems this is required in the designing phase, e.g. by ESA (European Space Agency) and IEC (International Electrotechnical Commission) standards.

Computationally speaking, model checking algorithms consist of three steps:

1. obtain the *transition graph* of the system $\mathcal{S}$ (a transition graph specifies how $\mathcal{S}$ may go from a *state* to another state);

2. compute the *reachable* states, starting from a given set of initial states (*reachability*);

3. verify ϕ for all reachable states.

The main obstruction for the verification via Model Checking is in the reachability step. In fact, even if the formal description of $\mathcal{S}$ has a reasonable size, the number of states in $\mathcal{S}$ is exponential in the size of the description of $\mathcal{S}$. This quickly fills up all the available computation resources, especially the memory. This problem is often addressed to as the *state space explosion.*

It is however possible to design efficient model checking algorithms to verify particular (and limited) class of systems. In my research activity I focused on the verification of some classes of highly interesting concurrent systems, i.e. communication protocols and hybrid systems (that is systems which need both discrete and continuous variables to be described), and on an important class of properties to be verified, i.e. *safety* properties.

More in detail, I worked on the following themes:

Verification algorithms based on statistical properties of the systems

In this setting, my contribution consists of designing, implementing and experimenting new model checking algorithms, which achieve better performances (especially for the RAM memory usage) than the currently state-of-the-art ones, when dealing with safety properties of communication protocol and hybrid systems. These algorithms have been implemented in the tool *Caching Murphi* (CMurphi), based on the *Murphi* model checker (originally developed at Stanford). CMurphi turned out to be very effective, and it has been successfully used by *Intel* (more precisely, by the *Platform Architecture Research Team* of the *Microprocessor Technology Labs* of the Intel *Corporate Technology Group*), in order to verify certain *cache coherence protocols* which was not possible to verify using the preceding tools and algorithms. For more details, see (c18), (c14) and (j7).

Algorithms for the verification of stochastic systems

Starting from the second half of the 90's, *Probabilistic Model Checking* has been developed to verify probabilistic properties of stochastic systems. Probabilistic Model Checking has the roughly the same obstructions of Model Checking, thus the problem is again P-SPACE complete.

In my research activity I focused on the verification of some classes of highly interesting stochastic systems i.e. probabilistic communication protocols and stochastic hybrid systems which can be described or approximated by Discrete Time Markov Chains. As for the properties to be verified, I focused on finite horizon properties, i.e. on properties which take into account only a finite number of execution steps in the system $\mathcal{S}$ to be verified (i.e. of transitions in the transition graph of $\mathcal{S}$).

In this setting, my contribution consists of designing, implementing and experimenting new probabilistic model checking algorithms, which achieve better performances (especially for the RAM memory usage) than the currently state-of-the-art ones, when dealing with finite horizon properties of probabilistic communication protocol and stochastic hybrid systems. These algorithms have been implemented in the tool *Finite Horizon Probabilistic Murphi* (FHP-Murphi). For more details, see (c16), (c17), (c19) and (j5).

Parallel Model Checking algorithms

One of the traditional methods used to cope with state explosion in Model Checking is to use parallel versions of the Model Checking algorithms, in order to take advantage of the availability of *computer clusters*. Note that here, rather than RAM memory usage (which is larger since it is distributed on many computers), the real bottleneck is in the execution time, due to the necessary communications between computers in the cluster.

In this setting, my contribution, consists of designing, implementing and experimenting a new parallelization scheme, which gave encouraging preliminary results. For more details, see (c12) and (j2). Further improvements are currently being developed with the research group of Prof. G. Gopalakrishnan of the University of Utah.

Automatic synthesis of reactive programs from formal specifications

Given a formal model of the environment which a reactive program has to interface with, it is possible to use model checking based techniques in order to automatically synthesize the C code for the reactive program itself. This may be done by considering the reactive program as a *universal plan*. This approach allows to automatically build correct-by-construction code, and to perform requirements verification and validation at closed loop from the first design stages. For more details, see (c10), (j3) and (c9).

Technology transfer

In order to evaluate the effectiveness of the proposed approaches in an industrial context, I also worked on case studies and technology transfer. For more details, see (c20), (c11), (c5), (c4), (c2) e (j1).

BAR protocols verification

BAR (Byzantine Altruistic Rational) protocols are cooperative systems (e.g. peer-to-peer protocols) with n agents, in which some agents may not follow the given protocol. Among these defiant agents, some (the *rational* ones) are guided by a prize-punishment mechanism defined by the protocol itself, others (the *byzantine* ones) model faulty agents.

In (c7) we present a symbolic model checking algorithm for verification of Nash equilibria in finite state mechanisms modeling BAR protocols, i.e. *Multiple Administrative Domains* (MAD) distributed systems.

Given a finite state mechanism, a *proposed protocol* for each agent and an *indifference threshold* for rewards, our model checker returns *PASS* if the proposed protocol is a Nash equilibrium (up to the given indifference threshold) for the given mechanism, *FAIL* otherwise.

In (c6) we extend by considering also agents coalitions.

Other researches

Finally, my research activity also includes some works not strictly regarding Model Checking. For more details, see (j4), (c15), (j6) e (c21).

Work in progress

SAT-based Bounded Model Checking *SAT-based Bounded Model Checking* takes in input the system and the properties to be verified, together with an integer n. Then, the reachability problem in at most n steps is translated in a CNF (Conjunctive Normal Form) formula, which is passed to a SAT-solver tool. This allows to exploit the major improvements in SAT-solver tools which have been done in the last years. Unfortunately, once the CNF is generated, we often have a huge file with about 10^8 clauses. This mainly affects the *Boolean Constraint Propagation* (BCP) performed by SAT-solver, which needs to repeatedly scan the whole CNF in RAM. We are studying how to perform BCP directly on disk (see (c8)) or to distribute it on several machines (by exploiting our experience in this field, see (c12), (j2) and (j7)).

Model Checking and controller synthesis for hybrid systems We are currently studying a new methodology for the verification and controller synthesis for hybrid systems described as Discrete Time Linear Hybrid Systems, i.e. modeled by a set of linear constraints. A first paper on this topic has been published at CAV 2010 (c3) and ICSEA 2011 (c1).