

Curriculum vitæ

FEDERICO MARI *

Contents

1	Current Position	1
2	Education	1
3	Publications	2
3.1	Refereed Articles in Conferences	2
3.2	Thesis	3
4	Software	3
5	International	4
5.1	Schools and Periods Abroad	4
6	Teaching	4
7	Research	5
7.1	Motivations	5
7.2	Contributions	7

1 Current Position

From January 1, 2010 Federico Mari is a Postdoctoral researcher at the Computer Science Department of Sapienza University of Rome (Italy).

2 Education

On February 2010 under the supervision of Prof. Enrico Tronci, he received his Ph.D. degree from Sapienza University of Rome, with a thesis titled *Verification and Synthesis for Discrete Time Linear Hybrid Systems*.

*Post-Doc at Sapienza University of Rome, Computer Science Department. Web: <http://www.dsi.uniroma1.it/~mari/>. Mail: mari@di.uniroma1.it. Tel: +39 06 4991 8352.

In 2006 under the supervision of Prof. Enrico Tronci, he received his Master degree in Computer Science from Sapienza University of Rome, with a thesis 3.2(2) titled *Automatic Hybrid Systems Verification via Satisfiability*.

3 Publications

3.1 Refereed Articles in Conferences

1. F. Cavaliere, F. Mari, I. Melatti, G. Minei, I. Salvo, E. Tronci, G. Verzino, and Y. Yushtein. Model checking satellite operational procedures. In *In: DAta Systems In Aerospace (DASIA), Org. EuroSpace, Canadian Space Agency, CNES, ESA, EUMETSAT. San Anton, Malta, EuroSpace.*, May 2011.
2. F. Mari, I. Melatti, I. Salvo, and E. Tronci. FROM BOOLEAN FUNCTIONAL EQUATIONS TO CONTROL SOFTWARE. *CoRR*, abs/1106.0468, 2011.
3. F. Mari, I. Melatti, I. Salvo, and E. Tronci. QUANTIZED FEEDBACK CONTROL SOFTWARE SYNTHESIS FROM SYSTEM LEVEL FORMAL SPECIFICATIONS. *CoRR*, abs/1107.5638, 2011.
4. F. Mari, I. Melatti, I. Salvo, and E. Tronci. QUANTIZED FEEDBACK CONTROL SOFTWARE SYNTHESIS FROM SYSTEM LEVEL FORMAL SPECIFICATIONS FOR BUCK DC/DC CONVERTERS. *CoRR*, abs/1105.5640, 2011.
5. F. Mari, I. Melatti, I. Salvo, and E. Tronci. SYNTHESIS OF QUANTIZED FEEDBACK CONTROL SOFTWARE FOR DISCRETE TIME LINEAR HYBRID SYSTEMS. In *Computer Aided Verification*, volume 6174 of *Lecture Notes in Computer Science*, pages 180–195. Springer Berlin / Heidelberg, 2010.
6. A. Bobbio, E. Ciancamerla, S. Di Blasi, A. Iacomini, F. Mari, I. Melatti, M. Minichino, A. Scarlatti, E. Tronci, R. Terruggia, and E. Zendri. RISK ANALYSIS VIA HETEROGENEOUS MODELS OF SCADA INTERCONNECTING POWER GRIDS AND TELCO NETWORKS. In *Risks and Security of Internet and Systems (CRiSIS), 2009 Fourth International Conference on*, pages 90–97, oct. 2009.
7. S. Mazzini, S. Puri, F. Mari, I. Melatti, E. Tronci. FORMAL VERIFICATION AT SYSTEM LEVEL. In *Proceedings of the International Space System Engineering Conference DAta Systems In Aerospace, DASIA 2009, Istanbul, Turkey, May 26-29, 2009*.
8. 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 *In Proceedings of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS '09, Lyon, France, November 3-6, 2009*. Springer, 2009.

9. F. Mari, I. Melatti, I. Salvo, E. Tronci, L. Alvisi, A. Clement, and H. Li. MODEL CHECKING NASH EQUILIBRIA IN MAD DISTRIBUTED SYSTEMS. In A. Cimatti and R. Jones, editors, *Proceedings of Formal Methods in Computer Aided Design, FMCAD 2008, Portland, OR, USA, November 17-20, 2008*. IEEE, 2008.
10. F. Chierichetti, S. Lattanzi, F. Mari, and A. Panconesi. ON PLACING SKIPS OPTIMALLY IN EXPECTATION. In M. Najork, A. Z. Broder, and S. Chakrabarti, editors, *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, pages 15–24. ACM, 2008.
11. F. Mari and E. Tronci. CEGAR BASED BOUNDED MODEL CHECKING OF DISCRETE TIME HYBRID SYSTEMS. In A. Bemporad, A. Bicchi, and G. C. Buttazzo, editors, *Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings*, volume 4416 of *Lecture Notes in Computer Science*, pages 399–412. Springer, 2007.

3.2 Thesis

1. Ph.D. Thesis. VERIFICATION AND SYNTHESIS FOR DISCRETE TIME LINEAR HYBRID SYSTEMS. *February 22, 2010*.
2. Tesi di Laurea. AUTOMATIC HYBRID SYSTEMS VERIFICATION VIA SATISFIABILITY. *March 15, 2006*.

4 Software

All listed software is available for download as open-source at <http://mclab.di.uniroma1.it/>.

- QKS (Quantized Kontrol Synthesizer) implements the algorithms for automatic synthesis of control software described in 3.1(5). 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

By appropriately modifying the NuSMV model checker, NASHMV implements the algorithms described in 3.1(9) and 3.1(8). This methods aim to verify whether or not a given protocol is a Nash Equilibrium in a distributed system in which nodes can behave rationally.

5 International

5.1 Schools and Periods Abroad

University of Texas at Austin (Sept–Oct 2008).

1. Collaboration with Prof. Lorenzo Alvisi.
2. Course in *Formal Verification and Semantics* (CS388S) held by Prof. E. Allen Emerson, one of the three recipients of the 2007 Turing Award.

MOVEP 2008: *Modeling and Verifying Concurrent Processes 2008*. Nouan-le-Fuzelier (Orléans), 23–27 June, 2008.

BiCi-SNS 2008: *International PhD School on Randomized Algorithms*. Scuola Normale Superiore di Pisa (February 4–8, 2008).

WEB BAR 2005: *Bertinoro International Ph.D. School on Advanced Retrieval and Web Mining*. Bertinoro (Forlì-Cesena), 29 August – 9 September, 2005.

6 Teaching

In Spring 2011, Federico Mari was a Teaching Assistant in the *Formal Methods for Software* (Prof. Anna Labella). During this course he gave a series of lectures on Alloy modeling language.

In Fall 2010, Federico Mari held a course on *Database design*, from the requirements (UML) to the relational schema (MySQL). The complete course was on Database design (MySQL) and Web development (PHP).

In Fall 2008, Federico Mari was a Teaching Assistant in the *Programming Languages* course (Prof. Nicola Galesi).

In Spring 2008, Federico Mari was a Teaching Assistant in the *Programming Languages Laboratory* (Prof. Enrico Tronci). During this course, for didactical purposes, he realized a minimal SAT Solver.

In 2006, Federico Mari was a Teaching Assistant in the *Software Engineering* course (Prof. Enrico Tronci), and gave a series of lectures on *Monte Carlo Model Checking* and *Modeling of Protocols*.

7 Research

Federico Mari's research interests mainly concern with *model checking and automatic synthesis of control software for discrete time linear hybrid systems*.

7.1 Motivations

In everyday life we are imperceptibly and discreetly surrounded by a great heterogeneity of useful devices. Among them, an important class is composed by *embedded systems*.

Embedded Technology has applications from micro to big dimensions. Examples of embedded systems are micro-controllers, music and video players, digital organizers, smart kitchen tools like high-tech refrigerators or self-cooking ovens. Examples can be found also in automotive industry. In fact, on-board computers are more and more complicated. For example, a modern car is equipped with parking assistants, satellite navigators, Antilock Braking System (ABS), just to mention some. Naval technology is also based on embedded systems. For example, in a warship we can find diver detection sonars, harbor surveillance systems, underwater acoustic signal analysis systems. In this fashion show, we cannot avoid to mention aviation related systems. In particular, present aircrafts are supplied with autopilots, radio altimeters, transponders and collision avoidance systems. Besides this, unmanned aircraft systems exist. Last but not least, we should not forget all the advanced embedded technology orbiting around our Earth, that is satellites on which present world communications actually rest. Most of this interesting devices can be conveniently modelled as hybrid systems.

Speaking of communications, *protocols over distributed systems* also represent a particular class of hybrid systems. As an example, we can think at peer-to-peer protocols like BitTorrent, eDonkey and Kad. Also ad-hoc networks among portable devices are increasingly spreading their interest among network and security research communities.

Hybrid systems Verification and Synthesis In a hybrid system there is a coexistence of a discrete part (e.g. the software on embedded systems, or protocol transitions) and a continuous component (e.g. the hardware, or net measures). The first is represented by using integer or boolean variables, whereas continuous variables realize the real part. The dynamics of the system is defined by discrete moves for integer quantities and by differential equations for the continuum.

Verification and *synthesis* for hybrid systems are both safety critical issues. Given a hybrid system and a property to be satisfied, verification consists in checking that the property holds in all possible evolutions of the system. For example, given the software realizing a collision avoidance protocol for an aircraft, we want to be guaranteed that no different aircrafts will eventually collide. As for synthesis, the goal is to build correct-by-construction software (controllers), aimed to satisfy desired behaviors. For example, given the model for an inverted pendulum we would like to be automatically supplied with a program moving and stabilizing the pendulum in its upright unstable equilibrium.

Problem relevance When not formally verified or synthesized, an embedded system is naturally and easily prone to *software bugs*. It is not a great social problem if our favourite music player stops working correctly while we are running, or if after jogging our “smart” oven ruins our so much desired delicatessen. That is annoying but not a great public deal. Nonetheless, software bugs can be more dangerous than that. They can even cause awful disasters and not required expenses. On August 20, 2008, Spanair Flight 5022, a McDonnell Douglas MD-82 crashes on takeoff at Barajas Airport in Madrid, Spain. Of the 172 people on board, 154 are killed [2]. The report released on the accident [3] confirmed that the software that should have prevented the crash failed to do so. Launched on January 3, 1999, the Mars Polar Lander sent the last telemetry just prior to atmospheric entry on December 3, 1999. Investigations show that the most likely cause of the failure was a software error [4]. Not to mention the subtle and costly Y2K problem [5], which cost has been estimated at over 300 billion US dollars [6], 15 times more than the 20 billion US dollars pledged by G8 in firm aid to poor nations (L’Aquila, Italy 2009 [7]).

Technical obstructions Verification and synthesis for hybrid systems are thus relevant issues. Several solutions have been proposed over the last twenty years. Nonetheless, there is not a solution which performs better than others in all situations. The motivation for this is that even the *reachability problem is undecidable* for simple classes of hybrid systems [1]. Verification and synthesis entails searching for reachable states, thus they are in general undecidable too.

Need for research The above considerations motivate the research on verification and synthesis of hybrid systems.

In fact, several investors are seeking for collaborations in order to develop new technologies and methods in this field. In particular, the European Space Agency (ESA) promotes many invitation to tenders (ITT) [8] for new research on Space, in which context many systems can be modelled as hybrid systems. Moreover, ESA together with the Canadian Space Agency sponsored the conference DASIA 2009, DAta Systems in Aerospace [9]. The European Union, in the context of the 7th Framework Programme, looks at contributions on Space research [10]. In particular, several activities entail the strengthening of the

foundations of Space science and technology. In these activities, security is a core problem.

Moreover, private companies maintain Research&Development sectors trying to find innovative solutions to these problems. For example, at Tokyo Motor Show 2009 [11] Honda presented an application of balance control: the monocycle U3-X. This device is completely automated. It stands in its unstable upright equilibrium and moves when the rider leans the upper body.

Need for efficient controller implementations Present technology trend moves toward installing embedded systems on smaller and smaller chips. For example, under skin chips containing personal information for medical use already exist. Besides security issues, the problem relating this new trend is to obtain even more compact control software implementations.

7.2 Contributions

Inspired by this state of affairs, our main research activity concerns with *verification (i.e. model checking) and synthesis for hybrid system and protocols in distributed systems*.

Many hybrid systems can be conveniently modeled as Discrete Time Linear Hybrid Systems (DTLHSs). In the following a list of contributions on DTLHSs is shown.

Model checking hybrid systems Given a system specification S and a temporal logic property φ , a *model checking* algorithm automatically verifies whether or not S satisfies φ . In particular, if S satisfies φ it returns YES. Otherwise it returns NO together with a path from an initial state to an error state, i.e. a state of S in which φ does not hold. If we are interested only in the correctness of those states reachable in at most k steps from an initial state, the problem is called *bounded model checking*.

It is known that bounded model checking for DTLHSs comes down to solve a Mixed Integer Linear Programming (MILP) feasibility problem. In 3.1(11) a SAT based bounded model checking algorithm for automatic verification of *safety* properties for DTLHSs is presented (extending results presented in 3.2(2)).

Using Counterexample Guided Abstraction Refinement (CEGAR) our algorithm gradually transforms a DTLHS verification problem into larger and larger SAT problems. Experimental results show that this approach can handle DTLHSs that are more than 50 times larger than those that can be handled using an MILP solver.

Automatic synthesis of control software for discrete time linear hybrid systems Let P model a physical system (*plant*) having an internal state, inputs u and outputs y . Given a *safety* property φ (*goal*), the *control problem* consists in finding a function K (the *controller* for P) s.t. *i*) the pair (K, P)

is a *closed loop* system, that is the input of K is the output y of P while the result $K(y)$ is used to instruct P , i.e. $u = K(y)$, and *ii*) thanks to the just described closed loop system (K, P) , the plant P eventually satisfies φ . In other words, the controller K drives the plant P to the goal φ by giving appropriate commands to P .

When K is required to be discrete valued (and this is the case for small on-chip software implementations), since the plant P is a physical system a *quantization* is needed. Namely, an analog-to-digital conversion is required for P 's output y whereas a digital-to-analog conversion is needed in order to instruct the input u with $K(y)$. In this case we call K the *Quantized Feedback Controller* (QFC) for P .

In the following we briefly describe our contribution in 3.1(2), 3.1(3), 3.1(4), 3.1(5), 3.2(1).

We presented novel techniques and algorithms that, given a DTLHS model P for a plant, a desirable controllable region and a goal region, returns a *correct-by-construction control software* K for P , along with a suitable representation for the set of states on which K is guaranteed to work (controllable region).

K is represented using a model checking data structure, called *Ordered Binary Decision Diagram* (OBDD). OBDDs represent boolean functions, thus they can be used to represent sets (in this case the set of control rules defining K). In order to obtain succinct software implementations of K , a ***translator from an OBDD to a C function*** is also presented .

Furthermore, we show that the so generated K has a *Worst Case Execution Time* (WCET) guaranteed to be linear in the number of bits of the quantization schema.

The algorithm has been implemented on top of the CUDD OBDD package and of the GLPK MILP solver. We present results on using such an implementation to synthesize C controllers for the Buck DC-DC converter. Experimental results show that the proposed approach is viable and the performances of the automatically synthesized controllers compare well with those of controllers designed using ad-hoc approaches.

Protocols as Nash Equilibria in MAD distributed systems In 3.1(9) and 3.1(8) we study a particular class of hybrid systems, that is protocols in Multiple Administrative Domains (MAD) distributed systems.

In MAD environments, each node is administrator of itself, meaning that it has power to break the protocol, e.g. modifying the software implementing it. If we look at the choice of the strategy to follow as a game among players, the protocol takes the role of a solution in the game-theoretical sense. In order for the players to follow the protocol, protocol designers must invent a reward and punishment mechanism such that the protocol results in an equilibrium for the game. With this view in our mind, in 3.1(9) we present a method to check if a given protocol is a Nash Equilibrium in the given system. Since this method does not cope with coalitions among agents, in 3.1(8) we present an algorithm checking if a given protocol is a Nash Equilibrium when agents can collude to

improve their power. Both methods take into account the presence of Byzantine agents behaviors.

The method presented in 3.1(9) can be seen as a particular case of the one presented in 3.1(8), where the maximum cardinality for coalitions is set to 1.

References

- [1] HENZINGER, T. A., KOPKE, P. W., PURI, A., AND VARAIYA, P. What's decidable about hybrid automata? In *STOC '95: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing* (New York, NY, USA, 1995), ACM, pp. 373–382.
- [2] Wikipedia. List of accidents and incidents on commercial airliners: http://en.wikipedia.org/wiki/List_of_accidents_and_incidents_on_commercial_airliners, 2009.
- [3] CIAIAC. 20-08-2008. EC-HFP. McDonnell Douglas MD-82. Aeropuerto de Barajas (Madrid).
- [4] Wikipedia. Mars Polar Lander: http://en.wikipedia.org/wiki/Mars_Polar_Lander, 2009.
- [5] Wikipedia: Year 2000 Problem: http://en.wikipedia.org/wiki/Year_2000_problem, 2009.
- [6] Y2K: Overhyped and oversold?, report from BBC News.
- [7] Reuters: G8 pledges \$20 billion in farm aid to poor nations. <http://www.reuters.com/article/topNews/idUSTRE5662VJ20090710>, 2009.
- [8] European space agency home page: <http://www.esa.int/esaCP/index.html>, 2009.
- [9] DASIA 2009. <http://pagesperso-orange.fr/eurospace/dasia.html>, 2009.
- [10] European Commission CORDIS. FP7: <http://cordis.europa.eu/fp7/>, 2009.
- [11] Tokyo motor show 2009. <http://www.tokyo-motorshow.com/eng/>, 2009.

Rome, October 12, 2011,

Federico Mari