Amir Pnueli and the Dawn of Hybrid Systems

Oded Maler
CNRS-VERIMAG, University of Grenoble
Centre Equation, 2 av de Vignate
38610 Gières, France
Oded.Maler@imag.fr

ABSTRACT

In this talk I present my own perspective on the beginning (I refer mostly to the period 1988-1998) of hybrid systems research at the *computer science* side, focusing on the contributions of the late Amir Pnueli, mildly annotated with some opinions of mine.

Categories and Subject Descriptors

A.0 [**General literature**]: General

General Terms

Verification

Keywords

Amir Pnueli, Hybrid systems

1. INTRODUCTION

Amir's contribution to hybrid systems consisted in suggesting one of the first verification-oriented hybrid system models, in formulating verification problems, in inspiring and encouraging others to look at the domain, and in participating in the founding of the HSCC (*Hybrid Systems: Computation and Control*) series to which he gave the name.

I think, however, that his major influence was of a more implicit nature via the impact he had on the design and verification of *discrete reactive systems*. His work contributed to bringing verification closer to other engineering disciplines that share a systems thinking, abstracting away from the details of programming languages. As a result, verification was better prepared for the intercultural challenge posed by hybrid systems.

Unlike many computer scientists, Amir came to hybrid systems with a solid background in continuous mathematics. His PhD thesis, under the supervision of Chaim Pekeris, dealt with partial differential equations for modeling ocean tides [36]. He remembered the days when the term *hybrid computation* referred to a mixture of digital and analog (differential analyzer) computers, and was aware of *hybrid simulation* which meant a blend of continuous and discrete-event simulation.

2. DISCRETE SYSTEMS

During his post-doc at Stanford, under the influence of Zohar Manna, Amir reinvented himself into the domain of semantics, logic and program verification where he made numerous contributions. The most celebrated among them, for which he received the Turing award in 1996, is the introduction of *temporal logic* as a specification language for expressing the acceptable behaviors of systems [33, 34]. Theoreticians may argue that temporal logic is nothing but a syntactic sugar for other logics already used in verification and elsewhere, but that would entirely miss the point. The impact of introducing temporal logic was in shifting the focus toward the *sequential ongoing input-output behavior* of a system, the input-output transducer that it realizes. This was in contrast to the central activity in program verification at that time which dealt with the logical description of the function computed upon termination by a data-rich program, for example, a sorting algorithm.

Later, in collaboration with David Harel who worked part-time as a consultant in avionics, he coined the term *reactive systems* [11] to denote systems whose major functionality is to maintain an ongoing and timely interaction with their external environment, rather than solving complex but *static* computational problems. Communication protocols (hardware and software alike), control systems, real-time systems, embedded systems, cyber-physical systems are all, in this sense, instances of reactive systems.

Evaluating a reactive system according to whether the sequences of states and events it produces satisfy a temporal property brings verification closer to other engineering disciplines that evaluate the performance of a system according to some measures on its *behaviors*, that is, the evolution of observable state variables over time. I believe that this twist in verification, taking it further away from the syntactic concreteness of software and closer to the abstract notion of a *dynamical system* paved the way to relating verification to other domains such as control [22] and optimization [23].

3. TIMED SYSTEMS

Like many others in the 1980s, Amir explored *quantitative* extensions of verification methodology, starting with *timed systems*, that is, a finer level of abstraction incorporating quantitative metric time. Amir interacted with numerous researchers working on these issues through conferences and workshops and later via European projects. He made extended visits to Stanford where he collaborated with Zohar and his students, most notably Tom Henzinger and Rajeev Alur. Amir suggested his variants of real-time temporal logic [35, 12] and a real-time system model, the *timed transition system* [15], which was the basis for his hybrid system model.

As it turned out, the less structured *timed automaton* model [3] became more popular, partly because its introduction was accompanied by a verification *algorithm*. The real-time temporal logics that prevailed are MTL [19], developed earlier, and its restriction MITL [4], for which Amir, very recently, participated in developing a new translation to timed automata [25]. This work is based

ACM X-XXXXX-XX-X/XX/XX.

on the idea of *temporal testers* that he advocated in the last couple of years as an alternative to the classical tableau construction.

This is perhaps the place to mention a particular cognitive trait of Amir. Many researchers need to translate the ideas and models of others to their own internal language, before being able to digest them. Perhaps the best analogy is the breaking up of external proteins into smaller molecules before synthesizing one's own proteins. Amir's mental digestion capability was remarkable as he could easily operate within models and notations introduced by others. He was very receptive to ideas raised by other people and this rare capability to *listen* made him even more popular among those who felt they have something important to say and sought feedback. Needless to say, this practice broadened his horizons even further.

4. HYBRID SYSTEMS

Amir was aware, of course, of the fact that on the other end of the reactive system, you find some physical reality, evolving according to its own rules, interacting with the computer via sensors and actuators. Hence, when I asked him one day, toward the end of my thesis, how one can verify that a robot, following some control program, behaves correctly in an environment, I was breaking through a widely open door. We discussed the topic several times and even wrote a research proposal that did not get through.

The next episode took place in the spring of 1991 when Amir showed up in Rennes (where I was doing my post-doc) on his way to an influential workshop in the Netherlands [8], telling me that he intended to present a hybrid system model in that meeting. His presentation, which was greeted with enthusiasm by the workshop participants, played an important role in shaping the computer science research on hybrid systems.

Amir's model, that he called a *phase transition system* [24], was essentially a hybrid automaton combining discrete instantaneous *transitions* with continuous *activities*, the latter defined by differential equations. A lot of effort has been invested in reconciling the common asynchronous interpretation of concurrency, realized by the interleaving semantics, with the more synchronous nature of differential equations. The semantics was defined in terms of runs that *alternate* between continuous evolution and discrete transitions, ranging over a "super-dense" time domain, to accommodate for several transitions that occur one after the other but at the same time instant. The paper concluded with a proof rule for invariance, demonstrated on a cat-and-mouse example, and some thoughts on the effect of discrete sampling on property satisfaction.

The model shared a lot of features with another paper Amir was writing in parallel about timed and hybrid StateCharts [17]. It contained additional ingredients taken from the general Manna-Pnueli framework for verifying concurrent programs [30, 31] such as fairness and justice conditions that, in retrospect, were less urgent in the hybrid context.

Few months later there were other contributions in this direction, most notably by Joseph Sifakis and his students and by Tom Henzinger who started developing an algorithmic approach for the verification of hybrid automata. Thanks to this interest in hybrid systems, Joseph offered me to move to Grenoble and contribute to the emerging field. Meanwhile, a hybrid systems movement, led mostly by Anil Nerode, started forming in the US, having a first hybrid system workshop in Cornell followed by a European workshop in Lyngby [10] to which my only contribution was the amusing "pamphlet" (as Amir called it) on real-world computations [21].

Around that time there was a general shift in focus from *deductive* to *algorithmic* verification, inspired by the success of *model checking* for discrete systems. The decidability results on timed automata motivated many to develop verification algorithms for hybrid systems with simple continuous dynamics in each of the discrete states [32, 2, 13, 1]. Amir participated in proving some positive and negative results concerning *stopwatch automata* [18] which are like timed automata but with clocks that can be stopped and resumed (such automata can model preemptive schedulers). Our joint technical contribution to the algorithmic analysis of hybrid automata was a paper submitted to *CAV'93* in which we proved decidability of reachability problems for PCD (piecewise-constant derivatives) systems on the plane. Not much later it was shown that one cannot go further in this direction as the reachability problem for PCD systems is undecidable for 3 dimensions (the positive and negative results are summarized in [7]). Similar limitations of the algorithmic approach can be found in [14].

Having concluded that these decidability games will not lead us far in terms of the original motivation, we focused for quite some time on timed systems. At that period Amir became a frequent visitor in Grenoble and our collaboration flourished, leading to various results such as the modeling of asynchronous circuits by timed automata [27] as well as an algorithm for controller synthesis for timed automata [29] in which we liberated synthesis from the use of tree automata (using them in the dense context would have been a nightmare). The algorithm has been embraced by the control community [20] and was extended to deal with hybrid systems with non-trivial continuous dynamics [5] and to solve time-optimal control problems [6].

In parallel to these scientific activities there were further developments such as conferences, joint projects, and other political affairs, culminating in the initiation of the *HSCC* series in 1998 [16]. Amir's presence in the steering committee of the conference contributed a lot to the credibility and popularity of the domain. He also co-chaired one of the conferences [28] but his own active interest in hybrid systems started declining – "the topic is in good hands" as he would say in his polite and diplomatic manner. Instead, he preferred to focus on the development of verification tools and on exploring new horizons in discrete verification, such as parameterized systems, abstraction, synthesis, compiler validation and many other topics.

Amir was lucky enough to see, during his lifetime, the adoption of various variants of temporal logic by the semi-conductor industry. He was particularly amused to observe the old fierce academic debates, concerning the adequacy of linear-time vs. branching-time logic, repeat themselves in industrial standardization committees, this time voiced by representatives of respected companies such as Intel and IBM. A European project around these specification languages, entitled *property-based system design*, gave me the last opportunity to collaborate with Amir on hybrid matters.

The hybrid niche in this project dealt with the extension of temporal logic to deal with analog signals [26], motivated by simulation-based verification (monitoring) of analog and mixed-signal circuits. The use of temporal logic to specify properties of real-valued signals complements traditional performance measures used in control, signal processing, robotics, circuit design and computational biology, which are not as good at specifying the progression of events in time. The adaptation to the continuous context of Amir's major contribution is an ongoing activity [9] that will hopefully lead to further insights concerning the interaction between discrete events and continuous change. The proliferation of temporal logic into such remote territories demonstrates the depth of Amir's insight and constitutes a prime example for the potential contribution of *computational thinking* to numerous scientific and engineering domains.

5. AFTERWORD

There is a Hebrew proverb saying roughly that "a stone thrown by a stupid person into a well will not be found by a thousand wise men". When the stone is thrown by a wise (and nice) person it can create and maintain whole scientific communities for lifetimes.

6. REFERENCES

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[2] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Grossman et al. [10], pages 209–229.

[3] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[4] R. Alur, T. Feder, and T.A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.

[5] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, 88(7):1011–1025, 2000.

[6] E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *HSCC*, pages 19–30, 1999.

[7] E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1):35–65, 1995.

[8] J.W. de Bakker, C. Huizing, W.-P. de Roever, and G. Rozenberg, editors. *Real-Time: Theory in Practice*, volume 600 of *LNCS*. Springer, 1992.

[9] A. Donzé and O. Maler. Robust satisfiability of temporal logic over real-valued signals. Submitted for publication, 2010.

[10] R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors. *Hybrid systems*, volume 736 of *LNCS*. Springer, 1993.

[11] D. Harel and A. Pnueli. On the development of reactive systems. In K. R. Apt, editor, *Logics and Models of Concurrent Systems*, NATO ASI Series, pages 477–498. Springer-Verlag, 1985.

[12] E. Harel, O. Lichtenstein, and A. Pnueli. Explicit clock temporal logic. In *LICS*, pages 402–413, 1990.

[13] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. In *CAV*, pages 460–463, 1997.

[14] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *STOC*, pages 373–382, 1995.

[15] T.A. Henzinger, Z. Manna, and A. Pnueli. Timed transition systems. In de Bakker et al. [8], pages 226–251.

[16] T.A. Henzinger and S. Sastry, editors. *Hybrid Systems: Computation and Control*, volume 1386 of *LNCS*. Springer, 1998.

[17] Y. Kesten and A. Pnueli. Timed and hybrid statecharts and their textual representation. In *FTRTFT*, pages 591–620, 1992.

[18] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs: A class of decidable hybrid systems. In *Hybrid Systems*, pages 179–208, 1992.

[19] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[20] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35:349–370, 1999.

[21] O. Maler. Hybrid systems and real-world computations. Unpublished manuscript, 1992.

[22] O. Maler. Control from computer science. *Annual Reviews in Control*, 26(2):175–187, 2002.

[23] O. Maler. On optimal and reasonable control in the presence of adversaries. *Annual Reviews in Control*, 31(1):1–15, 2007.

[24] O. Maler, Z. Manna, and A. Pnueli. From timed to hybrid systems. In de Bakker et al. [8], pages 447–484.

[25] O. Maler, D. Nickovic, and A. Pnueli. From MITL to timed automata. In *FORMATS*, pages 274–289, 2006.

[26] O. Maler, D. Nickovic, and A. Pnueli. Checking temporal properties of discrete, timed and continuous behaviors. In *Pillars of Computer Science, Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday*, pages 475–505, 2008.

[27] O. Maler and A. Pnueli. Timing analysis of asynchronous circuits using timed automata. In *CHARME*, pages 189–205, 1995.

[28] O. Maler and A. Pnueli, editors. *Hybrid Systems: Computation and Control*, volume 2623 of *LNCS*. Springer, 2003.

[29] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. In *STACS*, pages 229–242, 1995.

[30] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag New York, 1991.

[31] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer-Verlag New York, 1995.

[32] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An approach to the description and analysis of hybrid systems. In Grossman et al. [10], pages 149–178.

[33] A. Pnueli. The temporal logic of programs. In *Proc. 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 46–57, 1977.

[34] A. Pnueli. The Temporal Semantics of Concurrent Programs. *Theoretical Computer Science*, 13:45–60, 1981.

[35] A. Pnueli and E. Harel. Applications of temporal logic to the specification of real-time systems. In *FTRTFT*, pages 84–98, 1988.

[36] A. Pnueli and C.L. Pekeris. Free tidal oscillations in rotating flat basins of the form of rectangles and of sectors of circles. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 263(1138):149–171, 1968.