Ugo Montanari and Concurrency Theory

Roberto Gorrieri

Dipartimento di Scienze dell'Informazione, Università di Bologna
Mura A. Zamboni, 7, 40127 Bologna, Italy
email: gorrieri@cs.unibo.it

Introduction

This short note introduces the reader to the contributed papers in the area of concurrency theory that have been written in honour of Ugo Montanari. I also draw the attention to one of the contributions of Ugo in this area that, in my opinion, constitutes a pearl in theoretical computer science and would deserve to be investigated further.

Ugo Montanari is – undoubtedly – one the fathers of the theory of Concurrency. He started his research in this area about 25 years ago and contributed from the very beginning [29, 27, 25, 26] with many stimulating ideas and proposals. In my role of former student of Ugo, I had the chance and the honour to study first, and to follow then, some of these initial achievements. In my opinion, there is a *fil rouge* that links together many of Ugo's results in this area: the goal to find general, parametric, uniform, compositional and, possibly, universal theories that give comprehensive pictures of many subtheories that have been studied in isolation. The impressive list of papers that can be given to substantiate this claim includes at least the following [3, 7–10, 15, 16, 18]. Among the many specific lines of research in this setting, I would like to mention the goal to connect together distributed models of concurrency (such as Petri Nets and graph grammars) with the world of process algebras, whose theories were developed for automata-based models where concurrency is not a primitive concept. Many papers are devoted to reach this goal; among them, we mention at least the following [1, 2, 4, 6, 12–14, 19, 23]. In particular, in the last part of this note, I will draw the attention of the reader to one specific contribution of Ugo (and co-workers) that I personally consider a pearl in theoretical computer science: the definition of structural operational semantics for a process algebra that produces Petri nets instead of ordinary labeled transition systems, which is often known under the name of the *DDM approach*, after Degano, De Nicola and Montanari.

The contributed papers

The six papers that are collected in this section of the volume are valuable contributions that somehow span over Ugo's interests in this area.

– *On the Synthesis of Zero-safe Nets* by Philippe Darondeau is an interesting contribution about a class of Petri nets that Bruni and Montanari have proposed in [11], in order to compensate for the lack of mechanisms to force synchronization of transitions in P/T nets. In zero-safe nets it is indeed natural to define transactional behaviour where many transitions are executed atomically. While Bruni and Montanari have mainly studied the semantics of this class of nets, Darondeau here addresses the so-called synthesis problem: given the specification of transaction systems in the form of linear step automata in which the atomic steps between two stable states form a regular language, find the corresponding implementations producing and consuming resources in the form of bounded and well-behaved zero-safe nets. Darondeau proposes algorithmic techniques for this, resorting to a generalization of the theory of regions for zero-safe nets.

- The paper *Decomposition of Persistent Petri Nets*, by Eike Best, deals with persistent Petri nets, a class of nets characterized by the absence of conflicts in behaviour, while admitting shared input places among distinct transitions. The main result is a theorem stating that, in a bounded persistent net with a strongly connected marking graph, any cycle of the marking graph can be decomposed into a set of disjoint simple cycles. Starting from a known result by Keller, the author develops the proof of the main theorem, showing, along the way, several specific properties of persistent nets.
- In the paper *Secure Data Flow in a Calculus for Context Awareness*, Doina Bucur and Mogens Nielsen present a process calculus for modelling context-aware computing in the setting of ubiquitous computing systems. The process calculus, based on Mobile Ambients, is equipped with new operators in order to capture context provision and context discovery. The basic idea is to introduce macro definitions residing inside ambients and macro calls, that may cross the ambients boundaries. Moreover, since the network is highly reconfigurable, a type system is proposed in order to enforce distributed security policies of the hosting locations. Each ambient has a security policy in the form of a process type, which has to be satisfied by any internal ambient in the ambients hierarchy. The validation of security policies is realized by a combination of static and dynamic type checking.
- Paola Quaglia, a former student of Ugo, in the paper *On Beta-binders communications*, focuses on the Beta-binder calculus, which has been proposed in order to make easier the description and the analysis of biological systems. This kind of language follows the line initially proposed by Fontana and Buss in 1996, and subsequently applied by Regev and Shapiro in 2001 to use process calculi in the biological context. The beta-binders language is close to Ambient-like calculi developed by Cardelli, even if without nesting of ambients, with the distinguishing feature to permit a form of communication in which the matching of input and output is rather loose. The main motivation of the paper is to help understanding the differences between Ambient and Beta-binders. This is done by developing a novel labeled semantics for the beta-binders, which complements the one originally defined as a reduction semantics. Then Quaglia presents a formal relation between the two semantics, stipulating that the two are equivalent.
- The paper titled *On the asynchronous nature of the asynchronous π-calculus*, by R. Beauxis, C. Palamidessi and F.D. Valencia, supports the claim that

the asynchronous π-calculus is really a formalism for asynchronous communication, as commonly intended in the community of distributed systems. To this aim, a new π-based calculus with buffers is presented, whose semantics more clearly reflects the intuition of asynchrony, as all the communication among processes are mediated via buffers. This new calculus is three-fold, as buffers can be either bags, or stacks or queues. The first interesting result is that the bag π-calculus is in strict correspondence with the asynchronous π-calculus in either direction. Such a correspondence is so strict that no doubt can be raised about the fact the asynchronous π-calculus is indeed a calculus for asynchronous communication. On the contrary, the authors show that no similar result can be obtained when buffers are stacks or queue.

– The paper titled *StonyCam: a Formal Framework for Modeling, Analyzing and Regulating Cardiac Myocytes*, by E. Bartocci, F. Corradini, R. Grosu, E. Merelli, O. Riganelli and Scott A. Smolka, is a brief and clear overview of the StonyCam research program, jointly involving the Universities of Stony Brook and Camerino. The problem of cardiac electrical disturbances, such as atrial fibrillation, is introduced along with the need for a concretely useful analysis and prediction tools. Throughout the paper a bird's eye view is offered regarding all the related work and the main techniques used so far. In particular it is described how to model excitable cells using the theory of hybrid automata, obtaining models amenable to formal analysis with a tenfold speedup in simulation time with respect to classical systems of non-linear differential equations. As an example it is also described an interesting method for decting emergent behaviours.

One of Ugo's pearls

Many are the excellent contributions that Ugo has achieved during his brilliant and exciting research life. And I am not old (or good) enough to remember all of them! But for sure one of such results that I had the opportunity to apply in some of my initial papers, as a student exercise [24, 20], was the technique to give a Petri net semantics to process algebras, CCS in particular [28, 23, 21, 22]. This technique – Ugo developed in joint work with Pierpaolo Degano and Rocco De Nicola, and for this reason it has been sometimes called the *DDM technique* – attracted a lot of interest, as an attempt to reconcile two mainstreams of concurrency theory that were developed independently and somehow in antagonism: Petri nets and process algebras. Other researchers contributed to the success of this technique, in particular Ernst-Rüdiger Olderog, who made it popular with his significant textbook [17].

The key aspect of this technique is a non-trivial extension of Plotkin' Structual Operational Semantics [30, 5] (SOS for short), which was originally conceived in order to produce (labeled) transitions systems as semantic models of programs, to the more elaborate setting of Petri Nets. In the original Plotkin's approach, the terms of an algebra of programs are the states of the transition systems, while the transitions are generated by means of an inference system, defined inductively on the syntax of terms.

The very nice point is that the infinite (in states and transitions) labeled transition system is defined implicitly with two finite, generative structures: a grammar for the states/processes and a finite set of axioms and inference rules for the transitions (that represent the evolution of a global state into another global state). The transition system generated in this way is for the whole language, but of course, one can focus his/her attention only on the part that is reachable from a particular state/process. In the case of CCS, this SOS is particularly concise and elegant.

Degano, De Nicola & Montanari have transposed this idea to Petri nets by defining a grammar for generating the infinite set of CCS-oriented places of the net and a finite set of axioms and inference rules generating the infinite set of CCS-oriented net transitions. The global state (called marking in the Petri net terminology) of the net is represented as a (multi-)set of places and the transitions represent the evolution of a set of places (not necessarily the global state) into another (multi-)set of places. There is a precise mapping *dec* of CCS processes to markings of the net and the net associated to a CCS process E is the net reachable from the marking $dec(E)$. The resulting Petri net for CCS, on the one hand, offers two main advantages:

- it is now possible to investigate non-interleaving semantics for CCS, based on some notions of causality and conflict [21] that are somehow primitive in Petri nets while completely absent in ordinary transition systems;
- it is possible to apply well-known Petri nets analysis techniques also to CCS processes (e.g., reachability, coverability, net invariants).

On the other hand, this cross-fertilization between the area of Petri nets and process algebra produced benefits also for the area of Petri nets, where some ideas developed in the process algebra community were imported, notably bisimulation equivalence.

If one tries to look with a up-to-date perspective to what has been done, one immediately observes that there is a lot of work that could have been done and that still deserves to be done. In fact, the Petri net for CCS that DDM developed is just one example of a missing general theory of SOS for Petri nets. The theory of transition systems specification via SOS has a large body of results (e.g., formats that ensure that bisimulation is a congruence) that could conceivably approached also for a to-be-developed theory of Petri nets specification via SOS. For instance, it could be interesting to study which formats should have the inference rules to ensure that a Petri net is safe, or of a special class (elementary, rather than Place/Transitions or else), or such that history preserving bisimulation is a congruence. Another aspect is the meaning of negative premises (that might perhaps be better modeled with the use of inhibitors). Summing up, I think that the seminal papers that Ugo and co-workers have produced in this area should be generalized to start a new line of cross-fertilization between the Petri net area and the process algebra community.

Conclusion

I think that probably the most impressive distinguishing quality of Ugo is his ability to be a real *maestro*, in following the scientific work of many students, most of which have taken positions in various Italian and international Universities or research insitutions.

I would like to end this short note by quoting (part of) my dedication to Ugo I wrote in my PhD thesis, that somehow reflects this aspect of Ugo's character:

> *Writing a Ph.D thesis under the supervision of Ugo Montanari has been undoubtedly a unique experience. He is a shine researcher full of scientific interests which tries to instill in the mind of his students and collaborators with Socratic maieutic.*

And indeed, Ugo often acts as a new Socrates in guiding the initial steps of his many students. And he is so strongly affectionate to them, that he would never part from them! (And this explains why many of them still live in Pisa ...)

References

1. Paolo Baldan, Fabio Gadducci, Ugo Montanari: *Modelling Calculi with Name Mobility using Graphs with Equivalences.* Electr. Notes Theor. Comput. Sci. 176(1): 85-97 (2007)
2. Roberto Bruni, Hernán C. Melgratti, Ugo Montanari: *Event Structure Semantics for Nominal Calculi.* CONCUR 2006, LNCS 4137:295-309 (2006).
3. Ugo Montanari, Marco Pistore: *History-Dependent Automata: An Introduction.* SFM 2005, LNCS 3465:1-28 (2005)
4. Roberto Bruni, Ugo Montanari: *Concurrent models for Linda with transactions.* Mathematical Structures in Computer Science 14(3): 421-468 (2004)
5. Gordon D. Plotkin: *A structural approach to operational semantics.* J. Log. Algebr. Program. 60-61: 17-139 (2004)
6. Paolo Baldan, Andrea Corradini, Ugo Montanari: *Bisimulation Equivalences for Graph Grammars.* Formal and Natural Computing 2002, LNCS 2300, 158-190 (2002).
7. Andrea Corradini, Reiko Heckel, Ugo Montanari: *Compositional SOS and beyond: a coalgebraic view of open systems.* Theor. Comput. Sci. 280(1-2): 163-192 (2002)
8. Roberto Bruni, José Meseguer, Ugo Montanari, Vladimiro Sassone: *Functorial Models for Petri Nets.* Inf. Comput. 170(2): 207-236 (2001)
9. Gian Luigi Ferrari, Ugo Montanari: *Parameterized Structured Operational Semantics.* Fundam. Inform. 34(1-2): 1-31 (1998)
10. Gian Luigi Ferrari, Ugo Montanari, Miranda Mowbray: *Structured Transition Systems with Parametric Observations: Observational Congruences and Minimal Realizations.* Mathematical Structures in Computer Science 7(3): 241-282 (1997)
11. Roberto Bruni, Ugo Montanari: *Zero-safe nets, or transition synchronization made simple.* Electr. Notes Theor. Comput. Sci. 7: (1997)
12. Ugo Montanari, Marco Pistore:*Concurrent semantics for the pi-calculus.* Electr. Notes Theor. Comput. Sci. 1 (1995)
13. Roberto Gorrieri, Ugo Montanari: *On the Implementation of Concurrent Calculi in Net Calculi: Two Case Studies.* Theor. Comput. Sci. 141(1& 2): 195-252 (1995)

14. Ugo Montanari, Daniel Yankelevich: *Combining CCS and Petri Nets Via Structural Axioms.* Fundam. Inform. 20(1/2/3): 193-229 (1994)
15. Pierpaolo Degano, Rocco De Nicola, Ugo Montanari: *Universal Axioms for Bisimulations.* Theor. Comput. Sci. 114(1): 63-91 (1993)
16. Ugo Montanari, Daniel Yankelevich: *A Parametric Approach to Localities.* ICALP 1992, LNCS 623:617-628 (1992)
17. E.R. Olderog: *Nets, Terms and Formulas.* Cambridge Tracts in Theoretical Computer Science 23, Cambridge University Press (1991)
18. Gian Luigi Ferrari, Ugo Montanari: *Towards the Unification of Models for Concurrency.* CAAP 1990, LNCS 431:162-176 (1990)
19. Pierpaolo Degano, Rocco De Nicola, Ugo Montanari: *A Partial Ordering Semantics for CCS.* Theor. Comput. Sci. 75(3): 223-262 (1990)
20. Antonio Brogi, Roberto Gorrieri: *A Distributed, Net-Oriented Semantics for Delta Prolog.* TAPSOFT'89, LNCS 351:162-177 (1989)
21. Pierpaolo Degano, Rocco De Nicola, Ugo Montanari: *On the Consistency of "Truly Concurrent" Operational and Denotational Semantics.* LICS 1988, pages 133-141, IEEE CS Press (1988)
22. Pierpaolo Degano, Rocco De Nicola, Ugo Montanari: *Partial orderings descriptions and observations of nondeterministic concurrent processes.* REX Workshop 1988, LNCS 354:438-466 (1988)
23. Pierpaolo Degano, Rocco De Nicola, Ugo Montanari: *A Distributed Operational Semantics for CCS Based on Condition/Event Systems.* Acta Inf. 26(1/2): 59-91 (1988)
24. Pierpaolo Degano, Roberto Gorrieri, Sergio Marchetti: *An Exercise in Concurrency: a CSP Process as a Condition/Event System.* European Workshop on Applications and Theory of Petri Nets 1987, LNCS 340:85-105 (1988)
25. Pierpaolo Degano, Ugo Montanari: *A model for distributed systems based on graph rewriting.* J. ACM 34(2): 411-449 (1987)
26. Pierpaolo Degano, Ugo Montanari: *Concurrent Histories: A Basis for Observing Distributed Systems.* J. Comput. Syst. Sci. 34(2/3): 422-461 (1987)
27. Pierpaolo Degano, Ugo Montanari: *Liveness Properties as Convergence in Metric Spaces.* STOC 1984, ACM Press, 31-38 (1984)
28. Pierpaolo Degano, Rocco De Nicola, Ugo Montanari: *CCS is an (Augmented) Contact Free C/E System.* Mathematical Models for the Semantics of Parallelism 1986 (Advanced School), LNCS 280:144-165 (1986)
29. Ilaria Castellani, Ugo Montanari: *Graph Grammars for distributed systems.* In Graph-Grammars and Their Application to Computer Science 1982, LNCS 153:20-38 (1982)
30. Gordon D. Plotkin: *A structural approach to operational semantics.* Technical Report DAIMI FN-19, Aarhus University (1981)