

# THE CONCURRENCY COLUMN

BY

**NOBUKO YOSHIDA**

Department of Computing

Imperial College London

180 Queen's Gate, London, SW7 2AZ

[n.yoshida@imperial.ac.uk](mailto:n.yoshida@imperial.ac.uk), <http://mrg.doc.ic.ac.uk/>

# INTERVIEWS WITH THE 2020 CONCUR TEST-OF-TIME AWARD RECIPIENTS

Luca Aceto  
ICE-TCS, School of Computer Science,  
Reykjavik University  
Gran Sasso Science Institute, L'Aquila  
[luca@ru.is](mailto:luca@ru.is), [luca.aceto@gssi.it](mailto:luca.aceto@gssi.it)

Earlier this year, the International Conference on Concurrency Theory (CONCUR) and the IFIP 1.8 Working Group on Concurrency Theory decided to establish the CONCUR Test-of-Time Award to recognize important achievements in Concurrency Theory that were published at the CONCUR conference and have stood the test of time. All papers published at CONCUR between 1990 and 1995 were eligible for the first installment of the award, which will be presented at the 31st International Conference on Concurrency Theory (CONCUR 2020). The conference will be held on line in the period 1–4 September 2020, and has Igor Konnov and Laura Kovacs as chairs of the program committee.

The award winners for the CONCUR Test-of-Time Awards 2020, together with the award citations, may be found at

<https://concur2020.forsyte.at/test-of-time/index.html>

They were selected by a jury composed of Jos Baeten, Patricia Bouyer-Decitre, Holger Hermanns, Alexandra Silva and myself (as chair of the jury).

In order to give visibility to the CONCUR Test-of-Time Award, and to inspire young researchers to build on the work of the award recipients and to learn from their experiences, I decided to interview all the award recipients, namely Søren Christensen, Rob van Glabbeek, Hans Hüttel, Nancy Lynch, Davide Sangiorgi, Roberto Segala and Colin Stirling. I received positive answers from all of them, apart from Søren Christensen, who has worked in the private sector since he finished his PhD, and Colin Stirling, who has retired last spring.

The interviews can be accessed from my professional blog at

<https://processalgebra.blogspot.com/>

and from the above-mentioned conference web page. This write-up collects them under one roof and I hope that it will make them easily accessible to everyone. I encourage young researchers of all ages to read them and be inspired by the work of those distinguished colleagues of ours.

## Interview with Rob van Glabbeek

**Luca:** You receive one of the two CONCUR Test-of-Time Awards for the period 1990–1993 for your companion papers on “The Linear Time-Branching Time Spectrum”, published at CONCUR 1990 and 1993. Could you tell us briefly what spurred you to embark on the encyclopaedic study of process semantics you developed in those two papers and what were your main sources of inspiration?

**Rob:** My PhD supervisors, Jan Willem Klop and Jan Bergstra, were examining bisimulation semantics, but also failures semantics, in collaboration with Ernst-Rüdiger Olderog, and at some point, together with Jos Baeten, wrote a paper on ready-trace semantics to deal with a priority operator. I had a need to see these activities, and also those of Tony Hoare and Robin Milner, as revealing different aspects of the same greater whole, rather than as isolated research efforts. This led naturally to an abstraction of their work in which the crucial difference was the semantic equivalence employed. Ordering these semantics in a lattice was never optional for me, but necessary to see the larger picture. It may be in my nature to relate different solutions to similar problems I encounter by placing them in lattices.

**Luca:** Did you imagine at the time that your award-winning papers would have so much impact?

**Rob:** I didn’t think much about impact in those days. I wrote the papers because I felt the need to do so, and given that, could just as well publish them. Had I made an estimate, I would have guessed that my papers would have impact, as I imagined that things that I find necessary to know would interest some others too. But I would have underestimated the amount of impact, in part because I had grossly underestimated the size of the research community eventually working on related matters.

**Luca:** In your papers, you present characterizations of process semantics in relational form, via testing scenarios, via modal logics and in terms of (conditional) equational axiomatizations. Of all those characterizations, which ones do you think have been most useful in future work? To your mind, what are the most interesting or unexpected uses in the literature of the notions and techniques you developed in the “The Linear Time-Branching Time Spectrum”?

**Rob:** I think that all these characterizations together shed the light on process semantics that makes them understandable and useful. Relational characterizations, i.e., of (bi)simulations, are often the most useful to grasp what kind of relation one has, and this is essential for any use. But modal characterizations are intimately linked with model checking, which is one of the more successful verification tools. Equational characterizations also find practical use in verification, e.g., in normalization. Yet, they are most useful in illustrating the essential difference between process semantics, and thus form a guide in choosing the right one for an application. To me, the testing scenarios are the most useful in weighing how realistic certain semantic identifications are in certain application contexts.

**Luca:** How much of your later work has built on your award-winning papers? What follow-up results of yours are you most proud of and why?

**Rob:** About one third, I think.

- My work on action refinement, together with Ursula Goltz, extends the classification of process semantics in a direction orthogonal to the linear time - branching time spectrum. I am proud of this work mostly because, for a distinct class of applications, action refinement is a great criterion to tell which semantic equivalences one ought to prefer over others. These process semantics have also been used in later work on distributability with Ursula and with Jens-Wolfhard Schicke Uffmann. Distributability tells us which distributed systems can be cast as sequential systems cooperating asynchronously, and which fundamentally cannot.
- I wrote several papers on Structural Operational Semantics, many jointly with Wan Fokkink, aiming to obtain congruence results for semantic equivalences. These works carefully employed the modal characterizations from my spectrum papers. I consider this work important because congruence properties are essential in verification, as we need compositionality to avoid state explosions.
- I used a version of failure simulation semantics from my second spectrum paper in work with Yuxin Deng, Matthew Hennessy, Carroll Morgan and Chenyi Zhang, characterizing may- and must-testing equivalences for probabilistic nondeterministic processes. Probability and nondeterminism interact in many applications, and I think that may- and must-testing are the right tools to single out the essence of their semantics.

**Luca:** If I remember correctly, your master thesis was in pure mathematics. Why did you decide to move to a PhD in computer science? What was it like to work

as a PhD student at CWI in the late 1980s? How did you start your collaborations with Jos Baeten and fellow PhD student Frits Vaandrager, amongst others?

**Rob:** I was interested in mathematical logic already since high school, after winning a book on the foundations of logic and set theory in a mathematics Olympiad. I found the material in my logic course, thought at the University of Leiden by Jan Bergstra, fascinating, and when doing a related exam at CWI, where Jan had moved to at that time, he asked me to do a PhD on related subjects, I saw no reason not to. But he had to convince me first that my work would not involve any computer science, as my student advisor at The University of Leiden, Professor Claas, had told us that this was not a real science. Fortunately, Jan had no problem passing that hurdle.

I was the third member of the team of Jan Bergstra and Jan Willem Klop, after Jos Baeten had been recruited a few months earlier, but in a more senior position. Frits arrived a month later than me. As we were exploring similar subjects, and shared an office, it was natural to collaborate. As these collaborations worked out very well, they became quite strong.

CWI was a marvelous environment in the late 1980s. Our group gradually grew beyond a dozen members, but there remained a much greater cohesion than I have seen anywhere else. We had weekly process algebra meetings, where each of us shared our recent ideas with others in the group. At those meetings, if one member presented a proof, the entire audience followed it step by step, contributing meaningfully where needed, and not a single error went unnoticed.

**Luca:** Already as PhD student, you took up leadership roles within the community. For instance, I remember attending a workshop on “Combining Compositionality and Concurrency” in March 1988, which you co-organized with Ursula Goltz and Ernst-Rüdiger Olderog. Would you recommend to a PhD student that he/she become involved in that kind of activities early on in their career? Do you think that doing so had a positive effect on your future career?

**Rob:** Generally I recommend this. Although I co-organized this workshop because the circumstances were right for it, and the need great; not because I felt it necessary to organize things. My advice to PhD students would not be specifically to add a leadership component to their CV regardless of the circumstances, but instead do take such an opportunity when the time is right, and not be shy because of lack of experience.

This workshop definitively had a positive effect on my future career. When finishing my PhD I inquired at Stanford and MIT for a possible post-doc position, and the good impressions I had left at this workshop were a factor in getting offers from both institutes. In fact, Vaughan Pratt, who was deeply impressed with work I had presented at this workshop, convinced me to apply straight away for an assistant professorship, and this led to my job at Stanford.

**Luca:** I still have vivid memories of your talk at the first CONCUR conference in 1990 in Amsterdam, where you introduced notions of black-box experiments on processes using an actual box and beer-glass mats. You have attended and contributed to many of the editions of the conference since then. To your mind, how has the focus of CONCUR changed since its first edition in 1990?

**Rob:** I think the focus on foundations has shifted somewhat to more applied topics, and the scope of “concurrency theory” has broadened significantly. Still, many topics from the earlier CONCURs are still celebrated today.

**Luca:** The last forty years have seen a huge amount of work on process algebra and process calculi. However, the emphasis on that field of research within concurrency theory seems to have diminished over the last few years, even in Europe. What advice would you give to a young researcher interested in working on process calculi today?

**Rob:** Until 10 years ago, many people I met in industry voiced the opinion that formal methods in general, and process algebra in particular, has been tried in the previous century, and had been found to not scale to tackle practical problems. But more recently this opinion is on the way out, in part because it became clear that the kind of problems solved by formal methods are much more pressing than originally believed, and in part because many of the applications that were deemed too hard in the past, are now being addressed quite adequately. For these reasons, I think work on process calculi is still a good bet for a PhD study, although a real-world application motivating the theory studied is perhaps harder needed than in the past, and toy examples just to illustrate the theory are not always enough.

**Luca:** What are the research topics that currently excite you the most?

**Rob:** I am very excited about showing liveness properties, in Lamport’s sense, telling that a modeled system eventually achieves a desirable outcome. And even more in creating process algebraic verification frameworks that are in principle able to do this. I have come to believe that traditional approaches popular in process algebra and temporal logic can only deal with liveness properties when making fairness assumptions that are too strong and lead to unwarranted conclusions. Together with Peter Hoefner I have been advocating a weaker concept of fairness, that we call justness, that is much more suitable as a basis for formulating liveness properties. Embedding this concept into the conceptual foundations of process algebra and concurrency theory, in such a way that it can be effectively used in verification, is for a me a challenging task, involving many exciting open questions.

A vaguely related subject that excites me since the end of last year, is the extension of standard process algebra with a time-out operator, while still abstaining

from quantifying time. I believe that such an extension reaches exactly the right level of expressiveness necessary for many applications.

Both topics also relate with the impossibility of correctly expressing mutual exclusion in standard process algebras, a discovery that called forth many philosophical questions, such as under which assumptions on our hardware is mutual exclusion, when following the formal definitions of Dijkstra and others, even theoretically implementable.

## Interview with Hans Hüttel

**Luca:** You receive one of the two CONCUR Test-of-Time Awards for the period 1990–1993 for the paper “Bisimulation Equivalence is Decidable for all Context-Free Processes” you published with Søren Christensen and Colin Stirling at CONCUR 1992. Could you tell us briefly what the context for that work was and how it came about?

**Hans:** I was working on my PhD which was on the topic of decidability of behavioural equivalences for process calculi that allow processes to have infinite state spaces. Baeten, Bergstra and Klop had proved that strong bisimilarity is in fact decidable for the BPA calculus even though the class of trace languages for BPA is exactly the class of context-free languages. The result only held for normed BPA processes, that is, processes that could always terminate. Colin was my supervisor in Edinburgh, and he suggested that I try to find a simpler proof of that result (much can be said about the original proof, but simple it was not!). This turned out to be really interesting; I met Didier Caucal from IRISA who had found a nice, much shorter proof that relied on finite characterizations and I was able to use his notion of self-bisimilarity to prove that a tableau system for bisimilarity was sound and complete with respect to bisimilarity. Jan Friso Groote, who was a PhD student at the CWI at the time, and myself proved that all other known equivalences are undecidable for normed BPA (and therefore *a fortiori* also for full, unnormed BPA). But one problem that Colin and I were never able to solve was if the decidability result also held for the full BPA calculus.

Søren arrived in Edinburgh in 1990 and we shared a flat there, in St Leonards Bank just across from Holyrood Park. We hung out with the same people in the LFCS, many of whom were clever Italians such as Davide Sangiorgi and clever quasi-Italians such as Marcelo Fiore. It is no surprise that Søren and I often discussed our work or that he, given that Colin was also his supervisor, moved on to study decidability problems as well, the only difference being that Søren was mostly interested in the BPP calculus that has parallel composition instead of sequential composition.

When I left Edinburgh at the end of August of 1991, Colin and I had managed to make some progress on the decidability problem for unnormed BPA. We realised that a finite characterization of the maximal bisimulation á la Caucal was the way ahead but the actual characterization escaped us. When I returned for my viva in December, Søren had found an important lemma on finite characterizations and everything fell into place. The decidability result relied on proving that bisimilarity is semi-decidable using the finite characterization; non-bisimilarity was already known to be semi-decidable.

**Luca:** Both Søren Christensen and you wrote the award-winning paper as PhD students in Edinburgh. Could you tell us about the research environment in Edinburgh at that time, and the influence it had on you then and in the rest of your career?

**Hans:** The LFCS in Edinburgh was a great place to be at the time. Robin Milner was still around and busy working on the pi-calculus together with Davide Sangiorgi, who was his student at the time. Watching them at the blackboard was interesting, even though the rest of us often could not follow their discussions! Gordon Plotkin was there (and still is, fortunately). Rod Burstall was still active, and so was Mike Fourman. Plus many, many others. There was always someone to talk to, and it was perfectly legitimate to have an interest in basic research. Unlike the other professors, Colin had no background in maths – he was originally a philosopher! – and maybe that was why he was trying to avoid heavy mathematical lingo, trying to be simple and precise in his writing at the same time. I learned a lot from him, also in that respect.

After Søren returned to Denmark, he left academia and got a job in the private sector. He still lives near Aarhus. Sadly we lost touch with each other, in all likelihood because our lives turned out so different. We got back in touch recently, when we were told about the reward and we look forward to finally meeting each other again. Colin retired recently, and I really hope to see him again, when I am able to travel to Scotland again some day.

**Luca:** How much of your later work has built on your award-winning paper? What follow-up results of yours are you most proud of and why?

**Hans:** I worked on decidability issues for another few years after that but there was no-one to talk to in Aalborg, or rather, there was no-one there that shared my interest in the area at the time. What is more, the open problems that remained were major challenges. Some were only solved much later (such as the decidability of bisimilarity for pushdown automata, a result due to Sénizergues, the proof later greatly simplified by Colin) or remain open (such as the decidability of weak bisimilarity for BPA). Eventually I drifted down the slippery slope and became interested in other, related topics, and focused somewhat more directly on program

properties. The follow-up result that most directly relates to my paper with Søren and Colin is the result from 1994 that bisimilarity is also the only equivalence that is decidable for BPP. This is a result that I am also quite proud of. It grew out of discussions with Javier Esparza and Yoram Hirshfeld, when I was back in Edinburgh for a while in 1993. Ironically, there was a subtle flaw in the proof that Naoki Kobayashi discovered many years later. Naoki and one of his student found out how to repair the proof, and the three of us co-authored the journal version that came out many years later. The reason why Naoki became interested in BPP was that he was trying to use the calculus as a behavioural type system for a pi-calculus. As it happened, this was also the route that I had taken.

Those of my later results that I am most proud of have to do with this area: My work on type systems for psi-calculus (CONCUR 2011 and later) and a session type system for bounded name use in the pi-calculus (Acta Informatica 2019). The latter paper also has a CONCUR connection; it grew out of attending a talk by Roland Meyer at CONCUR 2013 and wondering why they were not trying to characterize notions of name-boundedness in the pi-calculus by means of a type system. It turned out that Meyer et al. were not familiar with type systems at all. It took me an awful long time to work out a type-based characterization (using session types), as you can probably tell.

As you can tell, I have not worked on bisimulation for quite some time. Not that there is anything wrong with bisimulation, of course, but if one wants results that are applicable for automated reasoning about program behaviour, decidability is important. One can then either choose to go for a less expressive model of computation (which is what researchers in the model checking community do) or keep the model of computation and go for sound, but incomplete characterizations of program properties (which is what researchers interested in type systems and related static analysis methods do). I ended up in the latter camp.

**Luca:** To your mind, what are the most interesting or unexpected uses in the literature of the notions and techniques you developed in “Bisimulation Equivalence is Decidable for all Context-Free Processes”?

**Hans:** BPA, BPP and similar process calculi can be used as the basis of behavioural type systems, and I am thrilled that my old research interests and my current research interests are so directly related.

In 2018 I discovered that Vasco Vasconcelos and Peter Thiemann had devised a session type system for a functional programming language in which session types were not regular expressions (which is essentially what they are in the original work by Honda, Kubo and Vasconcelos) but BPA processes. Vasco and Peter knew that type equivalence should be decidable but they were not so familiar with our results from the early 1990s. At POPL 2019 I attended a talk by Andreia Mordido, one of Vasco’s collaborators, and she mentioned our paper from CONCUR

1992! Later that day, I ended up talking to Andreia and Vasco about my work on BPA from all those years ago.

**Luca:** The last forty years have seen a huge amount of work on process algebra and process calculi. However, the emphasis on that field of research within concurrency theory seems to have diminished over the last few years, even in Europe. What advice would you give to a young researcher interested in working on process calculi today?

**Hans:** That they should still work on process calculi! There remains a lot of interesting work to be done. One reason why research topics drift in and out of focus is simply that researchers lose interest; it is hardly ever the case that a topic runs out of interesting research questions. Another reason is rather grim: Basic research is nowhere near as well-respected as it used to be. If you look at the funding schemes that we have today, only the very successful few can get funding for basic research; I am not among them and it does get frustrating quite often. Most topics these days deal with applied research. There is nothing wrong with applied research per se; some of what I do is on that side of things, but there has to be more to research than that.

**Luca:** What are the research topics that currently excite you the most?

**Hans:** I have slowly become more and more interested in programming language theory, and some of my current collaborators have taken a similar route, beginning in concurrency theory, drifting into the world of behavioural type systems and finally wanting to apply all of their skills to actual programming languages. Right now Mario Bravetti, Adrian Francalanza, António Ravara and myself are involved in working on a behavioural type system related to the Mungo tool for a subset of Java. I am fortunately enough to have had three exceptionally clever and productive MSc students involved as well, and this has been extremely helpful. Mungo is in many ways the work of Simon Gay and Ornela Dardha at Glasgow University, and they, too, began their careers working on process calculi.

**Luca:** You have a keen interest in teaching at university level. Is there anything you think we should do better in conveying the beauty and usefulness of theoretical computer science in general, and concurrency theory in particular, to our students today? Do you have any thoughts you'd like to share with us on how to combine digital resources and in-person meetings in the teaching of subjects in the theory of computing?

**Hans:** Personally I think there is a tendency to present any academic topic – and in particular topics in the mathematical sciences, broadly speaking – in such a way that the definitions and theorems appear as if they fell from the heavens, fully formed. As any researcher will know, that is certainly *not* how they came

about. In this way, we focus a lot on what Imre Lakatos and Karl Popper called the *context of justification*. A well-honed presentation of a theory may appear to be beautiful, but in my opinion the actual beauty is the one that one experiences when one has finally understood the theory and understands why the definitions and theorem turned out the way they did – that is, one must understand the *context of discovery*. I think problem-based learning (which is something that we talk about a lot at Aalborg University and at Roskilde University) is the key here, because it puts the focus on student-centered active learning.

I have lectured a lot over the years but since 2013 I have drifted away from traditional lectures towards flipped teaching in which I use pre-recorded podcasts (of my own making) that students can watch whenever they want; I then use the plenary sessions for activities that focus on active learning. I by far prefer having a dialogue with students that focuses on the problems that they encounter to the monologue style of a lecture. All my teaching activities are now flipped and I am happy to say that some of my colleagues are now thinking along similar lines. It is always good to have someone to talk to.

## **Interview with Nancy Lynch and Roberto Segala**

**Luca:** You receive one of the two CONCUR Test-of-Time Awards for the period 1992–1995 for your paper “Probabilistic simulations for probabilistic processes”, presented at CONCUR 1994. Could you tell us briefly what spurred you to develop the “simple” probabilistic automaton model introduced in that paper and how the ideas underlying that notion came about?

**Roberto:** We were studying randomized distributed algorithms and we noted that several algorithms suffered from subtle errors. We were looking for a formal, rigorous way to study the correctness of those algorithms and in particular we were trying to extend the hierarchical approach used for I/O automata. We knew that there was already an extensive literature within concurrency theory, but we were not able to use the theory to express even the simple idea that you may choose internally between flipping a fair or a biased coin; the existing approaches were extending labeled transition systems by adding probabilities to the arcs, losing the ability to distinguish between probabilistic and nondeterministic choice. The fact that the axioms for nondeterministic choice were not valid in most probabilistic calculi was further evidence of the problem. Starting from the observation that in the non-probabilistic world a step of a distributed algorithm corresponds to a transition of an I/O automaton, we tried to extend the notion of transition so that the process of flipping a coin could be represented by the “new” entity. This led to the idea of “replacing points with measures” within ordinary automata and/or ordinary labeled transition systems. The idea worked very well for our analysis of

randomized distributed algorithms, but we also wanted to validate it by checking whether it would simplify the concurrency-theoretical approach to probability. We decided to use simulation relations for our test, and indeed it worked and led to the CONCUR 1994 paper.

**Nancy:** For background for this paper, we should recall that, at the time, there was a divergence in styles of modeling for distributed systems, between different concurrency theory research communities. The style I had been using since the late 70s for modeling distributed algorithms involved interactive state machines, such as I/O automata, which are based on a set-theoretic foundation. Such models are good for describing distributed algorithms, and for analyzing them for both correctness and costs. On the other hand, the predominant style in the European concurrency community was more syntactic, based on logical languages like PCTL or various process algebras. These were good for formal verification, and for studying expressive power of different formalisms, but not great for analyzing complicated distributed algorithms. Our models were in the general spirit of the Labeled Transition System (LTS) models previously studied in Europe. When Roberto came to MIT, he and I set about to extend prior modeling work for distributed systems to the probabilistic setting. To do this, we considered both the set-theoretic and logical approaches. We needed to bridge the gap between them, which led us to the ideas in this paper.

**Luca:** How much of your later work has built on your CONCUR 1994 paper? What follow-up results of yours are you most proud of and why?

**Roberto:** Besides using the results of the paper for the analysis of several randomized distributed algorithms, we worked jointly as well as independently on the study of the theory and on the analysis of security protocols. In collaboration with Frits Vaandrager, we were able to discover different ways to analyze security in a hierarchical and compositional way. Furthermore, since in simple probabilistic automata probability and nondeterminism are well separated, it was easy to include computational complexity into the analysis of security protocols. This is work I did in collaboration with Andrea Turrini. The clear separation between probability and nondeterminism turned out to extend our approach to real-time models, leading to notions like Probabilistic Timed Automata and to several of the elements behind the first sketches of the probabilistic model checker PRISM in collaboration with the group of Marta Kwiatkowska.

**Nancy:** Roberto and I continued working on probabilistic models, in the set-theoretic style, for several years. As Roberto notes, Frits Vaandrager also became a major collaborator. Some of our more recent papers following this direction are:

- Compositionality for Probabilistic Automata, CONCUR 2003,

- Observing Branching Structure through Probabilistic Contexts, SIAM Journal of Computing 2007,
- Task-Structured Probabilistic I/O Automata, JCSS 2018.

This last paper provides a kind of compositional foundation for combining security protocols. The key idea was to weaken the power of the adversary, making it more “oblivious”. In other related work, my students and I have worked extensively on probabilistic distributed algorithms since then. The models are similar to those developed in this early paper. Examples include wireless network algorithms, and more recently, biologically-inspired algorithms, such as insect colony algorithms and neural network algorithms. I can’t pinpoint a specific result that relies heavily on the 1994 paper, but that paper certainly provided inspiration and foundation for the later work.

**Luca:** Did you imagine at the time that the CONCUR 1994 paper and its journal version would have so much impact? In your opinion, what is the most interesting or unexpected use in the literature of the notions and techniques you developed in your award-winning paper?

**Roberto:** We knew that the change of view proposed in the paper would have simplified the study and extension of several other concepts within concurrency theory, but we did not have any specific expectations. The collaboration on the PRISM project and on Probabilistic Timed Automata made more evident that there was a connection with Markov Decision Processes, which lead to a fruitful cross-fertilization between artificial intelligence, model checking, and concurrency theory. It was a long exchange of ideas between a world interested in existential quantification, that is, find an optimal policy to achieve a goal, and universal quantification, that is to make sure that under any scheduler, policy, adversary, a system behaves correctly. The consequences of such exchange were many and unexpected.

**Nancy:** No, I did not anticipate that it would have broad impact, though of course I thought the ideas were important. But in general, I am bad at predicting what will appeal to others and inspire them to further work.

**Luca:** The journal version of your paper appears in the Nordic Journal on Computing, which is not a prime venue. Did you ever regret not publishing that work somewhere else? What is your take on the trend of using the perceived quality of a publication venue to evaluate research quality?

**Roberto:** Within semantics I know of a few technical reports that are much more influential than most journal papers; I do not think that the venue of a paper should be given much importance. On the other hand, venue as an evaluation tool is more

objective, allows an evaluator to consider many papers without reading them, and protects the evaluator from any type of claim against decisions, which sometimes may have even legal consequences. Needless to say that I do not agree with any of the ideas above. I do not agree as well with the other emerging trend of counting papers to evaluate quality. One good paper is much better than ten mediocre papers. What worries me the most is that young researchers may have a hard time to concentrate on quality if they have to focus on venue and quantity. I feel I was lucky not to have to worry about these issues when we submitted our paper for the special issue of the Nordic Journal of Computing.

**Nancy:** I certainly never regretted the venue for this paper. In general, I haven't paid too much attention to choice of publication venue. The main thing is to reach the audience you want to reach, which can be done through prestigious journals, less prestigious journals, conferences, or even ArXiv technical reports and some publicity. It's good to get feedback from referees, though. For evaluations, say for hiring or promotion, I think it's fair to take many factors into account in evaluating research quality. Venues, number of citations, special factors about different fields, . . . all can be discussed by hiring and promotion committees. But I hope that those committees also take the time to actually read and consider the work itself.

**Luca:** The last thirty years have seen a huge amount of work on probabilistic models of computation and on the development of proof techniques and tools based on them. What advice would you give to a young researcher interested in working on probabilistic models in concurrency today?

**Roberto:** My first advice would be to keep things simple and to focus on key ideas, possibly under the guidance of multiple application scenarios. When we study probabilistic concurrency, especially in contexts where the external world is governed by non-discrete laws, the interplay between different features of the model may become overwhelmingly complex. Of course it is a nice exercise to dive into all the details and see what it leads to, or to extend known results to the new scenarios, but how about checking whether some simple changes to one of the building blocks could make life easier? Unfortunately, working on building blocks is risky. So, my second advice is to dive into details and write papers, but take some breaks and think whether there may be nicer ways to solve the same problems.

**Nancy:** Pick some particular application domain, and make sure that the models and techniques you develop work well for that application. Don't work on the theory in a vacuum. The models will turn out much better! Perhaps simpler, as Roberto says.

**Luca:** What are the research topics that currently excite you the most?

**Roberto:** Difficult question. There are many things I like to look at, but at the moment I am very curious about how quantum computing can fit in a nice and elementary way into a concurrency framework.

**Nancy:** Probabilistic algorithms, especially those that are flexible (work in different environments), robust to failures and noise, and adaptive to changes. This includes such topics as wireless network algorithms, biologically-inspired algorithms, and robot swarms.

## Interview with Davide Sangiorgi

**Luca:** You receive one of the two CONCUR Test-of-Time Awards for the period 1992–1995 for your paper “A Theory of Bisimulation for the pi-Calculus”, presented at CONCUR 1993. Could you tell us briefly what spurred you to develop open bisimilarity and how the ideas underlying that notion came about?

**Davide:** I developed the paper on open bisimulation during 1992 and 1993. I was in Robin Milner’s group, in Edinburgh. We were studying and questioning basic aspects of the theory of the pi-calculus. One such aspect was the definition of equality on processes; thus a very fundamental aspect, underlying the whole theory. The equality had to be a form of bisimulation, in the same way as it was for CCS. The two forms of bisimilarity that were around for the pi-calculus, late and early bisimilarity, are not congruence relations. In both cases, the input clause of bisimilarity uses a universal quantification on the possible instantiations of the bound name. As a consequence, neither bisimilarity is preserved by the input prefix (forbidding substitutions in the input clause would make things worse, as congruence would fail for parallel composition). Therefore, one has to introduce separately the induced congruence, by universally quantifying the bisimilarities over all name substitutions. In other words, the two bisimilarities are not fully substitutive (“equal” terms cannot be replaced, one for the other, in an arbitrary context). On the other hand, the congruences induced by the bisimilarities are not themselves bisimilarities. Hence in this case “equal” terms, after some actions, need not be “equal” anymore. Thus, for instance, such relations do not support dynamic modifications of the context surrounding related terms.

This situation was not fully satisfactory. The same could be said for the algebraic theory: there were proof systems for the two bisimilarities (of course, on finitary processes) but, because of the above congruence issue, there were no axiomatisations. (In those years I was also working with Joachim Parrow on axiomatisations of these relations.)

The universal quantification on substitutions in the input clause of the bisimilarities and in the definitions of the induced congruences was also unsatisfactory

because it could make checking equalities cumbersome.

All these were motivations for looking at possible variations of the definition of bisimulation. The specific hints towards open bisimulation came from thinking at two key facets of the pi-calculus model that were somehow neglected in the definitions of early and late bisimilarities. The first facet has to do with the pi-calculus rejection of the separation between channels and variables (“channel” here meaning a “constant identifier”). In the pi-calculus, there is only one syntactic category, that of names, with no formal distinction between channels and variables. This contributes to the elegance of the model and its theory. However, both in early and in late bisimilarity, the free names of processes are treated as channels, whereas the bound names of inputs are treated as variables because of their immediate instantiation in the bisimilarity clause. There was somehow a discrepancy between the syntax and the semantics.

The second facet of the pi-calculus that contributed to the definition of open bisimilarity is the lack of the mismatch operator: the pi-calculus, at least in its original proposal, has a match operator to test equality between names, but not the dual mismatch, to test for inequality. Mismatching had been excluded for the preservation of a monotonicity property on transitions, intuitively asserting that substitutions may only increase the action capabilities of a process. (Both facets above represented major differences between the pi-calculus and its closest ancestor—Engberg and Nielsen’s Extended CCS.) Thus I started playing with the idea of avoiding the name instantiation in the input clause and, instead, allowing, at any moment, for arbitrary instantiations (i.e., substitutions) of the names of the processes—the latter justified by the above monotonicity property of transitions. By adding the requirement of being a congruence, the definition of open bisimilarity came about.

Still, I was not sure that such a bisimulation could be interesting and robust. Two further developments helped here. One was the axiomatisation (over recursion-free terms). It was a pure axiomatisation, it was simple, and with a completeness proof that leads to the construction of canonical and minimal (in some syntactic sense) representatives for the equivalence classes of the bisimilarity. For other bisimilarities, or related congruences, obtaining canonical representatives seems hard; at best such representatives are parametrised upon a set of free names and even in these cases minimality is not guaranteed.

The other development has to do with a symbolic or “efficient” characterisation of the bisimilarity. The initial definition of open bisimulation makes heavy use of substitutions. In the symbolic characterisation, substitutions are performed only when needed (for instance, the unification of two names  $a$  and  $b$  is required if there is an input at  $a$  and an output at  $b$  that can interact), somehow echoing the call-by-need style of functional languages. Such a characterisation seemed promising for automated or semi-automated verification.

**Luca:** How much of your later work has built on your CONCUR 1993 paper? What results of yours are you most proud of and why?

**Davide:** The most basic idea in open bisimulation is to avoid the instantiation of the bound name of an input, possibly making such a bound name a free name of the derivative term. The use of substitutions, elsewhere in the definition, is necessary to obtain a congruence relation for the pi-calculus. I was surprised to discover, in the following years, that such substitutions are not necessary in two relevant subsets of the pi-calculus. I called the variant of open bisimulation without substitutions ground bisimulation (I think the name came from Robin). One subset is Honda and Boudol's Asynchronous pi-calculus, whose main constraint is to make outputs floating particles that do not trigger the activation of a continuation (other limitations concern sum and matching). The other subset is the Internal (or Private) pi-calculus, in which only private (i.e., restricted) names may be transmitted. I designed the Internal pi-calculus with ground bisimilarity in mind. People seem to have found this calculus useful in many ways, partly because of its expressiveness combined with its simple theory (in many aspects similar to that of CCS), partly because it allows one to limit or control aliasing between names, which can be useful for carrying out proofs about behavioural properties of processes, or for designing and reasoning about type systems, or for representing the calculus in logical systems.

Sometimes, the possibility of using ground bisimulation can considerably simplify proofs of equalities of terms. For instance, in my works on comparisons between pi-calculus and lambda-calculus, when I had to translate the latter into the former I have always used one of the above subcalculi (sometimes even combining them, e.g., the Asynchronous Internal pi-calculus), precisely for being able to use ground bisimilarity.

I consider both ground bisimilarity and the Internal pi-calculus spin-offs of the work on open bisimilarity.

While working on open bisimilarity for the pi-calculus, in a different paper, I applied the idea of open bisimilarity to the lambda-calculus. I kept the name "open" but the bisimulation is really "ground", as there are no substitutions involved. I remember Robin encouraging me to keep the name "open" because it conveyed well the idea of setting a bisimulation on open terms, rather than on closed terms as usually done. In open bisimulation for the lambda-calculus, a lambda-abstraction  $\lambda x. M$  yields an action with label  $\lambda x$  that should be matched (modulo alpha-conversion) by the same action by a bisimilar term. (Of course additional bisimulation clauses are needed when a free variable is found in evaluation position.) In contrast, in the ordinary bisimulation for the lambda-calculus, Abramsky's applicative bisimilarity, the bound variable of an abstraction has to be instantiated with all closed terms, which is heavy. In general, open bisimilarity is

finer than applicative bisimilarity and contextual equivalence (the reference equivalence in the lambda-calculus) but they often coincide in examples of concrete interest. Moreover, open bisimilarity does coincide with contextual equivalence in appropriate extensions of the lambda-calculus. In short, open bisimilarity offers us a technique for reasoning on higher-order languages using “first-order” tools, somehow similarly to what game semantics does.

This line of work about open bisimilarity in higher-order languages has been very fruitful, and is still studied a lot, for various forms of higher-order languages, sometimes under the name of “normal-form” bisimulation.

**Luca:** In your opinion, what is the most interesting or unexpected use in the literature of the notions and techniques you developed in your award-winning paper?

**Davide:** I mentioned the hope, when working on open bisimilarity, that its symbolic “efficient” characterisation could be useful for automated or semi-automated tools for reasoning about behavioural properties. Shortly after introducing open bisimulation, Björn Victor and Faron Moller, both in Edinburgh at the time, exploited it to design the Mobility Workbench. I also worked on an algorithm and a prototype tool for on-the-fly checking, with Marco Pistore.

However the most surprising applications in this direction have arrived later, when the “lazy” instantiation of bound names of open bisimilarity has been applied to languages richer than pi-calculus. For instance, Chaki, Rajamani and Rehof have used open similarity in their methods and tools for model checking distributed message-passing software. Others have applied open bisimilarity to languages for security and cryptography, like the spi-calculus and applied pi-calculus. These include S. Brias, U. Nestmann and colleagues at EPFL and Berlin. R. Horne, A. Tiu and colleagues in Singapore and Luxembourg have pushed significantly in this direction, with verification techniques and tools. For instance very recently they have discovered a privacy vulnerability for e-passports.

Similarly, Yuxin Deng and colleagues have applied the idea of open bisimulation to quantum processes, with analogous motivations — avoiding the universal quantification in the instantiation of variables, algorithmically unfeasible in the quantum setting as quantum states constitute a continuum.

Another line of work that I found interesting and surprising concerns abstract frameworks for concurrency, including logical frameworks. Here forms of open bisimulation are often the “natural” bisimulations that come up. These frameworks may be based, for instance, on coalgebras and category theory (e.g., works by M. Fiore and S. Staton, N. Ghani, K. Yemane, and B. Victor), category theory for reactive systems (e.g., works by F. Bonchi, B. König and U. Montanari), nominal SOS rule formats (e.g., works by M. Cimini, M. R. Mousavi, and M. A. Reniers), higher-order logic languages (e.g., works by A. Tiu, G. Nadathur, and

D. Miller).

There have been works that have pushed the idea of open bisimulation of avoiding the instantiation of bound names in interactions with an external observer one step further: such bound names are not instantiated even in interactions internal to the processes. The substitutions produced by the interactions are added to the calculus, producing particles sometimes called fusions. This mechanism resembles the explicit substitutions of the lambda-calculus, but it goes beyond that; for instance the addition of fusions leads to modifications of input and output prefixes that produce pleasant syntactic and semantic symmetry properties. Various people have worked on this, including B. Victor, J. Parrow, Y. Fu, C. Laneve, P. Gardner and L. Wischik.

I should also mention the recent work by K. Y. Ahn, R. Horne, and A. Tiu on logical interpretations of open bisimilarity. Remarkably, they explain the difference between the original (late and early) formulations of bisimilarity in the pi-calculus and open bisimilarity as the difference between intuitionistic and classical versions of modal logics.

Apologies for not mentioning everybody!

**Luca:** To your mind, how has the focus of CONCUR changed since its first edition in 1990?

**Daide:** I remember that in the early years of CONCUR there was a tremendous excitement about the conference. A forum for grouping the (fast-growing) community had been long-awaited. In Edinburgh, every year after the conference, the people with an interest in concurrency would meet (multiple times!) and discuss the contents of the proceedings. Several of us every year would attend the conference. Attending the conference was very useful and interesting: one was sure to meet a lot of people, hear about excellent papers, have lively discussions. We would try to go, even without a paper in the proceedings. I vividly remember the 1993 edition, where I presented the paper on open bisimulation. It had been organised by Eike Best in Hildesheim, Germany. It was an enjoyable and exciting week, I met and knew a number of people of our community, and learned a lot. (How sad and unwise that the Computer Science department in Hildesheim, so strong in concurrency at the time, was shut down a few years later.) Over the years, the CONCUR community has kept increasing its size. The conference has substantially broadened its scope, rightly including new emerging topics. Perhaps it is more difficult than in the past to (more or less) understand most of the presented papers, both because of the diversity of the topics and of their technical specialisation. On the other hand, there are now satellite events, covering a number of areas. Hence there are always plenty of interesting presentations and talks to attend (this definitely occurred to me in the last edition, in Amsterdam!). I should also mention here the activity on the IFIP WG 1.8 on Concurrency Theory,

currently chaired by Ilaria Castellani, that in many ways supports and promotes CONCUR.

The quality of the papers at CONCUR is still very high. This is very important. As a community we should strive to maintain, and possibly even increase, the excellence and prestige of the conference, first of all, by submitting our best papers to the conference. CONCUR must be a reference conference in Computer Science, which is essential for injecting new people and energy into the community.

**Acknowledgements** I thank the award recipients, who were kind enough to answer my questions and share their opinions with the community as a whole, the CONCUR Steering Committee and the leadership of the IFIP Working Group 1.8 on Concurrency Theory for setting up the CONCUR Test-of-Time Award and for the assistance they provided to the jury during its work, and Igor Konnov and Laura Kovacs for their support and for advertising the awards on the CONCUR 2020 web page very promptly. I am grateful to Ilaria Castellani, chair of the IFIP Working Group 1.8 on Concurrency Theory, who pointed out a number of typos in the original version of the interviews that I posted on my professional blog at <https://processalgebra.blogspot.com/>. Any remaining infelicity is solely my responsibility. Last, but by no means least, I thank Jos Baeten, Patricia Bouyer-Decitre, Holger Hermanns and Alexandra Silva for the work they did as members of the jury. The selection of the award-receiving papers was difficult, as there were many deserving articles we could have chosen. I have learnt much by discussing the candidate papers with them.