INTERVIEWS WITH THE 2022 CONCUR TEST-OF-TIME AWARD RECIPIENTS

Luca Aceto
ICE-TCS, Department of Computer Science,
Reykjavik University
Gran Sasso Science Institute, L'Aquila
luca@ru.is, luca.aceto@gssi.it

Orna Kupferman
School of Computer Science and Engineering
Hebrew University, Jerusalem
orna@cs.huji.ac.il

Mickael Randour
Faculty of Science, Mathematics Department
Université de Mons
mickael.randour@gmail.com

Davide Sangiorgi
Department of Computer Science, University of Bologna
Davide.Sangiorgi@cs.unibo.it

In 2020, the CONCUR conference series instituted its Test-of-Time Award, whose purpose is to recognise important achievements in Concurrency Theory that were published at the CONCUR conference and have stood the test of time. This year, the following four papers were chosen to receive the CONCUR Test-of-Time Awards for the periods 1998–2001 and 2000–2003 by a jury consisting of Ilaria Castellani (chair), Paul Gastin, Orna Kupferman, Mickael Randour and Davide Sangiorgi. (The papers are listed in chronological order.)

- Christel Baier, Joost-Pieter Katoen and Holger Hermanns. Approximate symbolic model checking of continuous-time Markov chains. CONCUR 1999.

- Franck Cassez and Kim Guldstrand Larsen. The Impressive Power of Stopwatches. CONCUR 2000.
- James J. Leifer and Robin Milner. Deriving Bisimulation Congruences for Reactive Systems. CONCUR 2000.
- Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Mariëlle Stoelinga. The Element of Surprise in Timed Games. CONCUR 2003.

This article is devoted to interviews with the recipients of the Test-of-Time Award. More precisely,

- Orna Kupferman interviewed Christel Baier, Joost-Pieter Katoen and Holger Hermanns;
- Luca Aceto interviewed Franck Cassez and Kim Guldstrand Larsen;
- Davide Sangiorgi interviewed James Leifer; and
- Luca Aceto and Mickael Randour jointly interviewed Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Mariëlle Stoelinga.

We are very grateful to the awardees for their willingness to answer our questions and hope that the readers of this article will enjoy reading the interviews as much as we did.

Interview with C. Baier, J.-P. Katoen and H. Hermanns

In what follows, BHK refers to Baier, Katoen and Hermanns.

Orna: You receive the CONCUR Test-of-Time Award 2022 for your paper "Approximate symbolic model checking of continuous-time Markov chains," which appeared at CONCUR 1998[1]. In that article, you combine three different challenges: symbolic algorithms, real-time systems, and probabilistic systems. Could you briefly explain to our readers what the main challenge in such a combination is?

BHK: The main challenge is to provide a fixed-point characterization of time-bounded reachability probabilities: the probability to reach a given target state

[1] See https://link.springer.com/content/pdf/10.1007/3-540-48320-9_12.pdf.

within a given deadline. Almost all works in the field up to 1999 treated discrete-time probabilistic models and focused on “just” reachability probabilities: what is the probability to eventually end up in a given target state? This can be characterized as a unique solution of a linear equation system. The question at stake was: how to incorporate a real-valued deadline d? The main insight was to split the problem in staying a certain amount of time, x say, in the current state and using the remaining $d - x$ time to reach the target from its successor state. This yields a Volterra integral equation system; indeed time-bounded reachability probabilities are unique solutions of such equation systems. In the CONCUR 1999 paper we suggested to use symbolic data structures to do the numerical integration; later we found out that much more efficient techniques can be applied.

Orna: Could you tell us how you started your collaboration on the award-winning paper? In particular, as the paper combines three different challenges, is it the case that each of you has brought to the research different expertise?

BHK: Christel and Joost-Pieter were both in Birmingham, where a meeting of a collaboration project between German and British research groups on stochastic systems and process algebra took place. There the first ideas of model checking continuous-time Markov chains arose, especially for time-bounded reachability: with stochastic process algebras there were means to model CTMCs in a compositional manner, but verification was lacking. Back in Germany, Holger suggested to include a steady-state operator, the counterpart of transient properties that can be expressed using timed reachability probabilities. We then also developed the symbolic data structure to support the verification of the entire logic.

Orna: Your contribution included a generalization of BDDs (binary decision diagrams) to MTDDs (multi-terminal decision diagrams), which allow both Boolean and real-valued variables. What do you think about the current state of symbolic algorithms, in particular the choice between SAT-based methods and methods that are based on decision diagrams?

BHK: BDD-based techniques entered probabilistic model checking in the mid 1990’s for discrete-time models such as Markov chains. Our paper was one of the first, perhaps even the first, that proposed to use BDD structures for real-time stochastic processes. Nowadays, SAT and in particular SMT-based techniques belong to the standard machinery in probabilistic model checking. SMT techniques are, e.g., used in bisimulation minimization at the language level, counterexample generation, and parameter synthesis. This includes both linear as well as non-linear theories. BDD techniques are still used, mostly in combination with sparse representations, but it is fair to say that SMT is becoming more and more relevant.

Orna: What are the research topics that you find most interesting right now? Is there any specific problem in your current field of interest that you’d like to see solved?

BHK: This depends a bit on whom you ask! Christel's recent work is about cause-effect reasoning and notions of responsibility in the verification context. This ties into the research interest of Holger who looks at the foundations of perspicuous software systems. This research is rooted in the observation that the explosion of opportunities for software-driven innovations comes with an implosion of human opportunities and capabilities to understand and control these innovations. Joost-Pieter focuses on pushing the borders of automation in weakest-precondition reasoning of probabilistic programs. This involves loop invariant synthesis, probabilistic termination proofs, the development of deductive verifiers, and so forth. Challenges are to come up with good techniques for synthesizing quantitative loop invariants, or even complete probabilistic programs.

Orna: What advice would you give to a young researcher who is keen to start working on topics related to symbolic algorithms, real-time systems, and probabilistic systems?

BHK: Try to keep it smart and simple.

Interview with Franck Cassez and Kim Guldstrand Larsen

Luca: You receive the CONCUR Test-of-Time Award 2022 for your paper "The Impressive Power of Stopwatches"[2], which appeared at CONCUR 2000. In that article, you showed that timed automata enriched with stopwatches and unobservable time delays have the same expressive power of linear hybrid automata. Could you briefly explain to our readers what timed automata with stopwatches are? Could you also tell us how you came to study the question addressed in your award-winning article? Which of the results in your paper did you find most surprising or challenging?

Kim: Well, in timed automata all clocks grow with rate 1 in all locations of the automata. Thus you can tell the amount of time that has elapsed since a particular clock was last reset, e.g., due to an external event of interest. A stopwatch is a real-valued variable similar to a regular clock. In contrast to a clock, a stopwatch will in certain locations grow with rate 1 and in other locations grow with rate 0, i.e., it is stopped. As such, a stopwatch gives you information about the accumulated time spent in a certain parts of the automata.

In modelling schedulability problems for real-time systems, the use of stopwatches is crucial in order to adequately capture preemption. I definitely believe

[2] See `https://link.springer.com/content/pdf/10.1007/3-540-44618-4_12.pdf`.

that it was our shared interest in schedulability that brought us to study timed automata with stopwatches. We knew from earlier results by Alur et al. that properties such as reachability was undecidable. But what could we do about this? And how much expressive power would the addition of stopwatches provide?

In the paper we certainly put the most emphasis on the latter question, in that we showed that stopwatch automata and linear hybrid automata accept the same class of timed languages, and this was at least for me the most surprising and challenging result. However, focusing on impact, I think the approximate zone-based method that we apply in the paper has been extremely important from the point of view of having our verification tool UPPAAL being taken up at large by the embedded systems community. It has been really interesting to see how well the over-approximation method actually works.

Luca: In your article, you showed that linear hybrid automata and stopwatch automata accept the same class of timed languages. Would this result still hold if all delays were observable? Do the two models have the same expressive power with respect to finer notions of equivalence such as timed bisimilarity, say? Did you, or any other colleague, study that problem, assuming that it is an interesting one?

Kim: These are definitely very interesting questions, and should be studied. As for finer notions of equivalences, e.g., timed bisimilarity, I believe that our translation could be shown to be correct up to some timed variant of chunk-by-chunk simulation introduced by Anders Gammelgaard in his Licentiat Thesis from Aarhus University in 1991[3]. That could be a good starting point.

Luca: Did any of your subsequent research build explicitly on the results and the techniques you developed in your award-winning paper? Which of your subsequent results on timed and hybrid automata do you like best? Is there any result obtained by other researchers that builds on your work and that you like in particular or found surprising?

Kim: Looking up in DBLP, I see that I have some 28 papers containing the word "scheduling". For sure stopwatches will have been used in one way or another in these. One thing that we never really examined thoroughly is to investigate how well the approximate zone-based technique will work when applied to the translation of linear hybrid automata into stopwatch automata. This would definitely be interesting to find out.

This was the first joint publication between me and Franck. I enjoyed fully the collaboration on all the next 10 joint papers. Here the most significant ones are probably the paper at CONCUR 2005, where we presented the symbolic on-the-fly algorithms for synthesis for timed games and the branch UPPAAL TIGA. And

[3]See `https://tidsskrift.dk/daimipb/article/view/6611/5733`.

later in a European project GASICS with Jean-Francois Raskin, we used TIGA in the synthesis of optimal and robust control of a hydraulic system.

Franck: Using the result in our paper, we can analyse scheduling problems where tasks can be stopped and restarted, using real-time model-checking and a tool like UPPAAL.

To do so, we build a network of stopwatch automata modelling the set of tasks and a scheduling policy, and reduce schedulability to a safety verification problem: avoid reaching states where tasks do not meet their deadlines. Because we over-approximate the state space, our analysis may yield some false positives and may wrongly declare a set of tasks non-schedulable because the over-approximation is too coarse.

In the period 2003–2005, in cooperation with Francois Laroussinie we tried to identify some classes of stopwatch automata for which the over-approximation does not generate false positives. We never managed to find an interesting sub-class.

This may look like a serious problem in terms of applicability of our result, but in practice, it does not matter too much. Most of the time, we are interested in the schedulability of a specific set of tasks (e.g., controlling a plant, a car, etc.) and for these instances, we can use our result: if we have false positives, we can refine the model tasks and scheduler and rule them out. Hopefully after a few iterations of refinement, we can prove that the set of tasks is schedulable.

The subsequent result on timed and hybrid automata of mine that I probably like best is the one we obtained on solving optimal reachability in timed automata. We had a paper at FSTTCS in 2004[4] presenting the theoretical results, and a companion paper at GDV 2004[5] with an implementation using HyTech, a tool for analysing hybrid automata.

I like these results because we ended up with a rather simple proof, after 3–4 years working on this hard problem.

Luca: Could you tell us how you started your collaboration on the award-winning paper? I recall that Franck was a regular visitor to our department at Aalborg University for some time, but I can't recall how his collaboration with the UPPAAL group started.

Kim: I am not quite sure I remember how and when I first met Franck. For some time we already worked substantially with French researchers, in particular from LSV Cachan (Francois Larroussinie and Patricia Bouyer). I have the feeling that there were quite some strong links between Nantes (were Franck was) and LSV on timed systems in those days. Also Nantes was the organizer of the PhD school

[4]See `https://doi.org/10.1007/978-3-540-30538-5_13`.

[5]See `https://doi.org/10.1016/j.entcs.2004.07.006`.

MOVEP five times in the period 1994-2002, and I was lecturing there in one of the years, meeting Olivier Roux and Franck who were the organizers. Funny enough, this year we are organizing MOVEP in Aalborg. Anyway, at some point Franck became a regular visitor to Aalborg, often for long periods of time—playing on the squash team of the city when he was not working.

Franck: As Kim mentioned, I was in Nantes at that time, but I was working with Francois Laroussinie who was in Cachan. Francois had spent some time in Aalborg working with Kim and his group and he helped organise a mini workshop with Kim in 1999, in Nantes. That's when Kim invited me to spend some time in Aalborg, and I visited Aalborg University for the first time from October 1999 until December 1999. This is when we worked on the stopwatch automata paper. We wanted to use UPPAAL to verify systems beyond timed automata.

I visited Kim and his group almost every year from 1999 until 2007, when I moved to Australia. There were always lots of visitors at Aalborg University and I was very fortunate to be there and learn from the Masters.

I always felt at home at Aalborg University, and loved all my visits there. The only downside was that I never managed to defeat Kim at badminton. I thought it was a gear issue, but Kim gave me his racket (I still have it) and the score did not change much.

Luca: What are the research topics that you find most interesting right now? Is there any specific problem in your current field of interest that you'd like to see solved?

Kim: Currently I am spending quite some time on marrying symbolic synthesis with reinforcement learning for Timed Markov Decision Processes in order to achieve optimal as well as safe strategies for Cyber-Physical Systems.

Luca: Both Franck and you have a very strong track record in developing theoretical results and in applying them to real-life problems. In my, admittedly biased, opinion, your work exemplifies Ben Schneiderman's Twin-Win Model[6], which propounds the pursuit of "the dual goals of breakthrough theories in published papers and validated solutions that are ready for widespread dissemination." Could you say a few words on your research philosophy?

Kim: I completely subscribe to this. Several early theoretical findings, such as the paper on stopwatch automata, have been key in our sustainable transfer to industry.

Franck: Kim has been a mentor to me for a number of years now, and I certainly learned this approach/philosophy from him and his group.

[6]See `https://www.pnas.org/doi/pdf/10.1073/pnas.1802918115`.

We always started from a concrete problem, e.g., scheduling tasks/checking schedulability, and to validate the solutions, building a tool to demonstrate applicability. The next step was to improve the tool to solve larger and larger problems.

UPPAAL is a fantastic example of this philosophy: the reachability problem for timed automata is PSPACE-complete. That would deter a number of people to try and build tools to solve this problem. But with smart abstractions, algorithms and data-structures, and constant improvement over a number of years, UPPAAL can analyse very large and complex systems. It is amazing to see how UPPAAL is used in several areas from traffic control to planning and to precisely guiding a needle for an injection.

Luca: What advice would you give to a young researcher who is keen to start working on topics related to formal methods?

Kim: Come to Aalborg, and participate in next year's MOVEP.

Interview with James Leifer

Davide: How did the work presented in your CONCUR Test-of-Time paper come about?

James: I was introduced to Robin Milner by my undergraduate advisor Bernard Sufrin around 1994. Thanks to that meeting, I started with Robin at Cambridge in 1995 as a fresh Ph.D. student. Robin had recently moved from Edinburgh and had a wonderful research group, including, at various times, Peter Sewell, Adriana Compagnoni, Benjamin Pierce, and Philippa Gardner. There were also many colleagues working or visiting Cambridge interested in process calculi: Davide Sangiorgi, Andy Gordon, Luca Cardelli, Martín Abadi,.... It was an exciting atmosphere! I was particularly close to Peter Sewell, with whom I discussed the ideas here extensively and who was generous with his guidance.

There was a trend in the community at the time of building complex process calculi (for encryption, Ambients, etc.) where the free syntax would be quotiented by a structural congruence to "stir the soup" and allow different parts of a tree to float together; reaction rules (unlabelled transitions) then would permit those agglomerated bits to react, to transform into something new.

Robin wanted to come up with a generalised framework, which he called Action Calculi, for modelling this style of process calculi. His framework would describe graph-like "soups" of atoms linked together by arcs representing binding and sharing; moreover the atoms could contain subgraphs inside of them for freezing activity (as in prefixing in the π-calculus), with the possibility of boundary crossing arcs (similarly to how ν-bound names in π-calculus can be used in deeply nested subterms).

Robin had an amazing talent for drawing beautiful graphs! He would "move" the nodes around on the chalkboard and reveal how a subgraph was in fact a reactum (the left-hand side of an unlabelled transition). In the initial phases of my Ph.D. I just tried to understand these graphs: they were so natural to draw on the blackboard! And yet, they were also so uncomfortable to use when written out in linear tree- and list-like syntax, with so many distinct concrete representations for the same graph.

Putting aside the beauty of these graphs, what was the benefit of this framework? If one could manage to embed a process calculus in Action Calculi, using the graph structure and fancy binding and nesting to represent the quotiented syntax, what then? We dreamt about a proposition along the following lines: if you represent your syntax (quotiented by your structural congruence) in Action Calculi graphs, and you represent your reaction rules as Action Calculi graph rewrites, then we will give you a congruential bisimulation for free!

Compared to CCS for example, many of the rich new process calculi lacked labelled transitions systems. In CCS, there was a clean, simple notion of labelled transitions and, moreover, bisimulation over those labelled transitions yielded a congruence: for all processes P and Q, and all process contexts $C[-]$, if $P \sim Q$, then $C[P] \sim C[Q]$. This is a key quality for a bisimulation to possess, since it allows modular reasoning about pieces of a process, something that's so much harder in a concurrent world than in a sequential one.

Returning to Action Calculi, we set out to make good on the dream that everyone gets a congruential bisimulation for free! Our idea was to find a general method to derive labelled transitions systems from the unlabelled transitions and then to prove that bisimulation built from those labelled transitions would be a congruence.

The idea was often discussed at that time that there was a duality whereby a process undergoing a labelled transition could be thought of as the environment providing a complementary context inducing the process to react. In the early labelled transition system in π-calculus for example, I recall hearing that P undergoing the input labelled transition xy could be thought of as the environment outputting payload y on channel x to enable a τ transition with P.

So I tried to formalise this notion that labelled transitions are environmental contexts enabling reaction, i.e. defining $P \stackrel{C[-]}{\rightarrow} P'$ to mean $C[P] \rightarrow P'$ provided that $C[-]$ was somehow "minimal", i.e., contained nothing superfluous beyond what was necessary to trigger the reaction. We wanted to get a rigorous definition of that intuitive idea. There was a long and difficult period (about 12 months) wandering through the weeds trying to define minimal contexts for Action Calculi graphs (in terms of minimal nodes and minimal arcs), but it was hugely complex, frustrating, and ugly and we seemed no closer to the original goal of achieving congruential bisimulation with these labelled transitions systems.

Eventually I stepped back from Action Calculi and started to work on a more theoretical definition of "minimal context" and we took inspiration from category theory. Robin had always viewed Action Calculi graphs as categorical arrows between objects (where the objects represented interfaces for plugging together arcs). At the time, there was much discussion of category theory in the air (for game theory); I certainly didn't understand most of it but found it interesting and inspiring.

If we imagine that processes and process-contexts are just categorical arrows (where the objects are arities) then context composition is arrow composition. Now, assuming we have a reaction rule $R \to R'$, we can define labelled transitions $P \overset{C[-]}{\to} P'$ as follows: there exists a context D such that $C[P] = D[R]$ and $P' = D[R']$. The first equality is a commuting diagram and Robin and I thought that we could formalise minimality by something like a categorical pushout! But that wasn't quite right as C and D are not the minimum pair (compared to all other candidates), but a minimal pair: there may be many incomparable minimal pairs all of which are witnesses of legitimate labelled transitions. There was again a long period of frustration eventually resolved when I reinvented "relative pushouts" (in place of pushouts). They are a simple notion in slice categories but I didn't know that until later....

Having found a reasonable definition of "minimal", I worked excitedly on bisimulation, trying to get a proof of congruence: $P \sim Q$ implies $E[P] \sim E[Q]$. For weeks, I was considering the labelled transitions of $E[P] \overset{F[-]}{\to}$ and all the ways that could arise. The most interesting case is when a part of P, a part of E, and F all "conspire" together to generate a reaction. From that I was able to derive a labelled transition of P by manipulating relative pushouts, which by hypothesis yielded a labelled transition of Q, and then, via a sort of "pushout pasting", a labelled transition $E[Q] \overset{F[-]}{\to}$. It was a wonderful moment of elation when I pasted all the diagrams together on Robin's board and we realised that we had the congruence property for our synthesised labels!

We looked back again at Action Calculi, using the notion of relative pushouts to guide us (instead of the arbitrary approach we had considered before) and we further looked at other kinds of process calculi syntax to see how relative pushouts could work there.... Returning to the original motivation to make Action Calculi a universal framework with congruential bisimulation for free, I'm not convinced of its utility. But it was the challenge that led us to the journey of the relative pushout work, which I think is beautiful.

Davide: What influence did this work have in the rest of your career? How much of your subsequent work built on it?

James: It was thanks to this work that I visited INRIA Rocquencourt to discuss process calculi with Jean-Jacques Lévy and Georges Gonthier. They kindly invited me to spend a year as postdoc in 2001 after I finished my thesis with Robin, and I ended up staying in INRIA ever since. I didn't work on bisimulation again as a research topic, but stayed interested in concurrency and distribution for a long time, working with Peter Sewell et al. on distributed language design with module migration and rebinding, and with Cédric Fournet et al. on compiler design for automatically synthesising cryptographic protocols for high level sessions specifications.

Davide: Could you tell us about your interactions with Robin Milner? What was it like to work with him? What lessons did you learn from him?

James: I was tremendously inspired by Robin.

He would stand at his huge blackboard, his large hands covered in chalk, his bicycle clips glinting on his trousers, and he would stalk up and down the blackboard—thinking and moving. There was something theatrical and artistic about it: his thinking was done in physical movement and his drawings were dynamic as the representations of his ideas evolved across the board.

I loved his drawings. They would start simple, a circle for a node, a box for a subgraph, etc. and then develop more and more detail corresponding to his intuition. (It reminded me of descriptions I had read of Richard Feynman drawing quantum interactions.)

Sometimes I recall being frustrated because I couldn't read into his formulas everything that he wanted to convey (and we would then switch back to drawings) or I would be worried that there was an inconsistency creeping in or I just couldn't keep up, so the board sessions could be a roller coaster ride at times!

Robin worked tremendously hard and consistently. He would write out and rewrite out his ideas, regularly circulating hand written documents. He would refine over and over his diagrams. Behind his achievements there was an impressive consistency of effort.

He had a lot of confidence to carry on when the sledding was hard. He had such a strong intuition of what ought to be possible, that he was able to sustain years of effort to get there.

He was generous with praise, with credit, with acknowledgement of others' ideas. He was generous in sharing his own ideas and seemed delighted when others would pick them up and carry them forward. I've always admired his openness and lack of jealousy in sharing ideas.

In his personal life, he seemed to have real compatibility with Lucy (his wife), who also kept him grounded. I still laugh when I remember once working with him at his dining room table and Lucy announcing, "Robin, enough of the mathematics. It's time to mow the lawn!"

I visited Oxford for Lucy's funeral and recall Robin putting a brave face on his future plans; I returned a few weeks later when Robin passed away himself. I miss him greatly.

Davide: What research topics are you most interested in right now? How do you see your work develop in the future?

James: I've been interested in a totally different area, namely healthcare, for many years. I'm fascinated by how patients, and information about them, flows through the complex human and machine interactions in hospital. When looking at how these flows work, and how they don't, it's possible to see where errors arise, where blockages happen, where there are informational and visual deficits that make the job of doctors and nurses difficult. I like to think visually in terms of graphs (incrementally adding detail) and physically moving through the space where the action happens—all inspired by Robin!

Interview with Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Mariëlle Stoelinga

In what follows, "Luca A." refers to Luca Aceto, whereas "Luca" is Luca de Alfaro.

Luca A. and Mickael: You receive the CONCUR Test-of-Time Award 2022 for your paper "The Element of Surprise in Timed Games," which appeared at CONCUR 2003[7]. In that article, you studied concurrent, two-player timed games. A key contribution of your paper is the definition of an elegant timed game model, allowing both the representation of moves that can take the opponent by surprise, as they are played "faster," and the definition of natural concepts of winning conditions for the two players—ensuring that players can win only by playing according to a physically meaningful strategy. In our opinion, this is a great example of how novel concepts and definitions can advance a research field. Could you tell us more about the origin of your model?

All authors: Mariëlle and Marco were postdocs with Luca at University of California, Santa Cruz, in that period, Rupak was a student of Tom's, and we were all in close touch, meeting very often to work together. We all had worked much on games, and an extension to timed games was natural for us to consider.

[7]See `https://pub.ist.ac.at/~tah/Publications/the_element_of_surprise_in_timed_games.pdf`).

In untimed games, players propose a move, and the moves jointly determine the next game state. In these games there is no notion of real-time. We wanted to study games in which players could decide not only the moves, but also the instant in time when to play them.

In timed automata, there is only one "player" (the automaton), which can take either a transition, or a time step. The natural generalization would be a game in which players could propose either a move, or a time step.

Yet, we were unsatisfied with this model. It seemed to us that it was different to say "Let me wait 14 seconds and reconvene. Then, let me play my King of Spades" or "Let me play my King of Spades in 14 seconds." In the first, by stopping after 14 seconds, the player is providing a warning that the card might be played. In the second, there is no such warning. In other words, if players propose either a move or a time-step, they cannot take the adversary by surprise with a move at an unanticipated instant. We wanted a model that could capture this element of surprise.

To capture the element of surprise, we came up with a model in which players propose both a move and the delay with which it is played. After this natural insight, the difficulty was to find the appropriate winning condition, so that a player could not win by stopping time.

Tom: Besides the infinite state space (region construction etc.), a second issue that is specific to timed systems is the divergence of time. Technically, divergence is a built-in Büchi condition ("there are infinitely many clock ticks"), so all safety and reachability questions about timed systems are really co-Büchi and Büchi questions, respectively. This observation had been part of my work on timed systems since the early 1990s, but it has particularly subtle consequences for timed games, where no player (and no collaboration of players) should have the power to prevent time from diverging. This had to be kept in mind during the exploration of the modeling space.

All authors: We came up with many possible winning conditions, and for each we identified some undesirable property, except for the one that we published. This is in fact an aspect that did not receive enough attention in the paper; we presented the chosen winning condition, but we did not discuss in full detail why several other conditions that might have seemed plausible did not work.

In the process of analyzing the winning conditions, we came up with many interesting games, which form the basis of many results, such as the result on lack of determinization, on the need for memory in reachability games (even when clock values are part of the state), and most famously as it gave the title to the paper, on the power of surprise.

After this fun ride came the hard work, where we had to figure out how to solve these games. We had worked at symbolic approaches to games before, and we followed the approach here, but there were many complex technical adaptations required. When we look at the paper in the distance of time, it has this combination of a natural game model, but also of a fairly sophisticated solution algorithm.

Luca A. and Mickael: Did any of your subsequent research build explicitly on the results and the techniques you developed in your award-winning paper? If so, which of your subsequent results on (timed) games do you like best? Is there any result obtained by other researchers that builds on your work and that you like in particular or found surprising?

Luca: Marco and I built Ticc, which was meant to be a tool for timed interface theories, based largely on the insights in this paper. The idea was to be able to check the compatibility of real-time systems, and automatically infer the requirements that enable two system components to work well together—to be compatible in time. We thought this would be useful for hardware or embedded systems, and especially for control systems, and in fact the application is important: there is now much successful work on the compositionality of StateFlow/Simulink models.

We used MTBDDs as the symbolic engine, and Marco and I invented a language for describing the components and we wrote by pair-programming some absolutely beautiful Ocaml code that compiled real-time component models into MTBDDs (perhaps the nicest code I have ever written). The problem was that we were too optimistic in our approach to state explosion, and we were never able to study any system of realistic size.

After this, I became interested in games more in an economic setting, and from there I veered into incentive systems, and from there to reputation systems and to a three-year period in which I applied reputation systems in practice in industry, thus losing somewhat touch with formal methods work.

Marco: I've kept working on games since the award-winning paper, in one way or another. The closest I've come to the timed game setting has been with controller synthesis games for hybrid automata. In a series of papers, we had fun designing and implementing symbolic algorithms that manipulate polyhedra to compute the winning region of a linear hybrid game. The experience gained on timed games helped me recognize the many subtleties arising in games played in real time on a continuous state-space.

Mariëlle: I have been working on games for test case generation: One player represents the tester, which chooses inputs to test; the other player represents the System-under-Test, and chooses the outputs of the system. Strategy synthesis algorithms can then compute strategies for the tester that maximize all kinds of objectives, e.g., reaching certain states, test coverage etc.

A result that I really like is that we were able to show a very close correspondence between the existing testing frameworks and game theoretic frameworks: Specifications act as game arenas; test cases are exactly game strategies, and the conformance relation used in testing (namely ioco) coincides with game refinement (i.e., alternating refinement).

Rupak: In an interesting way, the first paper on games I read was the one by Maler, Pnueli and Sifakis (STACS 1995)[8] that had both fixpoint algorithms and timed games (without "surprise"). So the problem of symbolic solutions to games and their applications in synthesis followed me throughout my career. I moved to finding controllers for games with more general (non-linear) dynamics, where we worked on abstraction techniques. We also realized some new ways to look at restricted classes of adversaries. I was always fortunate to have very good collaborators who kept my interest alive with new insights. Very recently, I have gotten interested in games from a more economic perspective, where players can try to signal each other or persuade each other about private information but it's too early to tell where this will lead.

Luca A. and Mickael: What are the research topics that you find most interesting right now? Is there any specific problem in your current field of interest that you'd like to see solved?

Mariëlle: Throughout my academic life, I have been working on stochastic analysis, with Luca and Marco, we worked on stochastic games a lot. First only on theory, but later also on industrial applications, especially in the railroad and high-tech domain. At some point in time, I realized that my work was actually centred around analysing failure probabilities and risk. That is how I moved into risk analysis; the official title of the chair I hold is Risk Management for High Tech Systems.

The nice thing is: this sells much better than Formal Methods! Almost nobody knows what Formal Methods are, and if they know, people think "yes, those difficult people who urge us to specify everything mathematically." For risk management, this is completely different: everybody understands that this is an important area.

Luca: I am currently working on computational ecology, on machine learning (ML) for networks, and on fairness in data and ML. In computational ecology, we are working on the role of habitat and territory for species viability. We use ML techniques to write "differentiable algorithms," where we can compute the effect of each input, such as the kind of vegetation in each square-kilometer of territory, on the output. If all goes well, this will enable us to efficiently compute which regions should be prioritized for protection and habitat conservation.

---

[8] See `https://www-verimag.imag.fr/~sifakis/RECH/Synth-MalerPnueli.pdf`.

In networks, we have been able to show that reinforcement learning can yield tremendous throughput gains in wireless protocols, and we are now starting to work on routing and congestion control.

And in fairness and ML, we have worked on the automatic detection of anomalous data subgroups (something that can be useful in model diagnostics), and we are now working on the spontaneous inception of discriminatory behavior in agent systems.

While these do not really constitute a coherent research effort, I can certainly say that I am having a grand tour of computer science—the kind of joy ride one can afford with tenure!

Rupak: I have veered between practical and theoretical problems. I am working on charting the decidability frontier for infinite-state model checking problems (most recently, for asynchronous programs and context-bounded reachability). I am also working on applying formal methods to the world of cyber-physical systems—mostly games and synthesis. Finally, I have become very interested in applying formal methods to large scale industrial systems through a collaboration with Amazon Web Services. There is still a large gap between what is theoretically understood and what is practically applicable to these systems; and the problems are a mix of technical and social.

Luca A. and Mickael: You have a very strong track record in developing theoretical results and in applying them to real-life problems. In our, admittedly biased, opinion, your work exemplifies Ben Schneiderman's Twin-Win Model, which propounds the pursuit of "the dual goals of breakthrough theories in published papers and validated solutions that are ready for widespread dissemination." Could you say a few words on your research philosophy? How do you see the interplay between basic and applied research?

Luca: This is very kind for you to say, and a bit funny to hear, because certainly when I was young I had a particular talent for getting lost in useless theoretical problems.

I think two things played in my favor. One is that I am curious. The other is that I have a practical streak: I still love writing code and tinkering with "things," from IoT to biology to web and more. This tinkering was at the basis of many of the works I did. My work on reputation systems started when I created a wiki on cooking; people were vandalizing it, and I started to think about game theory and incentives for collaboration, which led to my writing much of the code for Wikipedia analysis, and at Google, for Maps edits analysis. My work on networks started with me tinkering with simple reinforcement-learning schemes that might work, and writing the actual code. On the flip side, my curiosity too often had the better of me, so that I have been unable to pay the continuous and devoted attention to a single research field. I am not a specialist in any single thing I do or

I have done. I am always learning the ropes of something I don't quite know yet how to do.

My applied streak probably gave me some insight on which problems might be of more practical relevance, and my frequent field changes have allowed me to bring new perspectives to old problems. There were not many people using reinforcement learning for wireless networks, there are not many who write ML and GPU code and also avidly read about conservation biology.

Rupak: I must say that Tom and Luca were very strong influencers for me in my research: both in problem selection and in appreciating the joy of research. I remember one comment of Tom, paraphrased as "Life is short. We should write papers that get read." I spent countless hours in Luca's office and learnt a lot of things about research, coffee, the ideal way to make pasta, and so on.

Marco: It was an absolute privilege to be part of the group that wrote that paper (my 4th overall, according to DBLP). I'd like to thank my coauthors, and Luca in particular, for guiding me during those crucially formative years.

Mariëlle: I fully agree!

Luca A. and Mickael: Several of you have high-profile leadership roles at your institutions. What advice would you give to a colleague who is about to take up the role of department chair, director of a research centre, dean or president of a university? How can one build a strong research culture, stay research active and live to tell the tale?

Luca: My colleagues may have better advice; my productivity certainly decreased when I was department chair, and is lower even now that I am the vice-chair. When I was young, I was ambitious enough to think that my scientific work would have the largest impact among the things I was doing. But I soon realized that some of the greatest impact was on others: on my collaborators, on the students I advised, who went on to build great careers and stayed friends, and on all the students I was teaching. This awareness serves to motivate and guide me in my administrative work. The Computer Science department at University of California, Santa Cruz, is one of the ten largest in the number of students we graduate, and the time I spend on improving its organization and the quality of the education it delivers is surely very impactful. My advice to colleagues is to consider their service not as an impediment to research, but as one of the most impactful things they do.

My way of staying alive is to fence off some days that I only dedicate to research (aside from some unavoidable emergency), and also, to have collaborators that give me such joy in working together that they brighten and energize my whole day.

Luca A. and Mickael: Finally, what advice would you give to a young researcher who is keen to start working on topics related to concurrency theory today?

Luca: Oh that sounds very interesting! And, may I show you this very interesting thing we are doing in Jax to model bird dispersal? We feed in this climate and vegetation data, and then we. . . .

Just kidding. Just kidding. If I come to CONCUR I promise not to lead any of the concurrency yearlings astray. At least I will try.

My main advice would be this: work on principles that allow correct-by-design development. If you look at programming languages and software engineering, the progress in software productivity has not happened because people have become better at writing and debugging code written in machine language or C. It has happened because of the development of languages and software principles that make it easier to build large systems that are correct by construction. We need the same kind of principles, (modeling) languages, and ideas to build correct concurrent systems. Verification alone is not enough. Work on design tools, ideas to guide design, and design languages.

Tom: In concurrency theory we define formalisms and study their properties. Most papers do the studying, not the defining: they take a formalism that was defined previously, by themselves or by someone else, and study a property of that formalism, usually to answer a question that is inspired by some practical motivation. To me, this omits the most fun part of the exercise, the *defining* part. The point I am trying to make is not that we need more formalisms, but that, if one wishes to study a specific question, it is best to study the question on the simplest possible formalism that exhibits exactly the features that make the question meaningful. To do this, one often has to define that formalism. In other words, the formalism should follow the question, not the other way around. This principle has served me well again and again and led to formalisms such as timed games, which try to capture the essence needed to study the power of timing in strategic games played on graphs. So my advice to a young researcher in concurrency theory is: choose your formalism wisely and don't be afraid to define it.

Rupak: Problems have different measures. Some are practically justified ("Is this practically relevant in the near future?") and some are justified by the foundations they build ("Does this avenue provide new insights and tools?"). Different communities place different values on the two. But both kinds of work are important and one should recognize that one set of values is not universally better than the other.

Mariëlle: As Michael Jordan puts it: "Just play. Have fun. Enjoy the game."