# A.M. Turing Award Oral History Interview with Joseph Sifakis

**by Cris Calude**

**Athens, Greece**

**September 6, 2019**

**Calude:** This is a video interview with Professor Joseph Sifakis for the ACM Turing Laureates video interviews project. We are on the Friday, 6th of September 2019, in number 3 Stisichorou Street, Athens, Greece.

Joseph received the award in 2007. Here is the citation: "ACM, the Association for Computing Machinery, has named Edmund M. Clarke, E. Allen Emerson, and Joseph Sifakis the winners of the 2007 Turing Award, widely considered the most prestigious award in computing, for their original and continuing research in a quality assurance process known as Model Checking. Their innovations transformed this approach from a theoretical technique to a highly effective verification technology that enables computer hardware and software engineers to find errors efficiently in complex system designs. This transformation has resulted in increased assurance that the systems perform as intended by the designers."

I am Cristian Calude from the University of Auckland, and my role today is to suggest a path for understanding not only Joseph's long career and tremendous achievements, but also to put a human face on them.

In 2008, I was privileged to meet and chat with you in Liverpool at the meeting of Academia Europaea. I wouldn't have guessed for a second that you were a recent Turing Laureate.

Starting with the beginning, you were born in Heraklion, Crete. Heraklion is Europe's oldest city as well as one of Europe's fastest-growing tourist destinations. But also Heraklion is the place where Epimenides discovered the paradox that "All Cretans are liars," which managed to get into the New Testament.

**Sifakis:** Yes. First of all, I would like to say that Heraklion, in fact Knossos was the capital of the Minoan civilization, which is one of the oldest in the Mediterranean. And yes, so Epimenides has lived in Knossos in Crete six or seven centuries BC, and he said the famous "All Cretans are liars," you know this…

**Calude:** And he is a Cretan.

**Sifakis:** He is a Cretan. So this created a lot of problems to logicians after that, okay? The famous paradox.

Yes. I was born in Heraklion to a family of merchants, middle-class merchants. My grandfather was a merchant and my father also. They had a prosperous commerce of dried fruits. And I was one of the four children of the family, the first one.

**Calude:** I know that you are married but very little else about your family, and even Dr. Google seems to know very little. [chuckles]

**Sifakis:** Okay. All I can say is that I am married with Olga Ioannides, the name of my wife. And she's a jurist. She's a prosecutor, in fact. We live in France most of the year, sometimes also some few months also in Greece, and we travel a lot together.

**Calude:** What did you like to do when you were a kid?

**Sifakis:** Oh, as a kid, I enjoyed a lot of freedom, because I liked playing a lot outdoors and our house was in the suburbs of Heraklion and it was surrounded by vineyards and fields, and even the streets were not asphalted. So I had a lot of freedom and this I enjoyed as a kid, yes.

**Calude:** Did you have heroes in childhood?

**Sifakis:** Yes. But in our childhood, we did not have so many books. We did not have comics. TV did not exist at that time. We had some books, heroes about ancient Greeks or from Greek history mainly, and that's it.

As a child, I would like to say that I was very curious. I did not have enough books and I kept asking questions, especially my father, about how devices work, about natural phenomena. And I was not always satisfied [laughs] from the answers of my father. I remember I had questions about electric bulbs or a radio station – "How works a radio station?" – and I was not happy with the answers of my father I remember. And of course my father was really unhappy because he could not satisfy my curiosity.

**Calude:** What were your best and worst subjects in school?

**Sifakis:** Oh. Something I did not like was physical effort. I mean physical education. I hate physical education. I don't want to be forced to do something, although I like physical effort when I decide to do some physical effort.

**Calude:** I was witness two days ago for watching you swimming for two hours.

**Sifakis:** Yes. I like very much swimming. Diving I like very much. Walking. But you see, also the educational system when I was a child was a bit… I mean it was based on force, and this I did not like. So I had very bad grades in physical education, 11 for over 20, and I was a good student from there. I liked everything in school, especially history when I was young, and later when I went to high school, physics and mathematics, yes.

**Calude:** When was your first exposure to computers?

**Sifakis:** Oh. Very late, very late. When I was 22-23, when I went to France. I have to say I studied electrical engineering. I went to high school in Heraklion and then I was admitted to the National Technical University after entrance examinations, and there I studied electrical engineering. And as an electrical engineer, I have not seen any computer at that time.

**Calude:** A five years' program?

**Sifakis:** Five years' program. I graduated in '69. I had not seen any computer at that time. Of course, I knew that there were computers. And the closest course we had to computing was some course on control theory. So the first computer I encountered later when I went to France.

**Calude:** When did you go to France and why?

**Sifakis:** Oh yes. I have to say that I studied electrical engineering in Athens from '64 to '69. Then I wanted to continue my studies. At that time, I was passionate about physics, physics and mathematics, theoretical physics. And, okay, my motivation so was to continue. At that time, we had a dictatorship in Greece.

**Calude:** The Colonel.

**Sifakis:** The Colonel. So the putsch. They organized the putsch in '67. Also I would like to say that I was a kind of opponent to the regime, so I had some difficulty to leave the country after I graduated in '69. I wanted to go to the United States to continue my studies, but for some reason I could not get a passport for this. Finally after waiting for one year, *[0:10:00]* I got a passport to go to France, to go to… a passport with a three-month visa…

**Calude:** Tourist?

**Sifakis:** …to go to France. To three countries, France, Italy, or Germany. I have chosen France and I went to Grenoble, where I started studies in theoretical physics. And there I encountered the first computer. The courses started beginning of October and I decided to change [laughs] to enroll in informatics the same year.

**Calude:** Was there a particular teacher who inspired you?

**Sifakis:** Yes. In fact, I made this decision because I met a professor who was the director of IMAG. IMAG, I-M-A-G, Informatics and *Mathématiques Appliquées Grenoble*. This was Professor Kuntzmann, with whom I had frequent interactions and I was very much influenced by him. Finally, I made a decision to quit my studies in physics and start my studies in computer science. So this professor had a deep influence on my evolution, because he was my supervisor for my master's degree and also for my engineering thesis.

**Calude:** I remember you saying something about the advice Professor Kuntzmann told you, gave you at some stage – "Do not pollute your mind with reading too many things."

**Sifakis:** Yes, okay. Yes, yes. This was an advice he was giving to all students. He has given me a lot of small problems to start, and every time he was saying, "Find your own way first in research and then look at the literature and try to understand the state of the art." I think this was very quite reasonable advice, because it's important to find your own way, not to imitate other people. And this is an advice also I have been giving to my students, yes.

**Calude:** But you stayed in hardware only two years?

**Sifakis:** Yes. With Kuntzmann, I studied hardware. I mean hardware modelling, because hardware modelling at that time was a very hot issue. And I did my engineering thesis on modelling hardware systems. The idea at that time, the dominant idea was that hardware systems were of course logical circuits but dynamical systems. And I tried an avenue that proved to be not very interesting in practice – to describe hardware as a system of finite difference equations and, okay, by equations in finite fields. For this, I was influenced also by a book by Moisil, who was a Romanian…

**Calude:** My former professor in Bucharest.

**Sifakis:** Yes, yes. We discussed about that. So the approach was mathematically elegant, but the complexity, I mean the idea comes from electromechanical systems. Electromechanical systems are described by using systems of differential equations, so by analogy one could think that you use finite difference systems to describe logical. But logical systems, even the simplest ones are intrinsically nonlinear. Nonlinearity gives systems of equations that are not tractable. So this is an avenue we have abandoned, everybody has abandoned.

**Calude:** You didn't return from Grenoble too soon to Greece.

**Sifakis:** Yes. When I went to Grenoble, it was August 1970. I had a passport valid only for three months, so [chuckles] I wanted to continue my studies and the colonels, okay, wanted me back to Greece to join the army and things like that. So I decided not to go back, and in fact I stayed in France for four years without any passport, without papers. This was a very hard time in my life. But also I was obliged to reconsider many, many things in my life, to change. I had to work very hard to survive. And that's what I did in fact. And for this, Professor Kuntzmann has been very helpful. We wrote some papers together and I got a position at CNRS, which is the National Center for Scientific Research in France. So I got a position, I could earn my life, and I was independent from my family.

**Calude:** And become official in France?

**Sifakis:** Yes, okay. And then I got established in France, yes.

**Calude:** What did your parents want you to do when you grew up? How did they feel about you becoming a researcher? Before knowing that you'd be famous.

**Sifakis:** [laughs] Yes, okay. I should say that of course my parents left me a lot of freedom in the choices I made in my career. When I told my father I will apply to study electrical engineering, he said, "Okay, that's great. You can do that." But my father believed that he was rich enough and he had a business where all his children could be involved and live comfortably, so he did not care so much, not understand this. But of course he was proud of my studies. Now when I went to France and I could not go back to Greece, this was a problem for my parents of course. They did not like this situation, and I can understand this. And also because at that time, the regime was considering that opponents were kind of criminals or whatever. So it took some time to my parents to understand that this was a good choice for me at least, yes.

**Calude:** You said somewhere that, and I cite, "We are a generation that had visions. Most of us wanted to become scientists, we had ideals. At that time, science had another glamor and I wanted to be a researcher since childhood."

**Sifakis:** Yes. Since I was a child, I wanted to become a researcher, okay, to know something about… Okay, so I was obsessed by mathematics, physics. I should say also that I had a very good teacher in high school mathematics who influenced me a lot, because he was teaching mathematics by making reference also to ancient Greek philosophers, to Pythagoras. And, okay, he was trying to show the beauty of mathematics. At that time, I loved Euclidean geometry. I was a champion in solving exercises in Euclidean geometry. So I was very much motivated to study.

And all my generation, people wanted to become scientists. Physics had a huge prestige at that time, great physicists talking about quantum mechanics, quantum physics. I remember meetings discussing, "Well, what is an electron? An electron — is it a wave or a particle?" things like that. And my generation, because perhaps we were living in a society after World War II without problems of unemployment, we had a lot of freedom in the choice of our career. When for instance I changed, I was already an electrical engineer and I said I start my studies from scratch and I become computer scientist, of course my family was shocked, but this was something quite natural then. Later, students become much more reasonable. The students I have known as a teacher later were much more pragmatic. *[0:20:00]*

**Calude:** If you had not chosen engineering, what else would you have liked to do?

**Sifakis:** Well, when I was very young I wanted to become an archeologist, because my father's village was close to the city of Gortys. Gortys was a very important city in ancient Crete. It was Minoan, then Dorian, and then Roman – it was the capital of Roman Crete. There I was visiting ruins and I was very much influenced by that. I was interested in history, ancient history, and I thought archeologist would be a good job for me. [laughs] And, okay, as an adult and then even now, I like very much to read historic books about ancient history. I'm very much interested also in ancient Greek language.

**Calude:** Yeah, and etymology of words too.

**Sifakis:** Etymology of words, yes. I spent a lot of time in my life studying ancient Greek and I love ancient Greek and Greek language in general, yes.

**Calude:** Except these two professors of math, who was an important influencer of your activity both in Greece and in France?

**Sifakis:** Regarding scientific life, yes, when I went to Grenoble, I passed my state thesis and it was mainly about model checking. Then I connected very strongly with Amir Pnueli. Amir Pnueli, who was working temporal logics at that time. Amir has influenced a lot my career because he was not only a very good logician and computer scientist, but we had many points in common – in particular the interest in ancient history, in history, and also in philosophy. So with Amir, we had a long collaboration that lasted for many, many years, yes.

**Calude:** Twenty-five.

**Sifakis:** Twenty-five years, yes, because he visited us in Grenoble. And we've been working in parallel for a while, because he was taking a different approach in logic and verification.

**Calude:** You received the doctorate in 1974 and the state doctorate in 1979, if I'm not wrong.

**Sifakis:** Yes. And before I had to change completely direction because my engineering doctorate was about hardware and then for some reason I decided to change, to go to another team. That was the programming languages team. The reason I did this was because I found that what I was doing on hardware was not so deep in mathematics. So I decided to go to the programming team.

And, well, I got interested in program verification. At that time, it was a very hot topic. And, okay, I had to work a lot because I had a different background. I had this idea too of model checking, of which we have developed the theoretical foundations and also some methods about how to do that.

So I passed the state doctorate in '79.

**Calude:** So this was more computing science than engineering?

**Sifakis:** Yes. And I should say that when I passed my thesis, in the jury people were doubting about the interest of the theory. Because it was a nice piece of theory, very mathematical to their sense, to their taste, but they could not figure out or at that time could not figure out that this could be applicable because of the state explosion problem. Some people also, I remember a member of the jury said, "There exists no such thing as finite-state program." He was right, of course.

**Calude:** It's a model, yes.

**Sifakis:** Yes, okay. So I passed my thesis in '79. Then, as I am a stubborn guy, I worked a lot to find some applications of my results. And, okay, I was lucky because I have interacted with some people who had ideas about how to verify protocols, because protocols are finite state. You can find good abstractions, finite-state abstractions of protocols.

So we tried the tool we developed…

**Calude:** first

**Sifakis:** … our first… Yes. But of course we had at that time limitations. We were able to verify systems up to 20,000 states only. But this was the beginning of this adventure that was the model checking.

**Calude:** If I jump a little bit, your current work on autonomous systems we will discuss a bit later, is benefitting this from your engineering background and experience?

**Sifakis:** Yes. Okay, yes. Okay. I have this background in electrical engineering and, yes, this has been very useful in all my career, because… And I think also electrical engineering should be integrated in computing curricula, because I was always interested not in programs but in systems. Programs are running on hardware and systems interacting with environments, and this was the key direction in my research.

**Calude:** Did you code?

**Sifakis:** Not so much. [laughs] I don't… As a student, I had to code. I had to write simple programs. I remember I started with a PDP-8. This was a minicomputer of DEC. And we were writing programs on sheets that we were handing to operators, ladies that were working with…

**Calude:** Punching.

**Sifakis:** …these punching card machines. So you had the listing… So they have the execution listing eight hours later.

Then I had also to write some programs for hardware simulation. But at that time, the languages we were using were assembly languages, something like that. And then I also had to program in a language like ALGOL, a language that disappeared meanwhile. [laughs]

**Calude:** Yeah. For a while, it was useful as a theoretical tool, but not…

**Sifakis:** Yes, ALGOL. I mean these are the pure languages…

**Calude:** 68.

**Sifakis:** …yes, the pure languages that people design. And later I had to deal a bit with Ada for instance. But all these languages disappeared because they were designed by people who believed that theory was of some use or of some interest.

**Calude:** In 1993, you founded the Verimag labs, which… in a sense it's a mark of your philosophy of research – it's both theory and practice.

**Sifakis:** Yes.

**Calude:** Can you tell us more about…?

**Sifakis:** Yes, I can. Yes, okay. But first of all, I would like to say why I have created Verimag. Because, mid of '80s, we had the first results on applications of model checking that were quite encouraging, and also I had established a small team. I had my own team with 10 researchers and we had very interesting ideas at that time about how to build systems. Of course, at that time software was not so important. We had people working on software for business, for… yes, business applications, commerce, commercial software… *[0:30:00]*

**Calude:** COBOL.

**Sifakis:** COBOL and things like that. Also, in France you had industry interested in real-time applications. So we had the background that allowed us to understand the importance of these applications.

So we started Verimag cooperating with some industries, like… Okay, these industries, their names disappeared. There was Telemecanique, that became Schneider Electric, in Grenoble, and Aérospatiale, that became Airbus, at that time. These industries were demanding methods for the development of systems that were rigorous and were asking for guarantees. If you have a software running on a nuclear plant or on an aircraft, they wanted at that time to… the idea was to have fly-by-wire, they wanted guarantees. So they were interested in methods and they thought that our methods could be used.

So we've been approached by a company that was producing software at that time. The name of the company was Verilog. And we created Verimag, which is "Verilog" plus "IMAG." IMAG was the Institute of Informatics and Applied Mathematics. And this gave Verimag. So this is the way we started Verimag.

**Calude:** Could you please explain the concept of formalization of system design?

**Sifakis:** Oh. Yes. Now we are jumping to something else. Okay. Yes, but Verimag was about designing systems, in fact. Design is a universal concept. I mean you design artifacts in general – design houses, design bridges, design engines, whatever. So the idea of design is that you start from requirements that reflect your desires, your needs, and then you want to build an artifact. This process that leads from requirements to the artifact, this is called "design." And engineering is about designing, of course, artifacts. And design in fact involves two steps. One… System design. One step is you start from requirements, you write a software, application software, and then the application software, you run it on a piece of hardware, on a platform. You have these two steps – one is writing the software and the other is generating from the software the code and all the glue of course that will allow the system to interact with the environment. So system design is much more complicated than writing, designing software and developing software.

**Calude:** What does it mean in this situation, "correct-by-construction implementation"?

**Sifakis:** Yes. These ideas of building systems that are correct by construction, I developed them because I realized sometime in my career, mid ’90s, that verification was somehow hopeless. I mean you cannot develop complex systems based only on verification. Verification does not suffice, and I can explain why later. So the idea was that you have some recipes about how to build, to make things correct, and the recipes are like theorems that you apply. I think that the most useful part of computer science is algorithms, protocols, architectures, just recipes about how to do things. And of course you need some theory about how to combine these results.

This is the idea of correctness by construction. For instance, there are architectures. We know what is a token ring architecture. Token ring architecture is a structure that enforces the way that some components communicate, and this will allow the satisfaction of a property. Each good architecture enforces the satisfaction of a property. This is the idea of correctness by construction, which I believe is very basic in all engineering, I mean systems engineering discipline.

**Calude:** And somehow this has to certify, to give the user some trust in the product.

**Sifakis:** Yes. But this is something, yes, different. We should use techniques that will provide some evidence not only to the user but to the certification authorities, to the institutions that control the quality of the systems we develop that the system will behave as promised. This is true of all artifacts we build. If you buy, I don’t know, a toaster, this is certified. It is certified that if you use it properly, it will not kill you. For any product you buy, you have certification. So one more reason to have certification for flight controllers, for the critical systems.

**Calude:** In this guise, also this certification has to satisfy some obvious properties. Like certification should probably not be done by the designer itself.

**Sifakis:** Of course. No, no. Of course the certification should be done by authorities, by bodies that…

**Calude:** are independent.

**Sifakis:** … are independent bodies. Otherwise it’s not… I mean if the certification is done by the builder of the system, it’s completely… I mean the builder can cheat, of course. Yes, you need independent authorities. And the problem of the system developer and the system manufacturer is to prove, to provide evidence to the authority that the system will behave as expected, that it is correct in fact, has no major flaws or whatever, defects that can be critical.

**Calude:** You have done wonderful work with many colleagues, but specifically I would like you to tell us a little bit about the joint work with Maler and Pnueli.

**Sifakis:** Yes. This work we did with Pnueli and Maler was about… it's a continuation of the work we did on model checking. The idea is to synthesize controllers. What does a controller? A controller acts on an environment so as to enforce a property on the environment. So we have developed some techniques that start from the property and then a model of the environment, and in the model of the environment you distinguish between controllable and non-controllable actions. It's like a game. The controller acts on controllable actions and of course the environment can play on controllable actions, and the purpose of the game is that the controller wins when it satisfies the property. And we have developed some algorithms for time systems. So automata with timers, which are very important for real-time control.

This is a work we did with Amir and Maler, and Maler continued in fact in that direction. And also with Amir, I worked on hybrid systems. Hybrid systems, that are systems that combine continuous and discrete dynamics. So you have automata say and differential equations.

**Calude:** Time is continuous, probably.

**Sifakis:** Time is continuous. But what can be shown is that you can… even for continuous time, you have the concept of transition system, but these are dense transition systems. Okay, they have some properties, but that most of the theoretical results we had about model checking could be extended to hybrid systems and applied to hybrid systems. *[0:40:00]*

**Calude:** You are the Scientific Coordinator of the ArtistDesign European Network of Excellence.

**Sifakis:** Yeah. Now you are jumping to the times I worked on embedded systems. Having worked on verification and synthesis of systems, mid '90s I decide to quit verification and to be interested in design. At that time, the concept of embedded systems was emerging. So systems that, okay, if you have computers that continuously interact with physical environments.

I worked with other colleagues to start some research activities in the area, both in the United States and in Europe. As a result of this action, we launched the ARTIST Network of Excellence. That coordinated the European research on embedded systems for 8 years or 10 years. And I'm quite happy from the result because we brought together communities that were not interacting, real-time systems community, formal methods community, scheduling community, architecture community. So I think the result was quite interesting finally.

But I should say that, you see in research we have fashionable terms that appear. Mid of ’90s, we had “embedded systems,” and then 10 years later a new term appeared. That was “cyber-physical systems.” Cyber-physical systems are systems where you have electro-mechanical parts and computers, and so they are built from cyber-physical components. So you don’t have, as for embedded systems, the computer and the electro-mechanical environment, but each component is an embedded system with electro-mechanical environment, which makes the construction more modular and the interaction between computers and the cyber-physical environment more tight. You achieve better integration.

**Calude:** Can you tell us a bit about the rewards and challenges involved in the cooperation with big companies like Airbus or France Télécom?

**Sifakis:** Yes. Okay. I like to participate to big projects, and I have collaborated with companies like Airbus of course, but also companies that built the satellite systems and also STMicroelectronics, which is an important company in Grenoble area. I’ve been involved in some projects on critical systems design, starting from requirements up to the end. And I liked it. I learned a lot. By doing this, I realized also that I had some misconceptions about the interest of theory, or I mean the theoretical results, the application of theoretical results. Typically, I thought that you start from requirements and you refine the requirements and you find the system. Okay, you decompose. I mean this idea of V-model that has been very popular sometimes in systems engineering.

And I realized at the time that this is not practicable. Why? Because you never build a system from scratch. First of all, it’s very hard to understand the system requirements. If the system is complex, it’s not like the requirements for a program. It’s not a simple input-output relation. It’s very hard to understand system requirements and to formalize them. This is one problem. The other thing is that you realize that complex systems are not, are never built from scratch. You build the system, complex system because you know, you have known how to build a simpler system. So you never say, “I start from scratch and I will build a very complex system.” This does not work. [chuckles]

**Calude:** A plane.

**Sifakis:** No. Or even a flight controller. I mean even in engineering, what is very important is the idea of architecture. So you see, I’ve seen critical systems architectures, architecture for aircraft or for rocket. It’s the same architecture, and this architecture is used for decades just because it’s a principle and this allows you making things correct. So you cannot say, “I will start from requirements and I will reach some other different ideas.” It’s very hard. I mean systems developed as complex systems are just improvements or incremental modifications. It’s rare to build a system from scratch and succeed. This is what I learned.

**Calude:** We are going now to the hot areas in computing science – big data, AI, and quantum computing. In 2008, Chris Anderson, at that stage was the editor-in-chief of *WIRED* magazine, published an essay with a provocative title, "The End of Theory: The Data Deluge Makes the Scientific Method Obsolete." This was based on the idea that big data and machine learning will replace completely the scientific method. Do you agree?

**Sifakis:** Okay. [laughs] This is indeed a very provocative question. I cannot agree with this idea, because people do not even understand that there are different kinds of knowledge, and the knowledge is characterized by its degree of truthfulness. So you have mathematical knowledge. Some mathematical knowledge is not falsifiable. If you believe that, you accept the axioms, it's eternal knowledge. And then you have knowledge that is empirical knowledge. Empirical knowledge is knowledge about the physical world, and this also can be classified. Can we have facts, simple facts? You have like we know that today the temperature in Athens is 30 degrees. This is a fact. Then you have generalizations of facts. That is empirical, general empirical knowledge. And you have scientific knowledge. Scientific knowledge is empirical knowledge that is explainable by models. Now I trust scientific knowledge because it is validated by mathematical models, and we can… I mean it is explainable knowledge.

The problem with knowledge produced by machine learning systems is that it is just empirical knowledge. We don't understand how, we cannot explain how this knowledge is produced, and this is a big difference. So I cannot give the status of something I can trust to knowledge produced by machines. As a systems engineer, I cannot accept this idea.

**Calude:** Why should we bother about trust?

**Sifakis:** Oh. Because everything is built in human society on trust. I mean human society, you trust. There are rules and you trust that other humans will behave in a certain manner. And trust allows predictability in fact in society. So I trust that we'll be respecting stops when you drive, okay? So I don't brake *[0:50:00]* if there is a stop for somebody else, and I cross, and I trust that he will respect the rules. So in order to have performance in society and safety, we need the trust. And you trust that a bridge will not collapse. You trust that the bridge will not collapse because you have theory that guarantees that the bridge will not collapse, and you trust that the lift of the aircraft will not… yes, okay, the lift will not burn, whatever. So everything is… trustworthiness is very, very important in modern societies. The problem with extensive use of these new techniques is that we cannot have guarantees that these systems will perform as expected. That's the problem.

**Calude:** To trust, we need to understand. And I would like to go back to the first level of knowledge, the mathematical level you described so well. In mathematics, understanding is extremely important. For instance, if you will ask a normal mathematician, "If I tell you that the Riemann hypothesis is true, and I can give you reasons but I'm not giving you a proof, would you be satisfied?" and the answer would be no, just knowing…

**Sifakis:** Yes. But even if I have a proof, your proof should be readable. So…

**Calude:** Readable is not necessarily to be understandable.

**Sifakis:** Yes, okay. But if you read it and you proof-check it…

**Calude:** You should.

**Sifakis:** …you should, yes.

**Calude:** And of course it should be understandable for a person who has the right education.

**Sifakis:** Yes, yes. But you see, this is also another fact about knowledge and truth, because… okay, we know plenty of stories about that. If a mathematician says, "I have proven Fermat's tenth theorem," how many times people said, "I have proven it" and it was not correct?

**Calude:** Sure.

**Sifakis:** So it's not enough to say, "My system is good enough" or "My program is correct." You should also, some institution… For mathematics…

**Calude:** Community.

**Sifakis:** …it's the community of mathematicians that read the proof and say, "Oh, indeed, this guy has proven Fermat's theorem." Now when you write a software, when you develop a system, you should provide also some evidence to a third party, to an institution, to a certification authority who would say, "Oh, this software," because it's a critical software, "will behave…" I mean "has no major flaws. There are no major defects." So we need this guarantee that the system will behave as expected.

This is the role, in fact. In any organized society, this is the role of institutions in fact. Institutions decide… I mean say, "This is right," "This is wrong," "This is correct," "This is not correct." This is the role. In any modern society, we have institutions that play this role.

**Calude:** Okay. I would like you to comment on a short paragraph from an essay published by a mathematician, Strogatz, in *New York Times* last year, which reads as follows: "Maybe eventually our lack of insight would no longer bother us. After all, AlphaInfinity" – which is considered to be a hypothetical descendant of AlphaZero, the famous program of DeepMind – "could cure all our diseases, solve all our scientific problems and make all our other intellectual trains run on time. We did pretty well without much insight for the first 300,000 years or so of our existence as Homo sapiens. And we'll have no shortage of memory: we will recall with pride the golden era of human insight, this glorious interlude, a few thousand years long, between our uncomprehending past and our incomprehensible future."

**Sifakis:** This is a very common idea now, that we should leave machines decide about us and machines could replace humans in their mission in complex organizations. That's what we call "autonomous systems." But here I would like to make a comparison. You see, this is a great revolution, and in the past, a similar revolution was the discovery and use of mechanical tools. Mechanical tools allow us, allowed us to multiply our muscular force and to be able to build pyramids or bridges or whatever. All the mechanical, all the technical civilization is developed thanks to the use of machines.

Now with the advent of computers and especially of AI, this is something that extends our mental capabilities, not our muscular capabilities. And of course when you use a technology, you may lose some skills. For instance, my prehistoric ancestor living in caverns, using tools, he has lost some of his muscular skills. Now the question is whether we can use extensively AI and let machines decide for us, because what characterizes humans is their consciousness, their ability to judge and apply their free will. So we should think about that.

**Calude:** Following your analogy, would this be adequate to say that if we allow this type of new revolution, we should be prepared to pay with a diminished human thinking?

**Sifakis:** Yes. But take examples, normal simple examples. If now young children do not use computers just to do some multiplication, some division of numbers, so they don't learn basic ways of thinking about arithmetic and they lose skills, and of course we may lose also skills, I think that if we let machines decide for us, then we can lose fundamental skills. And also there is the risk that we cannot control what they are doing. If, for instance, we use machines to make decisions about how to distribute resources, about how to manage cities, how to manage transport systems…

**Calude:** Or health, our health.

**Sifakis:** …or health, or our health, there is a danger that the machines make decisions that we don't understand and we cannot control. This means that we will have to trust machines. And I think we will have also to trust the engineers that train machines to make these kind of decisions. This can be a very dangerous situation, yes.

**Calude:** So, if I follow up, this means we should be hard thinking about these issues till we have the ability of thinking.

**Sifakis:** Yes, yes. And this now raises also some ethical issues. And of course societies should think about the way these techniques are used.

**Calude:** Going to a more general level, can you tell us what are the projects or accomplishments you are most proud of?

**Sifakis:** The accomplishments I'm proud of or I got satisfaction from. Of course model checking. This is a piece of, okay, fairly good technical work, fairly complete.

**Calude:** We will discuss a little bit a little later about… *[1:00:00]*

**Sifakis:** Yes, we discussed, yes, the idea of… At that time, it was a really new idea and changed the state of the art in verification. Then I like very much all the work I did with Amir Pnueli, Oded Maler, and also Tom Henzinger on model checking of hybrid and time systems. This is work I developed for 10 years.

I like also very much the work we did in Grenoble and in Lausanne on BIP. BIP is a component framework where… this is a component framework with theory that allows you building systems that are correct by construction. We discussed a bit about that. This is a piece of work that is I think quite important but did not have yet the recognition of model checking, unfortunately. But this is quite interesting as results.

Then of course I consider also as an achievement, but this is not purely scientific, is setting up Verimag, managing teams, setting up ARTIST. I'm proud of this also.

**Calude:** I'm looking on my notes about the list of awards and prizes you have got, and…

**Sifakis:** Don't give the list.

**Calude:** I don't give the list. It's too long. How important these distinctions and recognition have been for your career?

**Sifakis:** I would like to say only one thing, because I know some people that I think in their research program, there were some parameters about how to achieve this to get this award. Personally, I have chosen my work direction because of my personal taste, because I like this type of work. I never thought about distinction and recognition. But of course I am sensitive to recognition, in particular to recognition from my peers.

**Calude:** What was the most exciting period or project in your career?

**Sifakis:** Yes. Of course, again, model checking. Yes, some projects that were industrial projects with some industry. Yes, projects on satellite systems and also projects on more multimedia systems with STMicroelectronics. We got very interesting results.

**Calude:** You have many collaborators from all around the world. Would you say that you work better in a team?

**Sifakis:** There is a type of work I would do with a team. I need a team, I need collaboration, I need engineers, in particularly the work around BIP, the work on tools. And there is work I like to do alone. And of course I don't like collaboration for the sake of collaboration.

**Calude:** Well, we have got here on this table your second book of poems. This is clearly alone. [laughs]

**Sifakis:** [laughs] Okay. It's here by chance? Yes. Why? The question is about poetry?

**Calude:** No, because it's a clear alone work. It's not a collaboration.

**Sifakis:** Ah, yes, yes, yes. Yes, yes. I write poems alone. Yes, yes. I like poetry very much indeed, yes.

**Calude:** If you could change a decision or an event in your life, what would it be?

**Sifakis:** Oh, this is a very hard question to answer, to change something in my life. I think that in my career as a researcher, I did many things I made wrong decisions or I had bad intuitions. For instance, during a period I tried to combine process algebras and temporal logics. And the technical problem is very hard and the interest is very limited. So I spent three years of my life doing this type of work which technically was quite hard, okay, and not very relevant. But we learn from our errors.

**Calude:** Did you ever doubt about yourself?

**Sifakis:** Doubting about myself. What it means, doubting about myself? Okay. Very often I question myself about the choices I make. So all of the values, my values. So yes, in some periods of my life, I even have undergone some kind of crisis when I was younger, questioning whether what I was doing was the right thing. Yes. And it's very useful. If you overcome the crisis, this can be very interesting for you, yes. Yes, I think you should question what you are doing every day, yes.

**Calude:** You have been involved in many projects in Greece, and one of the most interesting was to be President of the Greek Council for Research and Technology. Can you give us a glimpse of the work you have done…?

**Sifakis:** This was not a project, by the way. I mean this was just a mission to be president of this… yes. Yes, I had to… The role of this committee was to advise the government about the choices in management of research and technology. We formulated… For two years I was president of this committee, which is quite prestigious, or used to be quite prestigious in Greece. We formulated some proposals. I'm not sure that the proposals have been followed. You know that the country has been undergoing some deep crisis. So, okay, it was an interesting period of my life because I have reconnected also with Greece and I thought also that… I accepted to take this position just because I thought that I owed… this was something I had to do as a Greek, because my home country was undergoing this crisis. Now regarding the results, I'm not sure that the recommendations have been followed. But at least coming back to Greece, because you mentioned the poetry, okay, this inspired me a lot because I wrote two books, poems inspired by the situation in Greece. In fact, the crisis as a phenomenon, so economical crisis, but I think that any economical crisis hides another crisis that is more a crisis, a crisis of political life in a country.

**Calude:** And we have not one, not two, just a series of such crises nowadays.

**Sifakis:** Yes. Everywhere in the world, yes. That's a problem, yes.

**Calude:** At some stage, you said that from the perspective of being president of this advisory committee, that the level of research in Greece is comparable with top European research, but what is missing is the link between research and real economies, how…

**Sifakis:** Yes, yes, yes.

**Calude:** And you are mostly well-positioned to comment about this because of your work.

**Sifakis:** Yes. I think that in Greece you have very good scientists and you have a very strong Greek diaspora, scientific diaspora in the world. As a rule, we have very good professors here in the universities. What is missing is the link with economy. I believe this is very, very important. If today you want to… I believe that the research should be relevant somehow. And this has been a debate for many years in Greece, whether researchers should be…

**Calude:** Pure. [laughs]

**Sifakis:** …pure. I mean okay, these pseudo-dilemmas. And my point is of course that there is no… I mean you have either good research or bad research. *[1:10:00]* I mean it can be theoretical, it can be practical or practically motivated, applied. But you see this is a pseudo-dilemma that was at the center of a debate in the universities for decades in Greece. What I keep saying is that the best, the university that produced the best theoretical results develop the best applied research also and have startups, connection with industry, etc. Look at the bigger technological universities in the United States, for instance.

**Calude:** But I would also mention, because you have been so many times, Israel, which is also top and a very small country.

**Sifakis:** Yes, okay. Very small country, yes. And also in Europe you have universities that are very well positioned. I've been a professor at EPFL for five years. So these universities, ETH, Cambridge, okay, you have some universities in Europe that are top-level and they are producing top theoretical results, first-class theoretical research and also applied, and they have impact on a real economy.

**Calude:** Okay. Now we can focus on the Turing Award. If you could please explain to us in simple terms about the model checking and how this made this formidable link.

**Sifakis:** This is a question I had many times, so I will try to give an answer that is simple enough. Verification of a system is about checking that a model of a system satisfies a property, and the property is a correctness property, a desirable property.

Now when I started working on model checking, the dominant approach on verification was axiomatic verification. Axiomatic verification roughly speaking is about a reasoning, okay, you produce a proof that the model of the system satisfies the property by reasoning on the structure of the description of a system. And of course there are well-known limitations of non-decidability. So you can formalize proof rules, give proof rules, but achieving the proof depends on how much smart, intelligent is the guy who carries out the proof.

My idea was that we should have verification techniques that are automated, and to make verification automated, of course you should get rid of the limitation of undecidability. So I decided to restrict to finite-state systems. And in fact, the example of protocols was really motivating, because these are harder to verify and still finite state. I mean you can find reasonably good finite-state systems.

So what's the big difference between model checking and the other techniques, the previous techniques, the axiomatic techniques, is that you extract a model of the system and then you evaluate. In order to check the property, you evaluate the property on the model. And this… Okay, now I'll skip technicalities. This is like computing a fixed point of some monotonic function. And of course if the model of a system is finite, the computation will converge and then you have a proof math.

This method differs drastically from axiomatic verification because it's global. It can be applied so you know about the structure of the description, you generate the model, and flatten the system. So you should know the semantics, okay, semantic model of the system, and then you perform some evaluation. And if the model is not complex enough, then you… And also this method works for non-terminating systems. Because – you probably know that – for non-terminating systems, particular parallel systems, axiomatic techniques were not very easy to apply. This is quite general and this is applicable to non-terminating systems, which are in my opinion the hardest system to verify. This, so model checking has initially been applied to hardware systems, because all of these reasons…

**Calude:** finite automata

**Sifakis:** …finite state.

Then to software. The application to software is much harder, because the hard problem here is how to find a model of a software. So some abstract model of software. In order to do that, you should… if you have a program written in some programming language, say Java or C, the model is generated by taking into account the operational semantics of the language, and this of course involves a lot of technical issues. In particular, if you should decide perhaps the semantics of existing programming languages is not very rigorously defined, so you have to deal with that also. I had a student working on the semantics of Java, real-time Java. And then you see that Java, okay, you have a lot of issues, semantic issues that are left open, and this should be decided when you do this exercise.

Today model checking of software is very important for big companies. But for this, you need some technology that is heavy. I mean you need the compilers to models, abstraction techniques. It's fairly complicated technology.

And just to be complete, for systems, mixed hardware and software systems, it's very hard to define models. Modelling mixed hardware-software systems is really a problem.

**Calude:** To close the Turing Award work, what did you work after receiving the prize?

**Sifakis:** I said that in ’95, I decided to quit the verification community and go… Okay, I got interested in embedded systems in design. So I left the verification community. At that time, of course my colleagues did not understand why I was doing that. For me, this was the beginning of another adventure. Design. I think design is as important as science, because science studies laws that allows us understanding the world – so going from phenomena to the models. Then design is how to go from models to artifacts to the real world. So it’s really important as science.

And I think that system design deserves at least the attention of theoreticians. Now the problem is that theoreticians in computer science are interested in simple theoretical settings. You know this. It’s easier to write papers. And design, for design you need to mix many different considerations, many different things, hardware and software. And, okay, it’s not easy to produce papers about design.

**Calude:** Right. You need also…

**Sifakis:** Or at least theoretical papers, yes.

**Calude:** …yeah, a deep understanding of different parts.

**Sifakis:** Of different parts, yes, yes, yes.

**Calude:** On a more general level, we all made mistakes, and sometimes they are benign, sometimes they are not. Does any of your publications include a serious error with consequences?

**Sifakis:** Certainly my publications have errors. *[1:20:00]* I think that… There are benign errors. Typically it happened to me that I have the right intuition, so I understand that a proposition is true, and it’s true indeed. But perhaps you don’t give all the arguments or…

**Calude:** As a proof.

**Sifakis:** Yes, okay. Or you keep some arguments. And this happens very often I think.

Now, is there any paper that has serious flaws? Not to my knowledge, but perhaps others could discover later. [laughs]

**Calude:** [laughs] In mathematics, sometimes mistakes – and when I say “mistakes,” not mistakes made by students or in exams, I mean in research papers – have been the source of new ideas.

**Sifakis:** Yes.

**Calude:** Is it the case also for your areas?

**Sifakis:** Ah, okay, I don't know any example. But okay, some cases, for instance you have some flawed protocols that have been proposed and have been the object of a lot of discussion. Famous cases of protocols that have been used for years and then you discover flaws. Flaws also, I mean if you do something wrong, this can be a source of inspiration also. Yes, I agree.

**Calude:** Is computer science or computing science or, even perhaps better, informatics as a discipline as respected as mathematics or physics?

**Sifakis:** Well, it depends on what you mean by "respected," or respectability. I think that of course in the beginning, computer science was considered as something not very noble or a discipline that had the consideration of physics or mathematics or biology. But I think that progressively computing has acquired the status and respectability of other disciplines.

For me, computing is a domain of knowledge as important, as essential as physics or as biology. I'm considering that there are three basic domains of knowledge. Of course mathematics is apart because mathematics provides models for all disciplines.

**Calude:** It's not also a science?

**Sifakis:** It's not science. Also this is an interesting discussion. The reason I am talking about "domain of knowledge," not "science," because for me knowledge is about understanding the world, and science allows this, and also changing the world or solving problems, and this is engineering. So knowledge is a combination of these. And in fact computing has a scientific part and a scientific… yes, covers scientific aspects, scientific issues, and engineering issues. Scientific issues are about to study computers, to study phenomena of computation not only in computers but in biology, whatever, and the engineering part is about how to build effectively computers.

Now I said that there are three basic domains of knowledge. Physics – this is about understanding phenomena of the physical world and also the engineering, corresponding engineering counterparts – then biology of course, and computing. Computing is about computation. And biology is a domain apart because, in living organisms, we have physicochemical phenomena and we have computational phenomena, but these are very entangled so we cannot separate them. If someday we can separate them, I think biology would become not a basic domain. But, okay, I don't think they would be able to separate it.

To answer your question, I think that, yes, we have reached this respectability, but still it depends. Because it's respectability in the public opinion and respectability in… You know, I'm a member of academies, so sometimes I feel that my colleagues, physicists or mathematicians, do not consider yet that computing deserves the status of first-class discipline. But with time I think things will change.

**Calude:** Yes. In any discipline, take it mathematics, physics, biology, there are fashions and there are areas which are prominent at some stage and they disappear under the shadow. But it seems that in computer science, the life expectancy of, well, even theory and much, much faster application is smaller. Would you think that computer science or computing science will live as long as mathematics?

**Sifakis:** Oh yes, I'm sure about that. But, okay, as I said, mathematics is eternal. But yes, we need computers forever now, and I think computers is things that will stay with us and informatics – I prefer the term "informatics" – will remain a very active discipline as other basic disciplines. Yes, no doubt about that.

**Calude:** Kids know a lot about computers, smartphones, iPads, and so on. Is there any point in teaching informatics in schools?

**Sifakis:** Oh, that's a good question. Yes, I thought about that, because I am a member of some boards of foundations that are interested in that. And, okay, I think that computer science puts a lot of emphasis on programming only, and this appears also in curricula. I mean I think that the focus should be not on writing programs only but on developing systems, and I think that the curricula, computer science curricula should be broader to… and also when we talk about computers and teach computing in schools, we should motivate computing not by writing just simple programs but developing systems. Okay, so developing a small robot or robotic application, developing some sensors, some systems, some smart systems. So the focus should be on systems. And if this is introduced also in education, I mean elementary/primary school and high school, I think this should be a great revolution also in teaching.

You probably heard about STEM – science, technology, engineering, mathematics – which is a very fashionable term now in education. I think that the kids should learn not only how to program but to connect computers to all the other disciplines and build systems. Building systems is something very important. And this also should be the dominant approach in university education, not to produce simple programmers. I'm shocked that programmers do not understand anything about say signal processing, about… they don't understand, they don't know what is a Laplace transform, they don't know how to… So computer scientists should have a broader background.

**Calude:** So should the informatics be interdisciplinary?

**Sifakis:** No, no. This is a term I don't like, "interdisciplinarity." Okay. No, no, no. Okay. What I meant is that informatics now should… is much broader. The scope and the perimeter of informatics is broadening, and this should include… this should be related *[1:30:00]* to systems engineering in general. Now even if you design cars, you should know a lot about computers. Computers is related to everything, okay? We talk about cyber-physical systems. Cyber-physical systems, if you don't understand what is an electro-mechanical system, how you can design cyber-physical systems? So all this vision of IoT, Internet of Things, you need engineers and scientists with a broad spectrum of competencies and skills.

**Calude:** Is theoretical computer science of any use to informatics, or is it just an esoteric part of mathematics?

**Sifakis:** [laughs] Okay. Depends what you call "theoretical computer science." If theoretical computer science is only say complexity theory, decidability, things like that, of course you need this to understand what is possible, how much it would cost to solve this problem. But you don't learn how to write programs, we don't learn how to build systems. So I think this is a body of theory you need to know, but this is not enough. Of course the question is… I come from another area of theoretical computer science that is more semantics and logic. It's questionable whether all this is useful today. I think you need a theoretical background, solid theoretical background to understand what is possible or how to formalize things, but you need much more in order today to be able to build systems or I mean, yes, to design systems or to use systems effectively.

**Calude:** Does academia breed snobbery and arcane solutions which are irrelevant for real applications of informatics?

**Sifakis:** Okay. This is a question about academia. I think that academic institutions, okay, they involve people, and people try to survive in the system with their background. So people are often resistant to change. And in computing or informatics or IT, things are changing very, very fast, so people tend to resist to change. We've been talking about theoretical computer science or, okay, some schools in computing. They resist to change. And I think people should follow the evolution; they should not follow the fashion. But you see, there is a kind of forefront that is moving, and we should be interested in problems that are real problems for the evolution of the discipline. You have too many people in particular… Okay, not so much in the United States because in the United States, researchers, professors have to follow the movement because they are obliged to do so. But things are much more static in Europe.

**Calude:** Well, hot areas like big data, AI, quantum computing are marred by myth and hype. Is it something natural for any area to do, or for any successful area? What do you think?

**Sifakis:** Yes, I think that once you have a success somewhere, you have some over-expectations immediately, and so you have a period of hype. You've probably seen this "hype cycle" by Gartner. So you have discovered that, where expectations increase, you reach a peak and then you have a period of disenchantment or disappointment of people, and…

**Calude:** winter.

**Sifakis:** Winter. And this is the case I believe today for AI, but it happened for AI to have ups and downs. We remember this, so many winters. And also I know very well the winter now of formal methods. For formal methods also we had a lot of over-expectations, and today almost nobody talks about formal methods. I mean, yes.

So this is quite natural. I think that we should keep our mind cool and not to be… I mean not to follow this movement. And that's what I'm trying to do. I'm trying to resist to that and say, "Be cool," and just have a more, okay, reasoned position.

**Calude:** Preserve our human thinking?

**Sifakis:** Yes, yes. And especially for me as a systems engineer, yes, I think that. I resist to all the ideas that want that we solve *any* problem now by using AI.

**Calude:** What is the next big challenge for informatics?

**Sifakis:** Oh, that's a very good question. I think that for informatics, as I envision the future of informatics, I think that the big challenge is autonomy. Let me take some time to explain this idea of what is an *autonomous* system as opposed to an *automated* system.

We fairly well understand, we use automated systems – we use lifts, thermostats, flight controllers. These are automated systems. What is an autonomous system? An autonomous system is a system that replaces a human in his role, in his mission in a complex organization. So an autonomous system is a self-driving car. An autonomous system is a system that manages a factory, a smart factory. This is about autonomous systems – you replace workers by robots. So…

**Calude:** Autopilot would be?

**Sifakis:** No. An autopilot, no. Okay. I have a technical definition about that. What characterizes an autonomous system? First of all, an autonomous system manages many different goals. The goals may be conflicting. An autopilot has a single goal or a few goals. Or a thermostat has a very simple goal – just to keep temperature between two bounds. And the lift also has very simple goals. Now an autopilot for a car has many different goals, long-term goals and short-term goals, and this should be managed jointly to get the best possible result.

So an autonomous system differs from an automated system due to the fact that it has to manage many goals, not a single goal. A second difference is that it has to deal with complex environments, cyber-physical environments. And the third difference is that it has to cooperate with humans. So we should have a kind of symbiosis between humans and systems.

These are the main characteristics of autonomous systems. It's clear that, if you talk about a thermostat, this is not an autonomous system. It's an automated system – here's a single goal. While a self-driving system has many goals, has long-term goals, longer-term goals – "I'm here, I'm driving to another city" – but also very short-term goals, avoid collision and keep following some trajectory, pre-defined trajectory.

So the problem with autonomous systems is how to design them? They combine… I characterize them by four different functions they should combine. One is perception. Perception means that you receive *[1:40:00]* information about your environment and you identify concepts. So perception, and here machine learning is very good for that. You receive images and you say, "Here is a car," "a pedestrian," etc. This is perception.

Then you need the reflection. Reflection means that now I will build a model of my environment. Perception and reflection allows what I call awareness about your environment. This means that I can understand what happens in my environment. This is awareness about the environment.

Then you have two other functions. One is goal management, and planning. Goal management is about managing the different goals, and for each set of compatible goals, you should be able to plan them and generate action plans that will modify the state of your environment.

These four functions characterize autonomy. It's interesting also to say that, if I have a chess-playing robot, a chess-playing robot needs some of these functions, not all, because perception is not very important. I mean the environment is very static, okay.

**Calude:** What about the Go?

**Sifakis:** The goals? The goals also are well defined, well understood, okay?

**Calude:** No, I'm sorry. About the game of Go?

**Sifakis:** Ah. It's of the same type, but here you have… what is difficult is the planning. So you understand perfectly the goals, because for self-driving cars even the goals are hard to formalize. Here the goals are very well defined, the rules of the game are very well defined and understood. But we have an inherent complexity in the design of the planning, because planning is very hard, is much harder then for cars. So the hard problem.

So just summarize. For automated systems, the planning is very static. It's static. It's just an automaton. For autonomous systems, decision making is dynamic. You don't know statically what are the possible situations, or the possible configurations of the environment you will play with is extremely large like in chess playing, okay, so you have to dynamically decide this, how to plan. This is the big difference between autonomous systems and automated systems. And just summarize, I think the big challenge is how to build trustworthy autonomous systems today.

**Calude:** It seems that you are really into research in this area, not just interested about.

**Sifakis:** No, no, no. I'm very much into that. But I think also that we should agree on the basic concepts. It's very important to understand what we are talking about, because there is also a lot of confusion about what is an autonomous system, what are the technical difficulties we are facing. And in my work – and this is work I'm doing alone because I don't need… it's more conceptual work – I'm trying to characterize autonomy as a combination of functions, as I said, and these functions, I can define them mathematically and they are independent from any technology, I mean any solution. Perception can be model-based or can be machine learning–based. Of course it's much more efficient if it's machine learning–based. But I believe that all the decision-making process could be model-based, should not be based on machine learning as some industrial players want it to be.

**Calude:** So a clear example would be a driverless car?

**Sifakis:** Yes, exactly. This is really the important application. If we are successful in that, then I think many, many problems will be solved, yes.

**Calude:** And my take as a non-expert in this area was that the major problems are not technical, are more philosophical in terms of morality and who decides what. But I have the impression…

**Sifakis:** Okay. There are two types of problems. There are technical problems – what are the technical solutions we bring? – then of course how we make the systems decide, and this may involve some ethical issues. But first of all, the technical problem is how to build self-driving cars that are trustworthy enough. And trustworthiness means that they behave as they should, and the technical term of trustworthiness is… okay, I mean you can give an informal definition, but then it breaks down to technical requirements, because trustworthiness means that the software of course is functionally correct but all the devices, I mean all of the peripherals you use in the devices are also reliable enough.

So achieving trustworthiness is a very, very complex, hard problem. Already building trustworthy flight controllers, this is a problem I have extensively studied, it's a hard problem. But building trustworthy autopilots for automotive is an extremely hard problem because of the complexity of the environment, because of also the dynamics of a car… I mean the real-time problems you encounter for a car are much harder than the real-time problems for aircraft. And of course because of this collaboration you may need between humans and computers, and the systems, because you see we'll not go to fully automated cars immediately.

**Calude:** Yeah. There'll be a transition.

**Sifakis:** There will be a transition. This transition is anticipated by some standards or some regulations that distinguish different levels of autonomy. You probably know that there are levels of autonomy from zero to five. Zero means no autonomy at all, five means full autonomy. Level four means that the car is – these are levels of autonomy for the cars – the car is autonomous if you have specific equipment like in highways or I mean in geo-fenced areas. Then three means that the control is taken by the system, and when there is a problem, then the driver should take over. Which is, okay, I mean a situation…

**Calude:** A little bit fluffy...

**Sifakis:** Yes. I mean I think that here, this level should be skipped somehow, because we had a lot of accidents. Why? Because the system is driving and then you are relaxed and you are reading your newspaper…

**Calude:** And suddenly…

**Sifakis:** …say, "You should take over," and perhaps you have no time to take over.

By the way, this problem of collaboration between machines and humans is a very, very important problem. This is called… In NASA, they call "symbiotic computing." How… And this is not a problem of interface. It's not a human–machine interface problem. It is, in order to make machines safely cooperate with computers, you should have a very strict protocol and a way to exchange information. Because humans have information that is very synthetic, very abstract, and the computer's very detailed information. So if you have a diagnosis by a computer that is I don't know how many lines of symbols, it's not very easy to analyze by humans. This has been a problem. This is a big challenge, how to have what you call symbiotic computing, make machines cooperate with computers.

For this reason, some people believe that we will skip some levels and go to fully automated, *[1:50:00]* because fully automated perhaps is simpler. It would be simpler perhaps to have fully automated, fully autonomous cars everywhere than to have cars…

**Calude:** Combination of human and…

**Sifakis:** Combination, yes, yes. This is…

**Calude:** But is this not a bit a fantasy to believe that from one day to the other, all the old cars, the normally driven cars will be wiped out of the market and they are replaced with autonomous cars?

**Sifakis:** Yes, okay. That's a good question. But you know some people like Elon Musk say that there will be a day where driving your own car would be a crime. So perhaps someday some government decides that "In two years from now, all the cars will be autonomous." And this would solve… I mean would make the problem much simpler of course, instead of having mixed human-driven and autonomous… I mean robocars. That's a big debate.

Also the question is, we have discussed already, whether we should… we can trust all these systems, and this raises also the problem of certification of these cars. And I should mention this perhaps – that today in the United States, you don't apply strict certification rules to cars that have self-driving features, for instance to a Tesla car. A Tesla car is self-certified. I mean it's like a joke, being self-certified, because the manufacturer guarantees that the product is okay. So there is no…

**Calude:** no independent…

**Sifakis:** And it's not possible to apply the existing standards for safety-critical systems because the existing standards I mean require some model-based approach and we cannot apply this to machine learning systems.

**Calude:** Could we touch a little bit the relation between autonomous systems and consciousness?

**Sifakis:** [laughs] Okay. So, how much autonomous systems can approach human consciousness?

**Calude:** Yes. And you see, for instance, if we think about autonomous military devices which are capable and are designed to kill, consciousness which could be…

**Sifakis:** Okay. These are the ethical issues. But first of all, okay, in fact now with autonomous systems, we are approaching, we are developing a model that is closer to the model that our mind uses to make decisions, to decide. In fact, our mind combines two systems of thinking. This is a very well-known fact. The first time I've seen the discussion about that was in Kahneman's book, if you know Kahneman's book?

**Calude:** Yes, yes.

**Sifakis:** Okay. *Thinking, Fast and Slow*. Our mind combines two systems of thinking, one that is slow and procedural, and the other that is fast and non-conscious or automated.

**Calude:** Like driving the car.

**Sifakis:** Yes. I think there is an interesting analogy between this and machine learning and traditional computing, because traditional computing is based on logic and mathematics. So when you program, you are procedural – you understand what you are doing, you are model-based. It's like conscious thinking. And neural computing is closer to autonomous thinking, to fast thinking. I think this now we are approaching with this, I believe that neural computing, the underlying model is different from conventional computing.

Okay, perhaps I should not explain this here, but we have two models of computing and that approach the two systems of thinking. So this is a step toward understanding consciousness. I don't know how close we can get to that, but there are some missing links. [laughs] There are some missing links for the moment, okay? And if we don't understand how consciousness appeared, I mean through the evolution of humans, I don't think we will be able to understand, to build computers that approach human consciousness, because the key issue here is the appearance of language and this kind of creativity we have that cannot be approached either by neural systems or by conventional computing.

The key issue here is that… And in that we failed. So the key issue is how the neural systems, so the brain and the circuits, are connected to what some people call the mind, the concepts and all this. In our mind, we have a kind of semantic model of the world and the relationships. These are high-level concepts. Of course, they can have some implementation that are low-level at neural. For instance, if I'm thinking about what is a table, it might have some representation in terms of signals in the neural system. But there is… It's like… This is something I explain to my students also in that manner, that if I give you my laptop and you have all the instruments to monitor the signals, can you find out which function is computing my laptop?

**Calude:** Oh, this.

**Sifakis:** That's hopeless, okay?

**Calude:** It's undecidable.

**Sifakis:** So it's not by studying only the activity of the brain that we will… We need to bridge the gap between the brain and the world of concepts and of the language. This means that we can somehow formalize natural language, and this is a challenge in which we have failed so far.

**Calude:** Would it be wrong to say artificial intelligence in a way is an aspiration to model with computers the human intelligence?

**Sifakis:** No.

**Calude:** No?

**Sifakis:** No. I mean it's to imitate. I mean, okay, it's shown if you are failing Turing's test, okay…

**Calude:** Yeah. But what about, which I am much more fond of, that artificial intelligence is, how about human intelligence should not behave like a computer?

**Sifakis:** [laughs] Can you repeat this just to complete? How human intelligence should not be-…?

**Calude:** imitate a computer.

**Sifakis:** Yes, okay. I think that for the moment, I mean there are big differences between humans and computers. I mean it's ridiculous. And I think there are arguments, the Chinese room argument that shows that computers for the moment do not understand anything, just handle symbols, and understanding is something different.

**Calude:** Would you see a frontier to automation or to even more for autonomous systems?

**Sifakis:** Yes, okay. Yes. This is the idea of automation frontier. You have tasks that have been performed by humans, and now slowly they are taken by computers. So I have talked about automated systems – these are simple tasks – and autonomous systems. I think that for autonomous systems – autonomous systems are called to replace humans in missions – I think we should move progressively. There are tasks that for the moment you will not implement by using computers. *[2:00:00]* I mean you cannot replace a teacher by a computer. You cannot replace a doctor completely by a computer, or even a driver by a computer. But the idea is that progressively, some of the functions performed by humans will be performed by computers.

**Calude:** Or robots.

**Sifakis:** Or robots also, something like that. Now the interesting thing is what can be… So you have a role or a mission of a teacher – how much will be performed by the human and how much by the computer? Or you have the role of a medical doctor – how much will be performed? I think we should try this problem of division of work between humans and computers and how to make them collaborate to perform this. But it's clear that we cannot leave the computer these critical tasks. It would be too risky.

**Calude:** Do you think that famous people like you should play a significant social role?

**Sifakis:** This is the issue of our responsibility. I think that people who know, who have knowledge, have the responsibility also to inform other people who are less informed about say the stakes, the risks especially for the introduction of new technologies or new knowledge. And yes, I think we have this moral duty. Now the question is to what extent we can play this role? We can play this role by writing papers. But the problem is how to reach broad public?

And here I would like to say a few words about the role of media. Because the media have some preferred stars or media stars.

**Calude:** For the good or for the wrong reasons.

**Sifakis:** Yeah, for good or wrong reasons, as you say. So it's very hard to discuss seriously these issues for various reasons – because people do not also consider seriously like more sensational information about… I don't know, if you talk to people about Martians arriving tomorrow or the end of the world according to the calendar of Incas. I don't know. They are thrilled, they are interested in that. So you have some gurus that talk about technology, the future of technology, singularity. Singularity. You hear probably about singularity.

**Calude:** [laughs] Oh yes.

**Sifakis:** That's a completely crazy idea, that at some point computers will become more intelligent than humans just because a number of transistors… I mean based on completely…

**Calude:** Brute force.

**Sifakis:** …silly arguments. But these ideas find some echo in the media. Consider another case. Elon Musk, when he says something, he's taken by the media and he's echoed, and he says nonsense, but it's taken by the media and…

**Calude:** spread.

**Sifakis:** …spread. And the media also strangely forgets that he said nonsense two years ago.

So that's the game of the media. I don't understand. But I think we, informed people, we should try to explain the risks we are taking by introducing these new technologies and raise the issues that should be addressed by society, yes. By an *informed* society, yes.

**Calude:** With this high pace of automation, many, many jobs disappear, and of course others appear. I have two grandchildren, four and six. What career would you recommend or what should they study to not be lost in this? [laughs]

**Sifakis:** [laughs] I think a famous physicist was saying, "It's very hard to make predictions, especially about the future."

**Calude:** Yeah, I can't make it even for the past.

**Sifakis:** [laughs] I don't know. I mean I think that we should not direct children to make this or this choice. You should cultivate curiosity of children, and children should be taught to do what they like and to manage their freedom. That's the most important thing. That if you do what you like and you try to do your best, I mean to improve yourself every day, that's the most important thing. I mean accomplishment of your personality. To have fun in life. I mean to have pleasure in life. To find, to have a job. So there are many different jobs that are enjoyable. As the job they are doing, these gentleman here, I think that's very funny, that's very interesting. So…

**Calude:** And rewarding.

**Sifakis:** Yes. And rewarding, okay. So you find a job you like, you would like to do, because if you have a boring job, even if you have money, okay, what's the interest? Just earning money? And this is the rule I have adopted in my life. I try to do in the best possible manner what I like to do, and then all the rest will come. So I do not suggest a child to become say [laughs] a computer scientist. In the future, perhaps no computer scientists, I mean no computer science as we know it exists. So just cultivate the talents and try to have fun in life, which is unique.

**Calude:** And now to apply your advice, to give you free will to ask yourself the question you would have asked, have liked me to ask and I didn't. Is there any question?

**Sifakis:** No. I think we are at the end of the interview. [laughs] And I would like to thank you for coming from so far, and also thank you for organizing in such a nice manner, orchestrating the interview. And, okay, thank you for your friendship. I really appreciate what you did for me. Thank you.

**Calude:** Thank you. Thank you so much. It was an honor and a huge pleasure to be here to you and do this interview. Thank you very much.

**Sifakis:** Okay, thank you. Let's shake hands.

**Calude:** Thank you very much.

**Sifakis:** Okay. Thank you.

**[end of recording]**