Logical processes help computers crack complex mathematical problems, thus making them smarter and faster. Logic can even take human-machine communications to a whole new level. Christoph Weidenbach from the Max Planck Institute for Informatics has been developing such promising logical algorithms for thirty years, and is even testing them on his own race car.

TEXT: TIM SCHRÖDER

Christoph Weidenbach is a fan of fast cars. He owns an Opel Speedster, a roadster that weighs in at just 900 kilograms. He opens the garage door to reveal the immaculately clean room that houses his Speedster, gleaming except for a few splattered insects on the front grill. The vehicle is painted yellow and black. "Like Maya the Bee," he says with a smile. Weidenbach lives south of Mainz, in a small village that is known for its wines. Whenever he really wants to put his Speedster through its paces, he wends his way through the vineyards to the nearest highway, then heads up past Koblenz to the Nuerburgring in the Eifel region about an hour and a half away.

The so-called Nordschleife (North Loop) of the Nuerburgring, which snakes its way through 20 kilometers of hilly and forested terrain, is considered the most challenging racetrack in the world. All automobile manufacturers test their sports cars there and private citizens are also allowed to blast their own cars around the track. "The Nordschleife is like a roller coaster," says Weidenbach. "The only difference is that you have to steer it yourself. This year, I'll be taking my four-year-old daughter there for the first time." A few years ago, Weidenbach and his brother fine-tuned the Speedster, disassembling and rebuilding the entire car, including the engine. He added a supercharger that forces compressed air into the combustion chamber. The car used to be too slow for him, but it now produces nearly 300 hp – twice as much as the production model. His father owned an Opel dealership, so it is no wonder that he could already drive a car by the age of twelve.

Although he never became a car mechanic, his research over the years has always had some connection with cars. Weidenbach works at the Max Planck Institute for Informatics in Saarbruecken, where he heads up the independent "Automation of Logic" research group. Their key objective is to formulate logical descriptions of complex situations, so that they can be automated to enable computers to solve the associated problems independently. This is somewhat reminiscent of chess-playing computers, although Weidenbach's approach is much more complicated.

Generally speaking, the problems for which he and his team develop computer algorithms are generally "provably unsolvable," in that they are arbitrarily difficult and beyond the capacities of any computer. This is precisely why it is important to test the novel algorithms on real applications that are known to be solvable in principle. Weidenbach used pure logic to model his Speedster's engine control unit, so his research results accompany him every time he takes the car out for a spin. The engine control unit regulates the amount of air entering the engine, as well as the amount of gasoline that is injected. Conventional control units simply do what they are programmed to do. But thanks to logic, Weidenbach's version can monitor its own actions and evaluate in a fraction of a second whether they make good sense or not. Moreover,

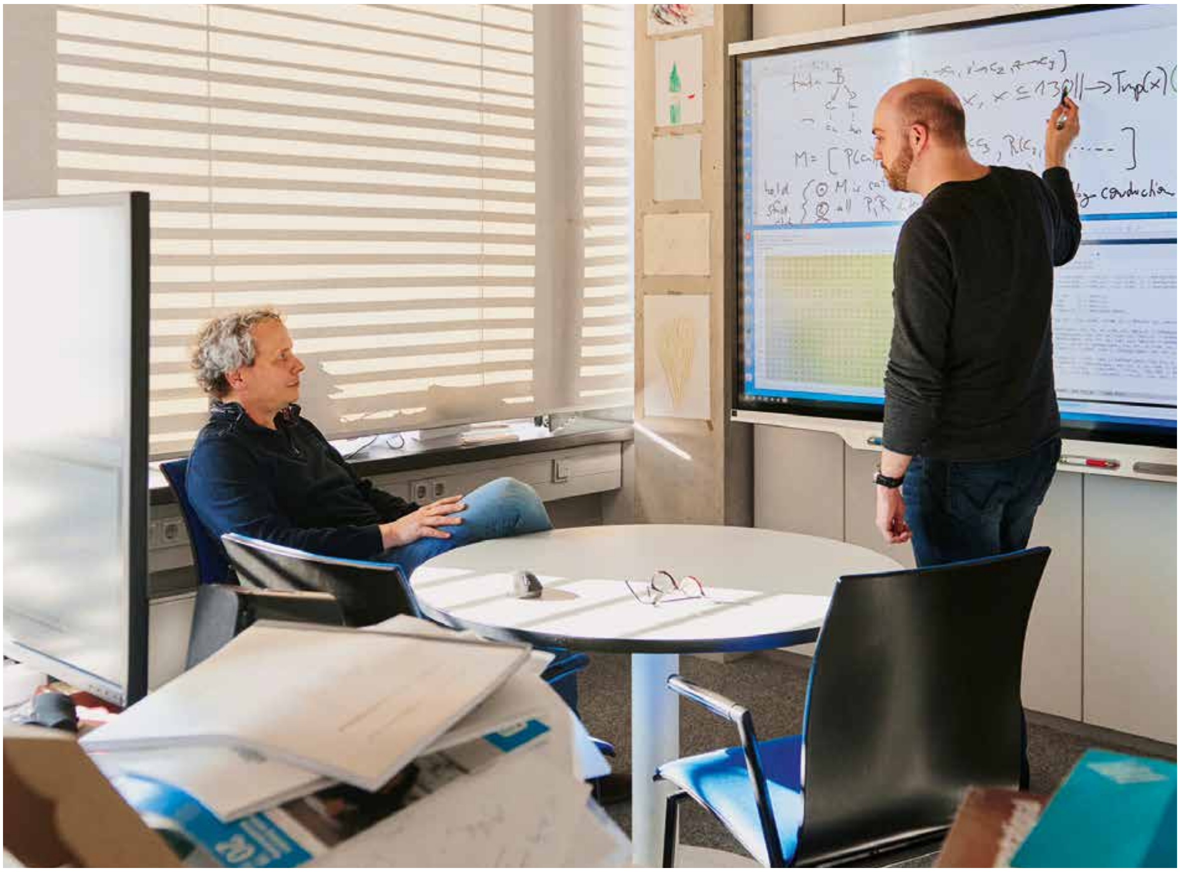
→

VISIT TO

CHRISTOPH WEIDENBACH

PHOTO:: ANNA ZIEGLER

Focused in the cockpit: Christoph Weidenbach and his team are developing algorithms for programs in logic that control complex technical systems in safer, more efficient, and self-explanatory ways – like the engine of an Opel Speedster.



PHOTOS: ANNA ZIEGLER

Think tank: Weidenbach and his colleague Martin Bromberger are modeling an engine control unit to come up with ways to expand their methods. Weidenbach uses a laptop to read out the data from this device, which he installed in his car.

his engine control unit can even explain its actions if necessary. As we know, computers are basically dumb: all they do is switch their transistors on and off to mindlessly perform a series of calculations, enabling them to add or subtract, for example. Drawing logical conclusions presents more of a challenge, as can be seen even in a simple syllogism such as: "All human beings are mortal, Socrates is a human being, therefore Socrates is mortal". All software programs and anything that a computer can calculate are based on these types of logical, If-Then relationships, i.e., rules.

The airbag is a good example: its control system continuously monitors whether it needs to be deployed or not, by analyzing such parameters as deceleration and it has to decide in an instant whether an accident has actually occurred. If it were to deploy and suddenly hit the driver in the face at the wrong moment, it could easily cause an accident. That is why systematic testing is carried out during the development of airbag control programs, to ensure that the software will always function correctly when installed in a car. Weidenbach's approach can be used to prove the correctness of the control program. The challenge for programs in logic simply consists of having to process enormous amounts of data in fractions of a second and their ability to do so is still extremely limited. But Weidenbach is already well ahead of the game. The algorithms he uses in his Speedster are still very new and are not yet being used in the industrial sector.

After completing his habilitation degree in the early 2000s, Weidenbach spent some time working for the General Motors Company (GMC). But back then, his job had little to do with logic. One of his managerial tasks was to assist with the introduction of the SAP resource planning system in Europe. "However," he says, "at that time, I had negotiated a contract that allowed me to continue my work at the Max Planck Institute for fourteen days a year. That's actually a pretty unique arrangement." That kept Weidenbach connected to the world of research over the years, and in 2005 he finally returned to work at the Institute full time, because he had been missing the clearcut, rational methodology. As a result, he is very familiar with both sides – the logical discipline of computer science and the needs of industry.

Another area in which the algorithms developed by Weidenbach are applicable is in the product kits used by automobile manufacturers. These kits consist of complex lists of all the components installed in the various car models, encompassing everything from the screws to the windshield. The ideal situation for a manufacturer is to be able to reuse existing components in new models, but it is not always easy to determine whether this is possible or not. For example, it is difficult to know whether a new model can be built to have certain, specific characteristics – for example, a top speed of 200 kmh, while using only 4 liters of fuel per 100 km (3.785 liters = 1 gallon). In this case, giving the computer a plan like, "Engine Z will be used in the new Model X," is not enough. The computer needs rules, such as "Engine Z will fit in any car that has an engine compartment volume of 1.10 meters wide and 0.6 meters deep."

"The great thing about the logical approach is that the devices in question can also explain why they do something."

The leading automobile manufacturers are already using these modeling techniques, but the problems still exist, due to the enormous number of variants. A specific car could be a coupe or a convertible and could be fitted with any one of ten different engines, five different transmission systems, and various wheel types. Some vehicles come with a trailer hitch, others with a high-end hi-fi system. The number of potential variations is in the billions, so simply listing all the possible variants is usually not an option. Modern logic algorithms solve the problem by deriving new insights from

→

the known rules. One such insight might be that no engine can achieve 100 km to the gallon in combination with the weight of the heavy-duty hi-fi system, which is a useful thing to know.

Another problem is that the number of these insights is still excessively large, because of the vast number of possible combinations. This means that the system could come up with quite a few useless statements, such as, “The trailer hitch cannot be combined with a bike rack,” which has precious little to do with the initial problem. So, current programs that employ this operating principle of learning by generating new insights are rendered ineffective by the sheer number of potential combinations and insights.

“The current solution,” Weidenbach explains, “is to delete unnecessary insights.” For example, if no connection can be made between the insight “the trailer hitch cannot be combined with a bike rack” and the original requirement “100 km to the gallon” via a few logical steps, then it is deleted. But as Weidenbach explains, “Even this approach will eliminate only about a third of the insights.” So, his goal is to use a combination of mathematics and logic to produce a set of rules that can be applied in practice and come up with a solution within a reasonable span of time – for modeling a new car while taking all available kit components and technical correlations into account, but also for solving a whole host of other problems.

Christoph Weidenbach and his team have developed the “Superlog” system, a software framework that uses so-called model assumptions to compute the required solutions. The program derives certain assumed facts from the given rules and calculates new information only for those areas where the existing logical model doesn’t yet work, which is akin to assuming at the outset that Socrates is a human being. This approach significantly reduces the computational effort required, so that huge amounts of data can be transferred efficiently into the basic rules.

One practical application of this technology already exists. The software for the engine control unit in Weidenbach’s Speedster records a series of engine parameters in just a few milliseconds and uses them to calculate what to tell the engine to do in real time. The speed of this logical analysis is unparalleled. “The great thing about the logical approach,” Weidenbach explains, “is that the devices in question can also explain why they have made a particular decision.” This differs from conventional control systems, which simply issue commands based on specific parameters.

Systems like this could make human-computer communications much easier and more secure. One example that Weidenbach likes to quote is the plane crash on June 1st, 2009, when an Air France plane suddenly fell out of the sky on a flight from Brazil to Paris. The crash occurred because the impact pressure sensors were iced over and sent conflicting information to the autopilot, which caused it to shut down halfway across the Atlantic. The conversation captured by the flight recorder shows that the pilots were confused by the behavior of the flight control system, which caused the on-duty pilot to pull the nose of the airplane up at too steep an angle without the other two pilots noticing. Eventually, the plane stalled due to the lack of speed and crashed into the sea. Had the plane’s control unit been modeled on the rules of logic, it could have simply explained that the joystick of the pilot pulling the plane’s nose upward had priority over that of the co-pilot who was trying to gain speed by forcing the plane downwards.

“I thought his approach was wrong and Ganzinger thought my doctoral topic was nonsense. He told me that to my face.”

The fact that Weidenbach ended up modeling non-trivial technical issues is basically due to a coincidence. While he was working on his doctorate in 1991, his supervisor relocated to Saarbruecken to join Harald Ganzinger, then Director of the Institute and head of the Logic of Programming working group. Ganzinger was interested in the concept of limiting and simplifying computational logic using so-called orders. “I thought his approach was wrong and

Ganzinger thought my doctoral topic was nonsense," says Weidenbach. "He told me that to my face. But we finally examined each other's theories in more detail and published some really good papers together."

During his teens, Weidenbach was an excellent saxophonist. He was only fifteen when he played in the state youth big band of Rhineland-Palatinate, which was a kind of talent incubator. "At that time, I was thinking of a career in jazz," Weidenbach says. On weekends, he and one of his musician friends used to earn money – to save up for a motorbike – by playing gigs in nearby villages and on club dance nights. "But, at some point I came to realize that you can't earn a long term living from that – and certainly not as a jazz musician." So, he decided to study computer science. The then Central Office for the Allocation of Places in Higher Education sent him to the Technical University of Kaiserslautern. He isn't exactly sure why he chose this particular field, but thinks the multiplication tables his father taught him might have had something to do with it. His cousin, who had studied mathematics in Bonn and had inspired his young mind with the logic of mathematical proofs, also had a major influence on his decision. To some extent, Christoph Weidenbach's career has been shaped by coincidence.

PHOTO: ANNA ZIEGLER

Innovative approach: computer science, as practiced by car buff Christoph Weidenbach, is by no means limited to theoretical studies; he also puts the results of his research into practice and tests them out in his 300 hp roadster, which he occasionally takes out for a spin on the Nuerburgring.

He currently helps young people to choose a career path and to achieve their goals more easily. He also heads the advisory board of the German National Computer Science Competition (Bundeswettbewerb Informatik – BwInf), which promotes young talent. Following two rounds of competition each year, thirty candidates are shortlisted from over a thousand high school students, who are then interviewed separately and given group tasks to solve. The best of them receive a scholarship from the German National Academic Foundation. In Weidenbach's view, it is particularly important to mentor young people: "We know where they teach computer sciences well and have interesting content and we ensure that the upcoming generation will be in the best hands – and it sometimes helps if we can give them a good letter of recommendation."

Weidenbach wants to ensure that their talent doesn't go to waste. "Kids simply get lost in the shuffle at huge universities. A work-study program doesn't do justice to the really good students. We ensure that our candidates don't lose their way when it comes to their studies," says Weidenbach, who adds that he was lucky to have landed with Ganzinger, who threw the door open in terms of his professional development. "Now I'm the one trying to point young people in the right direction."