The digital **language artist**

He describes himself as a bit unconventional. **Derek Dreyer** does, indeed, work with languages. Not, however, with everyday languages, but with programming languages, since they're more logical. The U.S.-born researcher works at the **Max Planck Institute for Software Systems** in Saarbruecken. Even though science is an important part of his life, he still makes room for passions like music. And then there's the thing about whisky – unconventional indeed.

TEXT **KLAUS JACOB**

Some offices are as impersonal as furniture store showrooms. Others can be read like a book. They reveal a lot about the people who work in them. That's the kind of office Derek Dreyer has. One of the end walls is dominated by a white board covered in cryptic equations that are only decipherable by insiders. Next to this are two colorful drawings by Derek's daughter. She's four years old. "No, four and a half," Dreyer corrects himself. With small children, every month counts.

The other end wall is completely covered by a shelf. Instead of books, the shelf contains row after row of whiskies, although not bottles, just their cardboard boxes. Dreyer is a big whisky enthusiast and knows almost every kind, but focuses on Scotch, not Bourbon.

The office feels open and comfortable, bright and spacious. Right in the middle there's a comfortable sofa, perfect for lounging on. The two long sides of the office are entirely paned with glass. One of these window facades provides a view of the wooded hillsides of Saarbruecken; the other, a complete view of the corridor. Dreyer, therefore, has an office all to himself, but the transparent walls (common to all the offices in this modern building), make the individual rooms feel almost like an open-plan office.

LOGIC AND MATH – DREYER'S STRONG SUIT

And that suits Dreyer. He's as open and friendly as the building, is willing to answer all my questions, and likes to laugh and engage with people. On his website, he writes that he can now be referred to as *"Herr Professor Doktor Derek Dreyer"* as he has been appointed an honorary professor at Saarland University. But he prefers just "DD", which is what his daughter Alma calls him.

Derek Dreyer is a computer scientist at the Max Planck Institute for Software Systems, where he researches programming languages. He is a member of the Institute management team and head of the independent group "Foundations of Programming", which comprises two postdocs and six doctoral researchers.

Logic and mathematics are Dreyer's strong suit. Spoken languages, on the other hand, are not. An American by birth, he's lived in Germany for eleven years, but you can't avoid speaking to him in English. This is primarily due to the fact that the working language at the Institute is English; most of the

An abstemious collection: Derek Dreyer collects unique whiskies, but in his office he only displays the empty boxes.

Photo: Wolfram Scheible

Please press the exit-button to switch OFF the system after use. Otherwise the monitor will turn on automatically the next time.
Arc Veri
0
1
2
{Arc(x)} y = x.clone
c(x)} x.print(); {
} x.drop(); {T

It's logical: Dreyer proves that some components of programming languages like Rust don't require stringent safety mechanisms.

researchers come from abroad. And his wife is also a U.S. citizen, although her German is much better. "*Es ist sehr peinlich*," [it's very embarrassing] says Dreyer in German with a thick accent, smiling. If there's a job that requires German to be spoken at home, his wife steps in.

Learning German is just too difficult, according to the researcher. There's a little playfulness in his modesty; he has a far better command of the language than he admits. At a minimum, he acknowledges that he knows the correct vocabulary for all food ingredients; good food means a lot to him, whether Japanese, Thai, French or Italian.

On business trips, he likes scouting for a good restaurant and invites a few friends and acquaintances to join him. "Give me any kind of food, and I'll tell you the German word for it," he says. He does, however, understand spoken German better than he speaks it. If he says something to his daughter in German, she laughs at him. For her, learning the new language in kindergarten is ... child's play.

Math on the other hand is a completely different matter. Dreyer mostly learned it effortlessly. He's an overachiever. With the help of his parents, he skipped several school grades because he was getting bored. He started university at the age of 13 instead of the usual 18. At 17 he moved to Carnegie Mellon University, one of the top universities for computer science in the U.S. Here too, he was five years ahead of his fellow students.

DREYER WASN'T PREDESTINED TO BE A COMPUTER SCIENTIST

His educational progress did hit a snag during his PhD studies, however. "I struggled at first," he says. Indeed, he almost had to drop out of graduate school in 2000, struggling to find his footing. He persevered, however, and was soon publishing his first scientific papers, finally earning his doctorate in 2005.

Derek Dreyer was born in 1980 in New York City, but soon moved with his parents into the suburbs of the metropolis, to Great Neck on Long Island. It's the area of New York made world-famous by the author F. Scott Fitzgerald with his novel *The Great Gatsby*, as Dreyer is fond of relating. Of his three siblings, he was the baby of the family, much younger than his two brothers and one sister.

His father is a pediatrician at the NYU Langone Medical Center, and has made a career for himself in numerous management positions in pediatrics. He became the President of the American Academy of Pediatrics, the largest pediatric association in the U.S. "He's very successful," Dreyer says about him, "and a great inspiration to me." His

Photo: Wolfram Scheible

» His expertise is in basic research. His proofs are universal and are helping to shape the programming languages of the future.

mother stayed at home and took care of the children and the household. Dreyer wasn't, therefore, predestined to be a computer scientist.

He did, however, love math at school and was intent on going on to study the subject. But his pragmatic parents encouraged him to enroll in computer science as well, as it would make finding a job easier. So, initially, Dreyer studied both mathematics and computer science, but then moved over fully to computer science. He's now glad he followed his parents' advice. In mathematics, he would miss real-world applicability. He likes the combination of theory and practice that computer science offers him.

Derek Dreyer hasn't just excelled in disciplines involving pure logic. He has also developed an active interest in the arts. Above all, he enjoys singing. As a pre-teen, he was singing in the chorus of the New York City Opera, even taking on some solo roles. Having developed into a tenor, he sang in the university church choir while he was living in Chicago. And music still has a strong hold on him, especially classical music and jazz. "My favorite composers are Bach, Britten and Shostakovich," he says. "I feel proud that this year I made the effort to explore Benjamin Britten's third cello suite. It's not easy music. I had to hear it about 20 times, but now I find it amazingly beautiful."

Dreyer never learned to play an instrument, however. "It's like the German language," he says with a smile, "you have to practice too many boring pieces before you can master an instrument." Maybe it's his perfectionism that makes him shy away from tackling things that he can't accomplish right away. He hasn't done much singing in recent years, but he likes to tap dance, even if not in public performances. To demonstrate, he jumps up and performs a few steps. It looks pretty professional ...

A GREAT OPPORTUNITY IN GERMANY

After studying and completing his doctorate at Carnegie Mellon University, Dreyer initially accepted a three-year independent research position at the Toyota Technological Institute at Chicago. The institution has nothing to do with cars – it's a basic-research institution funded by an endowment from Toyota. "It's like a mini Max Planck Institute," Dreyer says.

After three years, when his contract expired, he applied for positions internationally. The field of programming languages had now become his main focus, and there was little demand for it in the U.S. Although the community working in the field has grown considerably in recent years, it is still relatively small. The offer from Germany came at just the right time.

He didn't have to spend much time thinking about accepting the tenure-track position at the Max Planck Institute for Software Systems; if he achieved his scientific objectives, it offered him long-term prospects. He knew that experience at a Max Planck Institute is highly regarded in the U.S.

Dreyer reaches for his smartphone and shows me a worldwide ranking of all institutes involved in computer science and computer languages. The Max Planck Institute is near the top of the list. "It was a great opportunity," he says, especially because he had the chance to become involved in the initial development of the Institute. In 2008, he took up the post, initially on a short-term basis. Then, five years later, following an international evaluation, he was offered a permanent position with full scientific autonomy.

When asked whether the move from New York and Chicago to Saarbruecken was a culture shock, he provides a diplomatic answer: Max Planck is "great", and he works a lot, so the neighborhood isn't a major factor. "What's important for me is that I have fantastic students and colleagues, and Saarbruecken is a quiet place – a good place to work." For variety, he travels, and his job also takes him to many different places, even to major capitals such as Paris and London. So will Germany remain his second home? He has no plans to leave Germany in the near future, he says.

After that, he has no idea. He has considered other possibilities, but nothing comes close to his position at the Max Planck Institute. He can research whatever he wants here, and has a great team and excellent resources. He also finds the combination of research and teaching ideal. “Besides,” he says, “my wife Rose Hoberman has a great job at the Institute, teaching our doctoral researchers to write readable papers and give compelling lectures. And her office is two doors away from mine – beat that!”

One disadvantage of living in Saarbruecken is a universal fact of life for academics: the friendships he and his wife make are usually temporary. Most of their friends are students or postdocs from the Institute who leave after a few years. A career in science takes people all over the world.

Dreyer’s work is as incomprehensible to the layperson as the formulae on his blackboard. It’s ironic that his field is languages, as he has claimed not to be a keen linguist. But programming languages are a whole different ball game to German or Japanese; they are structured in an absolutely logical way.

RUST IS DESIGNED TO BE UNIVERSAL AND SAFE

Dreyer’s current focus is principally on the relatively young language known as Rust, which can be used to write complex programs. In 2016, he was awarded a European Research Council (ERC) Consolidator Grant of two million euros for his five-year project called RustBelt.

The programming language Rust was designed by Graydon Hoare, an employee of the software company Mozilla, best known for the web browser Firefox. Initially, he created the language alone and as a personal project. Mozilla has been supporting the development of Rust since 2009, and version 1.0 was released in 2015, along with the associated compiler, which translates the commands into machine code.

Rust is now widely used by Google, Microsoft, Facebook and Dropbox, among many other companies. It is designed to overcome the weaknesses of conventional languages. Up to now, languages have either been safe (they automatically check the work of the programmer for important classes of errors), or they have been flexible but unsafe (they offer programmers extensive freedom – including the freedom to commit errors).

But an error in a complex program that is only revealed during initial test runs is extremely troublesome. Trying to locate it can drive programmers to despair. Rust is designed to manage the balancing act of being both universally applicable and safe. Understanding how this can be achieved requires a little background knowledge.

Ultimately, all programming languages have a control mechanism. However, these can vary in their degree of rigor. The stricter the controls, the more they restrict the freedom of programmers. They’re like digital bouncers who sometimes interpret something as an error that is definitely useful, calling a foul on the programmer. Java is a language with strict controls. In contrast, C and C++ are, to a certain extent, the opposite.

Programming languages employ control mechanisms that go way beyond a document spell-checker. A particular example illustrates how they work. Most languages use pointers that refer to specific data in memory. Sounds pretty simple. Complex programs, however, require many pointers and many memory locations. It’s easy for a programmer to lose track of them, resulting in a pointer that could be used to call up memory that contains nonsense.

Rust has a number of mechanisms designed to prevent this. Each pointer, for instance, has only a limited lifetime, i.e. it is only active in a specific section of code and is automatically switched off afterwards when the associated data has become obsolete. Also, no two pointers can ever simultaneously perform modifications to the same memory location. Such duplication is a common source of errors, because the data in memory can be modified with every operation, potentially leading to bugs if accessed simultaneously.

SAFETY WITHOUT DIGITAL BOUNCERS

However, such built-in safety features come with a disadvantage: some operations cannot be carried out. For instance, the creation of complex data structures requires memory locations that can be simultaneously accessed by more than one operation. To ensure that Rust remains universally applicable, it thus incorporates the possibility of the “unsafe mechanism”. The programmer can specify that certain program components do not have to pass through the strict security gate. Using this mechanism, of course, runs the risk of errors creeping in – as with C or C++.

And that’s where the work of Dreyer and his team begins. He provides formal proof that typical “unsafe components” are, in fact, safe, so that they can be used with confidence, even without a digital bouncer. To achieve this, he

Music and dance: as well as mathematics, Dreyer is keen on music, especially classical music. He even used to sing at the New York City Opera. His tap dancing is just a private pursuit, however.

utilizes interactive proof assistants; in other words, his proofs are automatically verified by the computer. Ultimately, as a scientist, his work focuses on his favorite combination of mathematics and computer science.

However, it would be a mistake to suggest that Dreyer is only working on the fledgling programming language Rust. His expertise is in basic research. His proofs are universal and are helping to shape the programming languages of the future. His interest in Rust is merely as a practical application, although his involvement in the language primarily concerns the most complicated "unsafe components".

The ERC Consolidator Grant he received for the project is one award that he is particularly proud of. Another is the Robin Milner Young Researcher Award conferred by ACM SIGPLAN, the Association for Computing Machinery's Special Interest Group on Programming Languages; it is the most prestigious international prize for young scientists in the field of programming languages.

He dismisses other awards that he has received for outstanding publications with a wave of his hand. Like knick-knacks, they're stored on the shelf along with the whisky boxes. They're neither given pride of place, nor presented in a showy way. In any event, the line of boxes of exceptional whiskies is much more eye-catching. In addition to enjoying these spirits, exceptional food and, above all, music, he also loves fine wines and has even joined a wine club. "It's my way of getting acquainted with Germans," he says with a smile.

THE PROMISED LAND FOR LOVERS OF SCOTCH

The members of the club are acknowledged wine connoisseurs. About a dozen men and women meet every three to four weeks, all of whom, except Dreyer, are German. They invite each other to their homes and serve wine from their own cellars. The wine tastings are in no way drinking binges; the experts spit samples out after tasting them. "Part of the deal," proffers Dreyer.

As a Scotch connoisseur, he has a sensitive enough palate to distinguish fine notes: "Every vintage is completely different," he says. Nevertheless, he has a handicap: wine connoisseurs employ a broad vocabulary to describe the flavors of the different terroirs and vintages. As an American, he just can't keep up: "I'm always wanting to say something interesting, but what comes out is consistently boring due to my limited vocabulary." It means he mostly just listens. On one occasion, however, he did indeed take the floor. He guided the members through a tasting of four lesser-known whiskies: Clynelish, Ben Nevis, Springbank and Ledaig, which are "unconventional, a bit like me."

For a lover of Scotch like Dreyer, Germany is the Promised Land. In the U.S., Scotch whisky is not only much more expensive, but in many cases not even available. The shelves are full of them here. But occasionally, Dreyer travels to Scotland to buy directly from the distilleries. Does he have a favorite brand? He ponders for a while, then settles on a Black & White Blend from the 1960s. "Outstanding quality and, at 100 euros, very cheap" – a bargain. He bought a bottle yesterday, and he considers it his current favorite. Tomorrow, it may be another. ◂

Photos: Wolfram Scheible