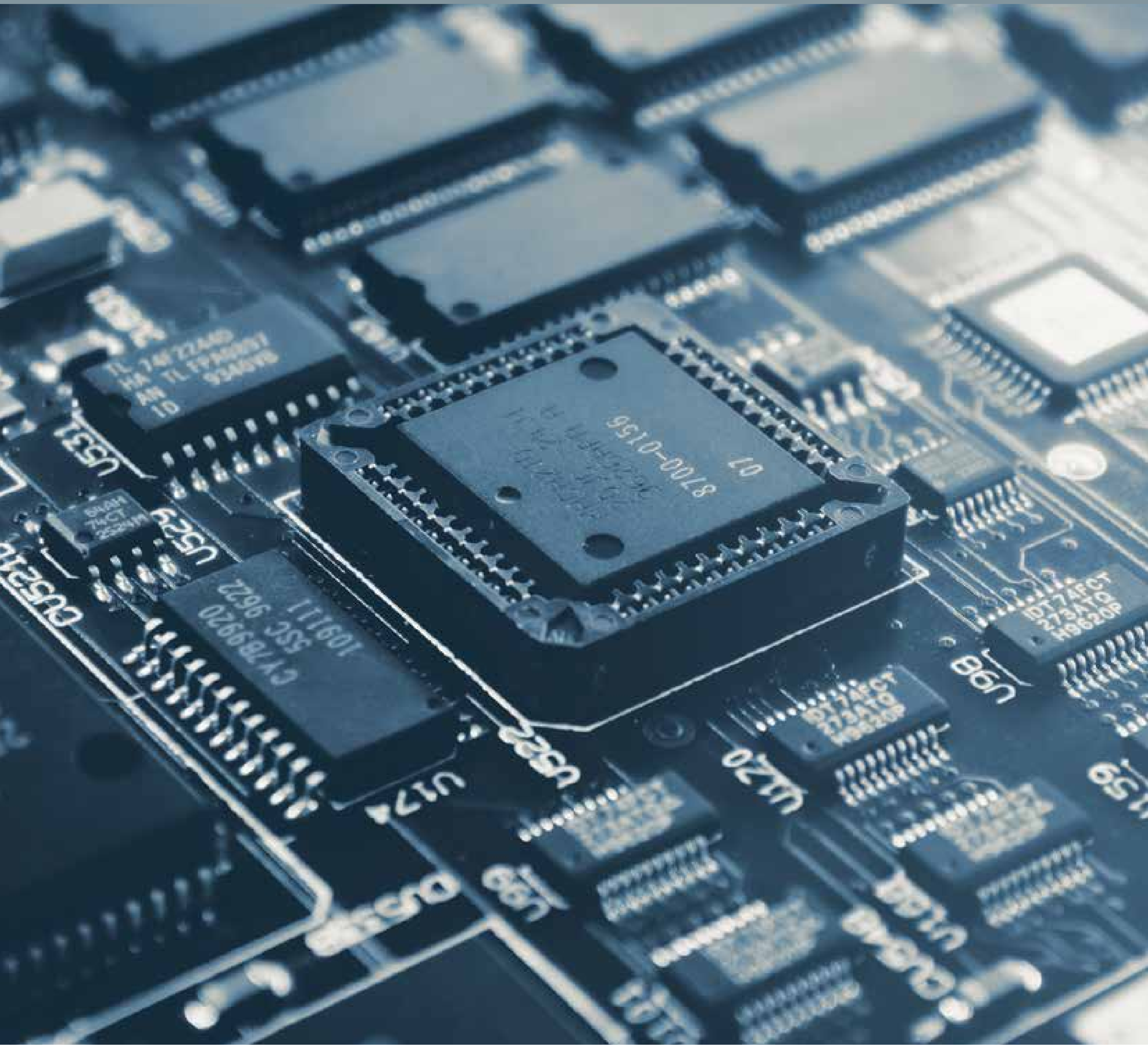
IMPACT

A NEW TYPE OF THEORY WITHIN COMPUTER SCIENCE

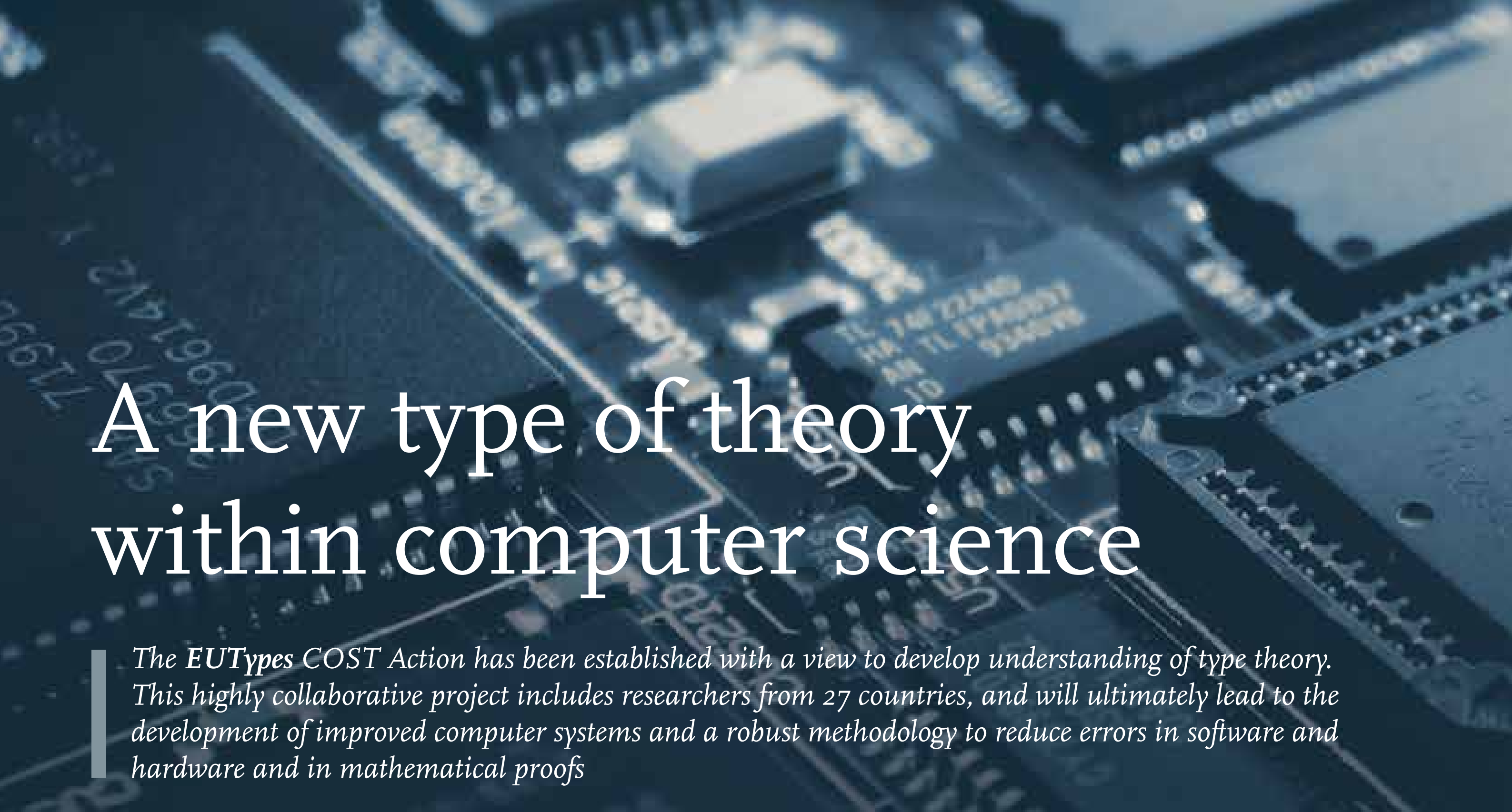


COST
EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY

COST is supported by the EU Framework Programme Horizon 2020

Radboud University

HTTPS://EUTYPES.CS.RU.NL/



A new type of theory within computer science

*The **EUTypes** COST Action has been established with a view to develop understanding of type theory. This highly collaborative project includes researchers from 27 countries, and will ultimately lead to the development of improved computer systems and a robust methodology to reduce errors in software and hardware and in mathematical proofs*

In both computer science and computer programming, a 'type' is a specific classification of data which explains to the interpreter how the programmer intends to use the data. The importance of types is shown by the fact that almost all programming languages include a notion of them. Indeed, a type defines a formal interface between software components and, in doing so, allows their connections to be automatically verified. Thus, both the robustness and reliability of computations and communications are greatly enhanced.

As technology has advanced, type systems have rapidly evolved, becoming increasingly able to capture new aspects of the behaviour of computer programs, like time or memory consumption, and side effects. Alongside this more practical focus comes the foundational study of type theory itself, which is the study of how types interact with one another, and how they can be organised into a consistent system of constructing objects and reasoning about them. A development of type theory will, therefore, enable an improved computer manipulation of mathematics.

THE NEED FOR IMPROVED SYSTEMS

In response to these developments, a COST Action has been established to accentuate research into type theory and its many applications in computer science. Chaired by Professor Herman Geuvers of Radboud University Nijmegen in the Netherlands, the highly collaborative, four-year 'European research network on types for programming and verification' (EUTypes) project involves participation from 27 different countries.

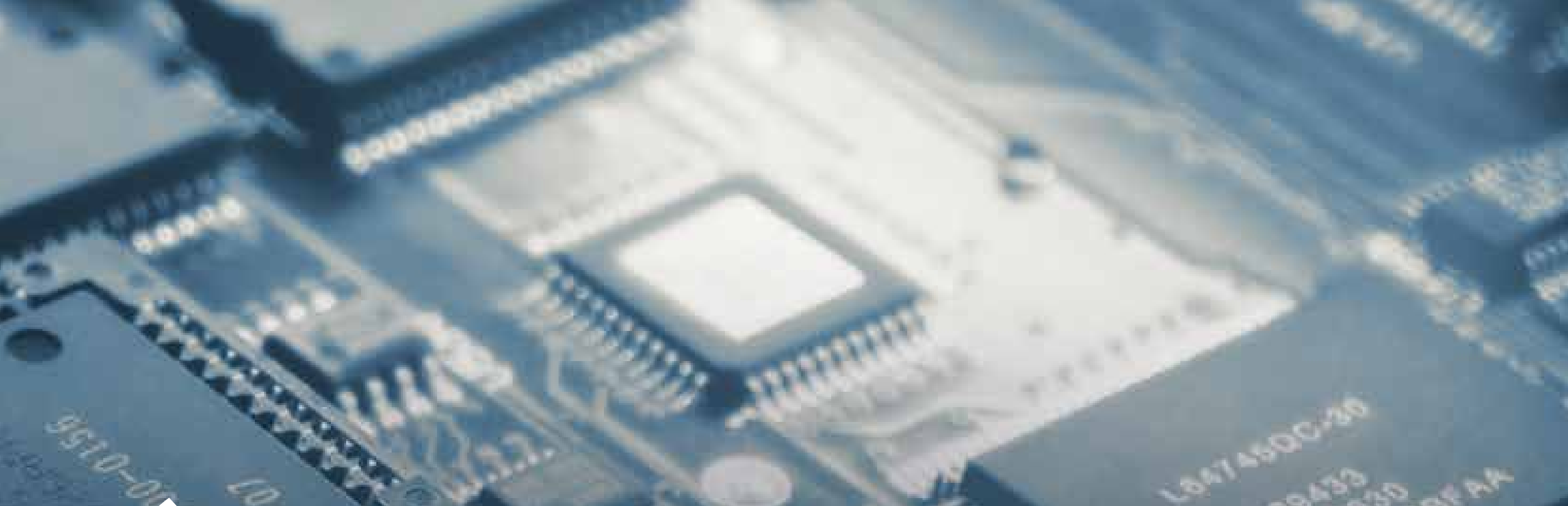
With the project now into its second year, the participants work with the notion of 'type' as the fundamental concept of computational thinking. Geuvers explains: 'Whenever we distinguish cats from dogs, or natural numbers from integers, we deal with types. It is sort of obvious that mistaking elements of one type with another may lead to undesirable consequences.'

For example, if a negative number is allowed to form part of a computer program in which positive numbers were essential, such as in computing the age of a living human being, a problem is likely to result as humans only ever get older. Such incompatibility presents a real problem in computer science and indeed in the real world too, which it often seeks to represent: 'There are many examples of computer malfunctions that were caused by a lack of systematic control over types of acceptable numbers,' Geuvers notes. 'For instance, the spectacular explosion of the maiden flight of Ariane 5 in 1996 was caused by a mismatch between the types of numbers that were used by different components of the control system.' Such errors occur because of insufficient type checks. This is one simple example where types would have helped. Type theory is a field which constantly seeks to invent new systems that can prevent more subtle errors, so the potential applications for the findings of the project are vast.

INTERDISCIPLINARY WORKING GROUPS

In order to achieve the aims of EUTypes, the action has been organised into four Working Groups (WGs). By managing the work in this way, two main purposes are served. Firstly, the main streams of efforts in the field are reflected; and second, the results of each contributor's studies are activated.

The community involved in the project is strong in theoretical investigations, so WG1 focuses on foundations and theory, with many of the activities concentrated on the study of relations between contemporary logic and geometry (expressed in terms of homotopy theory). 'Since Vladimir Voevodsky introduced homotopy type theory, researchers have come to understand that type theory is the ideal language to express notions from the abstract mathematical field of homotopy theory, and that type theory provides the ideal language to reason about homotopies in a more flexible, yet very precise way,' Geuvers says. It is exciting to see how ideas and concepts from homotopy type theory feed back into type theory as a language for programs and proofs. This has already resulted in the important concept of `higher inductive types'. In the coming years we will see better how this crossfertilization can result in improved programming and verification techniques. 'The interaction with geometry will be more fruitful when there are good computer tools that help in the development of formal mathematics,' explains Geuvers. 'This is where WG2 comes in – it is focused on the development of type-based computer tools.' Such tools are an important means of demonstrating the strength of the network's approach to the outside world.



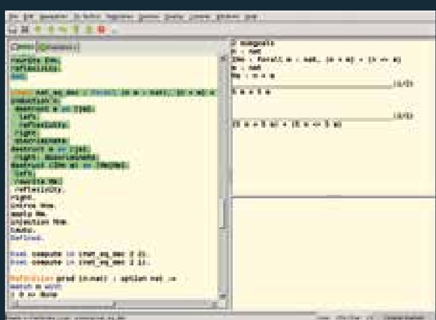
> *For me personally, the biggest result of the past few months has been the development and further understanding of a computation rule that reflects one of the most important geometrical principles to cover in this area, namely, the principle behind the univalence axiom*

Groups WG3 and WG4 are centred on programming languages and software verification. They deal with methods for the development of dependable software systems, albeit from two different directions. WG3: Types for Programming focuses on the development of stronger type systems and type-based programming methodologies that will guarantee the absence of more and more complicated programming errors (thereby directing the programmer in writing the correct programs); WG4: Types for Verification, meanwhile, focuses on making the process of property expression and proof easier and less time-consuming. So, while WG3 and WG4 are methodologically different, both serve to complement one another.

THE PRESENT BODES WELL FOR THE FUTURE

Despite the fact that the project is only a quarter of the way through it has already produced some exciting results. 'For me personally, the biggest result of the past few months has been the development and further understanding of a computation rule that reflects one of the most important geometrical principles to cover in this area,' says Geuvers. 'Namely, the principle behind the univalence axiom.' In lay terms, this principle states that types of the same structure, such as isomorphic types, should be considered equal. While this can be thought of a natural conclusion to draw, it is actually rather difficult to frame, as two types with the same structure can have considerably different descriptions. Perhaps this is best shown through the fact that natural numbers can be thought of as all numbers obtained by adding one several times to zero, but are also defined as all numbers represented in decimal notation.

Throughout the course of the COST Action, the researchers will employ innovative methods in order to conduct their investigations. Importantly, Geuvers and his colleagues try to formulate all of their theoretical works by using a proof assistant. This ensures the results they obtain are far more reliable, and practically eliminates the possibility of human error. In addition, this method enables the team to work on complex theoretical models that are far bigger than any other time in history. 'Formally verifying your mathematical results is an innovation that can slowly be seen spreading throughout the field,' says Geuvers. 'Proof assistant tools are improving and the acceptance of this technology is growing. I believe this is a technological revolution and we will see its benefits in the near future.' If the first year of EUTypes and the results that have been obtained so far are anything to go by, the future might be nearer than we think.



Proof checking in a proof assistant

Project Insights

FUNDING
EU Framework Programme Horizon 2020 COST Action CA15123

PARTICIPANT COUNTRIES
Austria • Belgium • Bosnia and Herzegovina • Czech Republic • Denmark • Estonia • Finland • France • FYR Macedonia • Germany • Greece • Hungary • Ireland • Israel • Italy • Lithuania • Netherlands • Norway • Poland • Portugal • Romania • Serbia • Slovenia • Spain • Sweden • Switzerland • United Kingdom

CONTACT
Herman Geuvers
Action Chair

T: +31 243652603
E: herman@cs.ru.nl
W: https://eutypes.cs.ru.nl/

ACTION CHAIR BIO
Professor Herman Geuvers is a Professor of Computer Science (Theoretical Computer Science) in the Software Science Section of the Institute for Computing and Information Science (ICIS) of Radboud University Nijmegen in the Netherlands, and head of the Foundations group within this Section. Since 2015 he has been the Research Director of ICIS. Geuvers was previously the Director of Education for the ICIS curricula at Radboud University Nijmegen. He is also the Chair of the Types Steering Committee, responsible for organising the annual Types Conference.

COST EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY

COST is supported by the EU Framework Programme Horizon 2020

Radboud University

COST (European Cooperation in Science and Technology) is a pan-European intergovernmental framework. Its mission is to enable break-through scientific and technological developments leading to new concepts and products and thereby contribute to strengthening Europe's research and innovation capacities. www.cost.eu

Impact Objectives

- Promote the synergy between theoretical computer scientists, logicians and mathematicians to develop new foundations for type theory
- Support the joint development of type theoretic tools as proof assistants and integrated programming environments
- Foster the study of dependent types for programming and its deployment in software development
- Support the study of dependent types for verification and its deployment in software analysis and verification

Improving techniques, methods and tools of automatic verification

*Professor **Herman Geuvers** is leading a project that seeks to provide a strong impetus to research on type theory. Here, he discusses the importance of the research involved, the benefits of collaboration and some of the challenges that have been faced in the project*

You are the Chair of the COST Action 'European research network on types for programming and verification' (EUTypes). Can you talk a little about what you hope will be achieved?

The network wants to bring European researchers in type theory together to develop and use expressive type systems as a basis for improved programming techniques and for methods and tools to implement computer artefacts and verify them. Types are pervasive in programming and information technology. A type defines a formal interface between software components, allowing the automatic verification of their connections, and greatly enhancing the robustness and reliability of computations and communications.

Type systems have rapidly evolved over the past years, becoming more sophisticated, capturing new aspects of the behaviour of programs and the dynamics of their execution. This COST action will give a strong impetus to research on type theory and its many applications in computer science.

In what ways does collaboration play an important role in the success of the network?

Because of the sheer size of the problems involved, all of the research questions require a joint effort. In addition, the open network will generate new original insights and fresh ideas. A result in the form of a tool that actually helps in the development of more dependable programs is a complicated artefact, and is also a long-term enterprise to develop it. It is far too complicated to conduct within one research team.

While large companies, such as Microsoft, Google and Facebook can afford to invest in type theoretic research and develop proof assistant tools, it is a far too long-term project for them to actually go through with it. In the types community various proof assistant tools have been developed in the past, and a range of new ideas have been invented, tested and incorporated into their systems. All of this has come about through an open exchange of ideas and implementations. This cooperative attitude is key to the success of this research in the long term.

Who will benefit from this research?

The work in the project will not have an immediate impact on society, but we do expect that it will improve the way computer programs are written. In particular, programs will be more robust and more secure. We have already seen that with past efforts of our community in the CompCert C-compiler and the seL4 operating system – both systems fully verified using tools that emerged from studies on type theory. This means that these systems are free from programming errors. More achievements of this kind will be possible with the help of tools and methods developed by members of the EUTypes community. Also, the verification of mathematical proofs will be enhanced considerably using the proof assistant tools that are developed by researchers in the project. We have already seen some impressive examples of that in the recent past: the formalisation of the proof of the four-colour theorem, and the formally checked proof of the Kepler conjecture.