

Programming approaches for the smart connected world

The world is turning into a global “supercomputer” consisting of sensors, smart devices, laptops, desktops, and (cloud) servers all connected via the Internet. A team led by computer sciences Professor Mira Mezini is developing new programming approaches that automatically deal with the complexity of this “world supercomputer” so that software developers can focus on solving real problems.

— By Boris Hänßler

Our brain does amazing things. This is evident, for instance, on our daily drive through the city. We should really be overwhelmed by all the signals that flow into our sensory organs from every direction – engine noises, sirens, construction machinery, traffic lights, advertisements. And yet we sit behind the wheel, generally unaffected by it all. We can even listen to an audio book or mull over the day as we operate the clutch and change gears. That’s because our brain processes most signals automatically, so we don’t really take much on board. It is only in the event of danger that it raises the alarm, and then we become highly focused.

These capabilities could well be used by software development teams in their solutions. The convergence of Edge, In-Network and Cloud Computing in a kind of “world supercomputer”, the availability of data and information, and advances in artificial intelligence may have tremendous potential for innovation in addressing urgent societal challenges. However, the complexity of the “world supercomputer” continues to grow, while the programming approaches that are familiar in the software development industry date back to the 1990s. As a result, software developers are often concerned with solving the technical challenges in the application in addition to modelling and solving the actual societal problems associated with them. Plus, their solutions to the technical challenges are often unreliable or suboptimal. A research team led by Professor Mezini of the Software Technology Group at the Technical University of Darmstadt is therefore developing a programming approach that independently solves some of the technical challenges of the “world supercomputer”. The current implementation of the approach is called REScala (www.rescala-lang.com).

The researchers fix some of the shortcomings of the previous programming techniques with REScala. These are basically geared towards a synchronous programming model. Put simply, in this model, software systems control the order in which their tasks are performed. But in today’s systems, such as a smart home or networked production, the process flow is no longer determined by the program itself and instead defined by the interaction with incoming data and events. In the traditional model, programs want to control when what data arrives and what happens to it, but external events do not wait for the program to become ready – the programs no longer control what is going to happen next.

A simple example: In a food warehouse, a program is supposed to continuously document the temperature using three sensors distributed in the room, and then calculate the average room temperature. The sensors automatically transmit their data to the software every time there is a change in temperature. For external events such as these, developers have hitherto used so-called callback functions. Unlike normal functions, which are always called at the same time during the execution of the program, a callback is not executed until an event occurs. This can be a mouse click on a website, or just the incoming signal from a sensor. In the example of food storage, this would mean that every time a temperature value arrived, the average temperature would have to be recalculated to ensure it is up-to-date.

“When data is coming in from hundreds of thousands of sources, it is possible that the overall state of the system may be inconsistent.”

“A callback solution is quite sufficient for a simple application, but when data is coming in from hundreds of thousands of sources, the overall state of the system easily becomes inconsistent. If you had about 5,000 sensors, a new value could come in while the computer was still busy calculating the

Information

Software Technology Research Group

Prof. Dr.-Ing. Mira Mezini
Email: mezini@st.informatik.tu-darmstadt.de
www.stg.tu-darmstadt.de

total value – which would then already be outdated. This is even more problematic for the “world super-computer” where there are billions of devices and no hope to coordinate between them. In medicine or finance, short-term false values could have dire consequences. To avoid this, the software providers build up complex, confusing program structures for each new application. The result is “accidental complexity” to be dealt with in the application. Such complexity is not due to the actual problem, but due to the challenges of the setting. Managing accidental complexity comes at the expense of focusing the attention to managing the inherent complexity of the problem – and thus at the expense of development time, and resources computing power, reliability and, ultimately, innovation.

REScala, at its core, works a bit like Excel. If you change a value in an Excel spreadsheet, all its dependent values automatically change. This means users can concentrate on modelling the connections in the data. How calculations are performed in the background to efficiently and continuously produce correct up-to-date results makes no difference to the users. They rely on the automation working reliably for them. REScala applies this idea to general-purpose programming by extending the Scala programming language. However, in doing so, it rethinks the idea to ensure automation in a world where data and computations are globally distributed. where calculations are concurrent without central coordination, and where the communication infrastructure is unreliable. Strict consistency is an unsolvable problem in such an environment. However, REScala ensures as much consistency as is necessary and possible in every situation – if no connection is available at the moment, then all the changes are recorded locally and consistency is restored at the next opportunity. This is called “eventual consistency”. REScala ensures this level of the application state consistency “by-design”. On top of this, it enables developers to declare application-level invariants that must hold no matter what from which is derives parts of the application, where stronger consistency is needed and automatically ensures it.

REScala developers simply describe how asynchronously arriving data and events are combined and processed by composition of functions, whose execution is activated automatically. Application developers are not concerned with the timing and order of execution of individual computational modules. REScala ensures that they execute automatically whenever necessary to adjust their state to



Photo: Fritjhof Kjer

Computer Science Professor Mira Mezini

the arrival of new data and events. In this process it affords automatic reliability and consistency, thus greatly simplifying the software.

REScala comes at the right time. Countries around the world are currently building the fifth generation mobile network, with research being carried out into the sixth – with potential transmission rates of up to one terabit per second. In addition, new accelerator chip sets will drive edge computing, and thus the Internet of Things, immensely in the coming years. As a result, more and more machine learning applications can process data locally. Decentralised intelligence is tremendously important, both from an economic and from an environmental point of view, as well as in terms of data sovereignty. It is likely to change society as radically as the smartphone did. REScala can help to pave the way.

The author is a technology journalist.

Awarded by the European Research Council

Professor Mira Mezini was awarded an ERC Advanced Grant by the European Research Council in 2012 for her project PACE (Programming Abstractions for Applications in Cloud Environments). Since 2019, the ERC has continued to support her research with a Proof of Concept Grant of 150,000 euros. In the REScala (A Programming Platform for Reactive Data-intensive Applications) project, she and her team are investigating how the knowledge generated by PACE can be transferred into novel programming languages and platforms for the development of data-intensive decentralised software systems.