SCIENCES & BIOENGINEERING SCIENCES

The Research Group

Software Languages Lab

has the honor to invite you to the public defense of the PhD thesis of

Christophe De Troyer

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:
A meta-level architecture for stream-based programming languages and its applications in cyber-physical systems

Promotors:
Prof. dr. Wolfgang De Meuter (VUB)
Prof. dr. Jens Nicolay (VUB)

The defense will take place on
Monday, June 27, 2022 at 17h in auditorium D.0.07

Please contact Christophe De Troyer at christophe.de.troyer@vub.be if you want to join the presentation remotely.

Members of the jury

Prof. dr. Ann Nowe (VUB, chair)
Prof. dr. Abdellah Touhafi (VUB, secretary)
Prof. dr. Elisa Gonzalez Boix (VUB)
Prof. dr. Walter Cazzola (Universita degli Studi di Milano)
Prof. dr. Hidehiko Masuhara (Tokyo Institute of Technology)

Curriculum vitae

Christophe De Troyer obtained his professional bachelor's in applied informatics at Ghent College in 2012, and his master's degree in computer science at the Vrije Universiteit Brussel (VUB) in 2015. He then started a PhD at the Software Languages Lab (SOFT), funded by the FWO-SBO project "D3-CPS", where he assisted with several courses, and supervised several bachelor's and master's thesis students.

His research has mainly focused on meta-programing for stream programming languages and its applications in context of Cyber-Physical systems to help developers in expressing software for large-scale and unreliable networks of heterogenous devices.
His work led to a peer-reviewed journal publication, a conference publication, and workshop publications.

Abstract of the PhD research

Contemporary computer networks no longer consist of just mainframes and desktop computers. Due to the miniaturization and mass production of computer chips, it has become feasible to embed computers into everyday objects, making it possible to integrate them into a larger cyber-physical system to achieve a common goal and improve the user experience of the physical device. We call these systems Cyber-Physical Systems (CPS).

Contemporary CPSs consist of up to thousands of heterogeneous devices connected to the same network. Devices continuously react to data and instructions emitted by other devices and emit new data and instructions in response, causing chain reactions throughout the system.

The different hardware and software traits of the devices affect the applications deployed on these networks. Some devices are battery-powered, making them unreliable in the long term. Some devices are connected using unreliable networks, making them unreliable in the short term. Some devices are functionally equivalent (e.g., thermometers) but differ in their non-functional aspects (e.g., data representations or protocols), affecting how they integrate into the system. And finally, some devices do not have sufficient resources to compute complex problems.

In this dissertation, we propose a two-pronged approach to design contemporary CPSs. First, we propose a novel distributed reactive stream-based architecture to design CPSs. Reactive stream applications are declarative transformations of possibly infinite streams of data. Distributing reactive streams over the network allows multiple devices to work together on a single computation. The distributed stream-based paradigm fits well to the event-driven large-scale event-driven nature of CPS.

Secondly, we propose a meta-level architecture for stream domain specific languages (DSL) to express the non-functional concerns of stream applications. We use the meta-level architecture to express the non-functional concerns and heterogeneous software and hardware traits of the devices in a CPS.

We show that the reactive stream paradigm is a good foundation to design CPSs. The reactive, distributed nature of stream programming enables the design of scalable, distributed, resilient, event-driven systems. The paradigm nudges the programmer to design extensible applications that deal with open systems. The meta-level architecture of the stream language separates the non-functional concerns and separates the logic necessary to integrate heterogeneous devices.

We evaluate our stream paradigm by implementing a prototype stream DSL called Creek. We develop a prototype meta-level architecture in Creek, called Creekμ. We evaluate the meta-level architecture by implementing shortcomings identified in the literature, such as pull-based semantics, logging, encryption, and operator fusion. Additionally, we evaluate the runtime performance impact of the meta-level architecture, and implement a rudimentary debugger based on the meta-architecture. Finally, we evaluate our stream-based architecture for CPS by implementing a prototype called Potato and use it to implement a smart building use case commonly found in the literature. We show that our approach reduces the accidental complexity associated with CPS and that it results in maintainable and extensible applications.