SCIENCES & BIOENGINEERING SCIENCES

The Research Group

Software Languages Lab (DINF-SOFT)

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

Janwillem SWALENS

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:

A multi-paradigm concurrent programming model

Promotor:

Prof. dr. Wolfgang De Meuter

The defence will take place on

Thursday September 27 2018 at 18:00h

in Auditorium D.0.07 at the Campus Humanities, Sciences and Engineering of the Vrije Universiteit Brussel, Pleinlaan 2 - 1050 Elsene, and will be followed by a reception.

Members of the jury:

Prof. dr. Viviane Jonckers (chairman)
Prof. dr. Beat Signer (secretary)
Prof. dr. Joeri De Koster (co-promotor)
Prof. dr. Jan Lemeire
Prof. dr. ir. Mira Mezini
(Technische Univ. Darmstadt)
Prof. dr. Hridesh Rajan (Iowa State University)

Curriculum vitae

Janwillem Swalens acquired his Master's degree in Applied Sciences and Engineering: Computer Science (specialization Software Engineering) in 2013. Afterwards, he started a PhD at the Software Languages Lab of VUB. His work focuses on the design and development of programming languages that optimally exploit the parallelism and concurrency of multi-core processors. The results of his research have been published in two full conference papers and three short conference papers, four of which as first author. He presented these results at international, peer-reviewed conferences.

Abstract of the PhD research

Since the introduction of multi-core processors, programmers must explicitly use concurrency to make their programs faster. This is notoriously difficult. To this end, developers use concurrency models: techniques that introduce parallelism in a controlled manner and provide guarantees to prevent common errors, e.g. race conditions and deadlocks. An empirical study has shown that existing programs often combine multiple such models. We study these combinations and show that they can annihilate the guarantees of their constituent parts. Hence, the assumptions of developers are invalidated and errors can resurface.

In this dissertation, we start from three concurrency models: futures, transactions, and actors. First, for the combination of transactions and futures, we create transactional futures: futures created in a transaction with access to the encompassing transactional context. Second, the combination of transactions and actors leads to transactional actors. These make it possible both to create transactions in actors, and vice versa, to send messages to actors in transactions. Finally, we combine all three models into one unified framework: Chocola (for "Composable Concurrency Language"), implemented as an extension of Clojure, formalized using an operational semantics, and evaluated in three applications.

This dissertation thus comprehensively studies the combination of three radically different concurrency models – futures, transactions, and actors – and specifies a semantics for their combinations that aims to introduce parallelism while maintaining their guarantees.