VUB SCIENCES & BIOENGINEERING SCIENCES

The Research Group

Software Languages Lab

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

Kevin De Porre

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:
When sequential code meets replicated data

Promotor:
Prof. dr. Elisa Gonzalez Boix (VUB)

The defense will take place on
Monday, December 19, 2022 at 17h in auditorium D.2.01

Please contact Kevin De Porre at kdeporre@vub.be if you want to join the presentation through Microsoft Team.

Members of the jury

Prof. dr. Bart de Boer (VUB, chair)
Prof. dr. Bas Ketsman (VUB, secretary)
Prof. dr. Nikos Deligiannis (VUB)
Prof. dr. Annette Bieniusa (Technical University of Kaiserslautern)
dr. Martin Kleppmann (Technische Universität München)

Curriculum vitae

Kevin De Porre obtained his MSc in Computer Science at the VUB in 2018 and got awarded the BrEA "Prijs voor de computerwetenschappen" for his master thesis. He then started a PhD at the Software Languages Lab (SOFT) supported by an SB fellowship from the Research Foundation - Flanders.

His research focused on programming languages and techniques that help programmers design, implement, and verify highly available distributed systems using replicated data. Kevin's research resulted in four publications in international peer-reviewed journals and conferences, and two contributions in international peer-reviewed workshops.

Abstract of the PhD research

Many applications today run atop a geo-distributed system, that is, a system that replicates data over several machines (called replicas) located at strategic positions around the globe. Such systems typically ensure good performance and high availability by allowing replicas to be updated independently. However, replicas may execute concurrent updates in different orders, which can lead to conflicts. Programmers must foresee and handle these conflicts, which is extremely difficult.

To avoid manual conflict resolution, researchers proposed replicated variants for sequential data types, called Replicated Data Types (RDTs). RDTs resemble common data structures but internally implement mechanisms to detect and solve conflicts. RDTs can have different semantics, depending on how conflicts are solved.

In this thesis, we argue that real-world applications require custom RDTs that are tailored to the needs of the application. However, existing RDTs cannot be customized as they exhibit hardcoded conflict resolution semantics. This leaves programmers no choice but to build their own RDTs or modify existing RDTs using ad-hoc conflict detection and resolution mechanisms. This is known to be error-prone and results in brittle systems. Moreover, these new or modified RDTs are seldom verified due to the complexity of software verification, and thus bugs are likely to go unnoticed.

To address the aforementioned issues, this dissertation explores programming language abstractions and techniques for the development of correct RDTs. This led to a solution that is twofold. First, we propose a general approach for replicating existing sequential data types instead of building dedicated RDTs for every use case. Our approach lets programmers augment sequential data types with a *declarative specification* of the desired conflict resolution semantics expressed through application invariants. We statically analyze the data type to detect all conflicts and find solutions that adhere to the desired semantics. At runtime, our novel replication protocol efficiently serializes operations such that replicas converge to the same state and maintain application-specific invariants with minimal coordination. To validate our approach, we successfully built an extensive portfolio of RDTs and several real-world applications that exhibit performance similar to state-of-the-art approaches.

Second, we devise VeriFx, a high-level programming language to implement and verify ad-hoc RDTs. Programmers implement custom RDTs in VeriFx and *automatically* get a proof of correctness or a counterexample if the implementation is wrong. Verified RDTs can be transpiled to mainstream languages to deploy them in an existing system. To demonstrate its effectiveness, we used VeriFx to implement and verify a portfolio comprising more than 40 RDTs, distilled from the literature on Conflict-free Replicated Data Types (CRDTs) and Operational Transformation (OT) and commercial databases.

Our results show two important insights. First, it is possible to build efficient RDTs with application-specific concurrency semantics, without having to manually handle conflicts. Second, the implementation and verification of RDTs can be unified in a high-level language such that software engineers without deep knowledge of verification can nevertheless implement RDTs and verify them automatically.