De Troyer Christophe

christophedetroyer.be
christophe@call-cc.be
+32 489 97 94 28
@cdetroyer
PDF version

PhD student, IoT, language design.

PhD student at Vrije Universiteit Brussel. Conducting research in reactive programming language design for the Internet of Things. MSc in Computer Science at VUB. Excited about Unix, language design, and, good music.

EDUCATION

2015 - Present	**University of Brussels** PhD Computer Science I am conducting research to find the best programming paradigm for data intensive Internet of Things applications. Currently I am working on my runtime for IoT devices, called Potato. Potato views an IoT system as a network of datastreams, and offers the programmer a toolset to easily compose, consume, and manipulate these streams.
2018 - Present	**University of Brussels** MSc Management
2012 - 2015	**University of Brussels** MSc in Computer Science (magna cum laude) Thesis: »Combining the Actor model with Software Transactional Memory« extensively studied the viability of integrating the actor model with software transactional memory and vice versa. Created proof of concept implementation in Clojure. Created operational semantics.
2009 - 2012	**Ghent College** BSc in Computer Science (programming major)

EXPERIENCE

VUB
Teaching Assistant
ECTS Card

2016 - Present

Teaching assistant "Structuur I".
Creating assignments, guiding, and grading.

VUB
Teaching Assistant
ECTS Card

2016 - Present

Teaching assistant "Multicore - OpenCL".
The course covers OpenCL, Erlang, and Clojure. I was the assistant for the OpenCL part.
Creating assignments, guiding, and grading.

VUB
Teaching Assistant
ECTS Card

2015 - Present

Teaching assistant "Programming Project I".
Creating assignments, guiding and grading.

14IT
Custom software solutions
14it.be

4/2015 - 1/2016

Internship

- Built real estate website using the Umbraco CMS (.Net) from scratch
- Connected website to legacy back-end

Computerparts BVBA
Computer Store
computerparts.be

2/2008 - 6/2008

Internship
Assembling, repairing computers.

TALKS

XAOP
Elixir/OTP

31 Jan 2017

Elixir/OTP Introduction
Gave a short technical talk by invitation on the Elixir/OTP framework.
»Slides«

IFL 2016
Extended Abstract

31 Aug 2016

Practical Session Types
On the requirements for the implementation of practical session types in contemporary languages.
»Slides«

PUBLICATIONS

CloudCom '18
Full Paper

13 Dec 2018

Building IoT Systems Using Distributed First-Class Reactive Programming
Contemporary IoT systems are challenging to develop, deploy, and maintain. This is because of their ever-increasing scale, dynamic network topologies, heterogeneity and resource constraints of the involved devices, and failures that may occur as a result of these characteristics. Existing approaches are either not at the right level of abstraction, require developers to learn specialized languages, or miss certain key features to address all these challenges in a uniform manner. In this paper we leverage reactive programming and code mobility to support the entire life-cycle of large-scale IoT systems. Our approach is based on existing programming technologies and offers simple and composable abstractions to developers. We implemented our approach in a middleware called Potato and used it to develop and deploy an IoT application on a Raspberry Pi cluster. We found that using Potato reduces much of the accidental complexity associated with developing and deploying IoT systems, resulting in clean and maintainable programs.
»PDF«

SPLASH '17
WIP

23 Oct 2017

First-Class Reactive Programs for CPS
Cyber-Physical Systems (CPS) are comprised of a network of devices that vary widely in complexity, ranging from simple sensors to autonomous robots. Traditionally, controlling and sensing these devices happens through API communication, in either push or pull-based fashion. We argue that the computational power of these devices is converging to the point where they can do autonomous computations. This allows application programmers to run programs locally on the sensors, thereby reducing the communication and workload of more central command and control entities. This work introduces the Potato framework that aims to make programming CPS systems intuitively easy and fast. Potato is based on three essential mechanisms: failure handling by means of leasing, distribution by means of first-class reactive programs, and intentional retroactive designation of the network by means of capabilities and dynamic properties. In this paper we focus on the reactive capabilities of our framework. Potato enables programmers to create and deploy first-class reactive programs on CPS devices at run time, abstracting away from the API approach. Each node in the network is equipped with a minimal actor-based middleware that can execute first-class reactive programs. We have implemented Potato as a library in Elixir and have used it to implement several small examples.
»PDF«

Programming '17
Position Paper

4 Apr 2017

Abstractions for Distributed Event-Driven Applications
The Internet of Things (IoT) requires us to rethink the way distributed event-driven applications are programmed. IoT applications differ from traditional distributed applications on a number of points. First, they are comprised of an order of magnitude more devices that operate within a dynamic network. Second, failure in large dynamic networks is no longer an exceptional state but a given and thus needs to be part of the core semantics when programming such networks. Third, the hardware in these networks is not homogeneous so that a common software stack is impossible. We believe that contemporary event-driven languages do not offer appropriate abstractions to write IoT applications. We propose a novel computational model for programming IoT applications by identifying four key abstractions for designating network nodes and handle failures that facilitate writing large-scale IoT applications.
»PDF«

INTERESTS

- Language design
- Type systems
- FP (Haskell, Clojure, Racket, Elixir, ...)
- Language implementation
- Unix systems
- Jazz, Hip-Hop, Rock
- Technical books
- Motorcycling

See https://github.com/m1dnight for personal projects.