



## Proceedings of the 14<sup>th</sup> Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software

Preface

**Castellani, Ilaria; Scalas, Alceste**

*Published in:*

Proceedings of the 14<sup>th</sup> Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software

*Link to article, DOI:*

[10.4204/EPTCS.378](https://doi.org/10.4204/EPTCS.378)

*Publication date:*

2023

*Document Version*

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*

Castellani, I., & Scalas, A. (2023). Proceedings of the 14<sup>th</sup> Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software: Preface. In *Proceedings of the 14<sup>th</sup> Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software*  
<https://doi.org/10.4204/EPTCS.378>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*arXiv:2304.05439v1 [cs.PL] 11 Apr 2023***DOI:**

10.4204/EPTCS.378

[EPTCS Site Map](#)[EPTCS Home Page](#)[Published Volumes](#)[Forthcoming Volumes](#)

ISSN: 2075-2180

# EPTCS 378

## Proceedings 14th Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software

Paris, France, 22 April 2023

Edited by: Ilaria Castellani and Alceste Scalas

*Preface*

Ilaria Castellani and Alceste Scalas

**Keynote Talk:** *VerCors & Alpinist: Verification of Optimised GPU Programs*

Marieke Huisman

**Keynote Talk:** *Thirty Years of Session Types*

Vasco T. Vasconcelos

*Presentations of Preliminary or Already-Published Work**Kind Inference for the FreeST Programming Language*

Bernardo Almeida, Andreia Mordido and Vasco T. Vasconcelos

1

*A Declarative Validator for GSOS Languages*

Matteo Cimini

14

*A Logical Account of Subtyping for Session Types*

Ross Horne and Luca Padovani

26

*Communicating Actor Automata - Modelling Erlang Processes as Communicating Machines*

Dominic Orchard, Mihail Munteanu and Paulo Torrens

38

*Choreographic Programming of Isolated Transactions*

Ton Smeele and Sung-Shik Jongmans

49

## Preface

This volume contains the proceedings of PLACES 2023, the 14th edition of the Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software. The workshop is scheduled to take place in Paris on 22 April 2023, as a satellite event of ETAPS, the European Joint Conferences on Theory and Practice of Software.

PLACES offers a forum for exchanging new ideas on how to address the challenges of concurrent and distributed programming, and how to improve the foundations of modern and future computer applications. PLACES welcomes researchers from various fields, and its topics include the design of new programming languages, models for concurrent and distributed systems,

type systems, program verification, and applications in various areas (e.g. microservices, sensor networks, blockchains, event processing, business process management).

The Programme Committee of PLACES 2023 consisted of:

- Marco Carbone, IT University of Copenhagen, DK
- Elias Castegren, Uppsala University, SE
- Silvia Crafa, Università di Padova, IT
- Francisco Ferreira, Royal Holloway, University of London, UK
- José Fragoso Santos, Universidade de Lisboa and INESC-ID, PT
- Paola Giannini, Università del Piemonte Orientale, IT
- Andrew K. Hirsch, State University of New York at Buffalo, US
- Sung-Shik Jongmans, Open University of the Netherlands, NL
- Luc Maranget, INRIA Paris, FR
- Andreia Mordido, Universidade de Lisboa and LASIGE, PT
- Violet Ka I Pun, Western Norway University of Applied Sciences, NO
- Emilio Tuosto, Gran Sasso Science Institute, IT
- Laura Voinea, University of Glasgow, UK

After a thorough reviewing process, the Programme Committee has accepted five research papers (out of seven submitted for review): such papers are published in this volume. The Programme Committee has also accepted seven talk proposal on preliminary or already-published work: the titles and abstracts of such talks are also listed in this volume (except for one, because the authors had to cancel their presentation). Each submission (research paper or talk proposal) was reviewed by three Programme Committee members and then discussed on the EasyChair platform.

We would like to thank everyone who contributed to PLACES 2023: this includes the authors of submissions, the Programme Committee members, the ETAPS 2023 organisers, the EasyChair and EPTCS administrators. We would also like to thank Marieke Huisman and Vasco T. Vasconcelos for accepting our invitation to give a keynote talk. Finally, a special thank you goes to the Steering Committee of PLACES, consisting of Simon Gay, Luca Padovani, Vasco T. Vasconcelos, and Nobuko Yoshida.

4 April 2023

Ilaria Castellani and Alceste Scalas

---

## VerCors & Alpinist: Verification of Optimised GPU Programs

**Marieke Huisman** (*University of Twente, NL*)

The VerCors verifier is a tool set for the verification of parallel and concurrent software. Its main characteristics are (i) that it can verify programs under different concurrency models, written in high-level programming languages, such as for example in Java, OpenCL and OpenMP; and (ii) that it can reason not only about race freedom and memory safety, but also about functional correctness. In this talk I will first give an overview of the VerCors verifier, and how it has been used for the verification of many different parallel and concurrent algorithms.

In the second part of my talk I will zoom in on verification of GPU programs, as they are widely used in industry. To obtain the best performance, a typical development process involves the manual or semi-automatic application of optimizations prior to compiling the code. To avoid the introduction of errors, we can augment GPU programs with (pre- and postcondition-style)

annotations to capture functional properties. However, keeping these annotations correct when optimizing GPU programs is labor-intensive and error-prone.

In my talk I introduce Alpinist, an annotation-aware GPU program optimizer. It applies frequently-used GPU optimizations, but besides transforming code, it also transforms the annotations. We evaluate Alpinist, in combination with the VerCors program verifier, to automatically optimize a collection of verified programs and reverify them.

---

## Thirty Years of Session Types

Vasco T. Vasconcelos (*University of Lisbon, PT*)

1993. Kohei Honda publishes *Types for Dyadic Interaction*. In the course of five years two further papers shaped a field that was to become known as Session Types. Session types discipline interactive behaviour in the same way that functional types govern applicative behaviour. What are session types? What are they good for? What sort of applications benefit from the discipline imposed by such types? What are the challenges ahead?

---

## Presentations of Preliminary or Already-Published Work

PLACES 2023 welcomed the submissions of talk proposals (describing preliminary or already-published work) that could spark interesting discussion during the workshop. This is the list of all accepted talk proposals.

### Concurrent Symbolic Execution with Trace Semantics in Coq

*Åsmund A. A. Kløvstad - Department of Informatics, University of Oslo, NO.*

Symbolic Execution is a technique for program analysis using symbolic expressions to abstract over program state, thereby covering many program states simultaneously. Symbolic execution has been used since the mid 70's in both testing and analysis, but its formal aspects have only recently begun to be explored and unified. We present a model of symbolic execution with trace semantics in a concurrent setting in Coq, utilizing syntactic contexts to succinctly deal with parallelism.

### MAG $\pi$ : Types for Failure-Prone Communication

*Matthew Alan Le Brun and Ornela Dardha - University of Glasgow, UK.*

This talk proposal is based on work accepted for publication at ESOP 2023. We introduce MAG $\pi$  - Multiparty, Asynchronous and Generalised  $\pi$ -calculus - an extension of generalised session type theory into a calculus capable of modelling non-Byzantine faults, for various physical topologies and network assumptions. Our contributions are: (1) a calculus and type-system enriched with timeouts and message loss semantics - capable of modelling the widest set of non-Byzantine faults; (2) a novel and most general definition of reliability, allowing MAG $\pi$  to model physical topologies of distributed systems; (3) a generalised theory capable of specifying assumptions of underlying network protocols; and (4) type properties that lift the benefits of generalised MPST into our realm of failure-prone communication.

## Functions as Processes: The Non-Deterministic Case

*Joseph Paulus - University of Groningen, NL.*

*Daniele Nantes-Sobrinho - Imperial College London, UK.*

*Jorge A. Pérez - University of Groningen, NL.*

Milner's seminal work on encodings of the lambda-calculus into the pi-calculus ("functions-as-processes") explains how interaction in pi subsumes evaluation in lambda. His work opened a research strand on formal connections between sequential and concurrent calculi, covering untyped and typed regimes.

In this talk, we review a recent series of works in which we extend "functions-as-processes" by considering calculi in which computation is non-deterministic and may lead to failures - two relevant features in programming models. On the functional side, we consider a resource lambda-calculus with non-determinism and failure, equipped with non-idempotent intersection types; on the concurrent side, we consider a session-typed pi-calculus in which non-determinism and failure are justified by logical foundations ("propositions-as-sessions"). We have developed correct encodings of the former into the latter; they describe how typed session protocols can codify sequential evaluation in which absence/excess of resources leads to failures.

Our work reveals a new connection between two different mechanisms for enforcing resource awareness in programming calculi, namely intersection types and session types. Our talk shall elaborate on the challenges involved in connecting these different type disciplines, and also how our encodings allow us to study confluent and non-confluent forms of non-determinism in the typed setting.

## Polymorphic Sessions and Sequential Composition of Types

*Diogo Poças, Diana Costa, Andreia Mordido, Vasco T. Vasconcelos - LASIGE, Faculdade de Ciências, Universidade de Lisboa, PT.*

Session types equipped with a sequential composition operator are known as context-free session types. The sequential composition operator poses new challenges not present in traditional, tail recursive types. The foremost challenge is probably deciding type equivalence. This problem has been studied in increasingly expressive systems, from first-order systems (where only base types may be exchanged), to higher-order systems; from Damas-Milner polymorphism to System F; and, more recently in the higher-order polymorphic lambda calculus. In all these systems, however, polymorphic types are of a functional nature, meaning that types cannot be exchanged on messages. We introduce polymorphic session types in a language of higher-order context free sessions and show that type equivalence is still decidable.

## Language Support for Implementing Algorithms on Low Level Hardware Components

*Mads Rosendahl, Maja H. Kirkeby, Mathias Larsen, Martin Sundman - Roskilde University, DK.*

*Tjark Petersen, Martin Schoeberl - Technical University of Denmark, DK.*

Future optimizations of algorithms will include hardware implementations targeting a field-programmable gate array (FPGA). However, describing hardware in a hardware description language like VHDL or Verilog is cumbersome compared to describing an algorithm in a software language like C or Java. An alternative is to use High-level synthesis to convert programs in C into hardware design.

We explore language extensions that can assist programmers in designing algorithms for FPGA components and be integrated into existing hardware designs. The aim is to give the programmer control over the parallelism while retaining the algorithmic aspects in the development process.

We compare hardware designs generated using the language extensions with designs written directly in hardware description languages.

## Multiparty Session Types Meet Message Sequence Charts

*Felix Stutz - Max Planck Institute for Software Systems, DE.*

Implementing communication protocols is a routine task for distributed software. However, verifying that a protocol is implemented correctly in an asynchronous setting is challenging. The implementability problem asks if a (global) protocol can be implemented locally and has been studied from two perspectives. On the one hand, multiparty session types (MSTs) provide a type-theoretic approach that restricts the expressiveness of protocols. Its projection operator is a partial function that, given a protocol, attempts to compute a correct-by-construction implementation. As a best-effort technique, it is very efficient but rejects implementable protocols. On the other hand, high-level message sequence charts (HMSCs) do not impose any restrictions on the protocols, yielding undecidability of the implementability problem for HMSCs. Consequently, model-checking can easily diverge but also suffers from high complexity. Our research aims to bridge the gap between both approaches. In this talk, we report on recent results from this endeavour. I will first visually explain classical MST projection operators and exemplify their shortcomings, showcasing sources of incompleteness for the classical MST projection approach. Then, I will elaborate on our decidability result for MST implementability. For this, we exploit our formal encoding from MSTs to HMSCs, generalise results for the latter, and prove that any implementable MST falls into a class of HMSCs with decidable implementability. Last, I will showcase techniques from the HMSC domain that become applicable in the MST setting with these results.

---