

API Generation of Multiparty Session Types

Telmo Ribeiro | José Proença & Mário Florido
Faculdade de Ciências da Universidade do Porto

Introduction

Multiparty Session Types (MPST) are a typing discipline ensuring that a message-passing process implements a given multiparty session protocol [1].

Session Types ensure the correctness of concurrent programs, where the validity of a protocol can be asserted through the well-typedness of its agents.

Multiparty describes a session with more than two agents in contrast with the parent concept of binary session types.

The MPST framework guarantees [2]:

- **Communication safety** i.e., absence of errors;
- **Progress**, i.e., deadlock-freedom;
- **Session fidelity**, i.e., protocol compliance

Objectives

- Achieve a framework capable of **generating APIs** allowing the implementation of different **MPSTs** in **Scala3 (APIGenMPST)**;
- Allow the choice between multiple **semantics**;
- Allow the choice between multiple **constructors**;

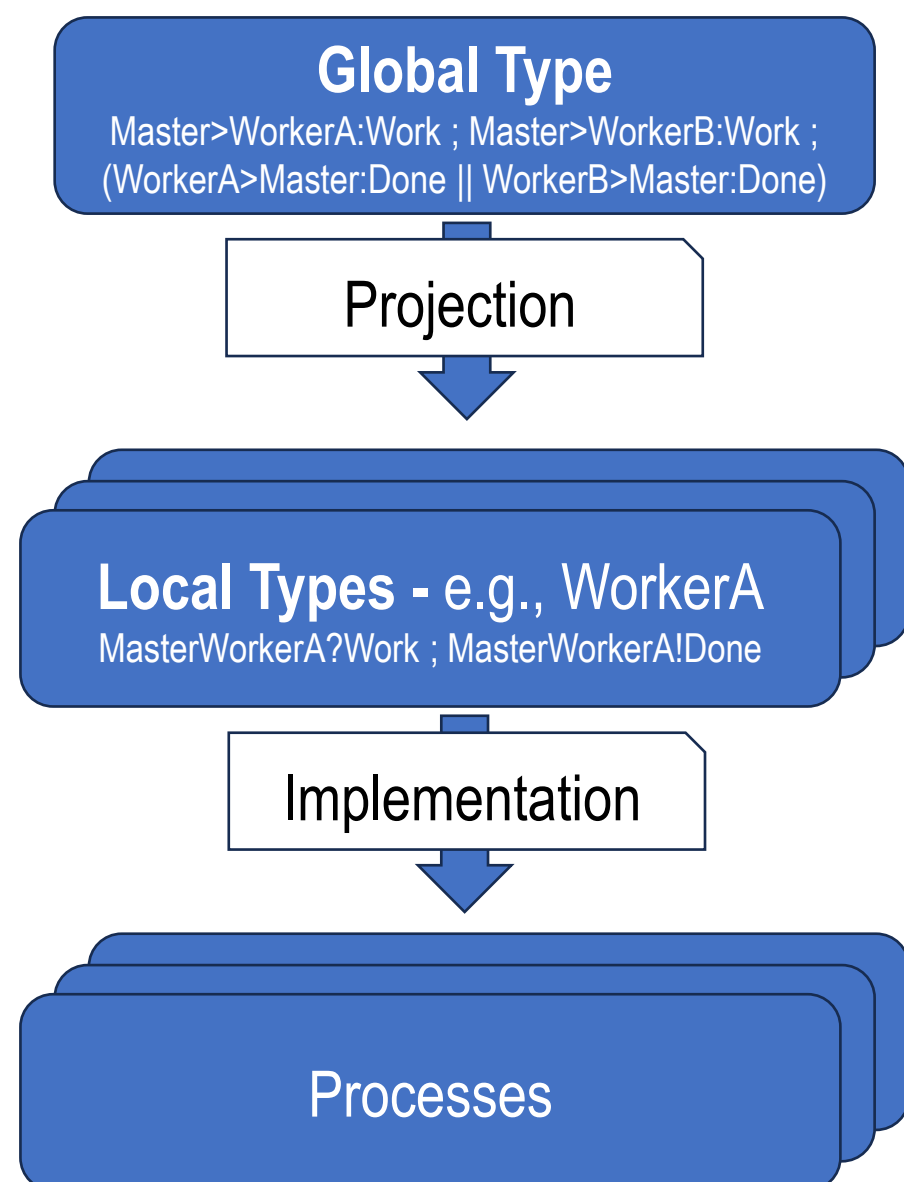
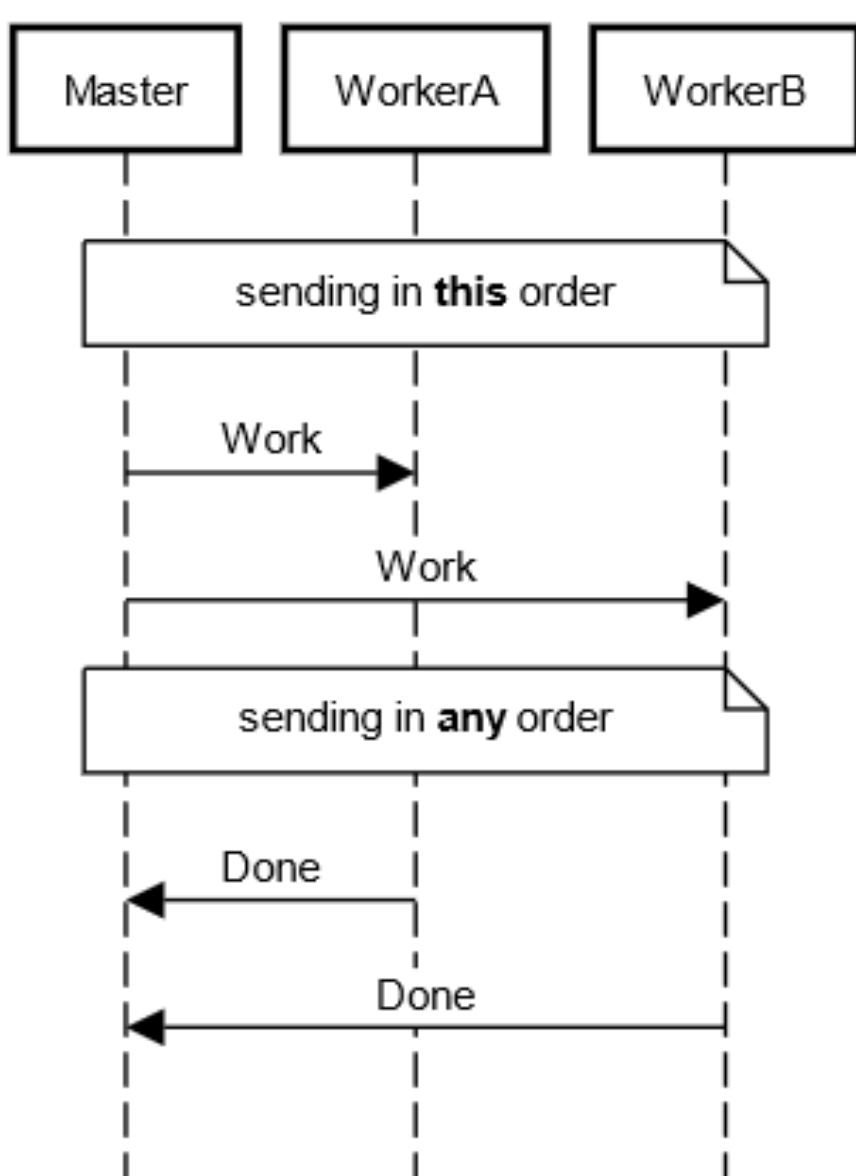
State-of-the-Art (SotA)

- Overview and roadmap of **different models of MPST**, with different expressivity and guarantees [1].
- **API generation** to enforce session types in different programming languages [3] [4].
- Use of **Scala 3**'s type system to generate compact APIs.

This work proposes **APIGenMPST** – a framework to generate APIs of **different MPST models** using **Scala 3**.

Examples

Master-Workers Workflow



Workplan

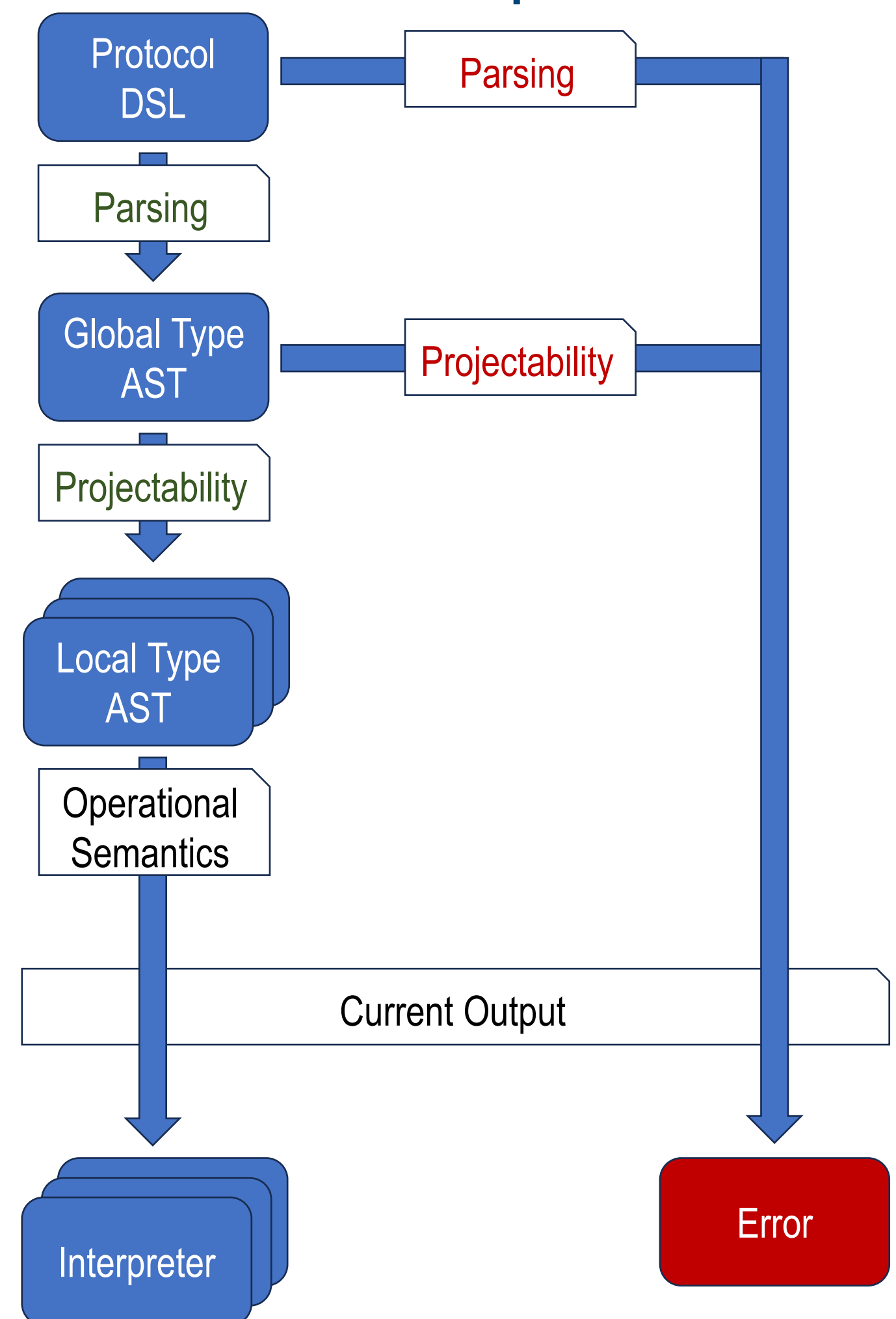
	10-23	11-23	12-23	01-24	02-24	03-24	04-24	05-24
Task 1								
Task 2								
Task 3								
Task 4								
Task 5								
Task 6								

Task 1 – Survey API Generation Engines
Task 4 – Explore Other Engine Variations

Task 2 – Categorize Engine Supported Concepts
Task 5 – Execute a Concurrent System

Task 3 – API Implementation for MPST
Task 6 – Write the dissertation

Tool Development



References

- [1] Scalas, A., & Yoshida, N. (2019). Less is more: multiparty session types revisited. *POPL*, 1-29.
- [2] Honda, K., Yoshida, N., & Carbone, M. (2008). Multiparty asynchronous session types. *POPL*, 273-284.
- [3] Hu, R., & Yoshida, N. (2016). Hybrid session verification through endpoint API generation. *FASE 2016*, 401-418.
- [4] Laguardie, N., Neykova, R., & Yoshida, N. (2020). Implementing multiparty session types in rust. *International Conference on Coordination Languages and Models*, 127-136.
- [5] Cledou, G., Edixhoven, L. J., Jongmans, S. S., & Proença, J. (2022). API generation for multiparty session types, revisited and revised using scala 3.