

# Typechecking Java Protocols with [St]Mungo

**Laura Voinea**, Ornela Dardha , Simon J. Gay  
School of Computing Science, University of Glasgow, United Kingdom

# Overview

- Toolchain Overview
- Mungo Typechecking
- HTTP Case Study
- <https://bitbucket.org/abcd-glasgow/mungo-tools/>

# The [St]Mungo toolchain

## StMungo(Scribble-to-Mungo)

- Java-based transpiler implemented using the ANTLR v4.5 framework
- Translates **Scribble local types** into **typestate specifications** for Java classes
  - The **Scribble specification language** is an implementation of multiparty session types
    - Communication protocols are expressed as global protocols/types
    - Global protocols are checked for validity,
    - And projected to local protocols for each participant
  - **Typestates** define valid sequences of operations that can be performed upon an instance of a given type
- Generates an API implementation for each participant, which follows its typestate specification

# The [St]Mungo toolchain

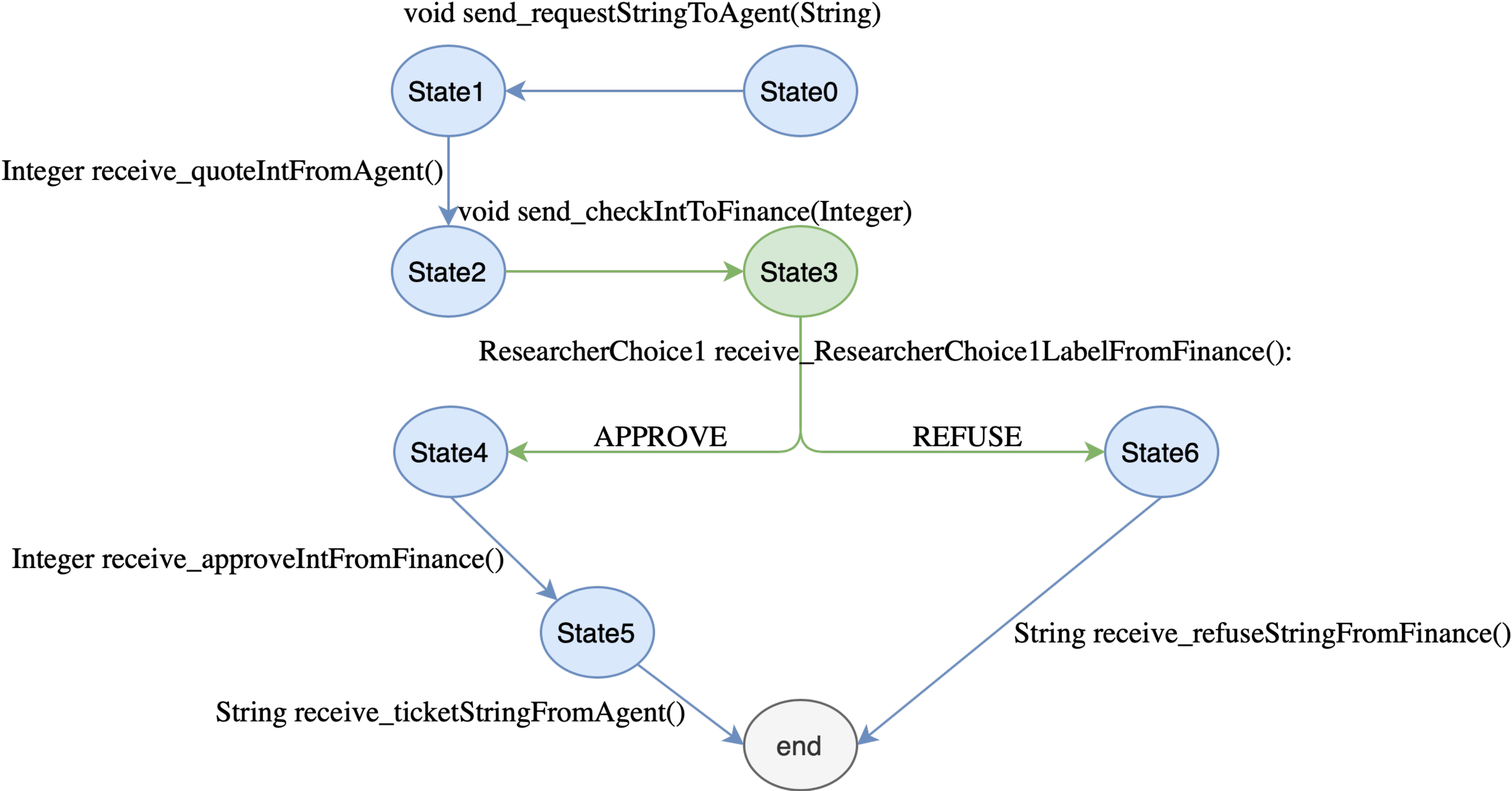
## Mungo

- A Java front-end tool used to statically typecheck typestate specifications
- Typestate specifications are expressed as annotations of Java classes
- Checks that method calls are performed following the object's protocol, as specified by its associated typestate
- If no errors are reported the code is compiled and run as standard Java code.
- The typechecker is formalised inspired by session types theory and the resulting type system is proved correct via the standard theorems of progress and subject reduction

# Travel Agency Example

- Models the process of booking a flight through a university travel agent as a communication protocol
- Three participants are involved:
  - a **Researcher**, who intends to travel;
  - an **Agent**, who is able to make travel reservations;
  - and **Finance**, who approves expenditure from the budget.
- We first represent this protocol in the Scribble language as a **global type**

# Researcher typestate



# Typechecking with Mungo

- Implemented in the extendj/JastAdd framework, a Reference Attribute Grammar (RAG) meta-compiler suite compatible with Java
- Typechecking for a subset of Java
- Tracks variables storing instances of classes with typestate specifications, through argument passing and return values
- Typestate inference system to eliminate the need for typestate declarations on parameters and return types

# Future work

- Mungo
  - Static typechecking of generics
  - Static typechecking of exceptions
  - Better error messages
- StMungo
  - keep it in line with the Scribble specification language



Thank you!