

# Java Pathfinder at SV-COMP-2019

**Cyrille Artho**

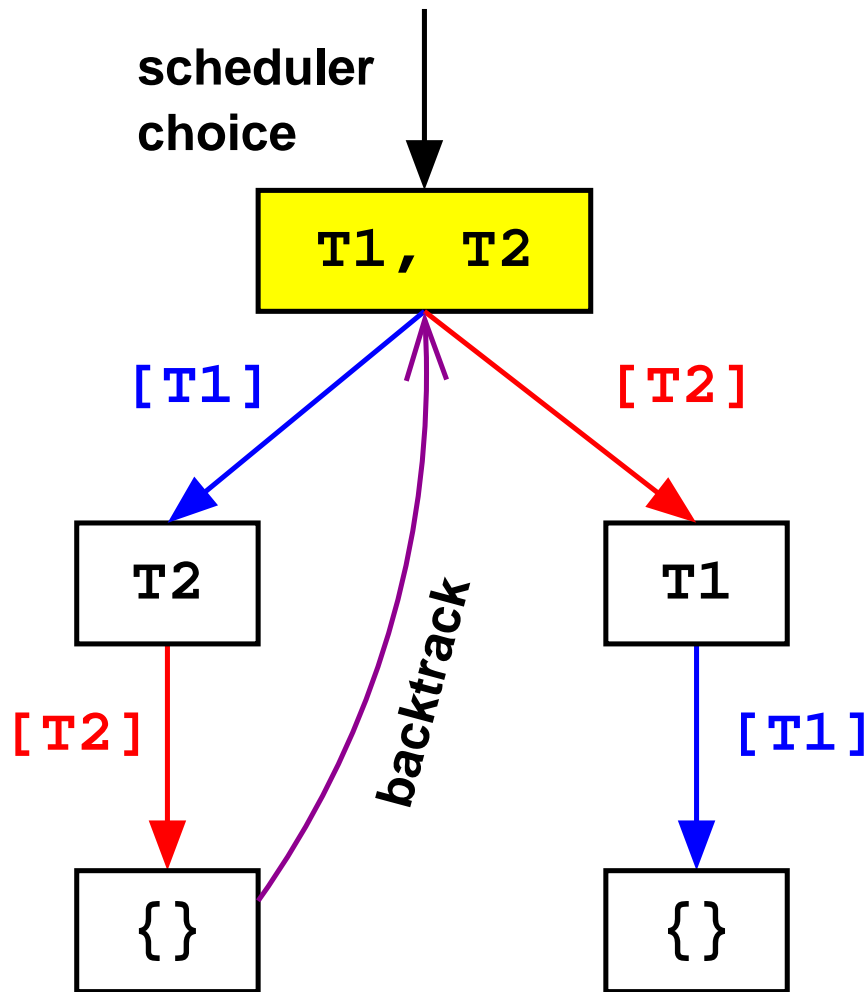
KTH Royal Institute of Technology  
Stockholm, Sweden

**Willem Visser**

Stellenbosch University  
Stellenbosch, South Africa

and many, many contributors

# What can software model checking do?



- **Backtrack** program execution.
- Explore all possible schedules!
- Find all program failures.

# What is Java PathFinder?

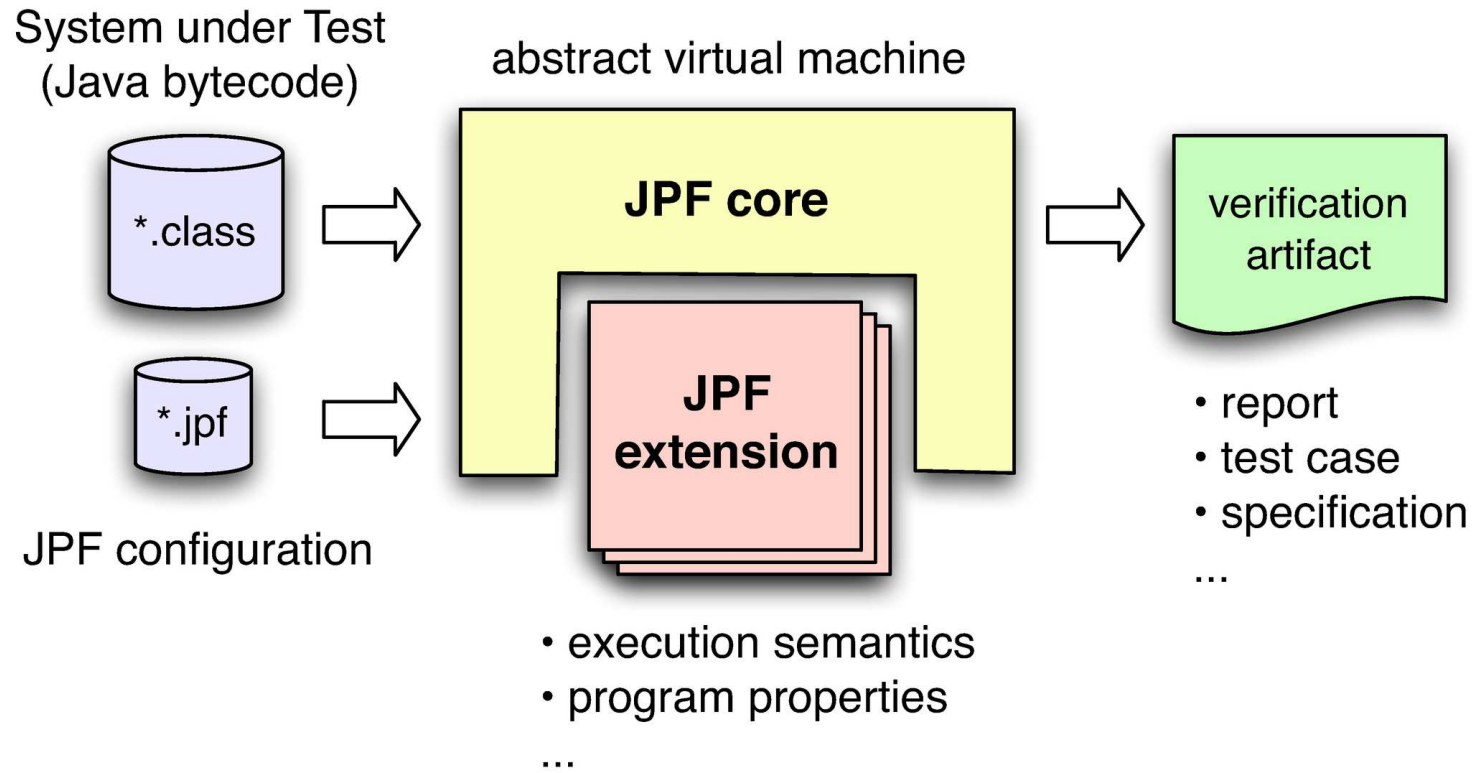


Image source: Java PathFinder tutorial,

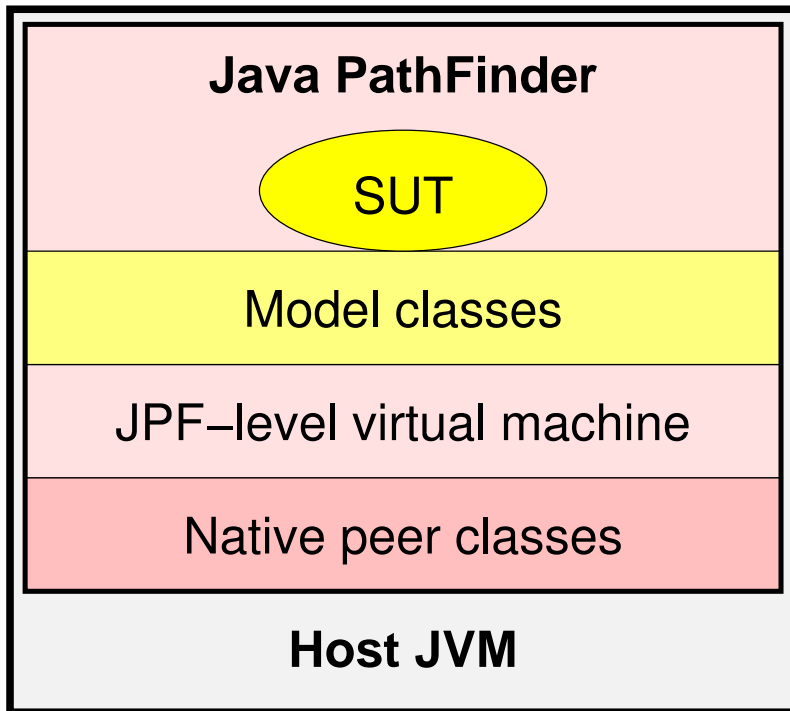
<http://babelfish.arc.nasa.gov/trac/jpf/attachment/wiki/presentations/start/jpf-graphics.z>

- Born as software model checker for Java bytecode.
- Evolved into multi-purpose tool framework.

# Architecture of JPF



# Architecture of JPF

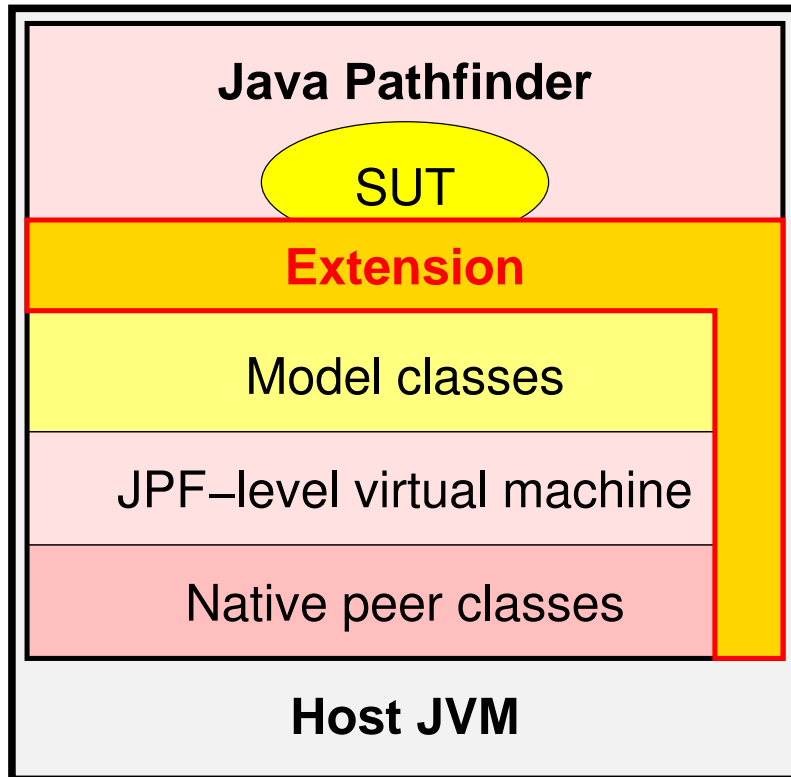


- **JPF: virtual machine on top of „host JVM”.**
- Explicit-state model checker.
- No native code in SUT + model classes.
- **Native peer classes** can access native code from underlying JVM.
- Native peers are not managed or backtracked by JPF.

# JPF extensions

## JPF extensions:

- Separate projects.
- Extend or override jpf-core.
- New features:
  - Native code bridge.
  - Network I/O.
  - Temporal logics.
  - Symbolic model checking.



# History of JPF

- **1999: Project started** at NASA Ames as front end to Spin model checker.
- **2000: Reimplementation** as concrete virtual machine.
- **2003:** Introduction of **extension** interfaces.
- **2005: Open sourced** on sourceforge.
- 2008: First participation in Google Summer of Code.
- 2009: Moved to own server, hosting extension projects and wiki.
- **2017: Moved to github.**
- Many collaborators/users:
  - Academia: > 20 univ. (USA, Canada, Japan, South Africa, Europe).
  - Companies: Fujitsu, other Fortune 500 companies.

# Summary/Outlook

- SV-COMP 2019 for Java just scratched the surface.
- **Wanted:**
  - interesting concurrent benchmarks;
  - programs using native code;
  - multi-process (networked) applications.

