Checking a Concurrent Multi-Agent Robotic Simulation with Java Pathfinder

Calden Wloka

York University, Toronto

4 April, 2011

Outline

- Review of implementation
- Plans for Java Pathfinder (JPF)
- Results
- Discussion

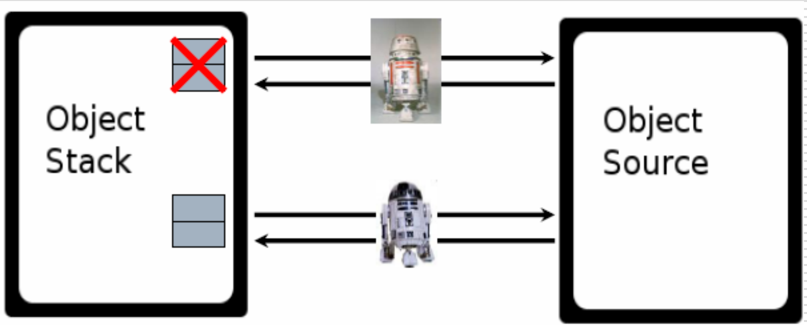
Review of Problem

Task 1:
See an object
Move to object
Pick up object
Move to home

~~**Task 2:**~~
~~Stack object~~

Task 1:
See an object
Move to object
Pick up object
Move to home

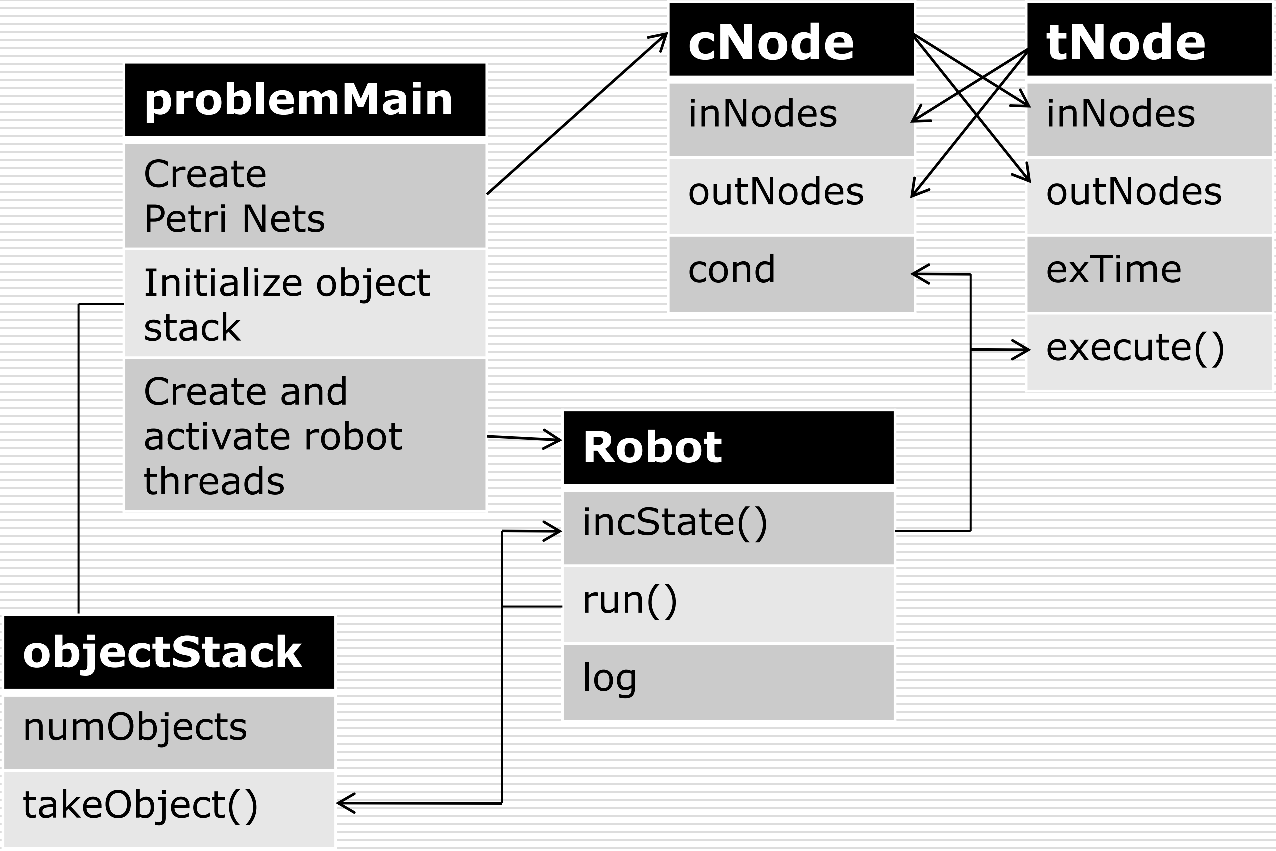
Task 2:
Stack object



Object Stack

Object Source

Review of Implementation



problemMain

Create Petri Nets

Initialize object stack

Create and activate robot threads

cNode

inNodes

outNodes

cond

tNode

inNodes

outNodes

exTime

execute()

Robot

incState()

run()

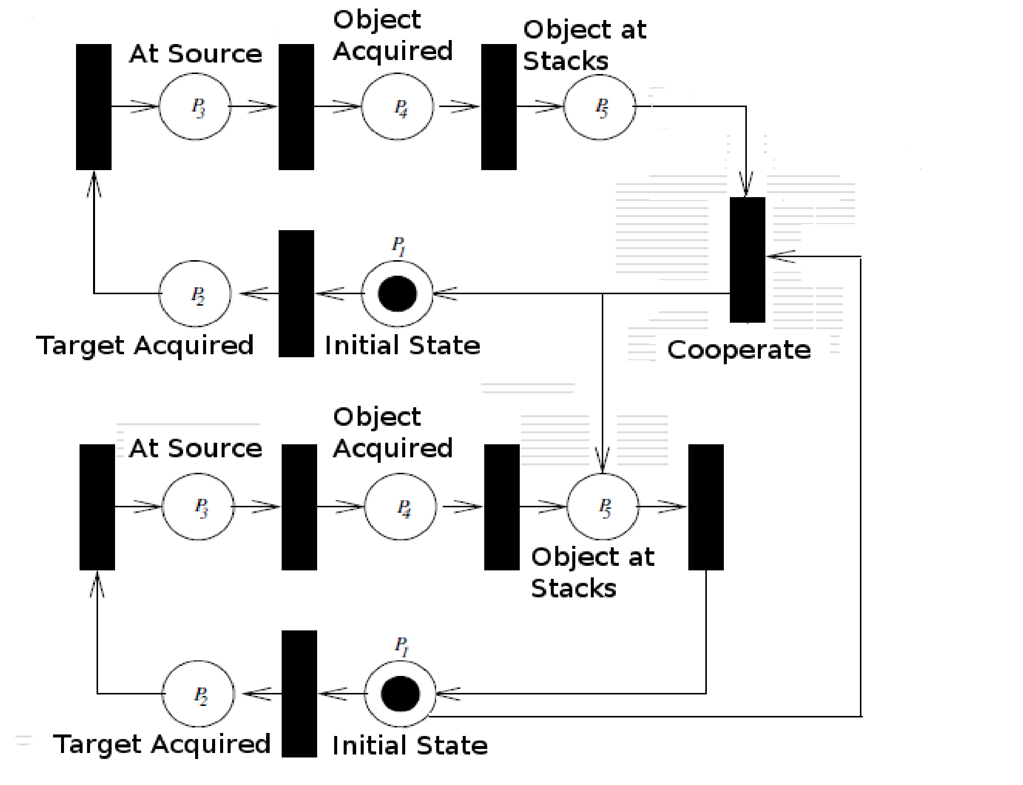
log

objectStack

numObjects

takeObject()

Review of Implementation



Java Pathfinder (JPF) Plans

Three items to investigate with JPF:

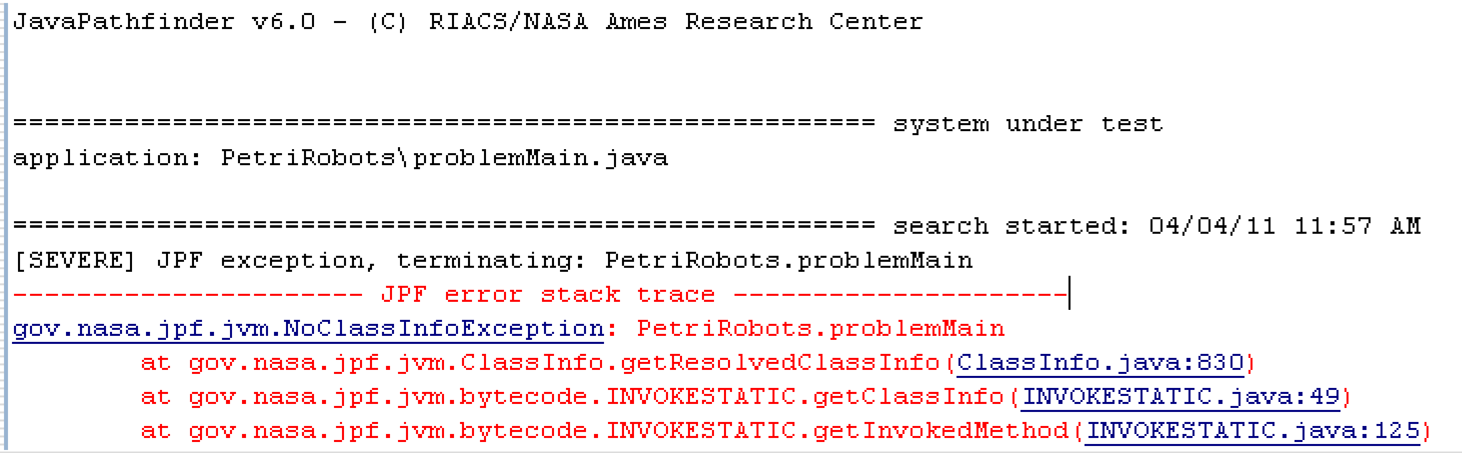
- Run JPF on main program with a small number of objects to deliver
  - Reduce the action times one one robot to increase the effect of noise on interleavings
  - Reduce the action times for both robots

Java Pathfinder (JPF) Plans

Three items to investigate with JPF:

- Remove the lock on the stack to see if the data race gets picked up by JPF
- Increase the number of robots to three (first two damaged robots, then two healthy robots) to use JPF to diagnose problems

Results



```
JavaPathfinder v6.0 - (C) RIACS/NASA Ames Research Center


======================================================= system under test
application: PetriRobots\problemMain.java

======================================================= search started: 04/04/11 11:57 AM
[SEVERE] JPF exception, terminating: PetriRobots.problemMain
---------------------- JPF error stack trace ---------------------
gov.nasa.jpf.jvm.NoClassInfoException: PetriRobots.problemMain
        at gov.nasa.jpf.jvm.ClassInfo.getResolvedClassInfo(ClassInfo.java:830)
        at gov.nasa.jpf.jvm.bytecode.INVOKESTATIC.getClassInfo(INVOKESTATIC.java:49)
        at gov.nasa.jpf.jvm.bytecode.INVOKESTATIC.getInvokedMethod(INVOKESTATIC.java:125)
```

Discussion

- Any ideas on what I did wrong?
- Assuming I could get things working, what other aspects of the robot system might be interesting to investigate with JPF?