

Schedulability Analyzer for Real-Time Systems

Petur Olsen

Aalborg University
<http://sarts.boegholm.dk/>

6. okt. 2008

Introduction

Safety critical systems

- Hard Real-Time Systems
- Loss of lives
- Loss of money
- Environmental damage

Contribution

Automatic Tool: SARTS

- Automatic Verification
- Greater precision
- Eclipse plug-in

Outline

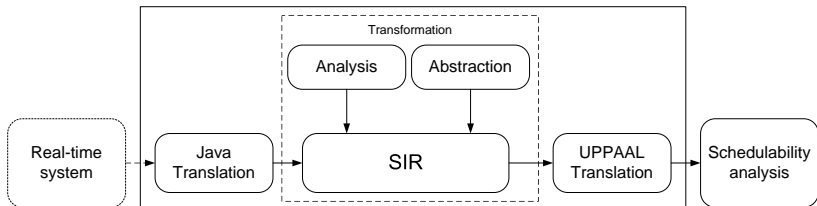
- 1 SARTS
- 2 Eclipse Plug-in
- 3 Conclusion

Schedulability Analyzer for Real-Time Systems

SARTS

- Generates Timed Automata
- Control flow graph with timing information
- UPPAAL Model-checker
- Check for deadlock

Design

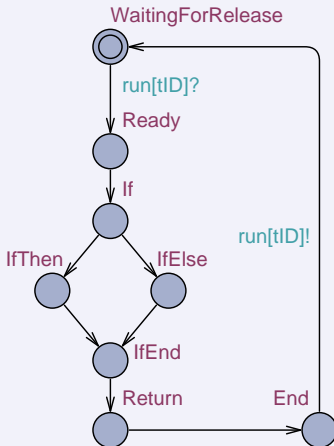


Example

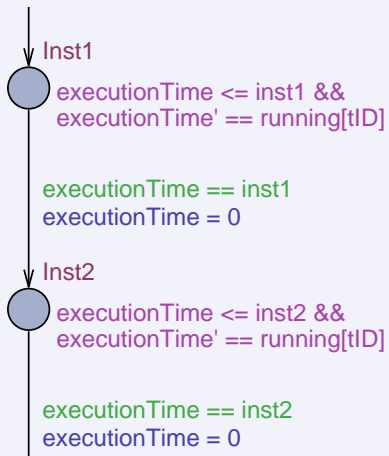
Java code

```
protected boolean run() {  
    if (condition){  
        //then statements  
    } else {  
        //else statements  
    }  
    return true;  
}
```

UPPAAL model



Basic Block



Pattern examples

Model Patterns

- Loops
- Monitor statements
- If statements
- Method invoke
- Sporadic task release

Platform / Framework

Supports

- Safety Critical Java profile (SCJ)
- Java Optimized Processor (JOP) (www.jopdesign.com)

Predefined models

- Scheduler
- Periodic Task
- Sporadic Task

Model-checking

UPPAAL

- Every generated model runs in parallel
- Verify property
- Generate error trace

Problems

- State space explosion
 - Branchings
 - Method invokes

Eclipse Plug-in

Implementation

- External build tool
- SARTS settings
- UPPAAL settings
- Run

Demonstration

Demonstration

RTSM

Data

- 2 periodic and 2 sporadic tasks
- 300 lines of code
- 17 methods
- Generates 65 UPPAAL models
- Verification time: 37 seconds

Future Work

Plug-in Improvements

- Incremental build
- Improved feedback
- Visualize WCET

Conclusion

- SARTS implementation
- More accurate analysis
- Automatic translation and verification
- Eclipse plug-in
- Needs optimization

Questions?