CERTIFIABLE JAVA FOR EMBEDDED SYSTEMS

Martin Schoeberl, Andreas Engelbredt Dalsgaard, René Rydhof Hansen, Stephan E. Korsholm, Anders P . Ravn, Juan Ricardo Rios Rivas, Tórur Biskopstø Strøm, Hans Søndergaard

OUTLINE

- Project Goals
- Two JVMs
- SCJ implementations
- Tools
- Applications
- Summary

CJ4ES PROJECT

- Danish technology and production research fund
    - 2 PhD plus some extra person resources
    - Duration 3 years
- Partners
    - Technical University of Denmark
    - Aalborg University
    - GomSpace

PROJECT GOALS

- SCJ Implementations (L0 and L1)
- On Java processor JOP and on top of RTSJ
- RT library
- Use cases - SCJ example applications
- Analysis tools: resource usage (memory and time)
- SCJ compliance checker
- Verification of safety properties

SAFETY-CRITICAL JAVA

- RTSJ derived and restricted Java platform
- Priority based scheduling of handlers
- Scoped memory with single scope stack
- Concept of missions
- Three compliance levels

JVMS USED

- Hardware near Virtual Machine (HVM)
  - Software JVM in plain C
    - Interpreter and compiler
    - Bare metal, no OS needed
- Java Optimized Processor (JOP)
  - Hardware implementation of the JVM
  - WCET tool available

WHAT IS AN SCJ IMPLEMENTATION?

- Scheduler
- Scoped memory
- IO
- Fill in RTSJ and SCJ defined classes

SCHEDULER

- Programmable timer with interrupt
- Interrupt handler
- Issue with scope violation
  - Interrupt handler is in immortal, handlers in mission memory

SCOPED MEMORY

- Simpler than RTSJ, single scope stack
- Use a single class to implement all three types
  - Immortal, mission memory, private memory
- Simple assignment testing with nesting level
- Issues: too much inherited from RTSJ
  - Need to cross package boundaries

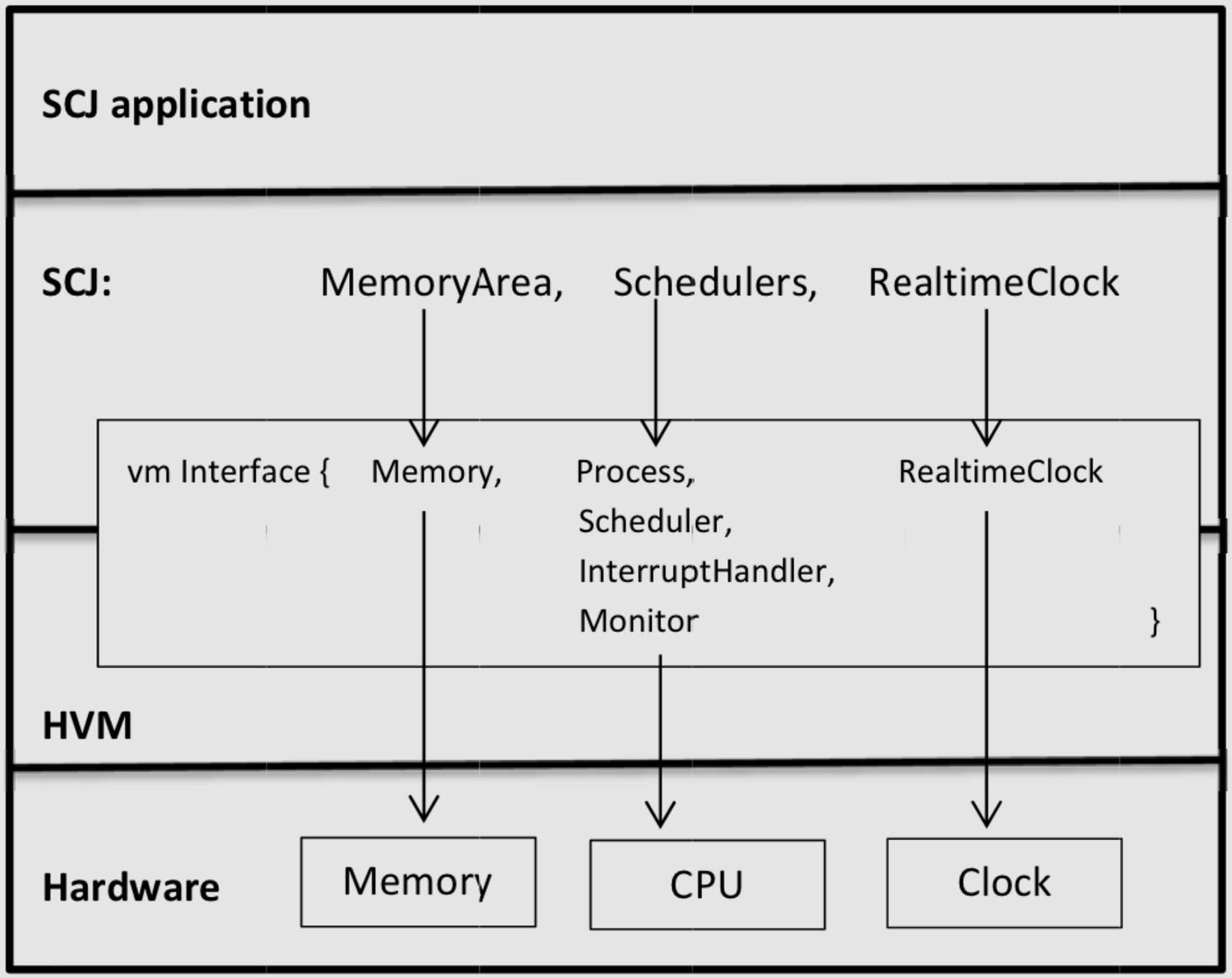
IO

- SCJ defines very many interfaces for memory mapped IO
- HVM and JOP provide also hardware objects
  - Nicer IO abstraction, but not SCJ
  - But IO is usually not portable ;-)

RTSJ AND SCJ CLASSES

- SCJ now less dependent on RTSJ classes
- But still some legacy
- Results in almost empty RTSJ classes
- Plus issue with package crossing
  - Where does the core implementation live ?

VM INTERFACE TO HVM



TOOLS

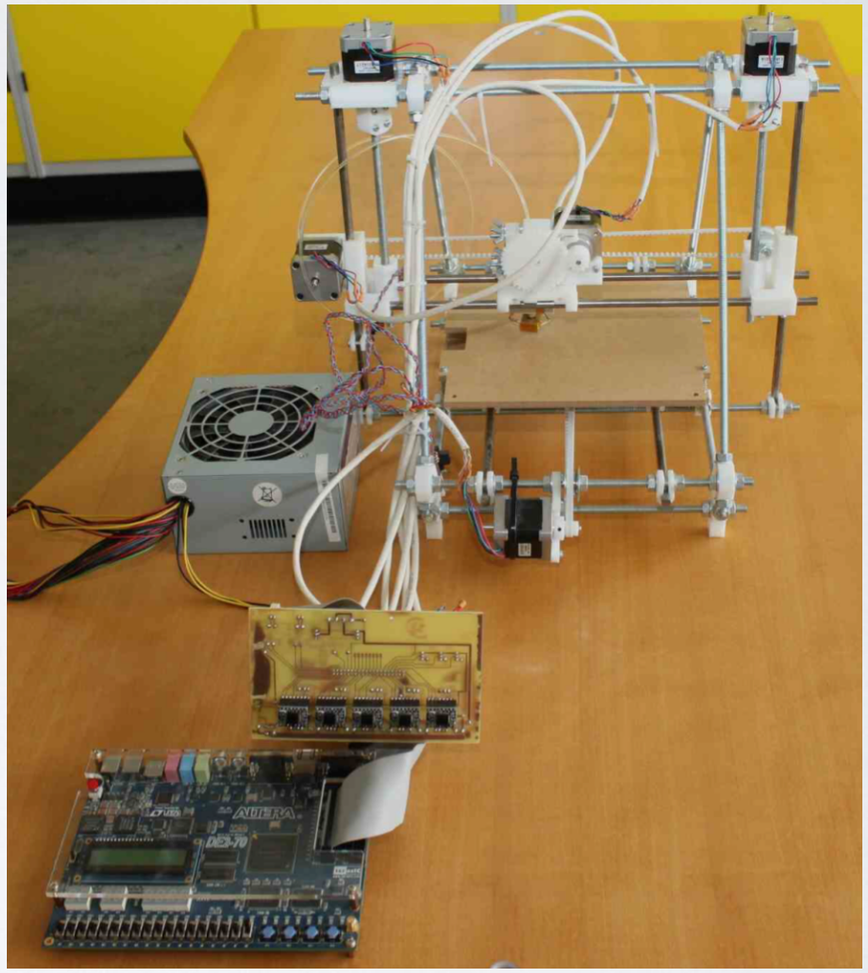
- WCET analysis
  - IPET based for JOP, model checking based for HVM
- Memory usage analysis
  - Easier with scopes than for GC collected heaps
- Static assignment checker

NANO SATELLITE

- Watchdog for nano satellite
- Cubesat space protocol
  - Similar to UDP/IP
  - Developed at Aalborg university
- Interrupt handler for CSP packets
- Periodic handler for watchdog ping packets
- Tested against a GomSpace satellite

RERAP 3D PRINTER

- DIY 3d printer (RepRap)
  - "Print" plastic parts
- Printer is a standard design
- Custom interface board
- 3 periodic handlers for:
  - Host comm, parsing, and controlling the motors



SUMMARY

- Two open-source implementations of SCJ
  - On the HVM and on JOP
- Assessed usability of SCJ with use cases
- Apply program analysis techniques to verify safety properties of SCJ applications
  - WCET, worst-case memory consumption, assignments

EXPLORE RESULTS

- All project results are open source
- HVM can run on a PC
- JOP has a simulator in Java
- Develop your SCJ application on a PC