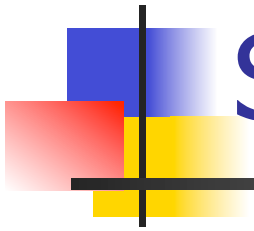
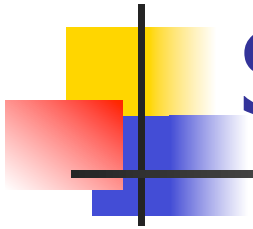


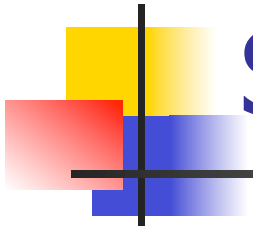
Safety-Critical Java

Martin Schoeberl



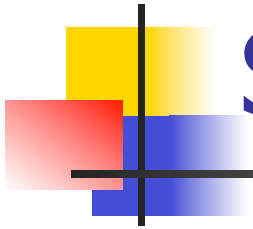
Safety Critical Java

- Certification for DO-178B level A
- Java Specification Request 302
  - Lead Dough Locke
- Restricted subset of RTSJ
  - More worst case analysis friendly
  - JSR 302 on-going work
- 3 different levels



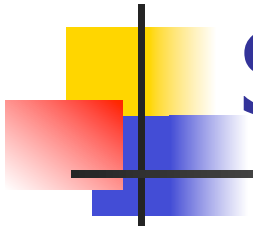
SCJ Levels

- L0 Single threaded
  - Cyclic executive
- L1 Static threads
  - Initialization and mission phase
  - Ravenscar like
  - No wait/notify
- L2 Multiple missions



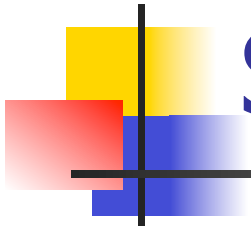
SCJ Memory Model

- No Garbage Collection
- RTSJ immortal memory
- RTSJ style scoped memories
  - Scopes are thread private
  - Communication via immortal
- Memory model still under discussion
  - Type system to avoid scope checks



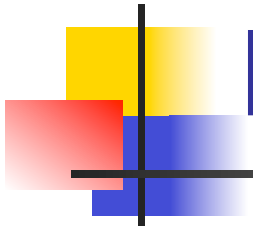
SCJ Execution Model

- Initialization phase - not time critical
  - Class initializing
  - Setup of all data structures and threads
- Mission phase
  - Mission can be restarted
  - Level 2: nested missions
    - More dynamic systems
    - Mode change not (yet) well defined



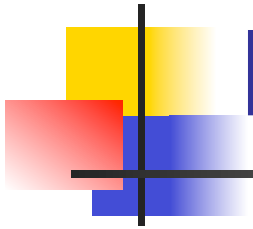
SCJ Tasks

- No threads at level 0
- Static threads/event handlers, priorities
- Event handlers
  - Time-triggered periodic
  - Event-triggered aporadic
- Single run method for all tasks
  - No waitForNextX()
  - No local state preserving



Periodic Tasks

- RTSJ
  - waitForNextPeriod()
  - Split of logic possible
- SCJ
  - Single run() method
  - Easier to analyze



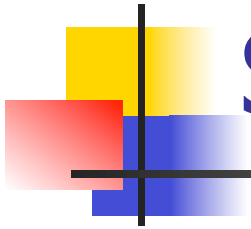
RTSJ Periodic Task

```
public void run() {

    State local = new State();
    doSomeInit();
    local.setA();
    waitForNextPeriod();

    for (;;) {
        while (!switchToB()) {
            doModeAwork();
            waitForNextPeriod();
        }
        local.setB();
        while (!switchToA()) {
            doModeBWork();
            waitForNextPeriod();
        }
        local.setA();
    }
}
```

- Possible abuse
  - Local state
  - Initialization
  - Split logic
- WCET analysis harder

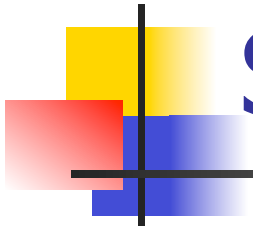


SCJ Periodic Task

- run() *executed*
  - Periodic (time triggered)
  - Sporadic (event triggered)
- No local state
- Single method for WCET analysis

```
new PeriodicThread(
    new RelativeTime(...)) {

        public void run() {
            doPeriodicWork();
        }
};
```



SCJ Summary

- Restricted Java/RTSJ
- Aiming for certifiable systems
- Specification is in final phase
- First implementations emerging
  - Could be your master project ;-)