WiP: An optimizing Framework for Real-time Scheduling

Sakthivel M. SUNDHARAM, Nicolas NAVET, Sebastian ALTMLEYER, University of Luxembourg

(RTAS 2016)
Vienna, Austria, April 12, 2016
Activation conditions (aka “guarded executions”) are for implementing functioning modes and executing event-triggered activities.
The choice of non-preemptive scheduling:
- No context-switch + no cache related preemption delays (CRPD) on the WCET + less memory usage
- No shared resources, easier to validate, less timing variability
- But .. reduced ability to meet tight deadline constraints

Currently FIFO, EDFNP, FPNP policy are available:
- FIFO Enforces event-order determinism, (i.e.) very predictable
- FIFO gives Work-conserving unlike static cyclic scheduling

Built-in support for WCET measurements at run-time

Everything is defined by the user ....
CPAL scheduler synthesis – Why?

- Scheduler synthesis: automatic selection and optimization of all scheduling parameters
- Application developer to take care on high level details of system development
- Low level details of the system development -> timing behavior is taken care by synthesis
- Interpretation engine will have the scheduling configuration synthesized
- development cycle time thus reduced because of automation
Conclusion & ongoing work

- CPAL: an interpreted language on a time-triggered execution engine - imperative programming in the functional domain - declarative programming in the non-functional domain
- Positive feedback about CPAL through industrial use-cases and teaching
- Code generation feasible for higher performance - hook to native code too

**Ongoing work:**

1. Timing equivalence between models in simulation and execution / SILx for the execution engine
2. Scheduler synthesis step to automate selection and optimization of scheduling parameters

*CPAL is free to use for academics (research works and industrial projects), Extensions to the language and toolset are welcome*