Time-Sensitive Software

Edward A. Lee¹ and Jim Woodcock²

¹UC Berkeley, Berkeley, USA and ²University of York, York, UK

Context

Timing of software execution is usually considered a performance property rather than a correctness property. But in software for cyber-physical systems, timing is often a critical feature of the execution of the software. Today, no widely used programming language specifies timing. Instead, timing is an emergent consequence of a particular implementation and is sensitive to every detail of the hardware on which the software runs and to what other software may be sharing the same hardware. Even a small change in the hardware or software context can lead to drastically different timing behaviour, making maintenance and upgrades difficult.

This question seeks contributions that turn this picture around. Just as a programmer delegates to the compiler and the microprocessor corrects execution of the program logic, we seek ways to similarly delegate the delivery of timing requirements. The contributions can include programming language enhancements, compilation techniques, innovative computer architectures, modelling methods and formal analysis techniques.

Some problems to consider include:

- How to model time (e.g., discrete, dense, super-dense; totally or partially ordered; linear vs. branching time; logical vs. physical time; hard vs. soft real time).
- How to analyse temporal properties (e.g., temporal logics, explicit-time logics, fundamental limits; temporal calculi).
- Programming language constructs (e.g., mechanisms for concurrency and composition; synchronous vs. asynchronous; comparisons of timing constructs in legacy programming languages; priorities vs. explicit timing requirements).
- Static analysis techniques (e.g., execution time analysis; feasibility analysis).
- Operating-system capabilities (e.g., real-time operating systems; clock synchronisation; distributed systems coordination; scheduling).
- Measurement and evaluation (e.g., testing frameworks; benchmarking; repeatability).

How to contribute to this Question

If you believe you can contribute to answering this Question with your research outputs, find out how to submit them in the Instructions for authors (https://www.cambridge.org/core/journals/research-directions-cyber-physical-systems/information/author-instructions/preparing-your-materials). This journal publishes Results, Analyses, Impact papers and additional content such as preprints and "grey literature". Questions will be closed when the editors agree that enough has been published to answer the Question so before submitting, check if this is still an active Question. If it is closed, another relevant Question may be currently open, so do review all the open Questions in your field. For any further queries, check the information pages (https://www.cambridge.org/core/journals/research-directions-cyber-physical-systems/information/about-this-journal) or contact this email (cps@cambridge.org).

Competing interests. The authors declare none.

References

On models of time:


On time in concurrent and distributed systems


On the physics of time

On microarchitectures and time