A survey of real-time scheduling tools

A Survey of Real-Time Scheduling Tools

TIK-Report No. 72

November 1998

Martin Naedele *

Abstract

Based on the corresponding publications, a survey of tools for the evaluation of the timing behavior of real-time systems is presented. Both analytical and simulation-based tools are surveyed.

*ETH Zurich, Computer Engineering and Networks Laboratory, CH-8092 Zurich, email: naedele@tik.ee.ethz.ch

1 Introduction

This survey of real-time analysis tools and scheduling simulators reflects the status of November 1998. Except for the commercial version of PERTS/DRTSS, none of the tools described below was at that time available for public evaluation, nor were there any independent evaluation results or case studies reported, so that the descriptions given here had to rely on papers published by the tool authors[1].

2 ASSERTS

ASSERTS (A Software Simulation Environment for Real-Time Systems) [10] targets distributed and heterogeneous systems. It allows the modeling of non-standard systems by a combination of parameter boxes and specification of the task body with a pseudo-code language that also contains control flow and communication primitives. The system allows multiple CPUs and the scheduling and resource access protocols associated with each CPU can be different. The system is specified by means of parameter files. Several system components are provided as building blocks: Futurebus, Ethernet, and fully-connected interconnection for communication, as well as RMS, GRMS, EDF, and Cyclic Executives for scheduling. The scheduling algorithms can be customized with scheduling computation times following a linear model. It is not clear whether new communications protocols can be added by a user. ASSERTS also contains some simple analytical tests, mainly focused on rate-monotonic scheduling.

The authors of ASSERTS see the main difference to PERTS/DRTSS in the fact that pseudo-code is used for the task body specification instead of C^{++}. STRESS also uses pseudo-code, but is said to have a less comfortable user interface and no built-in interconnection models and schedulers. Also the general approach to the modeling of intertask communication is different in STRESS and ASSERTS.

The system has been developed in cooperation with Lockheed-Martin Corporation. It does not seem to be publicly available and further development appears to have been discontinued.

3 DET/SAT/SIM

Design tool (DET), schedulability analyzer (SAT) and scheduling simulator (SIM) are part of a tool suite for time critical applications which also contains a code worst-case execution time estimator and a run-time execution tracer [2, 1, 3].

The real-time system is described on two levels: System level and node level. The system structure is constructed from nodes and communication links. Tasks are then manually mapped to nodes.

Tasks can be classified as critical, hard, or soft with respect to the consequences of failure to meet the deadlines. Periodic tasks as well as soft aperiodic

[1] Comparisons between specific tools given by some of the tool authors do not necessary reflect the personal view of the author of this survey.

tasks can be modeled. Further attributes for tasks are deadline, computation time, period, importance, and how much the task may miss its deadline.

Supported algorithms for the scheduling analyzer are RMS, DMS, and EDF for processor scheduling as well as PIP, PCP, DCP, and SRP for resource access. The analysis result is presented as yes/no indication only.

The simulator contains a generator for random aperiodic loads, specified by random distributions. The output of the simulator is a Gantt-chart like display. It appears as if no particular customization support is provided. The authors see the major difference to PERTS in the fact that PERTS does not contain a code worst-case execution time estimator.

Also part of this toolset seems to be the scheduling simulator GHOST (General Hard-real-time Oriented Simulator Tool)[16] for uniprocessor systems. The input task set is specified with a small language in the form of basic blocks, the execution times of which follow certain distributions. There does not seem to be a provision for nondeterministic or data-dependent branching in the control flow. The scheduler modules are written in C, and new algorithms can be added by creating C-functions that follow certain conventions. Similar to DRTSS, GHOST has a experiment control unit which allows to vary input (load) variables and calculate statistical values for certain output events. The authors see the main differences to STRESS in the fact that STRESS is only intended for hard real-time systems, and to DRTSS in that fact that DRTSS is not oriented towards the performance analysis of scheduling algorithms.

The tool set was developed within the European Community ESPRIT programs TRACS and MORIS. It is currently not publicly available.

4 PERTS tool set

The PERTS (Prototyping Environment for Real-Time Systems) tool set [18, 17, 14, 15, 23, 24, 20, 19] consists of a schedulability analyzer and a simulator.

4.1 PERTS SAT

For the schedulability analyzer [14, 15, 23] the model of the real-time system is specified using a graphical editor for task graph and resource graph. The resource graph can specify several nodes with processors and other resources onto which the tasks are then mapped automatically or manually. Tasks can be periodic or aperiodic, for the latter case case several periodic server options(polling server, persistent server, priority-exchange server, deferrable server, sporadic server) are available. Tasks are described by a number of attributes which also include optional parts and specification of intervals for resource access. Resources can be global or local. Implemented scheduling algorithms are RM, DM, and EDF. Resources are accessible via the PCP and SBP protocols. The output of the schedulability analyzer shows the total processor utilization and indicates for each task whether it is schedulable or not. There is also a graphical display of requested processing time versus processor capacity. The scheduling algorithms and resource access protocols must be identical for all nodes. The PERTS SAT is the most comprehensive scheduling analyzer among all the tools in this survey, however it does not seem to extendible.

4.2 DRTSS/PERTSSim

Major features of DRTSS (PERTSSim in the commercial version) [18, 17, 24, 20, 19] are the output analysis unit, the parameter variation (search) engine, and the *live task model*.

The output analysis unit can be programmed with a proprietary language to trigger on and process primitive simulator events (such as task started, task preempted, etc.) to extract the particular information one is interested in for the model under observation. This filtered and processed output can also be used to direct the parameter variation of a design space exploration run.

In the *live task model*, tasks and resources are represented by C++-classes and can contain real code. This allows on the one hand to implement new scheduling and resource access protocols, and on the other hand a stepwise refinement of the functionality of a task, until the final, fully functional code of the system to be developed is emulated by the simulation framework.

The last research version of the PERTS tool set can be obtained from the University of Illinois at Urbana-Champaign. A commercial version is distributed by Tri-Pacific Software, which also continues the development.

5 SEW

SEW (System Engineer's Workbench) [9, 8] is an analysis tool without simulation component, targeted at the evaluation of large-scale data stream oriented applications. The analysis functionality is based on the design and analysis methodology developed by Chatterjee in his PhD thesis. The system to be analyzed is modeled using large-grain building blocks for computers, networks, buses, and disks. These building blocks also encapsulate the access algorithms for the respective resource. It is not clear whether and how a new resource type can be customized or added. The analysis function computes bounds for end-to-end delay, jitter, buffer requirements, and resource utilization. The tool is in internal use in several companies.

6 AFTER

AFTER (Assist in Fine-Tuning for Embedded Real-time Systems) [5, 6, 4] is a tool to analyze traces of real-time programs and suggest improvements. The input for the tool are timing profiles extracted from an existing system. These timing profiles are analyzed for possible problems, and some modifications such as changing task period, changing task execution time, switching between fixed and dynamic priority scheduling, and switching between handling of aperiodic events as interrupts or aperiodic servers, can be made and their respective consequences evaluated. It seems [6], as if AFTER can only be used for systems developed on the basis of the VRTX-32 real-time operating system.

7 Tool of the Universite Libre de Bruxelles

The work described in [11, 25] consists of a textual simulation language for real-time systems, and the corresponding simulator. As with STRESS, the simulation language can be used to specify task set, resources and scheduling algorithms. The authors see the difference to STRESS in that they have a higher language level without implementation details, and that their language is better suited to describe new scheduling approaches [25]. The simulation output is displayed as Gantt-chart. The tool is not publicly available.

8 Framework of the Oregon State University

In [13] a C^{++} framework for scheduling simulation with emphasis on concurrency control policies is described. The framework consists of classes for schedulers, objects (passive resources), tasks, and performance monitoring. Schedulers for RMS and EDF are provided by default, others need to be added as program code. In an input file, tasks are described as sequences of several basic instructions (e. g. computer, read, write), following one predefined task structure. It is not clear from [13], whether the simulator needs to be recompiled for each simulation run with a different scheduler, or whether a configuration can be specified in the input file. As output, a stream of predefined events is generated.

9 CAISARTS

CAISARTS [12] is a rule-based expert system for schedulability analysis. Its main feature is the ability to provide context-sensitve advice for design improvements. The current status is unknown.

10 STRESS

The STRESS [7] scheduling analyzer and simulator is based on a single textual description language for resources (also network protocols etc.), application task structure, scheduling algorithms, and resource access protocols.

For schedulability analysis, DMS and RMS scheduling algorithms with the PCP resource access protocol, as well as EDF and LLF scheduling are supported. For simulation, any additional algorithms can and are intended to be specified using the STRESS language. The simulation result is presented in the form of a Gantt-chart.

The authors see the difference to PERTS in the fact that PERTS is directed towards the generation of actual application code from the simulation tools whereas STRESS is intended for the experimentation with novel scheduling approaches.

Development of the STRESS simulator has been discontinued and the tool is not available any more.

11 Scheduler 1-2-3

Scheduler 1-2-3 [22, 21] is a standalone schedulability analyzer/simulator as part of the ARTS project at the Carnegie Mellon University.

Tasks can be periodic or aperiodic. For periodic tasks, RM scheduling with the PI resource access protocol is assumed [22], the degree of support of for EDF and FCFS is not clear [21]. Aperiodic tasks are scheduled by means of a deferrable server. Context switching overhead can be taken into account.

The analytical schedulability analysis is augmented by a simulator for average response times of aperiodic tasks with randomly distributed arrival following a given distribution. This simulator makes use of an internal workload generator the output of which can also be accessed independently.

The current status is unknown. From [22] it is not entirely clear whether the simulator features are actually implemented in the described version of the tool.

References

[1] Paolo Ancilotti, Giorgio Buttazzo, Marco Di Natale, and Marco Bizzarri. A flexible tool kit for the development of real-time applications. In *Proceedings RTAS 96*, 1996.

[2] Paolo Ancilotti, Giorgio Buttazzo, Marco Di Natale, and Marco Spuri. A development environment for hard real-time applications. *International Journal of Software Engineering and Knowledge Engineering*, 6(3):331–354, 1996.

[3] Paolo Ancilotti, Giorgio Buttazzo, Marco Di Natale, and Marco Spuri. Design and programming tools for time critical applications. *Real-Time Systems Journal*, 14:251–267, 1998.

[4] Gaurav Arora. Automated analysis and prediction of timing parameters in embedded real-time systems using measured data. Master's thesis, Graduate School of the University of Maryland, 1997.

[5] Gaurav Arora and David Stewart. After: A CASE tool to assist in fine-tuning of embedded real-time systems. In *IEEE Real-time Systems Symposium 96 - Work-in-progress section*, 1996.

[6] Gaurav Arora and David Stewart. A tool to assist in fine-tuning and debugging embedded real-time systems. In *Proceedings Workshop on Languages, Compilers, and Tools for Embedded Systems, LCTES 98*, 1998.

[7] N. Audsley, A. Burns, M. Richardson, and A. Wellings. Stress: a simulator for hard real-time systems. *Software - Practice and Experience*, 24(6):543–564, June 1994.

[8] Saurav Chatterjee, Kevin Bradley, Jose Madriz, James Colquist, and Jay Strosnider. SEW: A toolset for design and analysis of distributed real-time systems. In *In Proceedings RTAW 97*, 1997.

[9] Saurav Chatterjee, Kevin Bradley, Jose Madriz, and Jay Strosnider. System engineer's workbench: A toolset for the design and evaluation of distributed real-time systems. Technical Report CMUCSC-96-4, Carnegie Mellon University, 1996.

[10] Kanad Ghose, Sudhir Aggarwal, Pavel Vasek, S. Chandra, Amritansh Raghav, Abrahijt Ghosh, and David Vogel. Asserts: A toolkit for real-time software design, development, and evaluation. In *Proceedings Euromicro Workshop on Real-Time Systems*, 1997.

[11] J. Goossens and Ch. Hernalsteen. A tool for statistical analysis of hard real-time scheduling algorithms. In *Proceedings Simulation 98*, 1998.

[12] Marty Humphrey and John Stankovic. Caisarts: A tool for real-time scheduling assistance. In *Proceedings RTAS 96*, 1996.

[13] Chih Lai and H. Rebecca Callison. A framework for simulation of concurrency control policy in real-time systems. In *Proceedings IEEE Real-Time Technology and Applications Symposium*, 1996.

[14] Jane W. S. Liu, Juan-Luis Redondo, Zhong Deng, Too-Seng Tia, Riccardo Bettati, Ami Silberman, Matthew Storch, Rhan Ha, and Wei-Kuan Shih. Perts: A prototyping environment for real-time systems. Technical report, Department of Computer Science, University of Illinois, Urbana, 1993.

[15] Juan-Luis Redondo. Schedulability analyzer tool. Master's thesis, Graduate College of the University of Illinois at Urbana-Champaign, 1993.

[16] Fabrizio Sensini, Giorgio Buttazzo, and Paolo Ancilotti. GHOST: A tool for simulation and analysis of real-time scheduling algorithms. In *Proceedings IEEE Real-Time Systems Education Workshop 1997*, 1997.

[17] Ami Silberman. *RTM - Design and Implementation*. PhD thesis, Graduate College of the University of Illinois at Urbana-Champaign, 1997.

[18] Matthew Storch. *A Framework for the Simulation of Complex Real-Time Systems*. PhD thesis, Graduate College of the University of Illinois at Urbana-Champaign, 1997.

[19] Matthew Storch and Jane W.-S. Liu. DRTSS: A simulation framework for complex real-time systems. In *Proceedings Complex Systems Design and Synthesis Technology Workshop*, 1994.

[20] Matthew Storch and Jane W.-S. Liu. A simulation environment for distributed real-time systems. Technical report, Department of Computer Science, University of Illinois, Urbana, 1994.

[21] Hideyuki Tokuda and Makoto Kotera. A real-time tool set for the ARTS kernel. In *Proceeding IEEE Real-Time Systems Symposium*, 1988.

[22] Hideyuki Tokuda and Makoto Kotera. Scheduler 1-2-3: An interactive schedulability analyzer for real-time systems. In *Proceedings COMPSAC '88*, 1988.

[23] Tri-Pacific Software, Alameda, California 94501. *PERTS Version 3.0 User Manual*.

[24] Tri-Pacific Software, Alameda, California 94501. *PERTSSim Version 1.1 User Manual*.

[25] Stephane De Vroey, Joel Goossens, and Christian Hernalsteen. A generic simulator of real-time scheduling algorithms. In *Proceedings Simulation 96*, 1996.