Experiencing the multi-path schedulability analysis capabilities in MAST 1.5

Julio L. Medina, Michael González Harbour, J. Javier Gutiérrez, José Carlos Palencia, César Cuevas, Patricia López Martínez, and José M. Drake

Departamento de Ingeniería Informática y Electrónica, Universidad de Cantabria, 39005-Santander, SPAIN
{medinajl, mgh, gutierjj, palencij, cuevasce, lopezpa, drakej}@unican.es

Abstract

The MAST model includes the possibility of having multipath end-to-end flows where an event flow can be divided into several paths with control nodes like fork or branch , and paths may be joined using join or merge nodes. The analysis of these multipath end-to-end flows uses the holistic analysis technique, and is implemented in MAST 1.5, the latest version of the tool, which will be demonstrated. This paper describes the basic features it enables and summarizes the various capabilities that can be experienced with it.

1. Introduction[1]

Here we describe the basic characteristics of MAST [1], a Modeling and Analysis Suite for Real-Time Applications. MAST provides an open source set of tools that enable engineers developing real-time applications to perform various kinds of schedulability analysis on them.

Distinctive aspects of MAST are the following:

- A very rich model of the real time system is used. It is an event-driven model in which complex dependence patterns among the different tasks can be established. For example, tasks may be activated with the arrival of several events, or may generate several events at their output. This makes it ideal for analyzing or simulating real-time systems designed using languages like UML/MARTE [3][4] or similar event driven formalisms.
- The latest offset-based schedulability analysis techniques are used to enhance its results. These techniques are much less pessimistic than previous ones.
- Schedulers may be composed in a hierarchical way, which enables the formalization and analysis of mixed-critical systems (IMA, and EDF within FP for example).
- The toolset is open source and fully extensible. This has allowed other teams to provide enhanced versions [2].

1. This work has been funded in part by the Spanish Government and FEDER funds under grant TIN2011-28567-C03-02 (HI-PARTES) and by the European Commission under project FP7-611146 (CONTREX). This work reflects only the authors' views; the funding organisms are not liable for any use that may be made of the information contained herein.

When timing requirements must be met even in the worst case conditions, testing methods are insufficient because there is no guarantee that the real worst case is tested. This is why mathematical methods, called schedulability analyses, are needed to obtain these guarantees.

As inputs, the analysis needs a model of the timing behaviour of the application and its timing requirements, together with the worst-case execution times (WCETs) of the different blocks of sequential code. As a result, schedulability analysis provides the answer to the question: will a set of dynamically-scheduled concurrent tasks meet its timing requirements under all possible circumstances?

To create such model we take into account that real-time systems have a reactive architecture in which software reacts to timing and external workload events, executing specific tasks in response. These events may be periodic, perhaps triggered from a hardware timer, or aperiodic, triggered from a human operator input, a message arriving through a communications network, or a hardware interrupt generated by a sensor reading a signal from the environment. Tasks may synchronize among themselves to accomplish a coordinated result and may share resources in mutual exclusion. In distributed systems, tasks will exchange messages among them using a communications network. These messages carry information and implement a control flow by which tasks in one processor may trigger the execution of tasks in other processors. All these behaviours and platforms are captured in the MAST input model.

A simple pass/fail answer from the analysis is generally not enough for the application developer. If the system meets its timing requirements we would like to know how much space we have for growth. Similarly, if the system is not schedulable we would like to know where the timing bottlenecks are, and what parts of our system we can change to achieve schedulability. These answers can be obtained from the sensitivity analysis that MAST offers.

Real-time systems theory has developed a large number of scheduling policies together with their corresponding schedulability analysis techniques. Engineers trying to develop industrial real-time systems need tools that allow them to model their systems and apply these techniques.

MAST was created at the University of Cantabria to serve both as an engineering tool and as a research platform for developing such modelling techniques and the associated analysis techniques, focusing on distributed systems.

2. The MAST Tools

A summary view of the set of tools in the MAST environment is shown schematically in Figure 1.

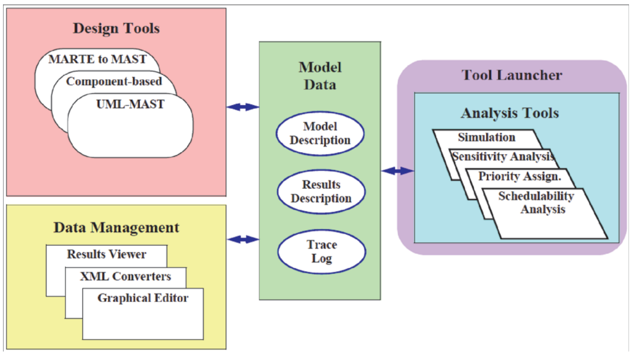

Figure 1. MAST toolset

The basic toolset encompasses the following elements:

- The model and the results are specified through an ASCII description that serves as input and output of the analysis tools. Two ASCII formats have been defined: a text special-purpose format and an XML-based format.
- Graphical editors and other tools generate the system using one of these ASCII descriptions. They can then invoke the analysis tools.
- A parser converts any of the ASCII descriptions of the system into an inner Ada data structure that is used by the tools. A module is included to convert the Ada data structure back into the chosen ASCII description.
- The XML format provides the designer with capabilities to use free standard XML tools to validate, parse, analyze, and display the model files.
- A module to generate the models from en Eclipse Ecore model is available, with tools to transform the real-time system model into the MAST XML format, and the results of the analysis tools back into an ecore model.
- A results viewer is available to view the analysis results in a convenient way.

The MAST environment integrates the following tools:

- The schedulability analysis tools perform different kinds of worst-case analysis to determine the schedulability of the system. Blocking times relative to the use of shared resources are calculated automatically.
- The priority assignment tools are able to make an automatic assignment of priorities and priority ceilings, using optimum priority assignments when available, and heuristics or optimization techniques when the optimum assignment is not available.
- The Deadline assignment tools are able to make an automatic assignment of deadlines to the individual tasks of distributed end-to-end flows given the end-to-end deadlines that they must meet.
- The sensitivity analysis tools calculate the system, processing resource, or end-to-end flow slacks, which tell the engineer which are the parts of the system that have more space for growth, or that need to be modified to make the system schedulable. This is done by repeating the analysis in a binary search algorithm in which execution times are successively increased or decreased.
- The simulation tools are able to simulate the behavior of the system to check soft timing requirements and generate temporal traces of the simulated execution.

3. Conclusion

The capabilities of the currently implemented MAST tools are condensed in the following tables. The tools with dark shading are still under development.

Table 1. Fixed-priority schedulability analysis tools

Technique	Single-Processor	Multi-Processor	Simple Transact.	Linear Transact.	Multipath Transact.
Classic Rate Monotonic	☑		☑		
Varying Priorities	☑		☑	☑	
Holistic	☑	☑	☑	☑	☑
Offset Based	☑	☑	☑	☑	

Table 2. EDF schedulability analysis tools

Technique	Single-Processor	Multi-Processor	Simple Transact.	Linear Transact.	Multipath Transact.
Single Processor	☑		☑		
EDF_Within_Priorities	☑		☑		
Holistic_Local	☑	☑	☑	☑	☑
Holistic_Global	☑	☑	☑	☑	☑
Offset Based	☑	☑	☑	☑	

To know more please visit `http://mast.unican.es`

References

[1] M. González Harbour, J.J. Gutiérrez, J.C.Palencia and J.M.Drake, MAST: Modeling and Analysis Suite for Real-Time Applications, in Proc. of the *Euromicro Conference on Real-Time Systems*, June 2001.

[2] Marco Panunzio and Tullio Vardanega, Schedulability analysis of Ravenscar systems with MAST+. http://www.artist-embedded.org/docs/Events/2011/Models_for_SA/06-MAST+Marco_Panunzio.pdf.

[3] Object Management Group, UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems, version 1.1, OMG document formal/2011-06-02, 2011.

[4] http://mast.unican.es/umlmast/marte2mast