

# MAST

## Modeling and Analysis Suite for Real-Time Applications

Grupo de Computadores y Tiempo Real. Universidad de Cantabria

<http://www.ctr.unican.es>

### MAST MODELING

The real time model is built up using primitive components defined in the MAST suite. At present, it defines the root abstract classes and a number of concrete ones derived from them. MAST is open to new concrete classes required by new analysis methods and tools.

Main abstract modeling components defined in MAST:

- **Real-Time Situation:** Execution mode of the system that provides the context for the analysis.
- **Transaction:** Sequences of activities that may be executed in response to a pattern of events.
- **Activity:** It models the execution of a piece of code or the transmission of a message.
- **External event:** It models the stimulus from external devices or timers.
- **Timing requirement:** It models a requirement imposed on the instant of finalization of an activity.
- **Processing Resource:** It models an active resource (processor or network).
- **Scheduling Server:** It models a schedulable entity hosted in a processing resource. It has a scheduling policy assigned.
- **Operation:** It models the amount of processing capacity that is required for executing a piece of code or transmitting a message.
- **Shared Resource:** It models a passive-protected resource.

The MAST model is not dependent on the methodology used in the development process.

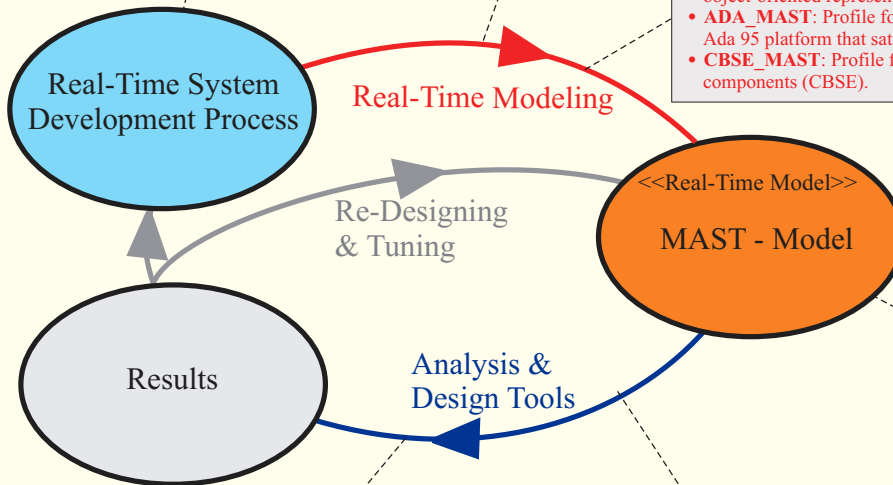
- Estimative RT-models help in the architecture design phase.
- Code-based RT-models help in the implementation phase.
- Measured-based RT-models help in validating the final product.

### PROFILES

They offer higher levels of modeling abstraction for some specific software methodologies or environments. Each profile defines a lot of new modeling components that implement concepts and artifacts introduced by the semantics of the associated software methodology.

At present three profiles have been defined:

- **UML\_MAST:** Profile for modeling real-time systems that are designed using object-oriented representation techniques and UML-CASE tools.
- **ADA\_MAST:** Profile for modeling real-time distributed applications, built with Ada 95 platform that satisfy both, Annex D and Annex E of the Ada 95 standard.
- **CBSE\_MAST:** Profile for modeling real-time systems based on software components (CBSE).



### MAST FILE

The real-time MAST model is described through a formatted ASCII description that serves as the input to the analysis tools.

In order to facilitate the integration of new tools, the MAST suite offers an interface with:

- A parser that converts the ASCII description of the system into a data structure that is accessible from the tools.
- An open repository and its interface.

### ANALYSIS AND DESIGN TOOLS

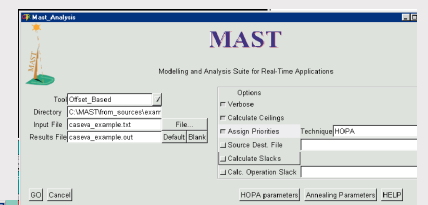
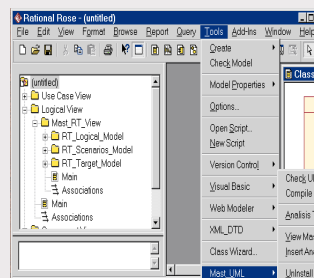
The MAST toolset is open and extensible. Its object-oriented design and the available managing interface facilitate that other design teams enhance it with new tools.

The present version is implemented for fixed priority systems. Today, MAST handles single-processor, multiprocessor and distributed systems based on different scheduling strategies and it is able to model most real-time operating systems and languages (like POSIX and Ada).

Technique	Single-Processor	Multi-Processor	Simple Trans. One priority	Simple Trans. many priorities	Linear Trans.	Nonlinear Trans.
Classic Rate Monotonic	X		X			
Varying Priorities	X		X	X		
Holistic	X	X	X	X	X	
Offset Based Unoptimized	X	X	X	X	X	
Offset Based	X	X	X	X	X	
Non-Linear	X	X	X	X	X	X

### AUTOMATED ENVIRONMENTS

**ToolDriver:** is a GUI based application that may be executed as an autonomous tool. It works directly with the ASCII files defined in MAST.



**UML\_MAST Framework:** is a "pseudo Add-In" that enhances the use of a UML graphical tool by including a modeling framework with the MAST\_RT\_View.

This framework has been implemented on Rational Rose UML CASE tool.