



Modeling and Analysis Suite for Real Time Applications (MAST 1.2)

Restrictions

By:	José María Drake	drakej@unican.es
	Michael González Harbour	mgh@unican.es
	José Javier Gutiérrez	gutierjj@unican.es
	José Carlos Palencia	palencij@unican.es

Copyright © 2000-2001 Universidad de Cantabria, SPAIN

1. Introduction

The MAST model has some general restrictions, that affect all application models, and other restrictions that affect specific analysis tools. Some of the restrictions are marked as temporary, because they will be eliminated in the near future.

In addition to the restrictions, there are some consistency checks that are made to determine if a MAST description is correct. They are based on a set of rules.

2. Consistency Checks

They are used to determine whether a MAST description is correct and if it covers the most basic general restrictions. They are based on the following set of rules, which are checked for each transaction in the system:

1. At least one external event
2. Each external event link directed at one event handler, and with an external event.
3. Each internal event link comes from an event handler
4. Each simple event handler has an input event link and an output event link
5. Each input event_handler has 2 or more input event links and an output event link
6. Each output event handler has 2 or more output event links and an input event link
7. No circular dependencies in the transaction graph
8. No isolated event links
9. No isolated event handlers
10. All activities have an operation
11. All activities have a scheduling server
12. All scheduling servers have a processing resource
13. All scheduling servers have scheduling parameters
14. All rate divisors, offset and delay event handlers are only followed by activities



3. Restrictions Related to the Use of Shared Resources

The following restrictions ensure a proper usage of shared resources through the transaction graphs. The following rules are checked for each transaction in the system:

1. All locked resources are unlocked
2. No resource is locked if it was already locked
3. No resource is unlocked if not previously locked
4. All locked resources in a segment are unlocked in that segment¹

The following rules have been added to facilitate the calculation of remote blocking terms:

5. All global shared resources (i.e., those that are shared by operations executed by different processors) use the *Immediate_Ceiling_Protocol*, for mutually-exclusive synchronization.
6. The priority ceiling of all global shared resources is such that the tasks that are involved in the calculation of the remote blocking (i.e., those tasks that have priorities above or the same as the priority ceiling) do not use shared resources.

The latter restriction is imposed because, otherwise, there would be a circular dependency among the remote blockings of the different tasks, which would make calculations very complex. But we allow some degree of remote blocking, for example by interrupt service routines, or by very high priority tasks that do not need shared resources.

When the *Immediate_Ceiling_Protocol* is used, if the tools are invoked with the “Calculate Ceilings” option set, then the optimum priority ceilings are calculated for each shared resource using this protocol. If the option is not set, then a check is made to determine that all priority ceilings are correct:

7. Consistent priority ceilings: no task locks an immediate ceiling resource having a priority lower than the ceiling.

In addition, we have a temporary restriction caused by the fact that no analysis techniques have been developed to calculate remote blocking effects in multiprocessors, when the blocking terms in one processor depend on the results of the blocking calculations in the other processors. We expect to eliminate this restriction in the near future:

8. All resources locked in a code segment are unlocked in that segment. A *segment* is defined as an uninterrupted sequence of activities that are all executed by the same scheduling server (and thus, by the same processing resource).

4. System-Kind Restrictions

The following are system restrictions that apply to some of the tools only:

- *Monoprocessor Only*: only one processing resource, which is a *Fixed_Priority_CPU*

1. A segment is a set of consecutive activities executed by the same scheduling server



- *Fixed Priorities Only*: All Scheduling Servers have Fixed Priority Parameters, all overridden parameters in operations are Fixed Priority, and all priorities are within the appropriate ranges for their processing resources
- *PCP_Or_Priority_Inheritance_Only*: all resources are PCP (i.e., Immediate_Ceiling_Resource) or Priority Inheritance resources
- *Referenced_Events_Are_External_Only*: No internal events are referenced by global timing requirements.
- *Simple_Transactions_Only*: Checks that every transaction has only one segment. A segment is a continuous sequence of activities executed by the same server.
- *Linear_Transactions_Only*: checks that every transaction only has one external event and that its event handlers are all Activities.
- *Linear_Plus_Transactions_Only*: Checks that every transaction is regular and has no *Concentrators* or *Delivery_Servers*, or *Query_Servers* or *Multicasts*, or *Barriers*
- *Multiple_Event_Transactions_Only*: Checks that every transaction is regular.
- *No_Permanent_Overridden_Priorities*: Checks that there are no operations with permanent overridden priorities.
- *No_Permanent_FP_Inside_Composite_Operations*: Checks that there are no operations with overridden priorities inside composite operations. If this would be required, the composite operation would have to be broken up into several activities.

All tools require *Fixed_Priorities_Only*, and *PCP_Or_Priority_Inheritance_Only*. The worst-case analysis tools (but not the simulation tools) require *Referenced_Events_Are_External_Only*. All tools except *Varying_Priorities* require *No_Permanent_Overridden_Priorities*. In addition, some tools require other restrictions. The following table shows the system-kind restrictions that are additionally required for each of the analysis tools:

Tool	Restrictions
Classic_RM_Analysis	Monoprocessor_Only, Simple_Transactions_Only
Varying_Priorities_Analysis	Monoprocessor_Only, Linear_Transactions_Only, No_Permanent_FP_Inside_Composite_Operations
Holistic_Analysis	Linear_Plus_Transactions_Only
Offset_Based_Unoptimized_Analysis	Linear_Plus_Transactions_Only
Offset_Based_Analysis	Linear_Plus_Transactions_Only
Multiple_Event_Analysis	Multiple_Event_Transactions_Only
Monoprocessor_Priority_Assignment	Monoprocessor_Only, Simple_Transactions_Only
Linear_HOPA	Linear_Plus_Transactions_Only
Multiple_Event_HOPA	Multiple_Event_Transactions_Only
Linear_Simulated_Annealing_Priority_Assignment	Linear_Plus_Transactions_Only
Multiple_Event_Simulated_Annealing_Priority_Assignment	Multiple_Event_Transactions_Only



Tool	Restrictions
Distributed_Simulation	Multiple_Event_Transactions_Only
Calculate_Ceilings_For_PCP_Resources	Multiple_Event_Transactions_Only
Calculate_Blocking_Times	Multiple_Event_Transactions_Only