JSimMAST user guide

César Cuevas Cuesta cuevasce@unican.es

José M. Drake Moyano drakej@unican.es

Patricia López Martínez lopezpa@unican.es

CTR -Computers and Real-Time Group

Electronics and Computers Department

University of Cantabria (Spain)

JSimMAST user guide

Contents

List of figures

1 Introduction

JSimMAST is a tool for simulating the temporal behaviour of real-time systems that have been previously modelled according to the MAST 2.0 meta-model. The simulator has been designed following the OOP paradigm and has been implemented using the Java programming language on top of the Eclipse platform.

1.1 Operational scheme

JSimMAST takes as input the MAST 2.0 model (serialized in XML format according to the W3C-schema Mast2_Model_JSimMAST.xsd) for the analyzed system and after setting some parameters (simulation profile, time units, limit time, etc.) the simulation can be executed. Depending on the selected simulation profile, this execution generates different information about the system, e.g. results data (serialized in XML format according to Mast_Result.xsd) that describe statistically the behaviour of the system or traces data (serialized in XML format according to Mast_Traces.xsd) that describe the most relevant events that have occurred in the executed scenario. In addition, relevant data are live displayed on the simulator GUI, like fulfilment of timing requirements set in the model or the utilization of passive and active resources.

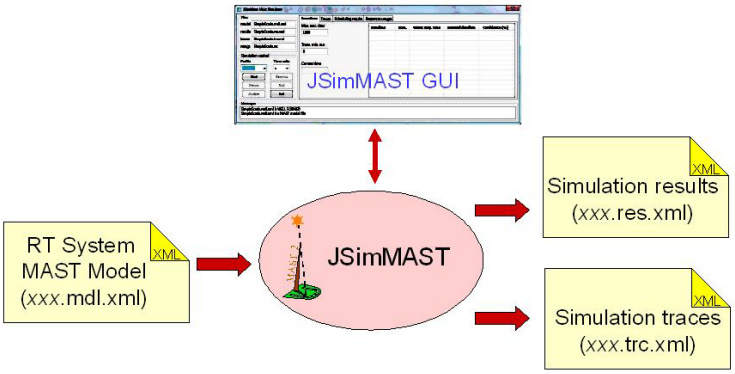

Figure 1. JSimMAST inputs and outputs.

1.2 Objectives

The JSimMAST tool has been developed with the following use cases:

- Estimating the behaviour of the modelled system in those situations in which no alternative analytic tools are available.
- Providing information about:
  - The satisfaction of the timing requirements specified in the model.
  - The worst / average / best case timing responses of end-to-end-flow transactions.
  - The percentage of resources utilization.
  - The evolution of the system, through the generation of traces.
- Validating analysis strategies under development, identifying, if any, the failure situations they present.

- Evaluating the correctness and efficiency of other available design and analysis methods and tools.

1.3 Possible users

- Designers of real-time applications, who need to estimate the behaviour of the applications under development.
- Automatic design tools, which successively process different design alternatives within a strategy aimed at finding the optimal one.

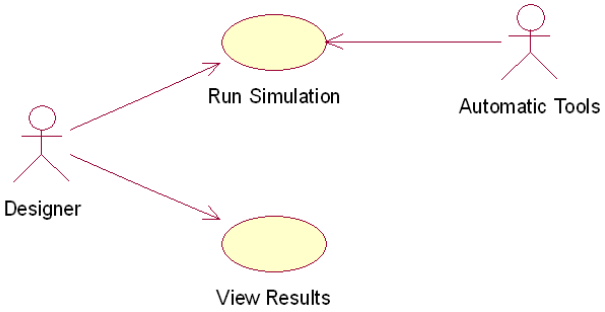

Figure 2. Possible users of JSimMAST.

2 Functionality

JSimMAST simulates the execution of systems that are described by means of models built according to the structure and semantics defined in the MAST 2.0 meta-model. The current version of the simulator is subject to some functional as well as structural restrictions on the MAST 2.0 meta-model definition, i.e. the MAST 2.0 models representing the systems to simulate must satisfy some restrictions in order to be valid for JSimMAST. These restrictions or limitations are enumerated below[1]:

2.1 MAST 2.0 elements not supported:

The following MAST 2.0 elements **are not yet supported**. Therefore, MAST 2.0 models meant to be simulated must not contain elements of these types:

- Subtypes of *Processing_Resource*
  - RTEP_Network
  - AFDX_Link
  - AFDX_Switch
- All subtypes of *Driver*
  - RTEP_Packet_Driver
  - Character_Packet_Driver
- Subtypes of *Scheduler*
  - Secondary_Scheduler
- Subtypes of *Scheduling_Policy*
  - Timetable_Driven_Policy
  - Timetable_Driven_Packet_Based_Policy
  - AFDX_Policy
- Subtypes of *Schedulable_Resource*
  - Virtual_Schedulable_Resource
  - Virtual_Communication_Channel
  - DEFAULT_COMMUNICATION_CHANNEL
- Subtypes of *Scheduling_Parameters*
  - Resource_Reservation_Params
  - Timetable_Driven_Params
  - AFDX_Virtual_Link
- Subtypes of *Synchronization_Parameter*
  - SRP_Params
- Subtypes of *Operation*
  - Composite_Message

[1] A special version of the MAST 2.0 W3C-schema has been developed for JSimMAST, including these restrictions. This W3C-schema –Mast_Model_JSimMAST.xsd– is provided in the Downloads section of the JSimMAST web page and input files for JSimMAST must comply to it.

- Subtypes of *Mutual_Exclusion-Resource*
  - SRP_Mutex
- Subtypes of *Event_Handler*
  - Queried_Branch
- Subtypes of *Observer*
  - Composite_Observer

2.2 Additional restrictions

Besides the list of non-supported MAST 2.0 elements, the following **restrictions** must also be observed.

- Class Mast_Model
  - The *System_PIP_Behaviour* attribute is not considered.
  - Only the FIFO option is implemented for the *Event_Queueing_Policy* attribute.
- Class Clock_Synchronization_Object
  Its functionality is limited because in this version only a single time scale is considered, i.e. time increases uniformly in the simulator. The processing resources that reference these objects introduce on their associated timers a random offset based on the accuracy of the Clock_Synchronization_Object instance. This is totally different to the underlying model established in the meta-model, i.e. when a processing resource does not have a Clock_Synchronization_Object assigned, its time is exactly the base time of the simulator (it is perfectly synchronized and with zero accuracy).
- Class Packet_Based_Network
  - The *Transmission_Kind* attribute only supports the value HALF_DUPLEX.
- Class Thread**:**
  - It does not admit aggregated elements of type Synchronization_Parameters**.**
- Class Step:
  - It does not admit the attribute *Hold_Schedulable_Resource*. The thread is released through the concept of segment, i.e. the thread is released when the step is followed by any other event handler different from another step executed in the same thread.

3 Simulation results

This section deals with the results data generated by JSimMAST when a simulation is executed under the appropriate simulation profile (any of the available ones apart from *traces*). Since the produced results model conforms to the MAST Results meta-model, if the reader is not familiar with it, he/she is encouraged to consult the document MAST_Results.pdf.

Similarly to the way a MAST 2.0 model aiming to feed JSimMAST must satisfy certain restrictions because the simulator does not cover the entire MAST 2.0 meta-model, the generated results model

conforms to a subset of the MAST Results meta-model, i.e. not all types of results are generated. The covered ones are presented below.

3.1 Subset of MAST Results meta-model covered by JSimMAST

3.1.1 Results related to a real-time situation:

- Name, Model_Date, Generation_Tool, Generation_Profile, Generation_Date.
- End_To_End_Flow_Results.
- Processing_Resource_Results.
- Mutual_Exclusion_Resource_Results.

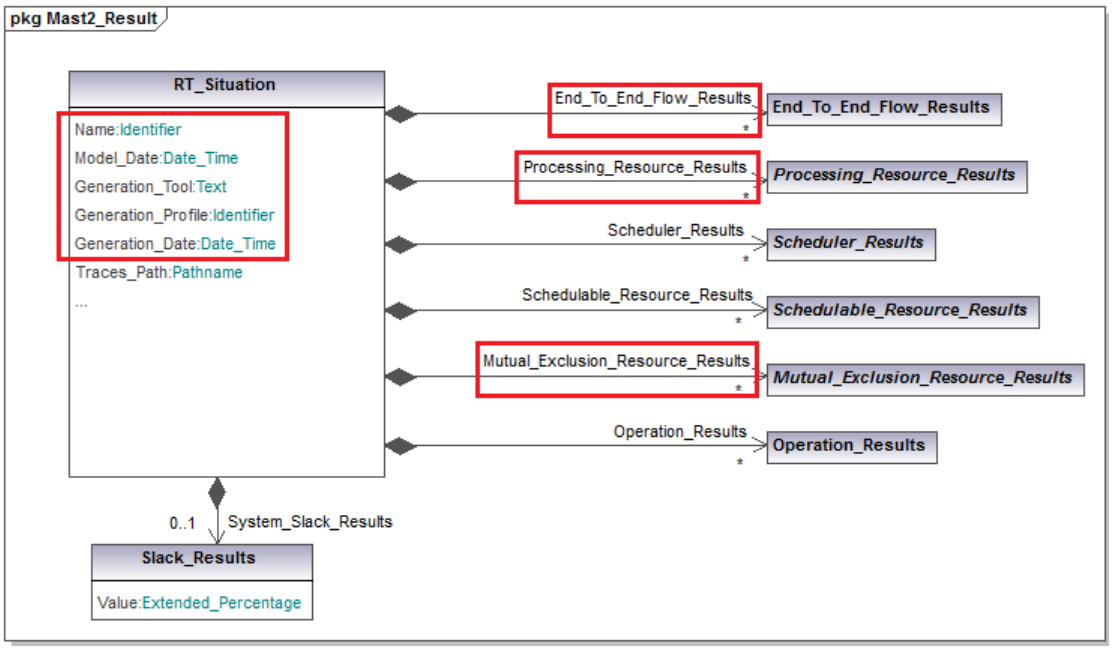

Figure 3. RT-Situation results.

3.1.2 Results related to an end-to-end flow transaction:

- Name.
- Simulation_Timing_Results:
  - Event
  - Worst / Avg / Best_Local_Response_Time
  - Worst / Avg_Blocking_Time
  - Max_Preemption_Time
  - (Max)_Suspension_Time
  - Num_Of_Queued_Activations
  - Throughputs
  - Worst / Avg / Best_Global_Response_Times
  - (Max)_Jitters
  - Local / Global_Miss_Ratios

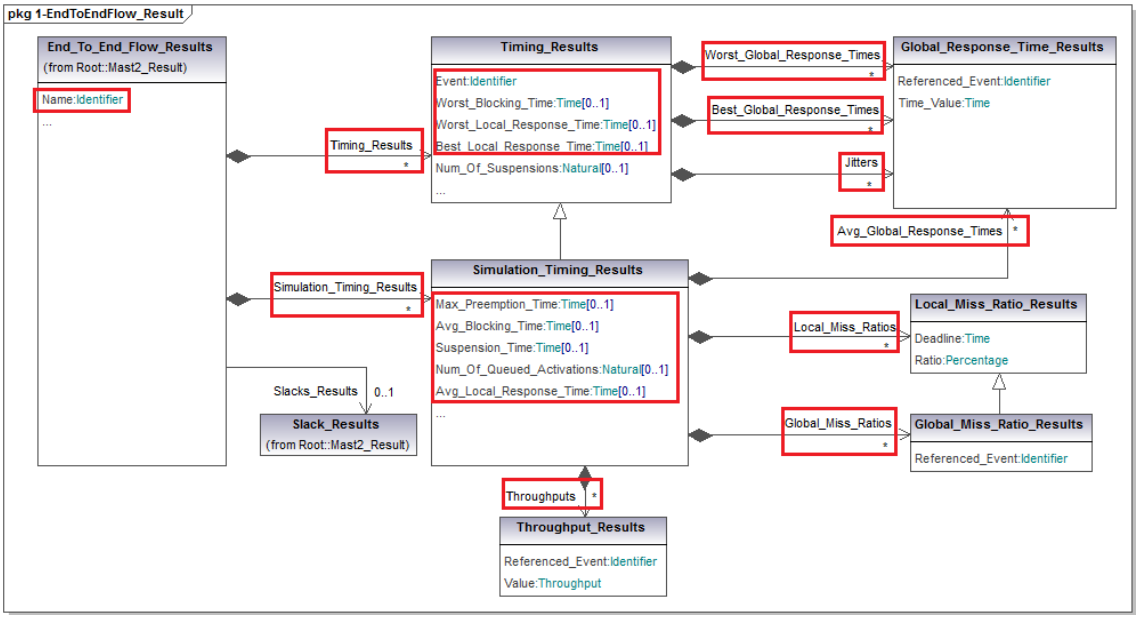

Figure 4. End_To_End_Flow results

3.1.3 Results related to a processing resource:

- Name.
- Utilization_Results (network switch case)
- Detailed_Utilization_Results (network case)
- Computing_Resource_Detailed_Utilization_Results (computing resource case)

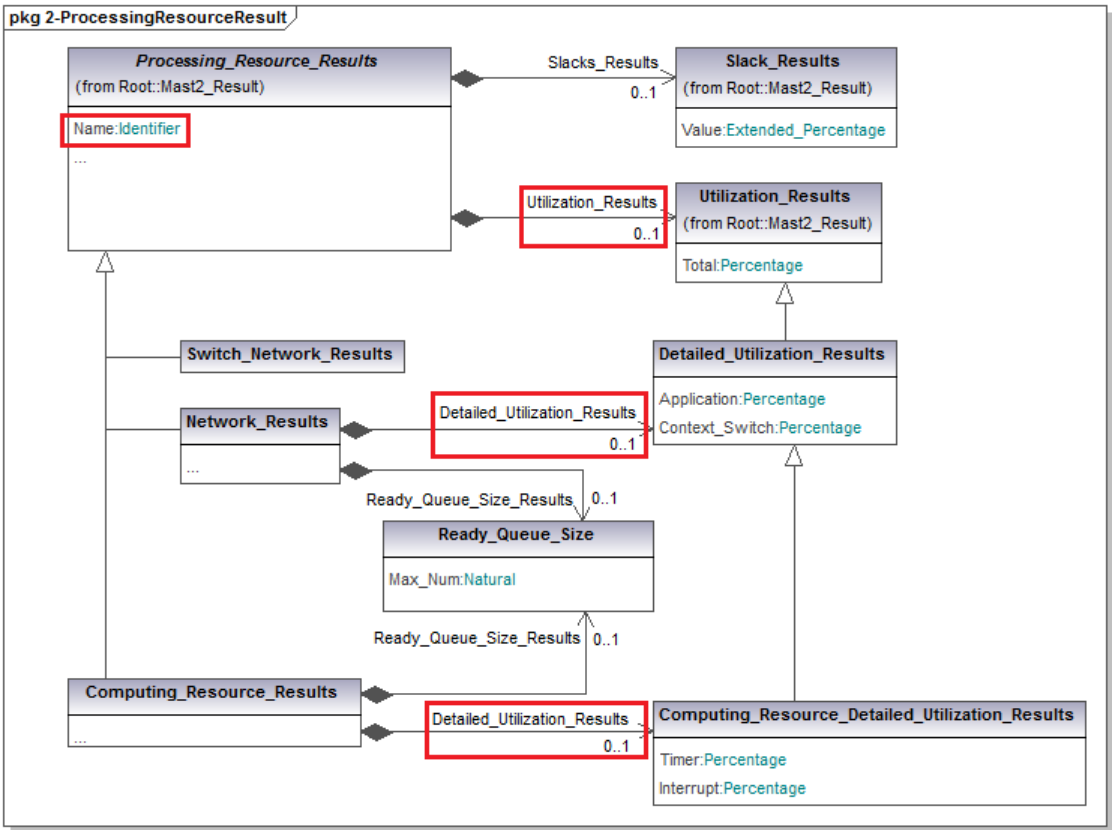

Figure 5. Processing_Resource results.

3.1.4 Results related to a mutual exclusion resource:

- Name.
- Utilization_Results.

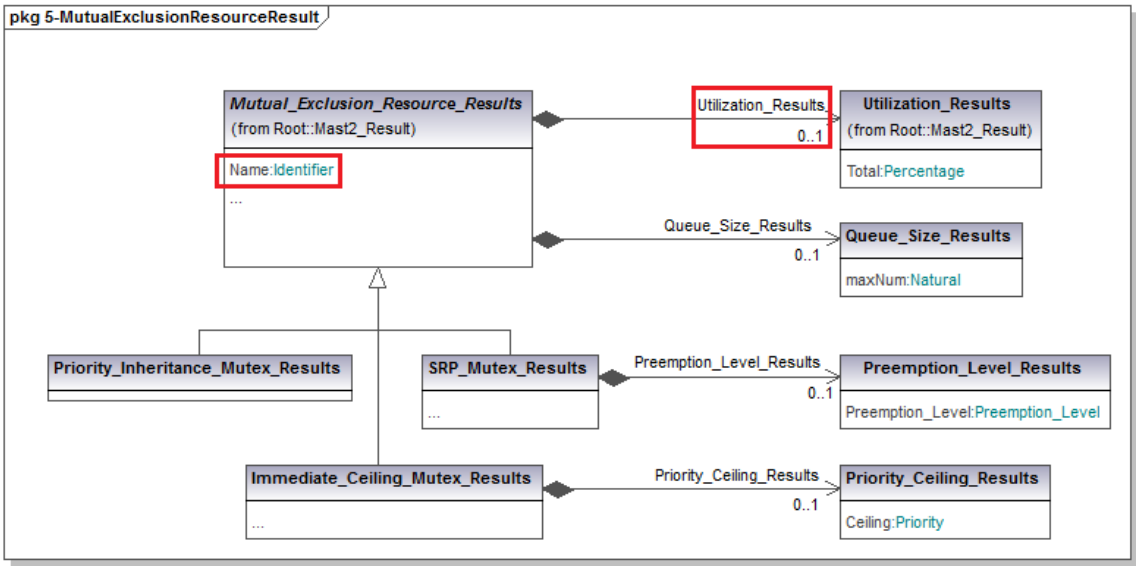

Figure 6. Mutual_Exclusion_Resource results.

4 Graphical User Interface specification (JSimMAST_GUI)

The JSimMAST simulator is launched and managed through a GUI program that allows the specification of the MAST 2.0 model that is going to be analyzed as well as of the start-up parameters and execution of the simulation. This GUI is shown in Figure 7 and the description of its components and its functionality appears below.

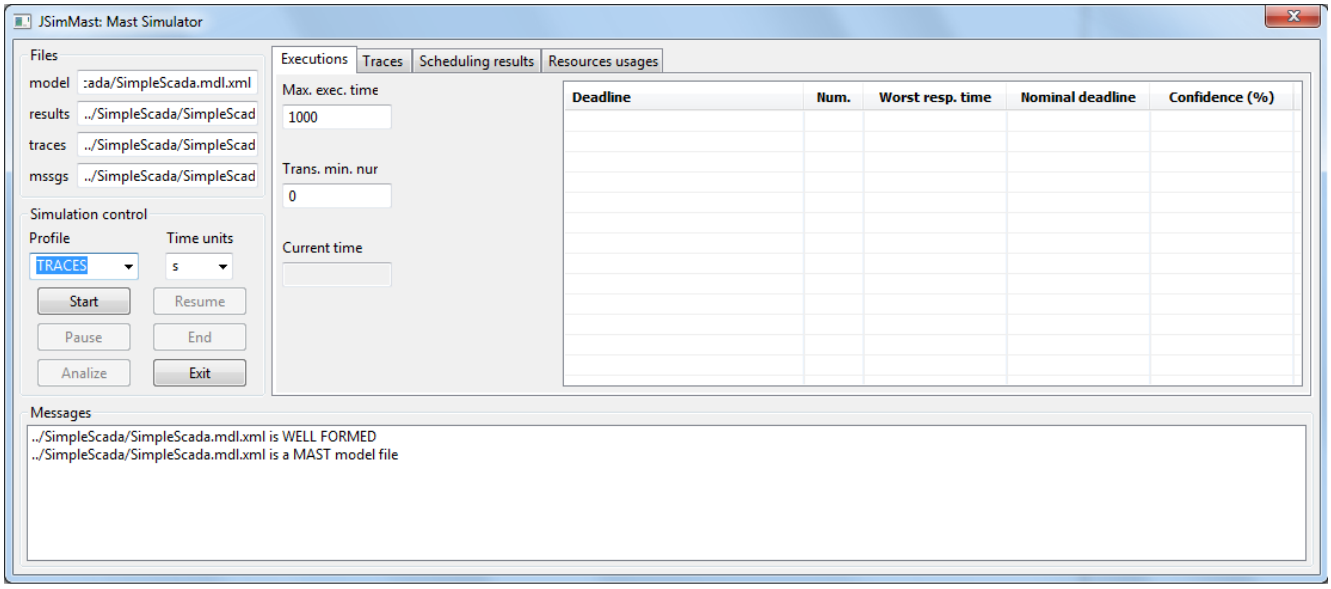

Figure 7. The JSimMAST GUI.

4.1 GUI widgets and controls

- **Files section:** It is composed of four text fields in order to hold the path of the files managed by the simulator.
  - model**:** path of the model file (*.mdl.xml).
  - results: path of the results file (*.res.xml) to be generated if appropriate.
  - traces**:** path of the traces file (*.trc.xml) to be generated if appropriate.
  - mssgs**:** path of the file for the generated messages (*.mss.xml).

  When the model file is introduced or changed, the paths of the other files are automatically initialized to the same value but with the appropriate extension. However, they can be modified afterwards.
- **Simulation control section**: it is composed of two drop-down menus (combos) and six buttons.
  - **Profile** combo: it allows choosing the simulation profile before the simulation execution starts. As will be explained in section 5, there are four profile options (*schedulability*, *performance*, *traces* and *verbose*) and the selection affects, among other things, the generated results as well as the criterion for the simulation end.

- **Time units** combo: The simulator assumes that model times are always expressed in seconds. The option selected in this combo only affects the display of the time data in the GUI.
- **Start** button: initially disabled, it is enabled when the file path of a valid model is set in the model text field and ENTER is pressed. Once it is enabled, pressing it builds the simulation model and the simulation execution starts. Besides, the Pause and End buttons are enabled in order to allow the manual suspension or finalization of the simulation.
- **Pause** button: initially disabled, it is enabled when the simulation is running, after pressing the Start button. It suspends the simulation. The Resume button is enabled, as well as the final time and trans. num. text fields.
- **End** button: initially disabled, it is enabled when the simulation is running, after pressing the Start button. When pressed, the simulation is terminated and the model and/or profile can be modified.
- **Resume** button: initially disabled, it is enabled when the simulation is suspended, after pressing the Pause button. Its function is to resume the simulation when another one has been executed and finished. The simulation continues from the step in which it was suspended without initializing the simulator state.
- **Analyze** button: a structural analysis of the model is performed. The errors and inconsistencies regarding the model itself as well as regarding the simulator capabilities are described.
- **Exit** button: it is always enabled and pressing it causes the GUI to shut down. If the simulator was running, the simulation results are lost.

When the simulation is finished or suspended, the corresponding files are generated and their values are passed to the GUI tabs.

- **Messages section:** it is a list in which all notifications generated by the simulator appear, namely:
  - [model file path] is a MAST model file – *if model file is *.mdl.xml* –
  - [model file path] is NOT a MAST model file – *if model file is not *.mdl.xml file* –
  - [model file path] is WELL-FORMED – *if model presents no inconsistencies* –
  - [model file path] is NOT WELL-FORMED – *if model presents no inconsistencies* –
  - Information about encountered inconsistencies, as for example:
    Host "LocalPro" for primary scheduler "LocalProc.Scheduler" is not found
  - SIMULATION MODEL IS BUILT – *right after pressing the start button* –
  - SIMULATION GOES ON – *right after building the simulation model* –
  - % {10, 20, 30, 40, 50, 60, 70, 80, 90, 100} reported – *during simulation* –
  - THE SIMULATION IS ENDED –*when the simulation is finalized*–
  - THE RESULTS HAVE BEEN GENERATED – *if the profile is different from traces* –
  - THE TRACES GENERATION IS ENDED – *if the profile is traces* –

4.2 Direct presentation of results in the JSimMAST GUI

The most important results from the simulation are presented in the four tabs contained in the GUI.

- **Executions:**

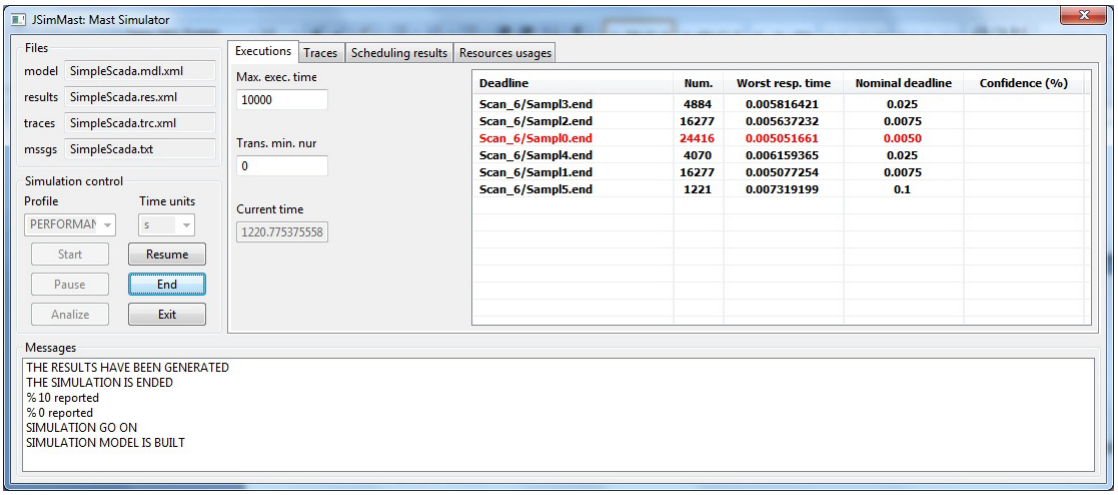

Figure 8. Executions tab.

Apart from the three text fields on the left-hand side of this tab showing the essential data about the state of the simulation execution, the tab also presents a table on the right on which the results corresponding to those events that have timing requirements (local or global) associated are shown live. The first column (*Deadline*) of this table is dedicated to the name of the event corresponding to each row, the second column (*Num.*) presents the number of times that the event has been evaluated, the third one (*Worst resp. time*) shows the largest response time evaluated while the fourth one (*Nominal deadline*) corresponds to the timing requirement that has been set. Finally, the fifth column (*Confidence (%)*) reports the % confidence level based on which the response time has been evaluated (it is not implemented in the current version).

- **Traces:**

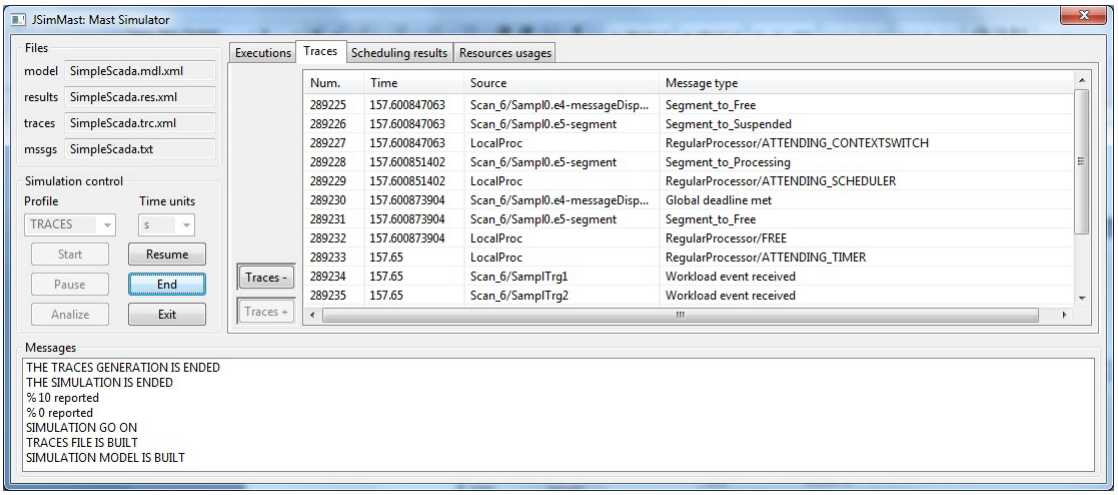

Figure 9. Traces tab.

This tab presents a table in order to show the traces that have been recorded during the simulation. Given that the traces data of a simulation can be constituted by massive information, its usage is only useful for systems simulated for a brief time. For every trace record, the table shows: the record ordinal (*Num.*), the instant in which the record has been generated (*Time*), the model element on which it has been generated (*Source*) and the message that describes the record nature (*Message type*).

- **Scheduling results:**

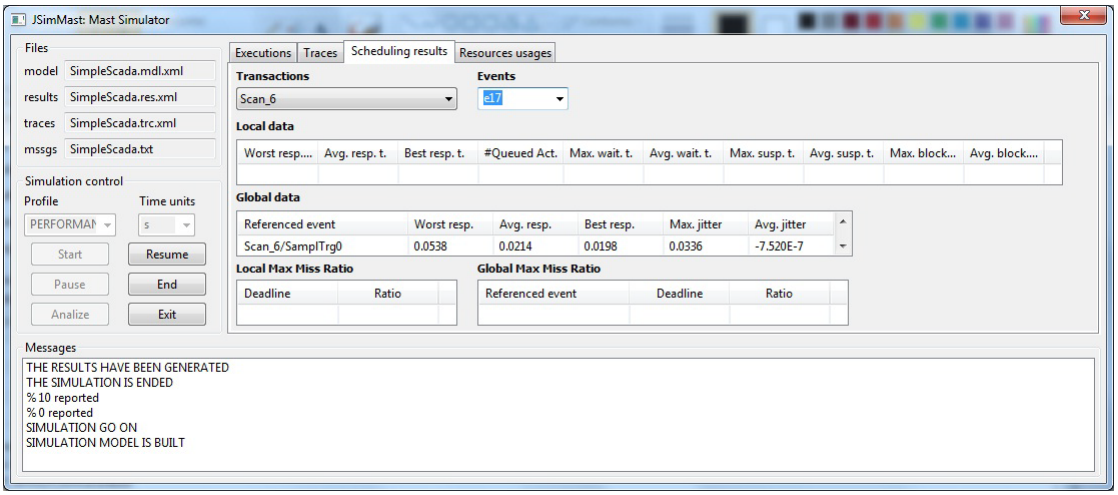

Figure 10. Scheduling results tab.

This tab shows statistical information about every event that has timing requirements associated. The event is selected through the combo *Events*, but previously it is necessary to select the transaction it belongs to, in the combo *Transactions*, which will enable the combo *Events*, making possible the selection of the event. The information shown about each event depends on the simulation profile that has been set and on the nature of the associated timing requirement.
In general, for an event it can be shown:

- **Local Data**.
  Statistical information relative to the execution in the segment for which the event is response:
  - *Worst / Avg / Best resp. t.*: the execution time, in the worst, average and best cases.
  - *#Queued Act*: the number of activations accumulated waiting for the segment execution.
  - *Max / Avg wait. t.*: the time that the activation has been waiting for the segment to be executed
  - *Max / Avg susp. t.*: the time that the segment execution has been suspended waiting for being scheduled in the corresponding processing resource, in the maximum and average cases.
  - *Max / Avg block. t.*: the time that the segment execution has been blocked waiting for a *mutex*, in the maximum and average cases.
- **Global Data.**

Statistical information relative to the time interval between the activation event of the transaction and the monitored event:

- *Worst / Avg / Best resp.*: time between the event that activates the transaction and the monitored event, in the worst, average and best cases.
- *Max. / Avg Jitter*: variability of the time between events, in the maximum and average cases.

- **Local Max Miss Ratio.**
  Statistical information about the non-satisfaction of a local timing requirement, i.e. a requirement about the execution time of the segment that ends when the monitored event happens:
  - *Deadline*: nominal value of the established requirement.
  - *Ratio*: the % of non-satisfaction.
- **Global Max Miss Ratio**:
  Statistical information about the non-satisfaction of a global timing requirement, i.e. a requirement about the time interval between the event that generates the transaction activation and the monitored event:
  - *Referenced event*: the external event that activates the transaction.
  - *Deadline*: nominal value of the established timing requirement.
  - *Ratio*: the % of non-satisfaction.

- **Resources Usages:**

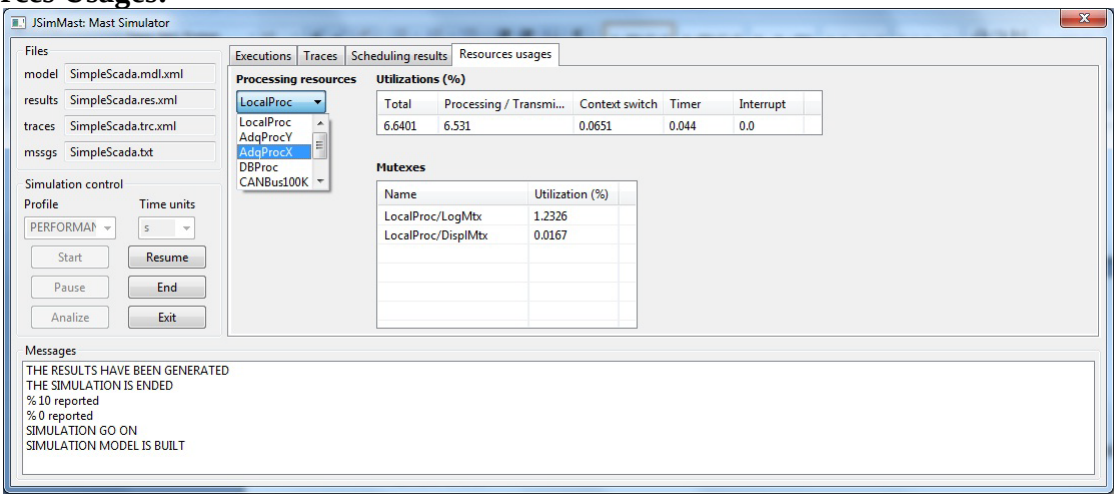

Figure 11. Resources usage tab.

This tab shows statistical information associated to the resources usage, both processing resources and mutexes. It incorporates the *Processing resources* combo that allows selecting the processing resource whose utilization will be presented in the table *Utilizations (%)*. The first column (*Total*) presents the global usage while the other columns show the decomposition in partial percentages:

- *Processing/Transmitting*: application process or message transmission.
- *Context switch*: management of application context switch changes.
- *Timer*: attention to the timer(s).

- *Interrupt*: interruption handling.

4.3 JSimMAST_GUI reactivity

The different GUI states are described, as well as the effect of the different controls exerts on them.

4.3.1 Ready to accept model file.

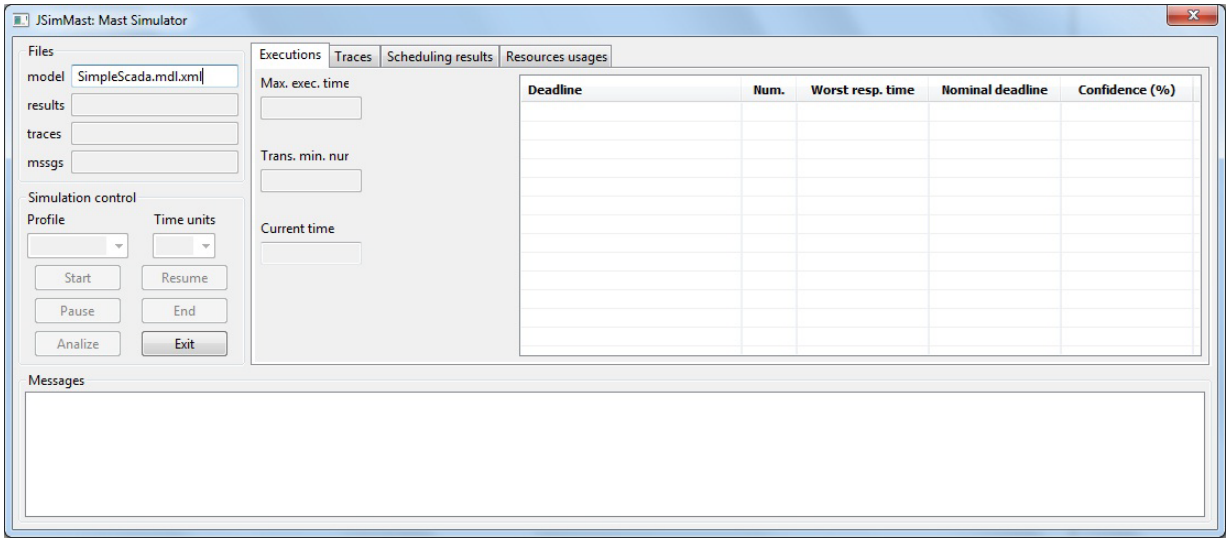

Figure 12. Ready to accept model file.

State in which the user can set the model file to simulate. This is the initial state when the GUI is launched and is featured by the following enabled/disabled state of the GUI elements.

- model file text enabled, with content and editable.
- traces / results / mssgs text fields disabled.
- Profile / Time units combos disabled.
- Analyze / Start / Resume / Pause / End buttons disabled.
- Exit button enabled.
- Max. Exec. Time / Trans. Min. Num. / Current time text fields disabled.
- Messages list disabled.

4.3.2 Ready to check model consistency.

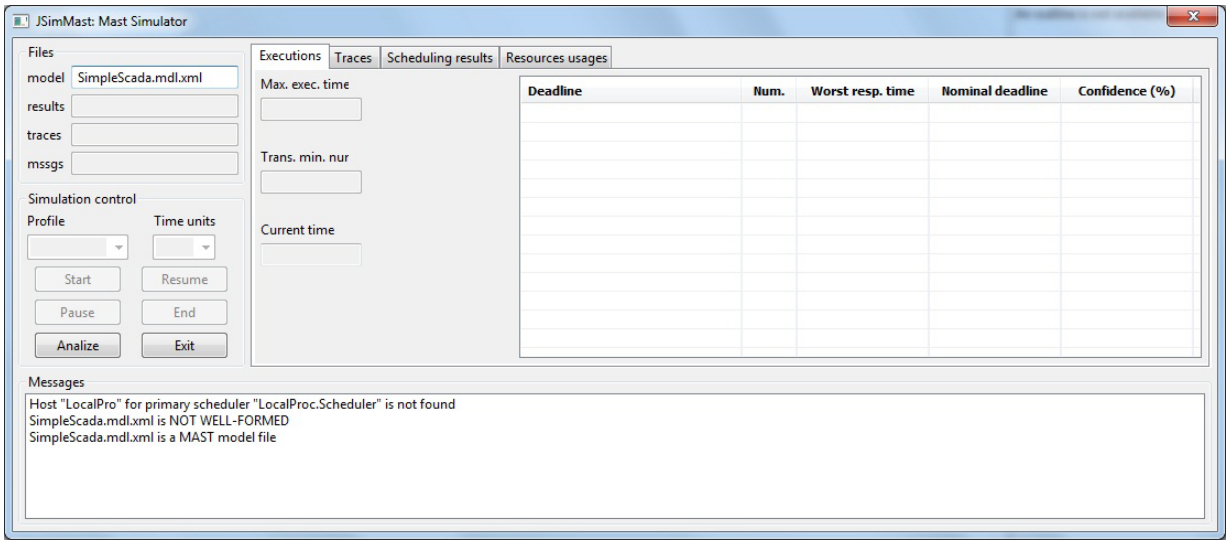

Figure 13. Ready to check model consistency.

State reached from the initial state when trying to set a file that happens to be inconsistent. Once in this state, supposing that the user has fixed those inconsistencies, he/she can attempt to validate the model again or even to enter a new model file. The enablement of the GUI elements is as follows.

- model field text enabled, with content and editable.
- traces / results / mssgs text fields disabled.
- Profile / Time units combos disabled.
- Analyze button enabled.
- Start / Resume / Pause / End buttons disabled.
- Exit button enabled.
- Max. Exec. Time / Trans. Min. Num. / Current time text fields disabled.
- Messages list enabled with content: “NON-WELL FORMED” + inconsistencies.

4.3.3 Ready to start simulation.

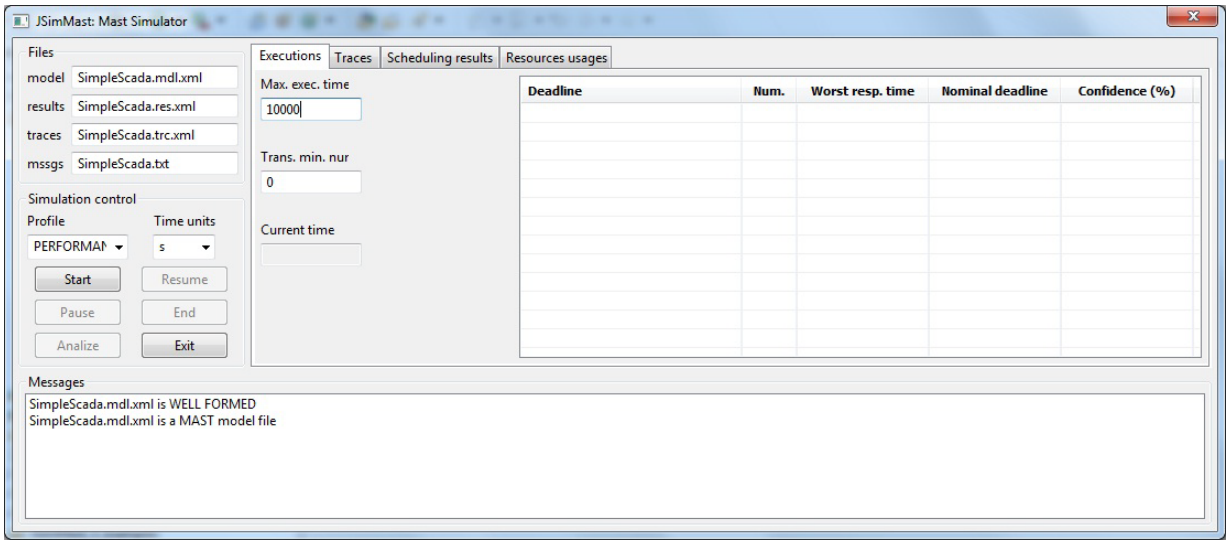

Figure 14. Ready to start simulation.

State reached from the initial state when setting a consistent model file. In this state the user can accept the default names for the files, based on the model file name or introduce different names, select the simulation profile and time units, set end conditions and finally start the simulation. It is also possible to enter a new model file and reinitialize the process as being in the initial GUI state.

- model field text enabled, with content and editable.
- traces / results / mssgs text fields enabled, with content (by default based on the content of the model field text) and editable.
- Profile / Time units combos enabled and with default content.
- Analyze button disabled (the model has already been checked as consistent).
- Start button enabled.
- Resume / Pause / End buttons disabled.
- Exit button enabled.
- Max. Exec. Time / Trans. Min. Num. text fields enabled, with default content and editable.
- Current time text field disabled and without content.
- Messages list enabled with content: “WELL FORMED”

4.3.4 Simulating.

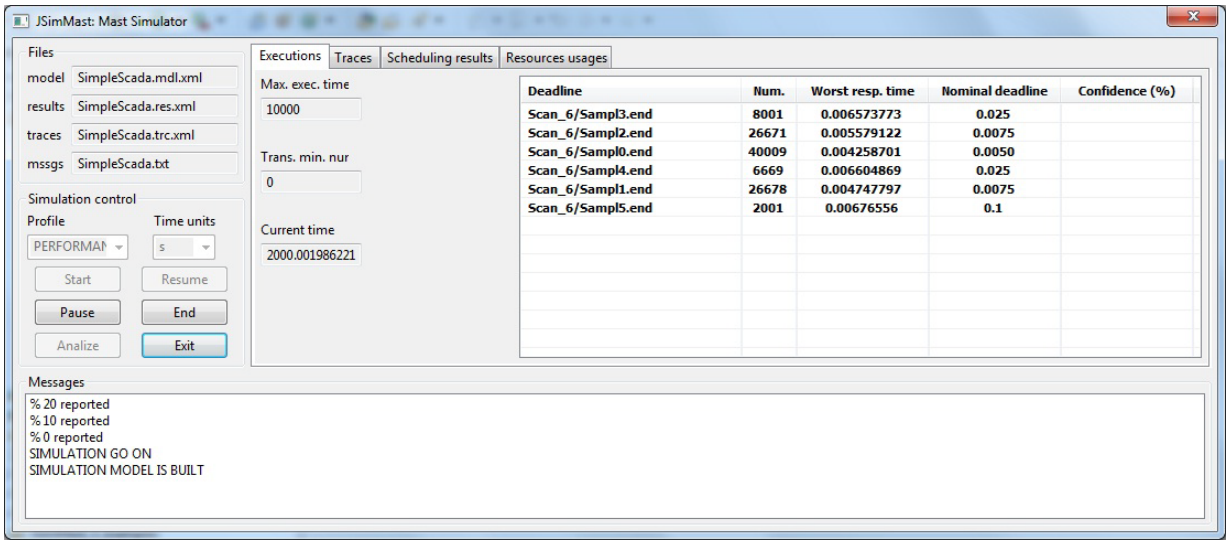

Figure 15. Simulating.

State in which a simulation is being performed. The user cannot change any parameter, he/she can only suspend or end the simulation.

- model / results / traces / mssgs field texts enabled, with content and not editable.
- Profile / Time units combos disabled and with content.
- Analyze / Start / Resume disabled.
- Pause / End / Exit buttons enabled.
- Max. Exec. Time / Trans. Min. Num. text fields enabled, with content and not editable.
- Current time text field enabled and updating.
- Messages list enabled and showing information about the simulation evolution.

4.3.5 Ready to resume simulation.

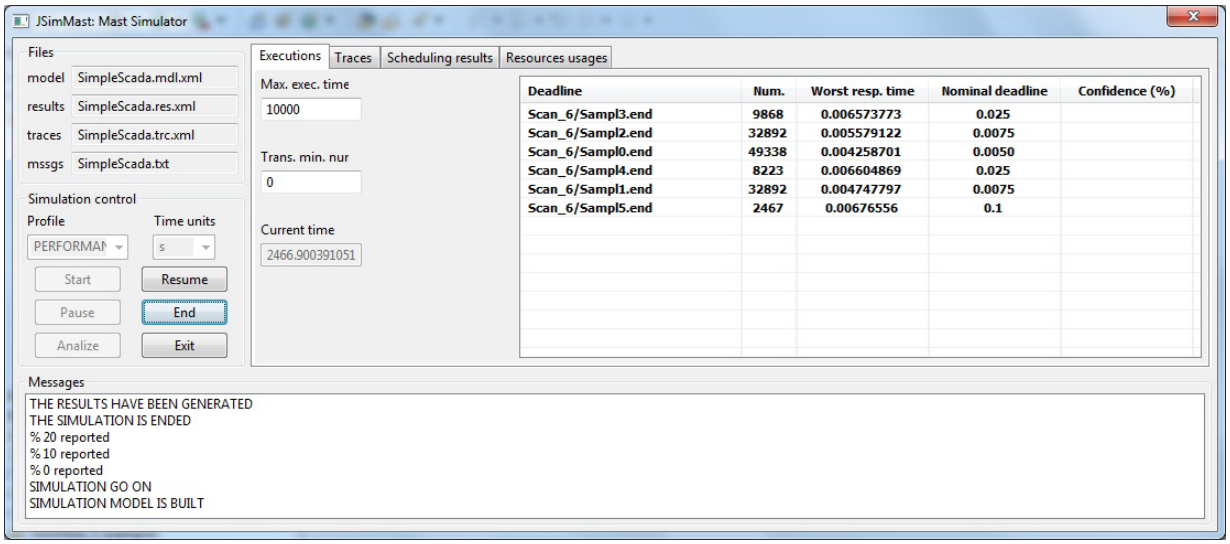

Figure 16. Ready to resume simulation.

State in which the simulation is stopped because one of the end conditions has been reached or because the user interaction and the corresponding files have been created. In order to resume the simulation, the user can change end conditions but cannot change neither names of the files nor profile or time units.

- model / results / traces / mssgs field texts enabled, with content and not editable.
- Profile / Time units combos disabled and with content.
- Analyze / Start / Pause button disabled.
- Resume / End / Exit buttons enabled.
- Max. Exec. Time / Trans. Min. Num. text enabled, with deprecated content and editable.
- Current time text field enabled, with content and not updating.
- Messages list enabled and showing information.

5 Simulation profiles

The execution of a simulation in JSimMAST can be performed under four different simulation profiles, namely *schedulability*, *performance*, *traces* and *verbose*, which are presented below. Each profile is featured describing when it should be used, according to its aim and specifying for each one the kind of results that are generated by the simulation, the different conditions that apply for terminating the simulation, the start-up parameters that must be set before executing the simulation, which informative messages are presented during the simulation and whether or not traces information is generated.

5.1 Schedulability

This profile should be used when the aim of the simulation is to analyze the system schedulability.

- Generated results:
  - Response times corresponding to the relevant events[2] of the transactions.
  - Miss-ratios of events containing timing requirements of type Max_Miss_Ratio.
- Termination criteria:
  - By timeout, when a pre-established maximum time for the simulation is elapsed.
  - All end-to-end-flow transactions are executed at least a pre-established number of times (the default value is 0).
  - Non-satisfaction of a timing requirement.
- Boot parameters:
  - Limit execution time (default value 0.1 s).
  - All segments are executed at least a pre-established number of times (the default value is 0).
  - Confidence interval of timing requirements (default value is 99%)
- Messages:
  - Information about Simulation evolution.
  - Information about the % of progress of the simulation.
- Traces: Not generated.

5.2 Performance

This profile should be chosen when the purpose of the simulation is to determine the resources utilization and identifying possible bottlenecks in the system. It is a profile very similar to the previous one, although it adds the analysis metrics corresponding to the % of resources usage.

- Generated results:
  - Response times corresponding to the relevant events of the transactions.
  - Miss-ratios of events containing timing requirements of type Max_Miss_Ratio.
  - Resources utilization and queues size. Processing resources and networks.

[2] Relevant events are those ones for which timing requirements have been specified or those ones end of transaction (sink events), independently of the specification or not of timing requirements for them.

- Termination criteria:
  - By timeout, when a pre-established maximum time for the simulation is elapsed.
  - All segments are executed at least a pre-established number of times (the default value is 0).
  - Non-satisfaction of a timing requirement.
- Boot parameters:
  - Maximum execution time (default value 0.1 s).
  - Minimum number of times that all segments must execute (default value is 0)
  - Confidence interval of timing requirements (default value is 99%)
- Messages:
  - Sequential stages in the simulation evolution.
  - Δ10% of executed simulation.
- Traces: Not generated.

5.3 Traces

This profile should be chosen when the purpose of the simulation is to generate a complete trace that allows visualizing the temporal evolution of the modelled system. The relevant events of the transactions are monitored, as well as the state changes of processing resources and synchronization resources. The strategy consists in using a continuous record of the trace based on turning the buffer over a file when it is filled.

- Generated results: none.
- Termination criteria:
  - By timeout, when a pre-established maximum time for the simulation is elapsed.
- Boot parameters:
  - Maximum execution time (default value 0.1 s).
- Messages:
  - Sequential stages in the simulation evolution.
  - Δ10% of executed simulation.
- Traces: complete traces for the total execution time.

5.4 Verbose

This profile should be used only during the debugging phase (because the simulator efficiency is considerably reduced), for checking that the simulator evolution is correct. Similar to the *traces* profile but in this case all transaction events are monitored, as well as all state changes of all resources in the system.

- Generated results:
  - Response times of all end-of-segment events.
- Termination criteria:
  - By timeout, when a pre-established maximum time for the simulation is elapsed.
- Boot parameters:
  - Maximum execution time (default value 0.1 s).

- Messages:
  - Arrival and departure of events in all event handlers.
  - State changes of the resources in the times that change.
- Traces: Not generated.