

# **UML/MARTE**

## **Methodology**

### **for Schedulability Analysis**

**April, 2015**



**Microelectronics Engineering Group**  
**TEISA Dpt. , University of Cantabria**  
**Authors: P. Peñil**

# Index:

<b>1</b>	<b>INTRODUCTION: MARTE2MAST METHODOLOGY .....</b>	<b>4</b>
<b>2</b>	<b>MODEL SPECIFICATION .....</b>	<b>4</b>
2.1	Application components .....	4
2.2	Processing Resource .....	5
2.3	Communicating Resource .....	5
2.4	Concurrency View .....	6
2.4.1	Tasks .....	6
2.4.2	Logic channels .....	6
2.5	Schedulability View .....	7
2.5.1	Analysis Context .....	7
<b>3</b>	<b>ANNEX I: METHODOLOGY STEREOTYPES .....</b>	<b>9</b>
<b>4</b>	<b>ANNEXO II: METHODOLOGY ENUMERATIONS .....</b>	<b>11</b>

## **Index of Figures:**

Figure 1 Application components and functions definition	4
Figure 2 Definition of shared application components	4
Figure 3 Definition of the execution host and the communication host in the elements of the <i>ArchitecturalView</i>	5
Figure 4 <i>GaCommChannel</i> specification	6
Figure 5 Specification of analysis context	7
Figure 6 Latency specification	8

# 1 Introduction: MARTE2MAST methodology

**Marte2Mast** is a tool that enables the extraction of schedulability analysis models created with a UML tool and their direct analysis using the tool MAST.

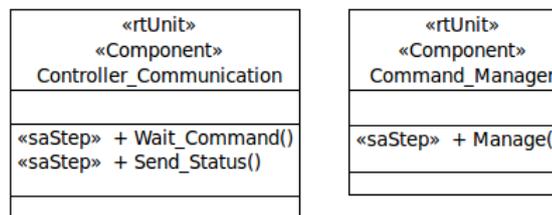
## 2 Model specification

### 2.1 Application components

The application components are modelled as *RtUnits* included in the *ApplicationView*. The entire functionality of these *RtUnits* is modelled as a set of representative functions which are characterized by timing properties in order to create analysis scenarios to be checked by the analysis tool.

These functions are modelled as UML operations owned by the *RtUnit*. Then, these functions are specified by the MARTE stereotyped <<SaStep>>. The attribute *execTime* defines the worst and the best execution time of the function:

- *execTime*=(worst=200, best=80, unit=ms)



**Figure 1 Application components and functions definition**

In addition to that, some application components have functions that are shared by other application and, thus, how these functions are accessed should be specified. An *RtUnit* instance defined in the application structure description can be specified by the MARTE stereotype <<SaSharedResource>>. This element should be used for shared resources in which the access to something has to be protected. The attributes used are:

- *protectKind*: ProtectProtocolKind.
- *ceiling*: NFP\_Integer



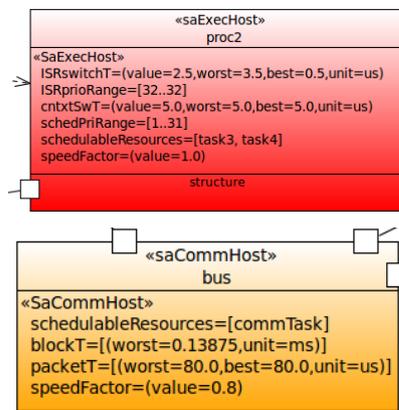
**Figure 2 Definition of shared application components**

## 2.2 Processing Resource

The processing resources considered in the methodology are the *HwProcessor* instances included in the platform description defined in the *ArchitecturalView*.

In order to enable the schedulability analysis the processor instances should have additional properties. These properties are included in the MARTE stereotype <<SaExecHost>>, which should be applied to these *HwProcessor* instances. The attributes that captured these new required properties are:

- **ISRswitch**: NFP\_Duration. *Context switch time of ISR (Interrupt Service Routines) interruptions.*
- **ISRprioRange**: IntegerInterval. *Range of ISR priorities supported by the platform.*
- **cntxtSwT**: NFP\_Duration. *Context switch time.*
- **schedPrioRange**: IntegerInterval. *The range of priorities offered by this processor.*
- **isPreemptible**: NFP\_Boolean.
- **schedPolicy**: SchedPolicyKind.
- **schedulableResources**: SchedulableResource. *Set of schedulable resources executed in this host.*
- **speedFactor**: NFP\_Real. *This number gives a linear approximation of the relative speed of the unit as compared to the reference one. The reference processing resource is determined as one with speedFactor equal to 1.0.*



**Figure 3 Definition of the execution host and the communication host in the elements of the *ArchitecturalView***

Figure 3 shows the specification of the processing host (the processor “proc2”) and the communication host (the bus “bus”). The figure shows the attributes aforementioned and how they have to be annotated.

## 2.3 Communicating Resource

The communicating resources considered in the methodology are the *HwBus* instances included in the platform description defined in the *ArchitecturalView*.

In order to enable the MAST analysis the bus instances should have additional properties. These properties are specified with the MARTE stereotype

<<SaCommHost>> (Figure 3), which should be applied to the *HwBus* component instances. The attributes that captured these new required properties are:

- schedPolicy: SchedPolicyKind.
- schedulableResource: SchedulableResources. *Defined the logic channels between schedulable resources managed by the communicating resource. In the methodology, these elements should be GaCommChannels.*
- elementSize: NFP\_Integer. *Size of the data transmitted.*
- transmMode: TransmModeKind
- blockT: NFP\_Duration. *Blocking time.*
  - (worst=value, best=value, unit= TimeUnit)
- packetT: NFP\_Duration. *Packet transmission time.*
  - (worst=value, best=value, unit= TimeUnit)
- capacity: NFP\_DataTxRate. *It defines the throughput.*
- speedFactor: NFP\_Real.

## 2.4 Concurrency View

The concurrent elements considered in the methodology are *SchedulableResource* components included in a new model view, the <<ConcurrencyView>>.

### 2.4.1 Tasks

This view includes the concurrent task to be scheduled. The tasks are modeled as UML components specified by the MARTE stereotype <<SchedulableResource>>.

The attributes considered of *SchedulableResource* are:

- schedParams: SchedParameters. *It defines priority*
  - fp(priority=(value=intValue, source=sourceType))
- host: Scheduler. *It defines the processing resource where the schedulable resource is executed.*

### 2.4.2 Logic channels

The logic channels define communicating elements for the data exchange between concurrent elements, specifically modelled as *SchedulableResources*. These channels are modelled as UML component specified by the MARTE stereotype <<GaCommChannel>>. The property considered is the priority of the channel that is captured in the *GaCommChannel* attribute *schedParams*. The value is annotated as:

- fp(priority=(value=intValue, source=sourceType))



**Figure 4 GaCommChannel specification**

Then, these logic channels are associated to the communicating resource, explained in the section “Communicating Resources”.

In the *host* attribute, the specific instance specified as *SaCommHost* should be included.

## 2.5 Schedulability View

The previous new stereotypes can be applied to elements included in the model with additional information required for the MAST analysis. However, new, different elements are needed for capturing a complete analysis model. These new modelling elements should be included in the new model view <<SchedulabilityView>>.

### 2.5.1 Analysis Context

An analysis context is a modelling element where specific analysis scenarios are captured.

In this methodology, the MARTE stereotype for defining these scenarios is <<SaAnalysisContext>>. In the attribute *context*, there are annotated some options for the tool:

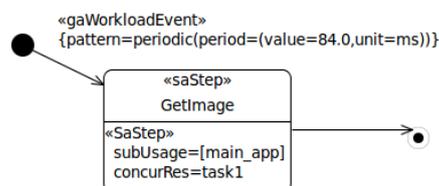
- (tool=mast,options=(invoke=true, recoverResults=true, overwriteResults=false, overwriteOutputModel=false,modelID="")))

In the *SaAnalysisContext* component, a set of UML Activities should be included. These activities are specified by the MARTE stereotype <<SaEndtoEndFlow>> in order to define the different analysis scenarios.

Each of these *SaEndtoEndFlow* analysis scenarios is described in an Activity diagram associated to the UML activity.

The activity diagram should be created as:

- An initial node. This element is specified by the MARTE stereotype <<GaWorkLoadEvent>>. In the *pattern* attribute, the characteristics of event that trigger the scenario are annotated (Figure 5). For instance:
  - sporadic (minInterarrival=(value=1, unit=s))
  - periodic (period=(value=500, unit=ms))
- A sequence of UML opaque actions connected by UML control flows. Each action is specified by the MARTE stereotype <<SaStep>>. The attributes of *SaStep* are:
  - *concurRes*: *SchedulabilityResource*. Identifies the *SchedulableResource* instance defined in the *ConcurrencyView* that executes this step.
  - *subUsage*: *ResourceUsage*. This attribute denotes the ordered set of functions defined in the *RtUnits* (included in the *ApplicationView*) that are executed in this stage.
- A final node to close the activity diagram



**Figure 5 Specification of analysis context**

In addition to that, each *SaEndtoEndFlow* activity has a UML constraint which is specified by the MARTE stereotype <<GaLatencyObs>> using the attribute *latency*

(value=Value, unit=unitTime). Then, this *GaLatencyObs* constraint is associated to the *SaEndtoEndFlow* activity in the attribute Precondition of the Activity.

```
«gaLatencyObs»
«GaLatencyObs»
latency=[(value=100.0,unit=ms)]
global_timing_requirements
{}
```

**Figure 6 Latency specification**

### 3 Annex I: Methodology Stereotypes

<b>Stereotype</b>	<b>Attributes</b>	<b>Profile</b>
SchedulabilityView		ESSYN
ConcurrencyView		ESSYN
SaStep	execTime: NFP_Time [1] concurRes: SchedulabilityResource [1] subUsage: ResourceUsage [*]	MARTE
SaSharedResource	protectKind: ProtectProtocolKind [1] ceiling: NFP_Integer [0...1]	MARTE
SchedulableResource	schedParams: SchedParameters [1] host: Scheduler [1]	
GaCommChannel	schedParams: SchedParameters [1]	MARTE
SaExecHost	ISRswitch: NFP_Duration [0...1] ISRprioRange: IntegerInterval [0...1] cntxtSwt: NFP_Duration [0...1] schedPrioRange: IntegerInterval [1] isPreemptible: NFP_Boolean [1] schedPolicy: SchedPolicyKind [1] schedulableResources: schedulableResource [*] speedFactor: NFP_Real [1]	MARTE
SaCommHost	schedPolicy: schedPolicyKind [1] schedulableResource: SchedulableResource [*] elementSize: NFP_Integer [0...1] transmMode: TransmModeKind [1] blockT: NFP_Duration [0...1]	MARTE

	packetT: NFP_Duration. [0...1] capacity: NFP_DataTxRate [1] speedFactor: NFP_Real [1]	
SaAnalysisContext	context: String [1]	MARTE
SaEndtoEndFlow		MARTE
GaWorkLoadEvent	pattern: ArrivalPattern [1]	MARTE
GaLatencyObs	latency: NFP_Time	MARTE

## 4 Annexo II: Methodology Enumerations

<b>Enumeration</b>	<b>Values</b>	<b>Profile</b>
ProtectProtocolKind	FIFO NoPreemption PriorityCeiling PriorityInheritance StackBased Undef Other	MARTE
TransmModeKind	Simplex half-duplex full-duplex	MARTE
SchedPolicyKind	EarliestDeadlineFirst FIFO FixedPriority LeastLaxityFirst RoundRobin TimeTableDriven Undef Other	MARTE