THALES

Research & Technology

Report / Technical Document

MARTE TO RAPIDRMA

Introduction to the Tri-Pacific RapidRMA gateway

DEPARTMENT / SERVICE	DOCUMENT NUMBER	PAGE
STI / LSE	61565273 305 6	1/11
	-	REVISION

DOCUMENT CONTROL

	- the : 06/06/08	A the:	B the:	C the:	D the:
Written by Signature	S. Demathieu				
Approved par Signature					

Revision index	Modifications
A	
B	
C	
D	

CONTENTS

1. PURPOSE

The UML profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) has been adopted by the OMG in June 2007. This initiative meets the needs of several Thales divisions, which develop real-time and embedded systems. Thales has been an active contributor to the MARTE submission through the ProMARTE consortium. It is also an expected end user of the profile when the latter will be implemented in commercial tools.

Thales Research & Technology has started a case study and a demonstrator related to MARTE. In that context, we needed to implement a gateway between MARTE models for IBM Rational Software Architect [MARTE for RSA] and the scheduling analysis tool TriPacific RapidRMA. These developments allowed us to validate the concepts introduced in MARTE. It has also provided us useful feedback on the limitation of the tools, with respect to the UML 2.1 and MARTE specifications.

This document introduces the distribution (software and related documentation) that results of this case study. It details the content of the distribution, provides an installation procedure and gives an overview of the available features.

2. DOCUMENTS

2.1. MANDATORY

[N/A] Not applicable

2.2. REFERENCE

[ATL]	Eclipse M2M project web site (http://www.eclipse.org/m2m/atl/download/archive.php, 2007-02-16 build)
[MARTE]	UML profile for MARTE, Beta 1 (http://www.omg.org/cgi-bin/doc?ptc/2007-08-04)
[MARTE for RSA]	MARTE for IBM Rational Software Architect (http://www.omgmarte.org/Tools.htm)
[OMG MARTE]	OMG MARTE official web site (http://www.omgmarte.org)
[RapidRMA]	TriPacific RapidRMA web site (http://www.tripac.com)

3. ACKNOWLEDGMENT

The UML profile for MARTE [MARTE] is the result of a collective effort performed by the ProMARTE partners. This consortium gathers end-users, tool vendors and academics willing to standardize a UML profile that addresses the real-time and embedded domain. More information on the specification and related work is available on the official web site [OMG MARTE].

The following persons contributed to this software and documentation: Sébastien Demathieu, Eric Maes, Laurent Rioux.

4. CONTENT OF THIS DISTRIBUTION

This software is delivered as a set of Eclipse plug-ins that can be deployed in the Eclipse directory (see next section for more details). The software is organised as following:

- *eclipse* directory
    - *plugins* directory
        - *com.thalesgroup.atl.marte.rsa*: Utility functions for ATL transformations with MARTE
        - *com.thalesgroup.atl*: Generic utility functions for ATL
        - *com.thalesgroup.java.log*: Logging tool used by the transformation plug-in
        - *com.thalesgroup.marte.menu:* Miscellaneous GUI
        - *com.thalesgroup.marte.rapidrma*: EMF implementation of the RapidRMA metamodel
        - com.thalesgroup.marte.rapidrma.edit*:* EMF.edit adapters for the RapidRMA metamodel
        - com.thalesgroup.marte.rapidrma.editor*:* EMF-generated editor
        - *com.thalesgroup.rapidrma.export:* MARTE to RapidRMA transformation

Note: source code and related documentation are located in the plug-in archive files, along with the binaries and resource files.

5. INSTALLATION PROCEDURE

5.1. Pre-requisites

The software included in this distribution has been developed and tested with IBM Rational Software Architect version 7.0.0 and TriPacific RapidRMA 5.3.3. No additional testing has been done to ensure its compatibility with other versions of these products, or related products.

The transformation plug-in that processes UML MARTE models into RapidRMA models depends on third-party software that needs to be installed in the IBM RSA implementation before the MARTE to RapidRMA plug-ins are deployed:

- MARTE for RSA 1.0.0: implementation of the MARTE profile for RSA [MARTE for RSA]
- Eclipse ATL 2006: transformation tool available in the Eclipse M2M project [ATL]

5.2. Installation procedure

Assuming that MARTE for RSA is already installed, an Eclipse extension called *MARTE_RSA* should be already created. This extension can be used to host the MARTE to RapidRMA plug-ins. It requires to know the location of this extension on the file system before moving forward.

The installation procedure is the following:

- Unzip the archive and copy the *eclipse* directory under the *MARTE_RSA* extension directory. Accept to override exiting files if required.
- Start Rational Software Architect.

- Go to Help > Software Update > Manage Configuration
- Disable and then enable the MARTE for RSA feature

6. FEATURE OVERVIEW [1]

6.1. Create a MARTE model

This feature allows one to create UML models with the all the MARTE sub-profiles that are automatically applied. A dedicated wizard has been developed for that purpose. One may access this wizard through the *File* > *New* > *Others* menu. It is then possible to select *New MARTE Model* in the *Modeling* Section (For more information on the implementation of MARTE for RSA, please have a look at [MARTE for RSA])

A new MARTE model can be created after the model name and location are entered by the user. All the MARTE sub-profiles required for scheduling analysis will be included.

6.2. Define an analysis context

The concept of “analysis context” in the quantitative analysis framework of MARTE is a wrapper for all the information used in a model-based analysis. It defines reusable building blocks that can be instantiated with specific parameters. In the current implementation, the GaAnalysisContext stereotype is applied to a structured class, which parts represent the workload behavior, the execution resources as well as the shared resources.

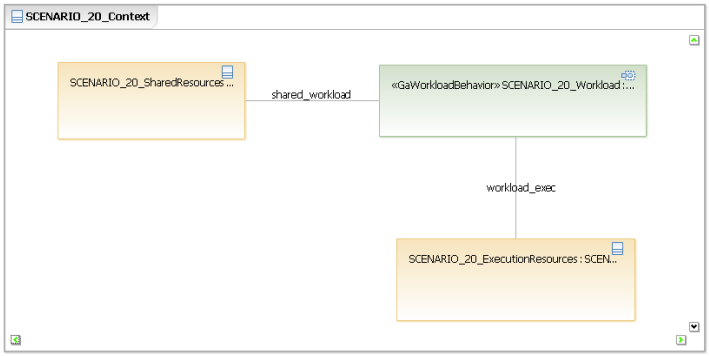

Figure 1: Analysis context diagram

6.3. Specify a workload

To analyse the timing behaviour of a software, it requires to identify the different internal and external events that stimulate the system. These events, identified by the GaWorkloadEvent stereotype, may follow different arrival patterns. They can be periodic (in this case a period should be known) or aperiodic (in this case they should follow an arrival law).

Activity diagrams are the preferred way here to specify a workload in the sense that they can easily express concurrency between the flows that are triggered by the different events. Each swim lane of the diagram defines an AcceptEvent action that is triggered by a TimeEvent. The time event carries the GaWorkloadEvent stereotype with a specific arrival pattern. The arrival pattern is a tuple type which value can be specified as a MARTE VSL textual expression. The syntax for valid VSL expression can be found in the MARTE

[1] Related examples are provided in this distribution in the *MARTE_RapidRMA_Example.emx* file

specifications. When the action is completed, it activates a CallBehavior action, identified by a GaScenario stereotype, which describes how resources are used (as described later.)

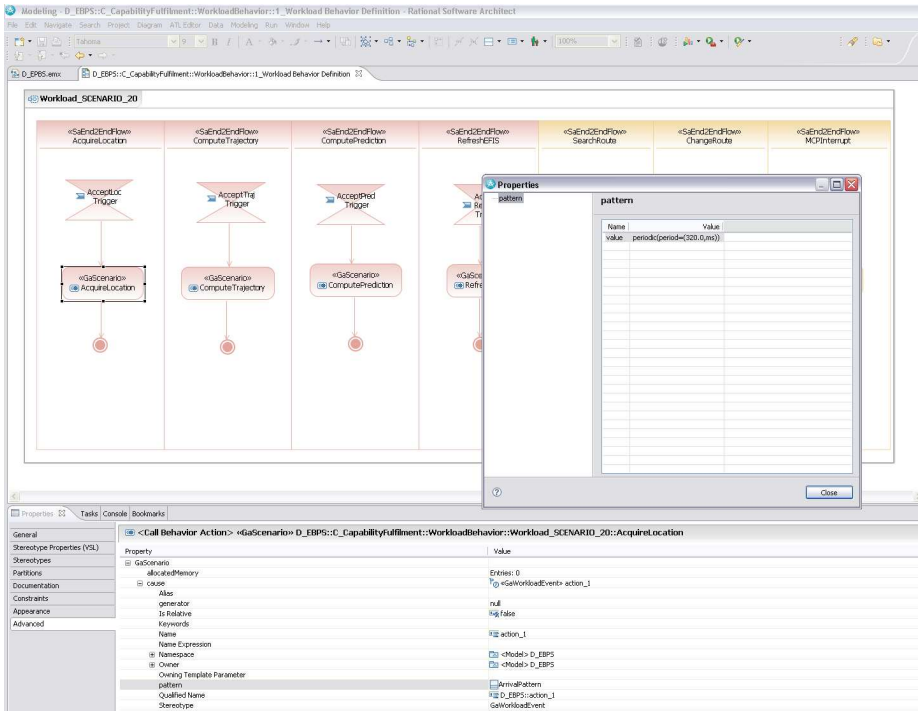

Figure 2: Identifying the workload events and their arrival patterns

In a similar way, it is possible to specify the deadline for each flow that has been defined as an activity swim lane. It requires to apply the SaEnd2EndFlow on a swim lane and to edit the end2endD stereotype attribute. The type of this attribute is a duration which value can be specified as a VSL textual expression as well. In the example, the deadline for acquiring the location information equals to 320 ms.

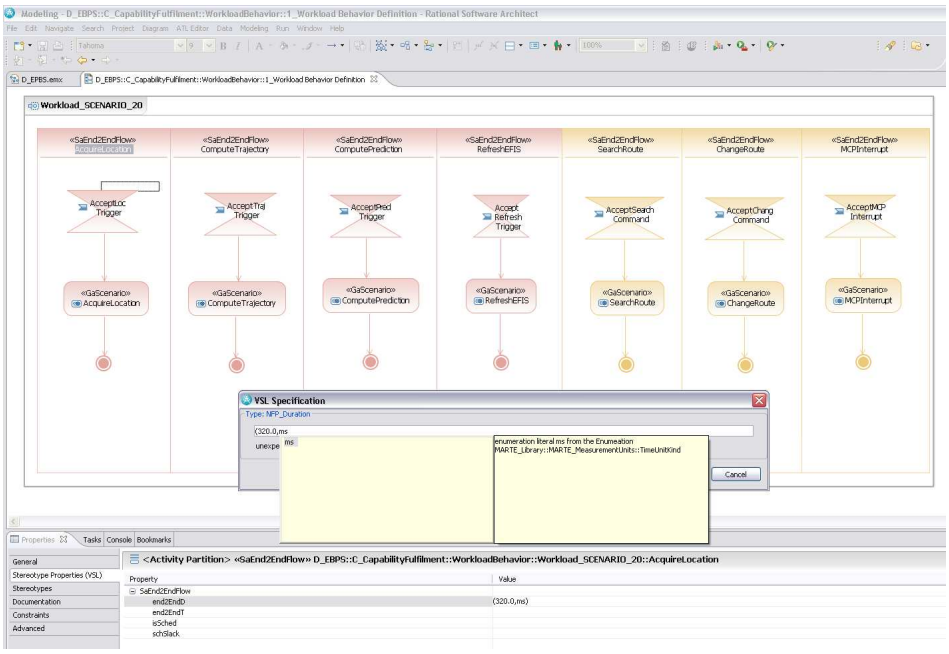

Figure 3: Specifying the deadline for end-to-end flows related to each events

6.4. Identify execution resources

When all the events are identified, one can define the execution resources that will handle these events. In the context of a scheduling analysis, we are going to deal with schedulable resources (SchedulableResource stereotype) that are managed by a scheduler. The later is own by an execution host (SaExecHost stereotype). In a multi-processor / distributed system, there would be several hosts.

Each schedulable resource defines schedulable parameters as a VSL expression. In the context of our gateway, the parameters can be only “fp” (for fixed-priority periodic tasks) or “server” (for fixed-priority periodic servers that handle non-periodic events.) Being a complex tuple type, it is possible to indicate children parameters, such as the priority to be computed, the server budget and/or period.

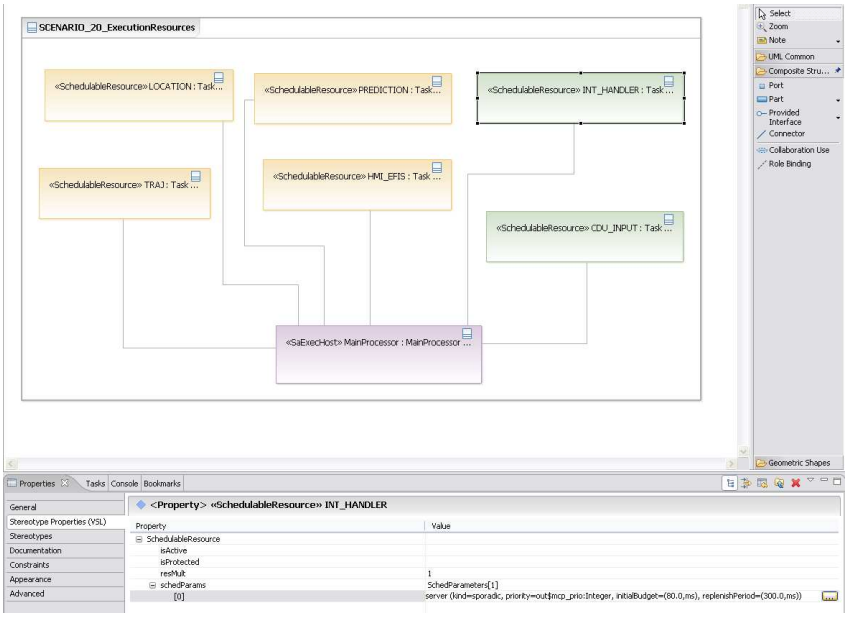

Figure 4: Identifying the execution resources and their related hosts

6.5. Identify shared resources

Shared resources need to be identified as well before the analysis is launched, as these elements have an impact on the system scheduling. The SaSharedResource stereotype is used for that purpose.

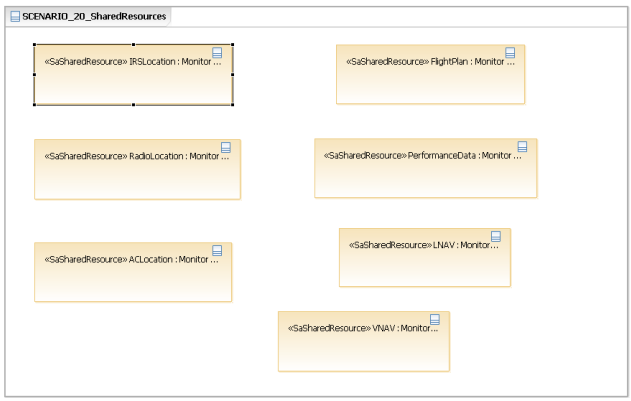

Figure 5: Identifying the shared resources

For each shared resource, it is possible to specify the kind of protection mechanism is that used to deal with concurrent accesses (e.g. priority inheritance, priority ceiling protocol…)

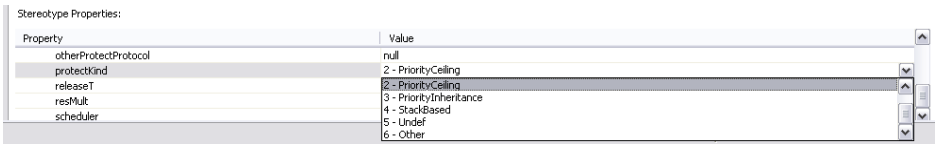

Figure 6: Specifying the protection mechanism for a shared resource

6.6. Describe resource usage scenarios

Once all the events have been specified and all the resources have been identified, it is then possible to describe how the system will use the resources in response to these events. This description is done with sequence diagrams using specific MARTE stereotypes. The scenarios are bound to events thanks to the CallBehavior actions created in the previous activity diagram.

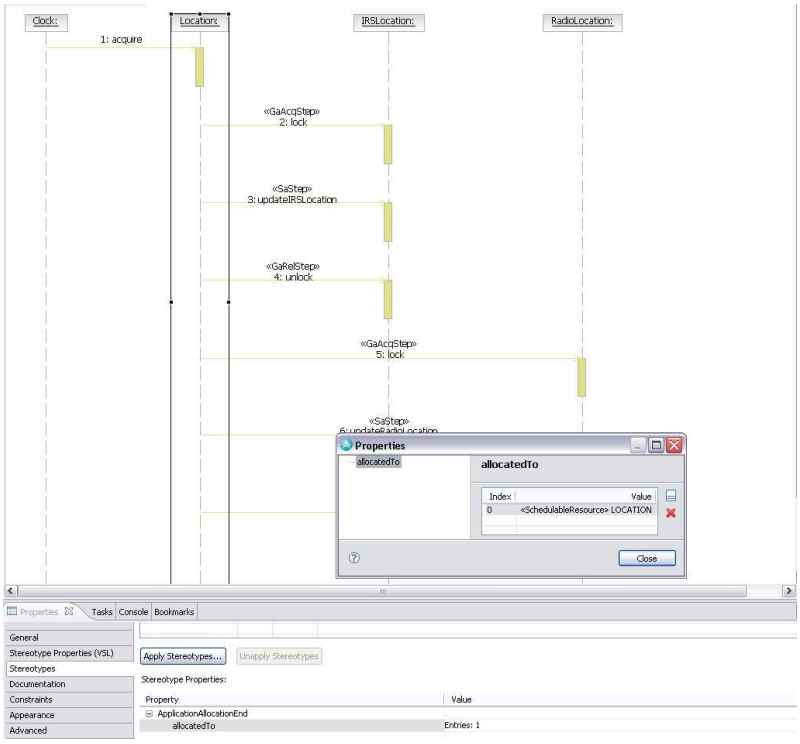

Figure 7: Description of a resource usage scenario in response to an event

All the sequence diagrams follow the same structure: an initiator (a clock in the case of periodic event or an interrupt handler in the case of a non-periodic event) sends a message at the beginning of the scenario to an execution resource, which will perform some computation in response to this message. The entity which receive the first message can be a execution resource (carrying the SchedulableResource stereotype) or an application element / a function that is allocated to an execution resource. Figure 7 describes the later.

In response to this message, the receiver will interact with the other entities of the system. Each of the resulting message will be stereotyped as GaAcqStep (acquisition of a shared resource), GaRelStep (release of a share resource) or SaStep (computation that uses the CPU with shared resource).

These MARTE stereotypes define attributes which provide the ability to specify execution times for all the steps defined in the context of this scenario. This information is also used to compute the time shared resources are used within a given a scenario.

6.7. Generate RapidRMA files

From the Eclipse Resource perspective, one can export the information stored in the MARTE model (.emx file) into files that can be read by RapidRMA. One needs to select the relevant .emx file and then to select the MARTE Export menu item. A series of files will be generated.

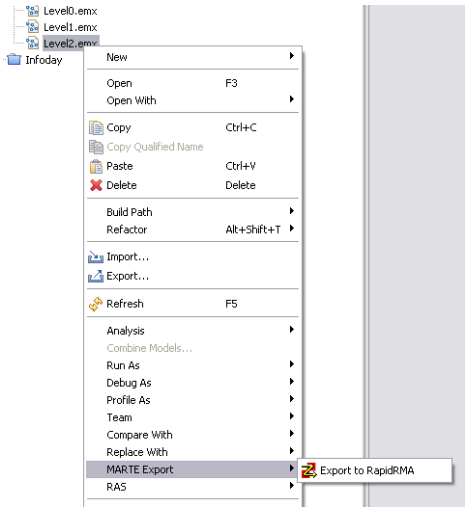

Figure 8: Export to RapidRMA menu

6.8. Launch a scheduling analysis with RapidRMA

Once the RapidRMA files are generated in the Eclipse workspace, it is possible to start the tool to open the file with the extension .rma. All the information stored in the MARTE models has been translated in a format that is understandable by the scheduling analysis tool. From there, it is possible to check whether the system design with UML MARTE models is schedulable, what the slack time is, and other scheduling information.

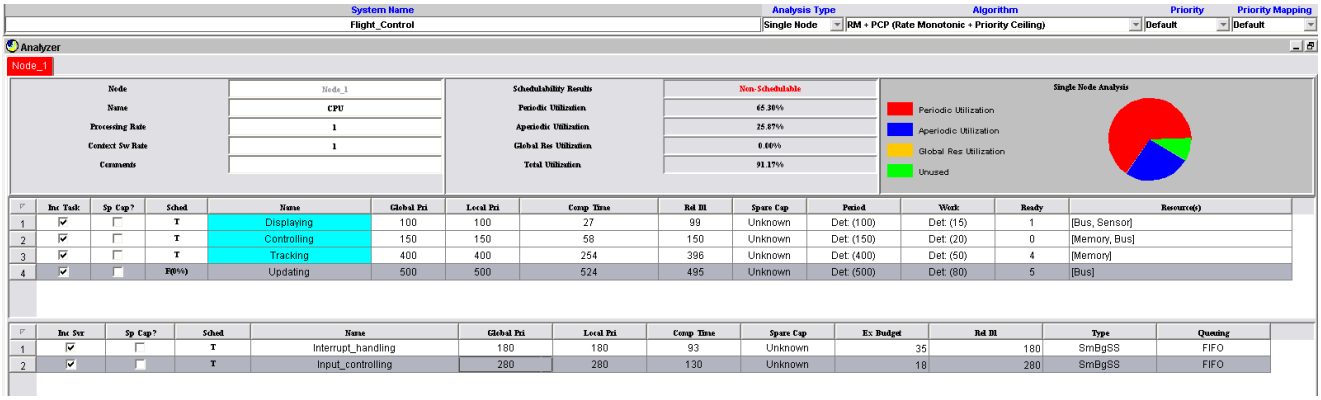

Figure 9: Scheduling analysis results with RapidRMA

7. KNOWN LIMITATIONS

The current implementation supports only scheduling analysis for mono-processors, with periodic and sporadic events (through sporadic servers). The current version does not provide support for multi-processors and distributed systems.

RapidRMA and IBM RSA are not integrated thought the GUI. Moreover, there is no automatic launch of RapidRMA after the input files are generated. It requires a manual operation.

The current implementation does not offer any feedback capabilities from RapidRMA to the UML modeling tool. All the analysis results can be exploited within the tool only (at least, with the version we used).

8. CONTACT INFORMATION

If you want more information about this software, or if you want to provide feedback, please contact:

- Sébastien Demathieu (sebastien.demathieu@thalesgroup.com)
- Laurent Rioux (laurent.rioux@thalesgroup.com)