THALES

Research & Technology

Report / Technical Document

MARTE FOR RATIONAL SOFTWARE ARCHITECT

Introduction to the UML profile and VSL editor

DEPARTMENT / SERVICE	DOCUMENT NUMBER	PAGE
STI / LSE	61565273 305 2	1/12
	-	REVISION

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

DOCUMENT CONTROL

	- the : 26/10/07	A the:	B the:	C the:	D the:
Written by Signature	S. Demathieu				
Approved par Signature					

Revision index	Modifications
A	
B	
C	
D	

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

CONTENTS

Pages

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

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 the UML profile for MARTE in modelling tools, such as IBM Rational Software Architect (RSA) 7.0. It has 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

[OMG MARTE]	OMG MARTE official web site (http://www.omgmarte.org)
[MARTE]	UML profile for MARTE, Beta 1 (http://www.omg.org/cgi-bin/doc?ptc/2007-08-04)
[Papyrus]	Papyrus UML modelling tool (http://www.papyrusuml.org)

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 Value Specification Language (VSL) editor for RSA, provided as part of this distribution, builds upon an initial implementation done by CEA-List and released under the EPL license in the Papyrus project [Papyrus]. Thales Research & Technology has completed the editor and extended its support for Rational Software Architect. Improvements made to the initial code base have been contributed back to the Papyrus project.

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

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

4. CONTENT OF THIS DISTRIBUTION

This software is delivered as an Eclipse extension that can be deployed in Rational Software Architect. It contains Eclipse features and plug-ins that allow one to create and edit UML models profiled with MARTE, along with a support for the VSL language in a dedicated editor. It is organised as following:

- *MARTE_RSA* directory
    - *eclipse* directory
        - *features* directory
            - *com.thalesgroup.marte.rsa*: Top-level feature (profile and VSL editor)
            - *com.thalesgroup.marte.rsa.nfp*: VSL editor feature
        - *plugins* directory
            - *com.cea.nfp.parsers*: VSL editor core functions (developed by TRT and CEA-List)
            - *com.thalesgroup.marte.rsa.helper*: a series of wizards and utility tools for MARTE
            - *com.thalesgroup.marte.rsa.nfp.ui*: VSL editor GUI and adapters for RSA
            - *com.thalesgroup.marte.rsa.profile*: profile implementation and model libraries
            - *com.thalesgroup.marte.vsl*: VSL editor core functions
            - *com.thalesgroup.vslview*: VSL AST viewer core functions
            - *com.thalesgroup.vslview.rsa*: VSL AST viewer for RSA
            - *org.odf.ctrte.nfp*: Miscellaneous GUI
        - *.eclipseextension* file

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

5. INSTALLATION PROCEDURE

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

The installation procedure is the following:

- Unzip the archive and copy the *MARTE_RSA* directory in your file system
- Start Rational Software Architect
- In the Help menu, go to *Software Updates* > *Manage Configuration*
- Right-click on *Rational Software Architect*
- Select *Add* > *Extension Location…*
- Browse your file system and select the *MARTE_RSA/eclipse* directory you just unzipped
- Restart Rational Software Architect

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

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 (See Figure 1.)

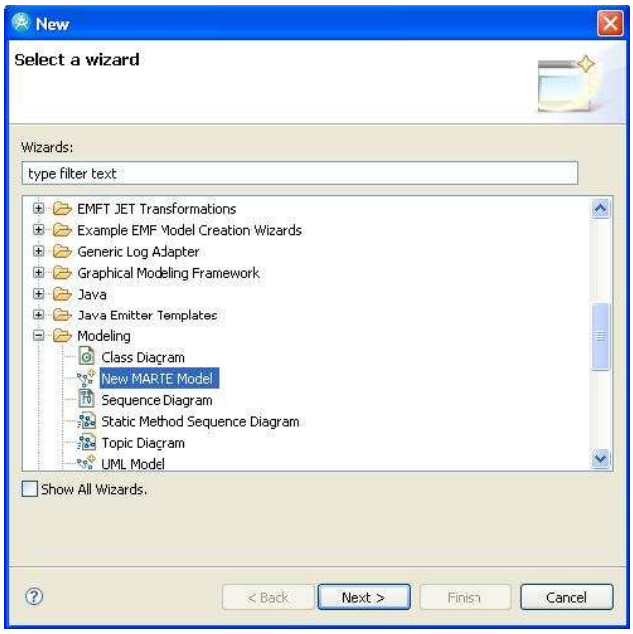

Figure 1: Access to the MARTE wizard

A new MARTE model can be created after the model name and location are entered by the user. All the MARTE sub-profiles will be applied, as illustrated in Figure 2.

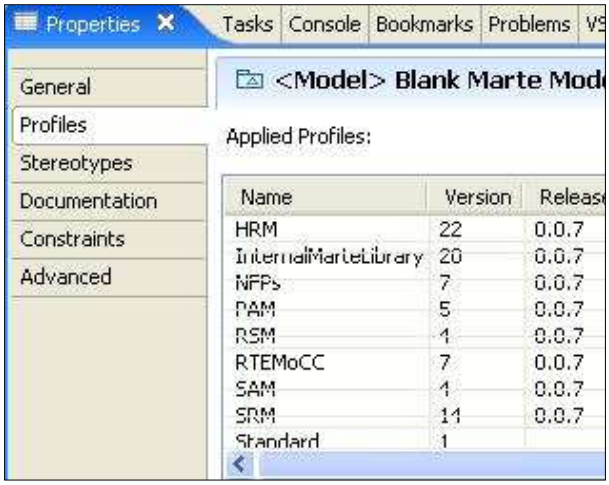

Figure 2 : List of sub-profiles applied on the MARTE model

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

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

6.2. Apply a MARTE profile on an existing UML model

The MARTE profile is broken down in a series of sub-profiles. It provides the ability to apply subsets of MARTE depending on different use cases. Provided that a plain UML model has been already created, it is possible to add only specific MARTE sub-profiles using the regular RSA mechanisms. The sub-profiles are deployed and ready to be used, as illustrated in Figure 3.

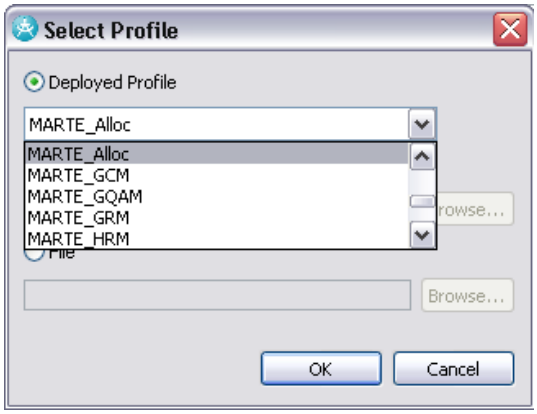

Figure 3: The list of deployed MARTE sub-profiles

6.3. Apply a stereotype and edit the stereotype attributes

The stereotype application mechanism is straightforward and leverages the default functions provided by RSA. However, there are two mechanisms to access and edit stereotype attributes:

- If the type of the related stereotype attribute is a VSL primitive type, a VSL enumeration or a VSL composite type (e.g. tuple, choice), then the user needs to use a dedicated tabbed panel in the property sheet. The tabbed panel is called *Stereotype Properties (VSL)* and gives access to the editor.

  Note: It is sometimes necessary to force a refresh of the property sheet so that *Stereotype Properties (VSL)* tabbed panel appears.

- Otherwise, the stereotype attribute can be edited like a regular attribute in the *Stereotype* tabbed panel.

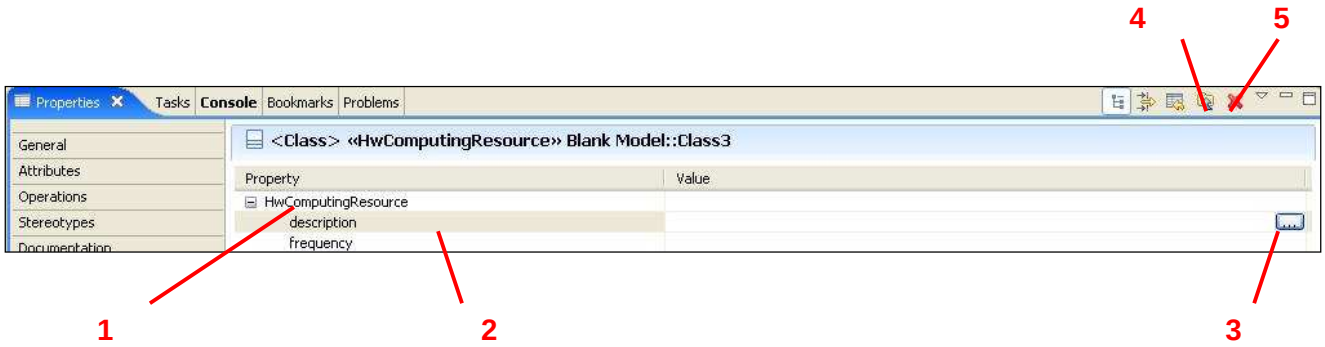

Figure 4: Details of the Stereotype Properties (VSL) tabbed panel

1. The list of applied stereotypes
2. The list of stereotype attributes that can be edited with the VSL editor
3. The button that triggers the VSL editor
4. The button that allows insertion of new values (for multiple-value properties)
5. The button that allows deletion of existing values (for multi-value properties)

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

The VSL language is applicable to any UML element (no only MARTE) that defines a UML Value Specification. As a consequence, the VSL editor can be used to edit values of stereotype attributes, but also to edit constraints, instance specification slots and property default values. For more details regarding the VSL editor features, you can have a look at section 6.4.)

6.4. Edit a constraint body using VSL

The VSL language can be used to edit the body of UML constraints. The mechanism is general enough to be used without any MARTE stereotype.

The VSL editor is a GUI that can be triggered on various kinds of UML elements. It provides content-assist, syntax checking, and type checking features. Errors are displayed using the Eclipse live validation mechanism.

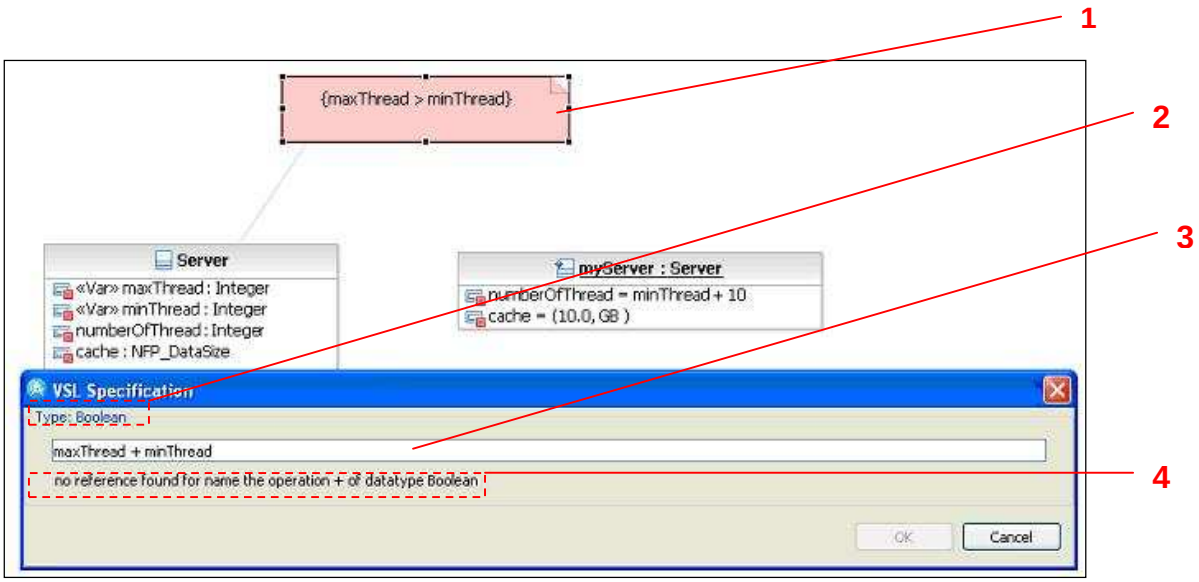

Figure 4: Usage example of the VSL editor

1. The UML element provided in input of the VSL editor
2. The expected type for the VSL expression
3. The input text field
4. The error message area

Figure 2 illustrates the content assist features of the VSL editor. It helps users completing references to model elements, references to types, operators, as well as some VSL syntax constructs.

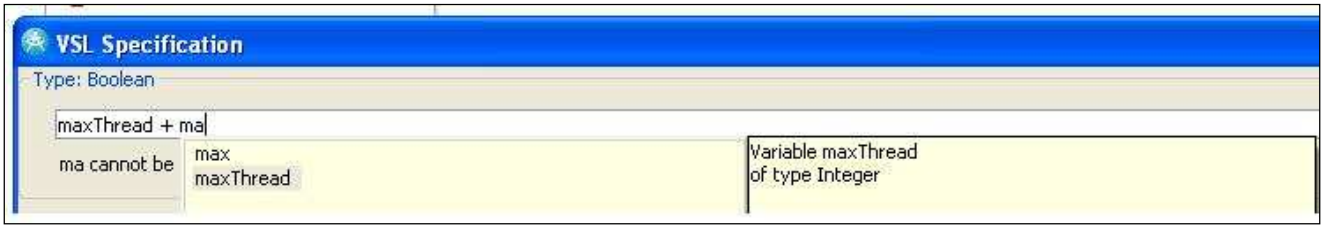

Figure 5: Example of content assist

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

The implementation of the VSL editor imposed some limitations on the constructs allowed by the language:

- In choice specifications, the choice names are not optional.
- In tuple specifications, the field names are not optional. A short syntax (e.g. “(10.0, ms)”) is provided for tuple types that introduce only a value and a unit. Its use is quite limited, though.
- Variable directions and types are not optional.
- All ambiguous references to model elements may lead to a typing error. In such a case, using fully qualified name is recommended.

To implement a complete type system, we made the following assumptions (that are not part of the specifications of the VSL language):

- InstantExpression is typed as NFP_DateTime.
- DurationExpression is typed as NFP_Duration.
- Jitter is typed as NFP_Duration.
- Variable declaration expressions have the type of the declared variable.
- VSL_Expression is the super-type of all VSL expressions. It implies that one can enter any kind of VSL expression without triggering type errors.

6.5. Edit a slot of an instance specification using VSL

The VSL language can be used to edit values of slots owned by UML instance specifications. The mechanism is general enough to be used without any MARTE stereotype. It just requires that the type of the related UML property is a VSL primitive type, a VSL enumeration or a VSL composite type (e.g. tuple, choice).

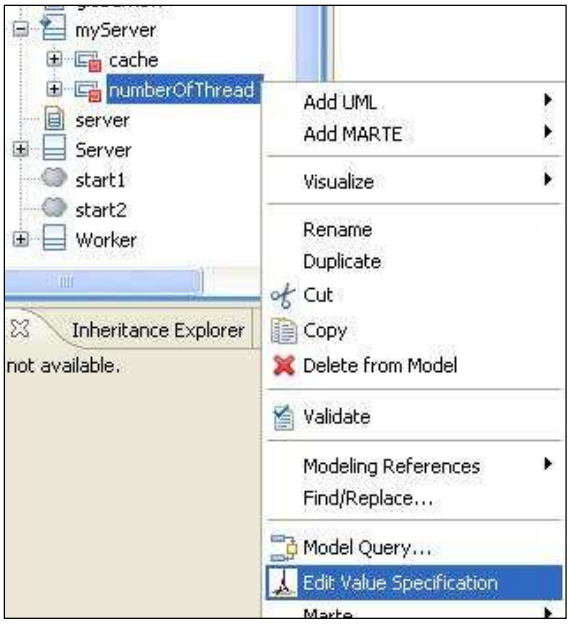

Figure 6: Triggering the VSL editor on an instance specification slot

Figure 6 illustrates how the VSL editor can be triggered on the slot “numberOfThread” of the “myServer” instance specification. A dedicated Eclipse action has been implemented for that purpose. In this example, we considered that “numberOfThread” is typed by the Integer type defined in the MARTE Library.

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

6.6. Edit a default property value using VSL

The VSL language can be also used to edit the default values of UML properties. The mechanism is general enough to be used without any MARTE stereotype. It just requires that the type of the related UML property is a VSL primitive type, a VSL enumeration or a VSL composite type (e.g. tuple, choice).

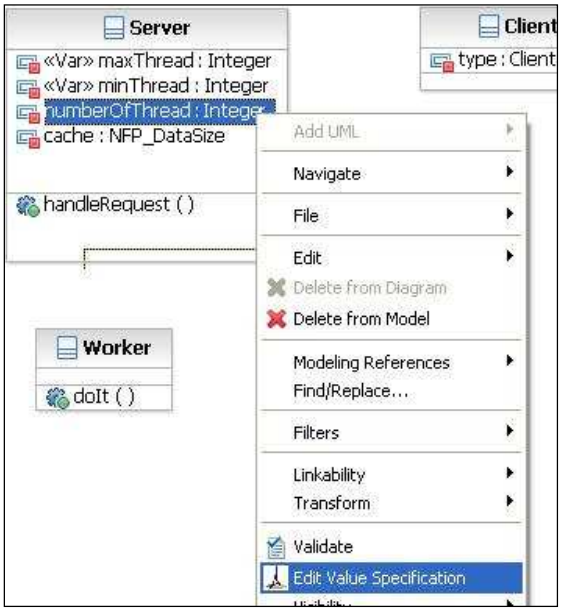

Figure 7: Triggering the VSL editor on a property to edit the default value

Figure 7 illustrates how the VSL editor can be triggered on the property “numberOfThread” in order to edit its default value. A dedicated Eclipse action has been implemented for that purpose. In this example, we considered that “numberOfThread” is typed by the Integer type defined in the MARTE Library.

Note: The property sheet does not always listen to the events that are triggered when the default value is updated. Therefore, it is necessary to force a refresh of the property sheet so that the display gets updated.

6.7. Visualize the AST of VSL expressions

The VSL editor can build an Abstract Syntax Tree (AST) of a VSL textual expression, corresponding to the meta-model defined in the MARTE specification. We created a dedicated Eclipse view that allows one to visualise these AST. We found this utility tool quite useful when implementing or debugging a model transformation that takes in input or output a model with VSL expressions.

One may access this view through the *Windows* > *Show View* > *Others…* menu. It is then possible to select *VSL View* in the *VSL* category.

The VSL view displays a VSL expression related to a selected element. It works only with UML constraints, slots and property default values (where the VSL editor is invoked through an Eclipse action). There is no support for stereotype attributes, at this time. Figure 8 illustrates the VSL AST related to body of a selected UML constraint.

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

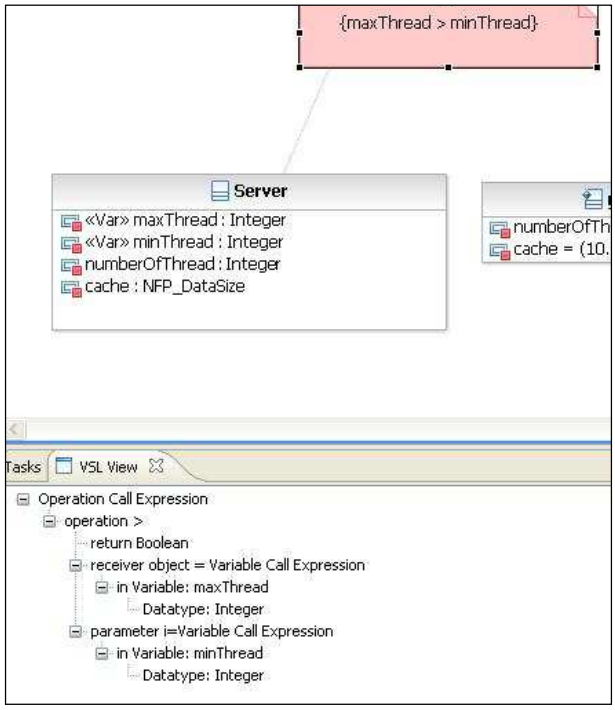

Figure 8: Abstract Syntax Tree of a VSL expression

6.8. Create new time observations and time event

The MARTE profile extends constructs defined in the CommonBehavior::SimpleTime package of UML (e.g. time observation, duration observation). It also requires accessing UML events defined along with sequence diagram messages. These elements cannot be created from the current RSA GUI although they are implemented in the underlying Eclipse UML2 meta-model. We added a specific “Add MARTE” menu to enable creation of these constructs on top of the existing tree-view of RSA (see Figure 9.)

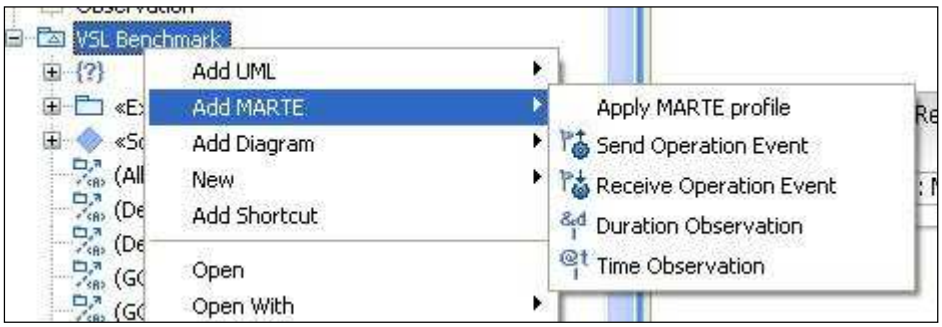

Figure 9: Add MARTE menu

Once they are added to the UML model, these elements can be handled as plain UML element in the tree-view, as well as the property sheet. Figure 10 describes how to access features of a duration observation using the RSA property sheet.

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.

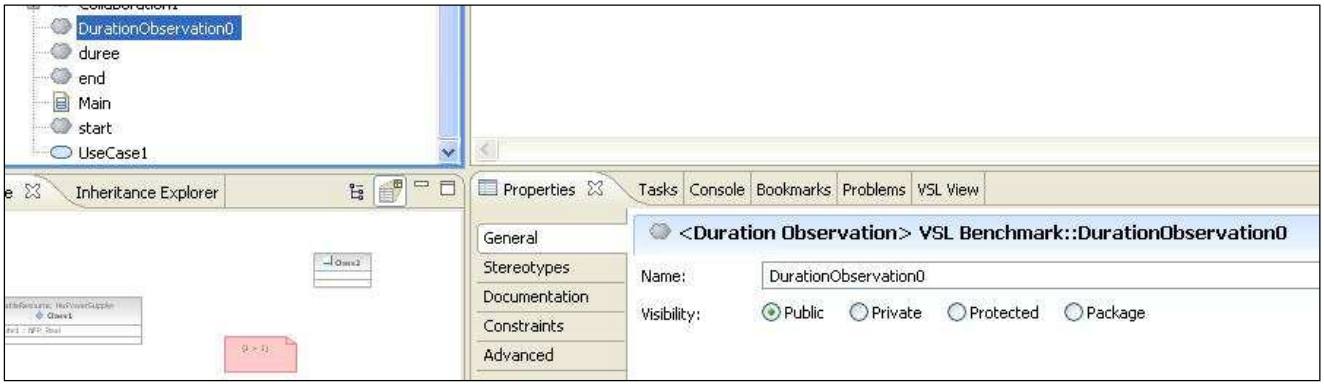

Figure 10: A duration observation in the property sheet

6.9. Display time observations in sequence diagrams

We added a support for UML observation display (time observation or duration observation) in sequence diagrams. It is based on the Eclipse decoration mechanism.

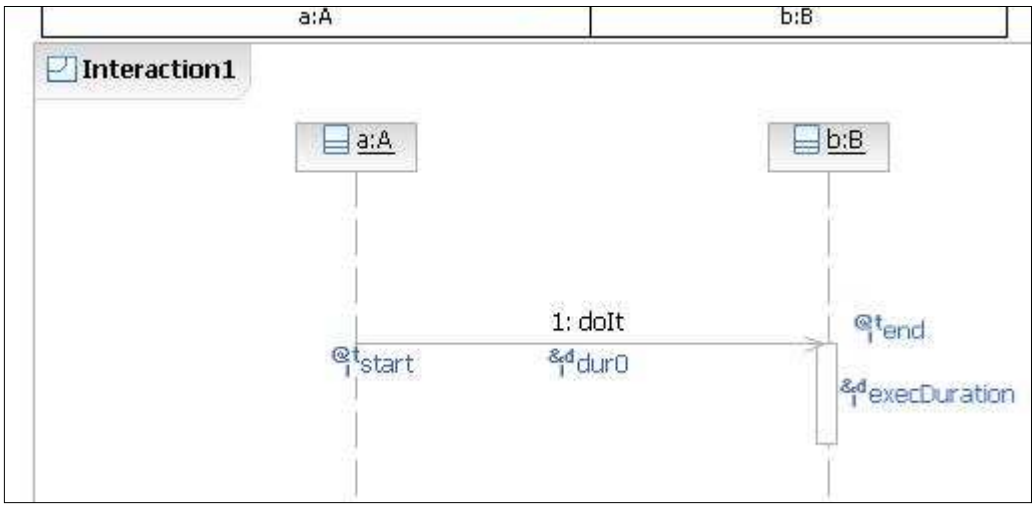

Figure 11: Instant observations and duration observations

This feature allows one to visualise observations related to messages and behaviour execution specifications. One can visualise time and duration observations that refer to a send operation event, receive operation event, send signal event, receive signal event (e.g. “start” and “end” in Figure 11.) It is also possible to visualise a duration related to events of a behaviour execution specification (e.g. “execDuration” in Figure 11.)

Note: Time observations and duration observations cannot be moved in the sequence diagram. It also requires saving and opening the diagram to ensure a proper display refresh.

7. 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)

THALES MARTE (Copyright (c) THALES 2007 All rights reserved) is free software; you can redistribute it and/or modify it under the terms of the Eclipse Public License as published in http://www.eclipse.org/legal/epl-v10.html

THALES MARTE is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details.