





Grant Agreement 224442

Model-based Analysis & Engineering of Novel Architectures for Dependable Electric Vehicles

4

Report type Report name

Deliverable D5.1.1 MAENAD Modeling Workbench

Dissemination level	PU
Status	Final
Version number	3.0
Date of preparation	2014-02-1

© 2014 The MAENAD Consortium

Authors

Editor	E-mail
Sara Tucci-Piergiovanni	Sara.Tucci@cea.fr
Authors	E-mail
Sara Tucci-Piergiovanni	Sara.Tucci@cea.fr
David Servat	David.Servat@cea.fr
Chokri Mraidha	Chokri.Mraidha@cea.fr

Reviewers	E-mail
Henrik Lönn	henrik.lonn@volvo.com
Frank Hagl	frank.hagl@continental-corporation.com

The Consortium				
Volvo Technology Corpo	pration (S)		Centro Ricerche F	iat (I)
Continental Automotive	(D) [Delphi/Mecel (S)	1	4S Group (I)
ArcCore AB (S)	MetaCas	e (Fi)	Systemite (SE)	CEA LIST (F)
Kungliga Tekniska Högs	kolan (S)	Technische L	Jniversität Berlin (D) University of Hull (GB)

Revision chart and history log

Version	Date	Reason
0.1	2010-12-06	Outline
1.0	2011-09-05	Intermediate release
1.1	2012-08-30	Update to EAST-ADL 2.1.10
2.0	2013-09-08	Update to EAST-ADL 2.1.11, instanceRef support and export to EATOP
3.0 prel	2014-02-14	Update to EAST-ADL 2.1.12 for review
3.0	2014-02-18	Final version following review

Approval	Date
Henrik Lönn	2014-02-20

Table of contents

Auth	ors	3
Rev	sion chart and history log	4
Tabl	e of contents	4
List	of figures	6
1	Introduction	7
2	Installation	8
3	Creation of a new model	10
4	InstanceRef support	14
5	EAXML export	17
6	References	19

List of figures

Figure 1: The bundled archive	.8
Figure 2: The workspace launcher	.9
Figure 3: Steps in creating a model from the wizard1	1
Figure 4 Model Structure (Model Explorer View) 1	1
Figure 5 Main Papyrus Views1	2
Figure 6: EAST-ADL abstraction levels 1	2
Figure 7 Class diagram use for Safety concepts of the Propulsion case study	13
Figure 8 Composite Diagram for Functional Architecture of the Propulsion case study1	13
Figure 9 Domain Model for Realization1	4
Figure 10 DataTypes for instanceRefs of Realization attributes	15
Figure 11 UML Profile for Realization1	15
Figure 12 UML model with TargetInstanceRef for FunctionalAllocation	6
Figure 13 EAXML export from the Papyrus model explorer1	17
Figure 14 generated EAXML file 1	17
Figure 15 Opening the EAXML file1	8
Figure 16 EAXML model in EATOP1	8

1 Introduction

This deliverable describes the UML-based modeling environment developed within the MAENAD project. This includes the UML modeler Papyrus and the UML profile for EAST-ADL developed for that tool [3]. This profile is currently compliant with M2.1.12 [4].

There are also extra plugins developed, which are described in D5.2.1 – MAENAD analysis workbench, see [1].

The workbench of EAST-ADL consists of a customized version of the Papyrus UML modelling tool, which is developed by CEA in the context of the Eclipse MDT project – see [2]. This is done by a dedicated EAST-ADL palette, which allows for direct creation of EAST-ADL stereotyped elements in the model.

The Papyrus tool provides a UML2 implementation that fully conforms to the OMG standards:

- 1. Papyrus conforms to the XMI format for saving models;
- 2. Papyrus conforms to the UML standard semantically and graphically;
- 3. Papyrus conforms to the Diagram Interchange (Di) standard to handle models graphical interoperability between tools.

To facilitate its extensibility, Papyrus is an Eclipse plug-in that uses other plug-ins such as UML2, EMF and GMF, ANTLR.

Papyrus 0.10 implements the following diagrams of UML2 standard:

- 1. Activity diagram
- 2. Class diagram
- 3. Composite diagram
- 4. Use Case diagram
- 5. Deployment diagram
- 6. State Machine diagram

Papyrus offers advanced functionalities for UML2 profiles support. Some of them are:

- 1. Profile diagrams
- 2. Hierarchical profiles
- 3. Complex typing of stereotype properties
- 4. Icons and shapes for stereotypes
- 5. Palette customization for applied profiles
- 6. Adding a popup menu to connect with an external tool

To create EAST-ADL entities, the user uses the dedicated EAST-ADL palette to create UML entities with the correct stereotypes applied to them. This feature is brought by a specific API which provides profile implementations (known as static profiles). This enables users to add specific functionalities associated to stereotypes, such as filters and automatic derivation of stereotype attribute values. Moreover, a model creation wizard is provided to help users have a well-configured model right from the start.

2 Installation

The Modeling workbench is provided as a complete bundle ready to be used, including some examples, via the link:

www.maenad.eu/public_pw/Tooling/papyrus0.10-eastadl2.1.12.zip

The installation of the tool and the profile is explained, and an empty EAST-ADL is created. There is also a more complete tutorial of the language in the project presentation material, which could be used as the next step, if the reader wants to develop a more complete EAST-ADL model.

Papyrus is based on Eclipse, and it can be used as a stand-alone RCP (Rich Client Platform), or on top of an existing Eclipse installation. This deliverable is based on the standalone version. There might be some small differences in the user interface compared with the plugin version.

Once unzipped the archive provides an executable: eclipse.exe, which launches an Eclipse application with a full Papyrus+EAST-ADL installation. The user is prompted with the choice of a workspace. You can select the workspace included in the bundle:

After unzipping the bundled archive, you get the following directory in [4].

	Rechercher Dossiers	•			
Adresse 🛅 C:\Documents and Setting	gs\David\Bureau\eclipse				~ 🔁
Assistant Recherche	Nom 🔺	Taille	Туре	Date de modification	
	Configuration		Dossier de fichiers	27/06/2011 16:41	
	🚞 dropins		Dossier de fichiers	27/06/2011 16:15	
	🛅 features		Dossier de fichiers	27/06/2011 16:32	
	☐ p2		Dossier de fichiers	27/06/2011 16:15	
	C plugins		Dossier de fichiers	27/06/2011 16:32	
	📄 🚞 readme		Dossier de fichiers	27/06/2011 16:15	
Que voulez-vous	workspace		Dossier de fichiers	27/06/2011 16:34	
	eclipseproduct	1 Ko	Fichier ECLIPSEPRO	27/06/2011 16:15	
Images, musique ou vidéos	artifacts.xml	392 Ko	Document XML	27/06/2011 16:32	
🔁 Documents (traitement de	eclipse.exe	52 Ko	Application	27/06/2011 16:16	
texte, feuilles de calcul,	🧕 💁 eclipse.ini	1 Ko	Paramètres de confi	27/06/2011 16:32	
	eclipsec.exe	24 Ko	Application	27/06/2011 16:15	
Tous les richiers et tous les dossiers	epl-v10.html	16 Ko	HTML Document	07/04/2011 05:13	
	Conotice.html	7 Ko	HTML Document	07/04/2011 05:13	
ordinacedis od personnes					
Informations du centre Aide et support					
Vous voulez peut-être également					
👰 Rechercher sur Internet					
Modifier les préférences					
<u>`@</u>					

Figure 1: The bundled archive

Double-click on eclipse.exe, you are prompted with a workspace selection, i.e. a directory where projects are stored on the hard drive.

🗁 eclipse			
Fichier Edition Affichage Favoris	Outils ?		A
🜀 Précédente - 🕥 - 🏂	P Rechercher		
Adresse 🛅 C:\Documents and Settings	David\Bureau\eclipse		💌 🛃 ок
Assistant Recherche	Nom	Type Date de modification	,
	Select a workspace Eclipse stores your projects in a folder called a work Choose a workspace folder to use for this session.	kspace.	-
Que voulez-vous rechercher ? Images, musique ou vidéos Documents (traitement de texte, feulles de calcul, etc.) Tous les fichiers et tous les dossiers Ordinateurs ou personnes Ordinateurs ou personnes Ordinateurs du centre Aide et support. Vous voulez peut-être également Rechercher sur Internet Modifier les préférences	Workspace: C:\Documents and Settings\David\Bur	reau\eclipse\workspace CK nd others, 2000, 2011. All rights reserved, tion,Tnc, Oracle and Java are registered to a ames may be trademarks of their respective	Browse Cancel Eclipse emarks owners.

Figure 2: The workspace launcher

In the following section it is explained how to create an initial model with the EAST-ADL profile applied.

Let us remark that updates of this platform can be obtained at any time after this installation, thanks to Eclipse software updates, simply select Help>Check for updates. You'll be prompted with a list of potential updates of plug-ins installed in your configuration, a check for availability and restrictions will be performed. After a restart, you will have an updated platform.

3 Creation of a new model

You can create a completely new model, by using the Papyrus creation wizard. In the navigator view, where files are shown, right click on the project and select New>Other, then Papyrus>Papyrus model. Choose a name, select UML file, then choose the diagrams you would like to be created and choose the EAST-ADL template for the model. See the following set of subfigures of Figure 3.

😤 Model Ex	φlorer 🕒 Project Explorer δ	3 🗖 🗖 moc	lel.di 🖾	
	· [
😑 😂 Exa	mple			
	mo New		Project	
			Ecore Diagram	
	Decte	Ctrl+C Ctrl+V	😭 Ecore Tools Project	
	X Delete	Delete		
	Remove from Context	Ctrl+Alt+Shift+Down		
	Move		T Example	
	Rename	FZ	📑 Other Ctrl+N	
	import			
	Export		_	
	🔊 Refresh	F5		
	Close Project	he .		
	Due As		-	
	Run As Debug As		•	
	Team		•	
	Compare With		>	
	Restore from Local Hist Configure	ory	gram 🔀 📴 NewDiagram	
🗄 Outline	Discovery		▶ es 🕅	
	Properties	ált-Enter		
			🖬 Info	
II. nple/model.di - Eclipse n - Papyrus Window - Help			☐ Info	
III nple/model.di - Eclipse n Pepyrus Window Help 留 - 11 th □ 12 III 12 III 12	: t _{@ 5} # t _n : → • @	凝•	 Info <td></td>	
Ple/model.di - Eclipse Papyrus Window Help •	ta := t. : → • @	∭ ≪ • • • • • • • • • • • • • • • • • • •	 Info <td></td>	
Re/model.di - Eclipse Pepyrus Window Help • ::::::::::::::::::::::::::::::::::::		滚 • 嗯 • 詩 • 饺 • ∉ ● New Select a wizar	G Info > · ↔ • 2* 强 • 100% ♥ C : Tahona I	. ×
/model.di - Eclipse Papyrus Window Help 1 1: 1	T ta st ta i → • 100 P model.di X	∰ • ® • ∰ • ੴ • ∯ E New Select a wizan	Info	
III Papyrus Window Help IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Na sta ti → Na Nodel.di X	⊗ • •8 • ∰ • °C • € E New Select a wizan	Info	
model.di - Eclipse apyrus Window Help I III Ta Ta Ta Ta Ta IIII Ta Ta Ta Ta Ta IIIIIIIIII	Tun st≉ tun i → • @)		Info	
III Papyrus Window Help IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Tue se tue i → • ⊡ ■ □ ■ □ ■ □ ■ □ ■ □ ■ □ ■ □ ■ □			
odel, di - Ectipse yyus Window Hep 다 문 이야하 X 문 역동	Tue sta te i → • Wi model.di X	Wizards: Vizards: Upp filter text. Image: Select a wizard Wizards: Upp filter text. Image: Select a wizard Image: Select a wizard <tr< td=""><td>Info Info Info</td><td></td></tr<>	Info Info	
del. di - Eclipse rus Window Heip Fa Ta Ta Ta Ta Euplorer X Ta Ta	te st te i → • •	Image: Select a wizards: Image: Select a wizard	Info Info	
nodel.di - Ectipse syrus Window Help	te st te i → v te i	Wizards: Uppe filer text Wizards: Uppe filer text Image: State Emilier Imag	Info Info	
aodel, di - Ectipse yrus Window Hep III III IIII IIIIIIIIIIIIIIIIIIIIIIII	tu ≄ tu i → · ⊕	Wizards: Uppe filter beat Wizards: Uppe filter beat Wizards: Uppe filter beat De Modeling De Modeling De Papyne Uppe filter beat	Info Info	
andel, di - Ectipse nyrus Window Help III III IIII IIIIIIIIIIIIIIIIIIIIIIII	tu ≄ tu i → · ⊕	X • • • • • • • • • • • • • • • • •	Info Info	
model, di - Eclipse aprus Window Help : 타 램 말 말 한 말 한 Ject Explorer X 금 양	Tun ≠ tun i → · Uni		Info Info	
annodel, di - Eclipse Papyrus Window Help : : : : : : : : : : : : : : : : : : :	The set the i → • the interval is a set of the set of	Image: Select a wizard Image: Select a wiza	Info Info	
III Papyrus Window Help · : : : ::::::::::::::::::::::::::::::	The state in the state of the	Wizards: Vype filter bast: Papy Syst Pugst Syst Pugst Viser Ass Vapard	Info Info	
III Papyrus Window Help · : : : : ::::::::::::::::::::::::::::	The state is in the state of t	Wizards: Vype filter text: Wizards: Vype filter text: Modeling Modeling Modeling Modeling System Papyrus: System Payrus: System Payrus: System Payrus: System Payrus: System Payrus: System Payrus: System Syste	Info Info	
ble/model.di - Eclipse Papyrus Window Help • i i i i i i i i i i i i i i i Project Explorer 2 E i i i Kton		Wizards: Vrpe filter text: Vrpe filter text: Modeling	Info Info	
ple/model.di - Eclipse Papyrus Window Help Project Explorer 2 Const Explor	NewClayam 23	Wizards: Vizards:	Info Info	
III Papyrus Window Help Voject England Ton Son	MewDlagram 25	Image: Select a wizards: Vizards: Vype filter text: Image: Select a wizards: Vype filter text: Image: Select a wizards:	Info Info	
model.di - Eclipse apyrus Window Help Jet Enplore 名 日本 日本 日本 日本 日本 日本 日本 日本 日本 日本 日本 日本 日本	MewOlogram 23	Image: Select a vizari Vizards: Vype filter text: Image: Select a vizari	Info Info	
nodel.di - Ectipse pyrus Window Help ⇒ ™™™™™™™™™ st Esplore: 20 ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩ ₩	Mexologram 23	Image: Select a wizards: Vizards:	Info Info	arcel
III	Menologram 23		Info Info Info Info Info Info Info Info Info Info Info Info Info Info Info Info Info	

New Papyrus Model	💶 🗖 🔀 👘 New Papyrus Model	
elect language of the diagram	Initialization information Select name and kind of the diagram	
Diagram Language:	Diagram Name:	
O 👘 SysML	NewDiagram	h
	Select a Diagram Kind:	<u> </u>
O Profile		
	Image: Structure Diagram Image: The UML Deployment Diagram	×
	You can load a template: A UHK model with basic primitive types (ModelW An EAST-ADL model with applied profile and syst	thBasicTypes).] em structure (initialmodel)
	/OA	
?) Stark Next >	Finish Cannel 1d ? <back mit<="" td=""><td>xxt > Finish Cancel</td></back>	xxt > Finish Cancel
	le le	

Figure 3: Steps in creating a model from the wizard

Now in the Project Explorer view, three files have been created, the .di file, the .uml file and the .notation file. The .di file contains the graphical information and the .uml file is where all model data is stored. At creation time the model is open, but since all diagrams are closed the diagram view is empty and the palette absent. In order to let show up the palette you need to open a diagram. To do this, scroll model elements from the model explorer and choose a diagram (see Figure 4).

🖶 Model Explorer 🛛	📃 🗉 🗄 🔛 🕒 💲 🗋	
a 🖾 InitialModel		
⊳ 📇 <package< th=""><th>Import> UML Primitive Types</th><th></th></package<>	Import> UML Primitive Types	
a 🛅 TopPacka	je	
a 🛅 «Syste	nModel» SystemModel	
⊳ 🖿 «Ve	hicleLevel» MyVehicleLevel	
A» 📃 «A	nalysisLevel» MyAnalysisLevel	
⊳ 📃 «D	esignLevel» MyDesignLevel	
«In	plementationLevel» MyImplementat	tionLevel
📑 Dia	gram SystemModel	
🛅 Commoni	Datatypes	
b 📩 AnalysisLe	velElements	
DesignLev	elElements	
🛅 Implemen	tationLevelElements	
EAST-ADL	ExtensionElements	

Figure 4 Model Structure (Model Explorer View)

Now you should obtain the layout shown in Figure 5.

MAENAD

D!	5.'	1.	1
_			-

Papyrus - testtest/model.di - Eclipse						
File Edit 🥠 Diagram Navigate Search F	apyrus Project	Run Window He	lp 自当\$\$`\$P → ▼ \$P 淡 × °B ▼ \$P * "	?** ↔ * + * * * * * * * * * * * * * * * *) \ \ \ - 0 - \ - \ - \ - \	A -
년 • 원 • ← ⇔ • → • 존 [Tahoma		+ 9 + I	3 I A + D + 2 +		Quick Access	🖁 Java 🌈 Papyrus 🏠 Resource
 Model Explorer ImitialModel ImitialModel ImitialModel ImitialModel CapPackage TopPackage CommonDatatypes AnalysisLevelElements DesignLevelElements DesignLevelElements EAST-ADLExtensionElements 	✓ model.di ∷ ✓ ■ BystemMode	Propulsion.dt	Myleticitikevel -Venicitikeveli Letiveral/fasture/todes()//Fasture/tode) -Lasture/todes/ -Massure/todes/ -Reserve/todes/ -Reserve/todes/	a	Diagram view	Image: Second state of the second
Model explorer view	Properties SystemM	J Model Validati	on 🐑 Error Log 🖹 Problems 🗿 History	Property View		
	UML Comments	Name Visibility	systemModel			•
۲ [۲	Profile	Package merge				00 +× /

Figure 5 Main Papyrus Views

Note that the model is created with the correct structure (Figure 4), as suggested in the specification of the EAST-ADL language, as shown in Figure 6.

	System Model							
Vehicle Level	VehideLevel TechnicalFeature	Model						
Analysis Level	AnalysisLevel AnalysisArchitect	ure		nent Model	onValidation	irement	ming	ndability
Design Level	DesignLevel FunctionalDesign HardwareDesign/	Architecture Architecture		Environn	Verificatic	Requ	Ē	Deper
Implementation Level	ImplementationLe AUTOSAR Application SW	Vel AUTOSAR Bæsic SW	AUTOSAR HW					

Figure 6: EAST-ADL abstraction levels

Let us note that only two types of diagrams are used to graphically build and represent EAST-ADL concepts, namely the Class diagram and the Composite diagram.

Note that both views are useful during the construction of an EAST-ADL model. For instance Class diagrams are suitable to set up Requirements, type definitions in general (AnalysisTypes, DesignTypes, DataTypes, etc) or Feature models, while Composite diagrams are necessary to show prototypes as internal parts as in the case of analysis and design architectures and error models. Figure 7 and Figure 8 shows the use of respectively Class diagram and Composite

diagram for safety-related concepts representation and functional architecture representation of the Propulsion case study.



Figure 7 Class diagram use for Safety concepts of the Propulsion case study



Figure 8 Composite Diagram for Functional Architecture of the Propulsion case study

4 InstanceRef support

The UML profile, up to the implementation of the EAST-ADL M2.1.10, has supported the instanceRef concept by adding a 'path' attribute to all stereotype's attributes that should represent an instanceRef. This support has been found very weak from a user point of view, but more importantly it has been found as not sufficient to correctly serialize instanceRefs in the EAXML format. Just to make an example, consider the case of the Realization concept, which has two <<instanceRef>> attributes: realized and realizedBy of multiplicity [0..*] as shown in Figure 9.



Figure 9 Domain Model for Realization

Up to the implementation of EAST-ADL M2.1.10, the Realization concept has been represented as <<Realization>> stereotype with a 'realized' attribute of type NamedElement to set the target of the instanceRef and a 'realized_path' attribute to set the instanceRef context. The problem arises when multiple 'realized' elements should be set, in which the context may not be set for all of them. In this case the correspondence between each target and its own context in the *_path attribute (maybe not defined) cannot be traced.

An alternative solution has been pursued in the implementation for EAST-ADL V2.1.11, now M2.1.12. The idea is to map each instanceRef to a corresponding DataType, defined in the profile. Figure 10 shows the two dataTypes corresponding to 'realized' and the 'realizedBy' instanceRefs.



Figure 10 DataTypes for instanceRefs of Realization attributes

Figure 11 shows that the realized and the realizedBy attributes are explicitly typed by the previously defined instanceRefs.



Figure 11 UML Profile for Realization

At the user level, data types defined in the profile will be instantiated on the fly. Figure 12 shows a sample model where the target attribute of a functionalAllocation is being set. In the properties view the Allocation Target and the Allocation Target context will appear and values for these properties can be set. Note that values must pre-exist in the model.

Allocation Target context

🖶 model.uml 🔀 ⁄ *model.di ⊿ 🖾 <Model> model Package Import> UML Primitive Types Package> 0_TopPackage <Package> 1_CommonDatatypes Package> 2_VehicleLevelElements Package> 3_AnalysisLevelElements a 🛅 <Package> FunctionalElements Electric Class> MyFunctionalDesignArchitecture
 Section 2.2
 Sec > ackage> Package1 a 🛅 <Package> Allocations <<Allocation>> <Class> MyAllocationTest ▲ {?} <<FunctionAllocation>> <Constraint> FunctionAllocation1 💀 <Literal String> ♦ Target Instance Ref Allocated Element Instance Ref Package> HardwareElements <Package> 5_ImplementationLevelElements Package> 6_EAST-ADLExtensionElements 🔲 Properties 🔀 Property Value 🗉 Hardware Component Prototype hda - model.uml Allocation Target

Figure 12 UML model with TargetInstanceRef for FunctionalAllocation

E H

Hardware Component Prototype class1 - model.uml

5 EAXML export

An EAXML export has been realized for the version of the UML profile for EAST-ADL v2.1.11 and then migrated to EAST-ADL 2.1.12. The export uses the serialization services of EATOP which means that the EAXML export will work only if EATOP is installed as well. Figure 13 shows how to run the export: the root model element must be selected in the Model Explorer, then 'Generate eaxml from EASTADL' option must be chosen.



Figure 13 EAXML export from the Papyrus model explorer

The EAXML file will be generated and stored in the model project folder as shown in Figure 14.



Figure 14 generated EAXML file

To open the generated file, you need to click-right on it and choose 'Open with EAST-ADL Example Editor as shown in Figure 15.

Resource - propulsion/propulsion	on di - Eclipse		
File Edit 🥠 Diagram Naviga	e Search Papyrus Project Run W	/indow Help	
11 - II @ 2 Ø IB B	🗂 🍫 ▾ 🖗 → ▾ 👫 💥 ▾ 여	+ 💱 + 🕃 +	⊕ • • • • ٩, ٣ 田 • / / ا ۹
Tahoma	▼ 9 . ▼ 10 / 1 A ▼ .9 ▼ .7	5	
Project Explorer 🔀	🖻 🍇 😵 🏱 🗖 🚺 🥠 *mo	del.di 🛛 🧑 *pr	opulsion.di 🔀
a 😂 propulsion	*		
propulsion			
PropulsionModel.	New	•	
	Open	F3 🏴	
🗄 Outline 🖾 📗 Task List	Open With	•	EASTADL Example Editor
	Сору	Ctrl+C	Text Editor
	Paste	Ctrl+V	System Editor
	Delete	Delete 📄	In-Place Editor
	Remove from Context Ctrl+Alt+S	hift+Down	Default Editor
	Mark as Landmark Ctrl+Alt	+Shift+Up	Other
	Move	-	note=Vehicle stopped at the
	Rename	F2	
اللأج	Import	-	— `x —
2	Export		· · · ·
		13	🗗 RequirementDiagram 🖬 Functi

Figure 15 Opening the EAXML file

Figure 16 shows the EAXML file content for the Propulsion case study:

⁄ *model.di	🤿 *propulsion.di 📄 PropulsionModel.eaxml 🔀
EASTADL	Contents Tree
Propulsion	Model.eaxml Contents
This section	n enables the contents of this element to be edited.
⊿ 🔶 EA	XML
♦	PropulsionModel
⊳	SystemPackage
4	DesignElements
	FunctionTypes
	A HarwareTypes
	A
	HardwareDesignArchitecture
	FunctionDesignArchitecture
4	BehavioralElements
	a 💠 Behavior
	a 💠 ModeGroup
	VehicleMode1
4	AnalysisLevelElements
	PropulsionAnalysisFunction
	panalysisFunctionPrototype
4	Dependability
	UnwantedPositiveTorque
	SuddenVehicleMovement
	♦ HE1
	PropulsionItem
	♦ SafetyGoal1
4	Requirements
	DeriveRequirement1
I	A DeriveRequirement?

Figure 16 EAXML model in EATOP

6 References

- [1] MAENAD Deliverable D5.2.1 MAENAD analysis workbench.
- [2] Papyrus MDT website, http://www.eclipse.org/modeling/mdt/papyrus/
- [3] EAST-ADL Profile specification for M2.1.12, http://maenad.eu/publications.html
- [4] EAST-ADL Domain Model Specification M2.1.12, http://maenad.eu/publications.html