

SystemWeaver for EAST-ADL

Concept Presentation

2011



Concept presentation roadmap

1. Short description: SystemWeaver in general
 - SystemWeaver – A system modeling platform
 - SystemWeaver – A collaborative environment
 - The Graph views: Tree View FAA, FDA and HDA
2. Concepts - Motivation of SystemWeaver concepts
3. How we have implemented EAST-ADL; examples:
 - The port concept as an example: Our solution building on ready "buildingblocks" (parts in systemweaver, which already have directions defined by its part type).
4. File exchange – choosing between 2 options
 - A XML-transformation technique
 - or an separate application/plugin.

1. SystemWeaver

-A System modeling platform

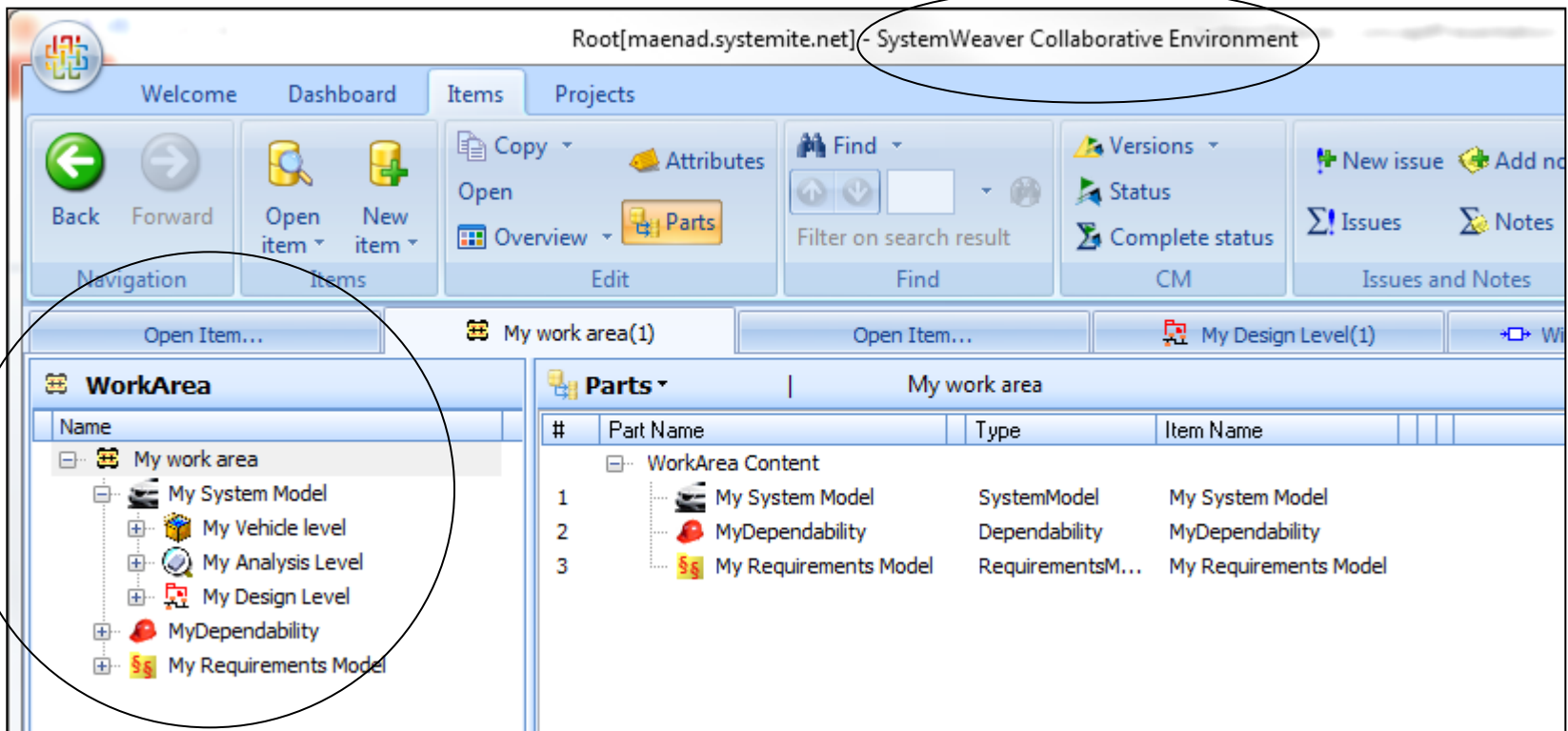
Short description:

- SystemWeaver is a generic System modeling platform.
- Almost every need of general System modeling is met in the SystemWeaver platform.
- The general rule in the System modeling is: 1 effective solution of implementation concept, for every need that is met.
- This is followed by the fact that the full formal EAST-ADL meta model not always is fully supported in terms of basic syntax and semantics of SystemWeaver.
- However in most cases there is a suitable modeling construct in SystemWeaver. Also the mapping is simple enough to support bidirectional exchange with EAXML without loss of information.

SystemWeaver

- A collaborative environment

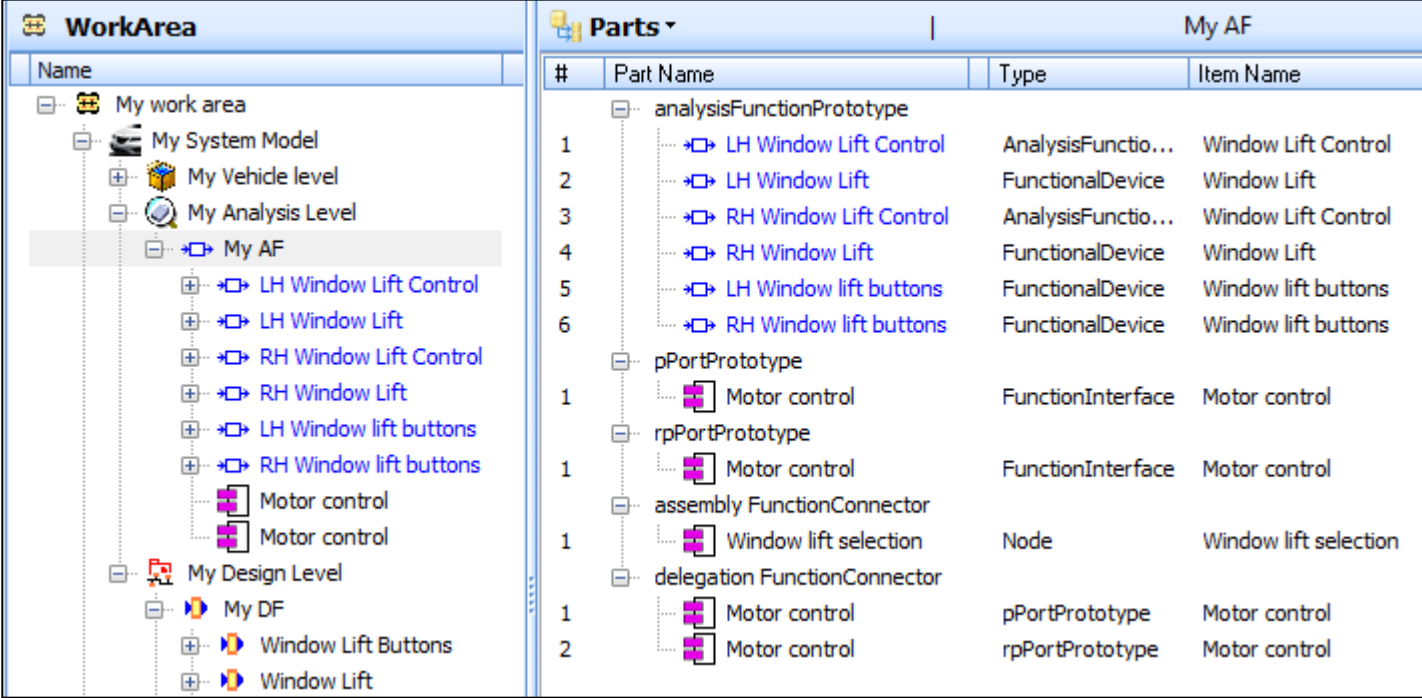
- Many users – everyone will see everything right now



#	Part Name	Type	Item Name
1	My System Model	SystemModel	My System Model
2	MyDependability	Dependability	MyDependability
3	My Requirements Model	RequirementsM...	My Requirements Model

Tree view

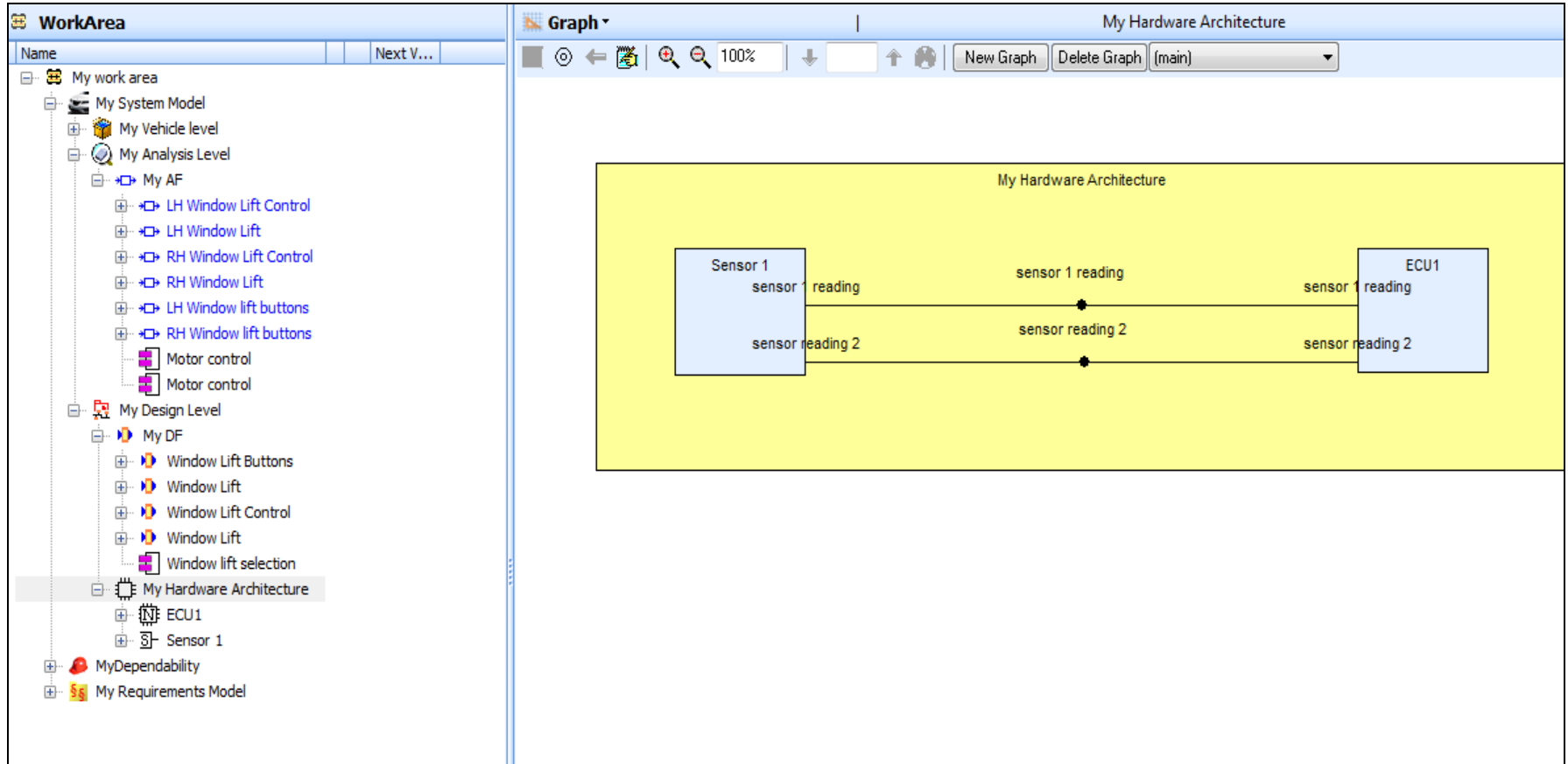
- When marking the “My AF” (FAA) for an example: You also get info on all prototypes on the right hand side. Where information about all instantiations (parts in SystemWeaver) on that level is listed.



The screenshot shows the SystemWeaver interface with two main panels:

- WorkArea:** A tree view showing the project structure. The selected node is "My AF" under "My Analysis Level".
- Parts:** A table listing all parts and prototypes associated with the selected "My AF".

#	Part Name	Type	Item Name
analysisFunctionPrototype			
1	LH Window Lift Control	AnalysisFuncio...	Window Lift Control
2	LH Window Lift	FunctionalDevice	Window Lift
3	RH Window Lift Control	AnalysisFuncio...	Window Lift Control
4	RH Window Lift	FunctionalDevice	Window Lift
5	LH Window lift buttons	FunctionalDevice	Window lift buttons
6	RH Window lift buttons	FunctionalDevice	Window lift buttons
pPortPrototype			
1	Motor control	FunctionInterface	Motor control
rpPortPrototype			
1	Motor control	FunctionInterface	Motor control
assembly FunctionConnector			
1	Window lift selection	Node	Window lift selection
delegation FunctionConnector			
1	Motor control	pPortPrototype	Motor control
2	Motor control	rpPortPrototype	Motor control

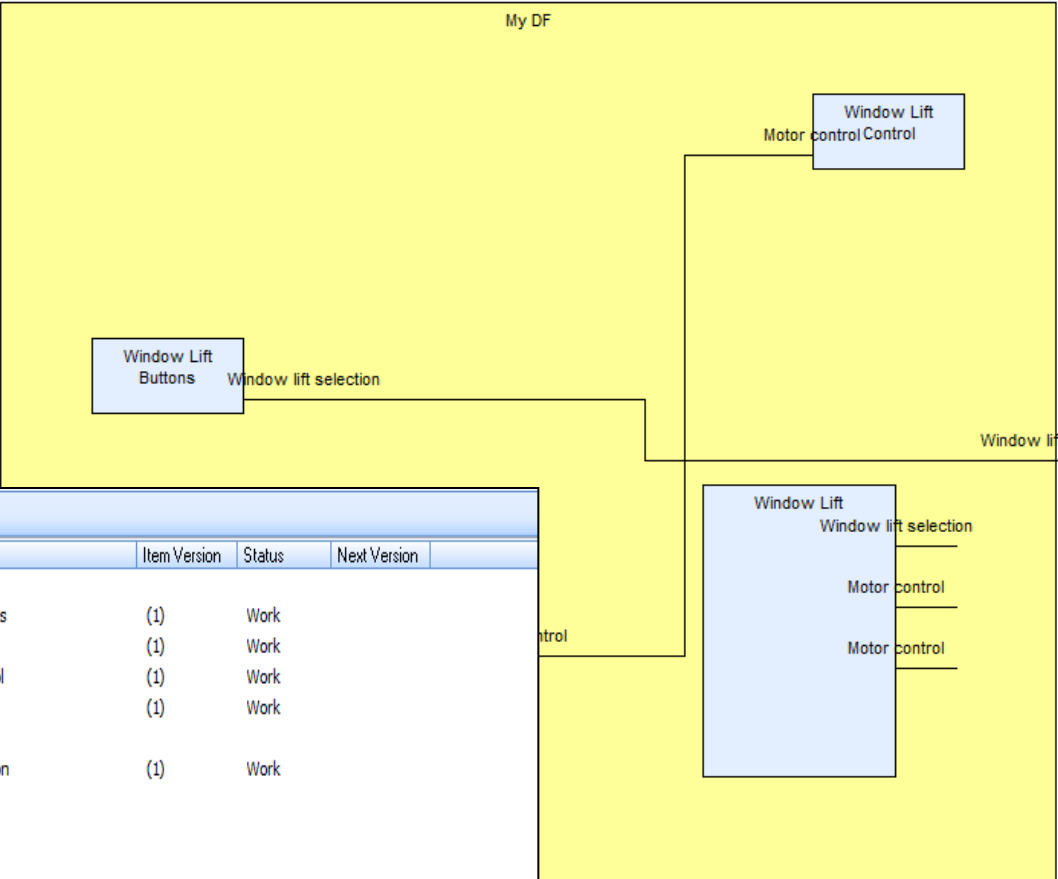


Open Item... My work area(1) Open Item... My Design Level(1) Window lift buttons(1)

WorkArea

Name	Next V...
My work area	
My System Model	
My Vehicle level	
My Analysis Level	
My AF	
LH Window Lift Control	
LH Window Lift	
RH Window Lift Control	
RH Window Lift	
LH Window lift buttons	
RH Window lift buttons	
Motor control	
Motor control	
My Design Level	
My DF	
Window Lift Buttons	
Window Lift	
Window Lift Control	
Window Lift	
Window lift selection	
My Hardware Architecture	

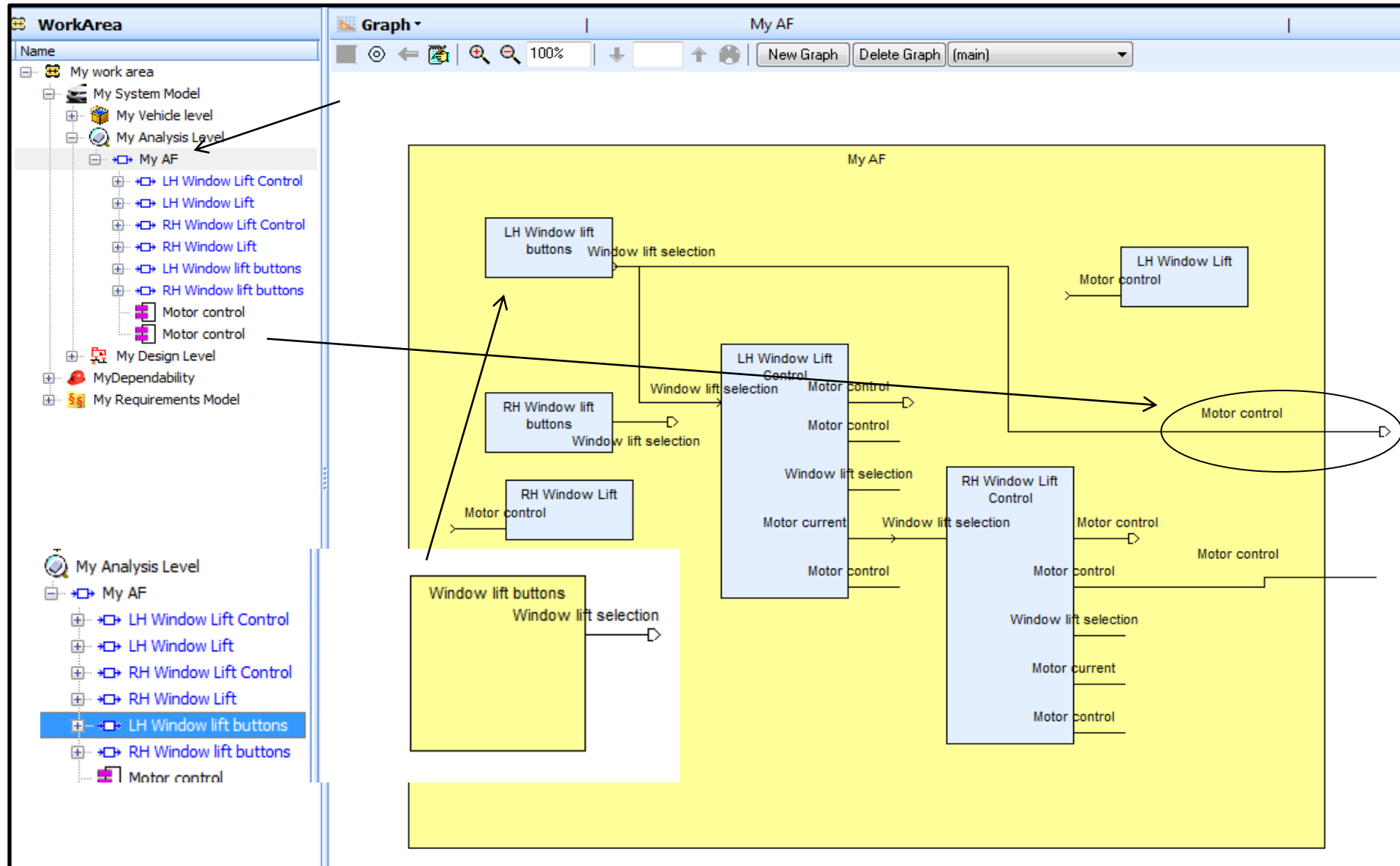
Graph | My DF



The diagram shows a hierarchical structure of design functions. At the top level is 'Window Lift Buttons', which connects to 'Window Lift selection'. 'Window Lift selection' then branches into 'Window Lift Control' and 'Motor control'. 'Window Lift Control' further connects to 'Motor control'. The diagram is set against a yellow background.

Parts | My DF

#	Part Name	Type	Item Name	Item Version	Status	Next Version
designFunctionPrototype						
1	Window Lift Buttons	DesignFunction...	Window Lift Buttons	(1)	Work	
2	Window Lift	DesignFunction...	Window Lift	(1)	Work	
3	Window Lift Control	DesignFunction...	Window Lift Control	(1)	Work	
4	Window Lift	DesignFunction...	Window Lift	(1)	Work	
rpPortPrototype						
1	Window lift selection	FunctionInterface	Window lift selection	(1)	Work	
assembly FunctionConnector						
1	Motor control	Node	Motor control			
delegation FunctionConnector						
1	Window lift selection	rpPortPrototype	Window lift selection			



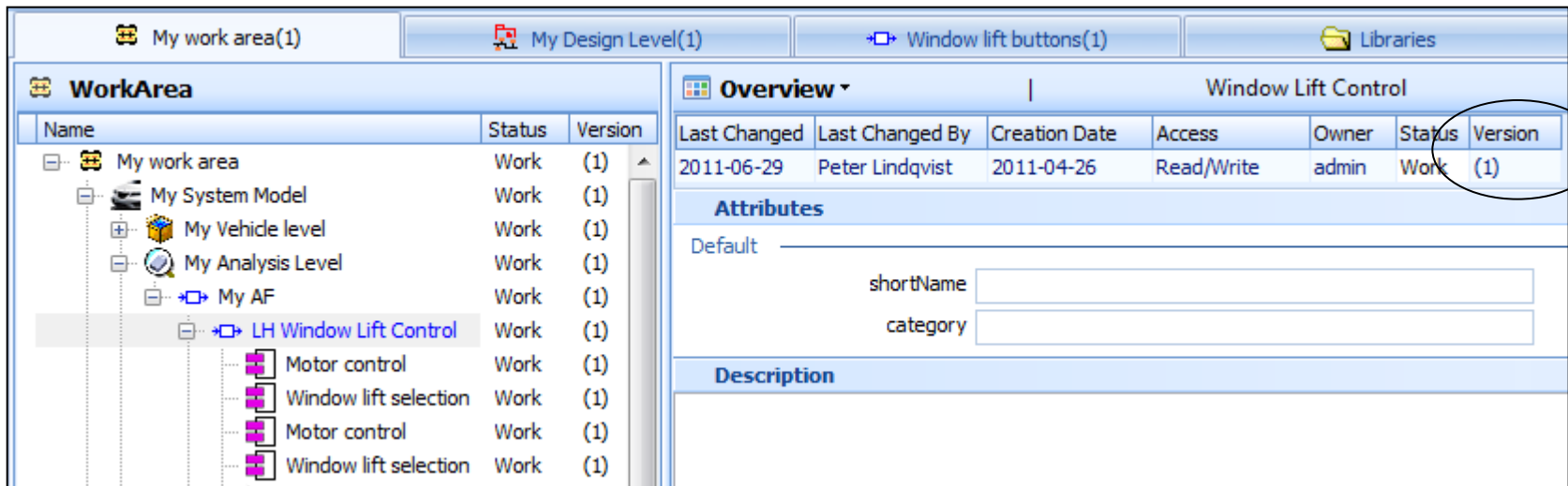
2. Concepts of SystemWeaver

Motivation of the SystemWeaver concepts:

- Fine grained versioning
- Every item has its own versioning
- Specific and self-sufficient contexts/building blocks

Every item has its own versioning

- Every item has its own versioning, where every item including its contained structure of parts (“prototypes in EAST-ADL”)
- Every of this versions forms a separate configuration
- This is useful in some years old projects/meta models



The screenshot shows a software interface with a hierarchical tree on the left and an overview table on the right.

WorkArea Table:

Name	Status	Version
My work area	Work	(1)
My System Model	Work	(1)
My Vehicle level	Work	(1)
My Analysis Level	Work	(1)
My AF	Work	(1)
LH Window Lift Control	Work	(1)
Motor control	Work	(1)
Window lift selection	Work	(1)
Motor control	Work	(1)
Window lift selection	Work	(1)

Overview Table (Window Lift Control):

Last Changed	Last Changed By	Creation Date	Access	Owner	Status	Version
2011-06-29	Peter Lindqvist	2011-04-26	Read/Write	admin	Work	(1)

The 'Version' column in the Overview table is circled in the image.

Attributes:

Default

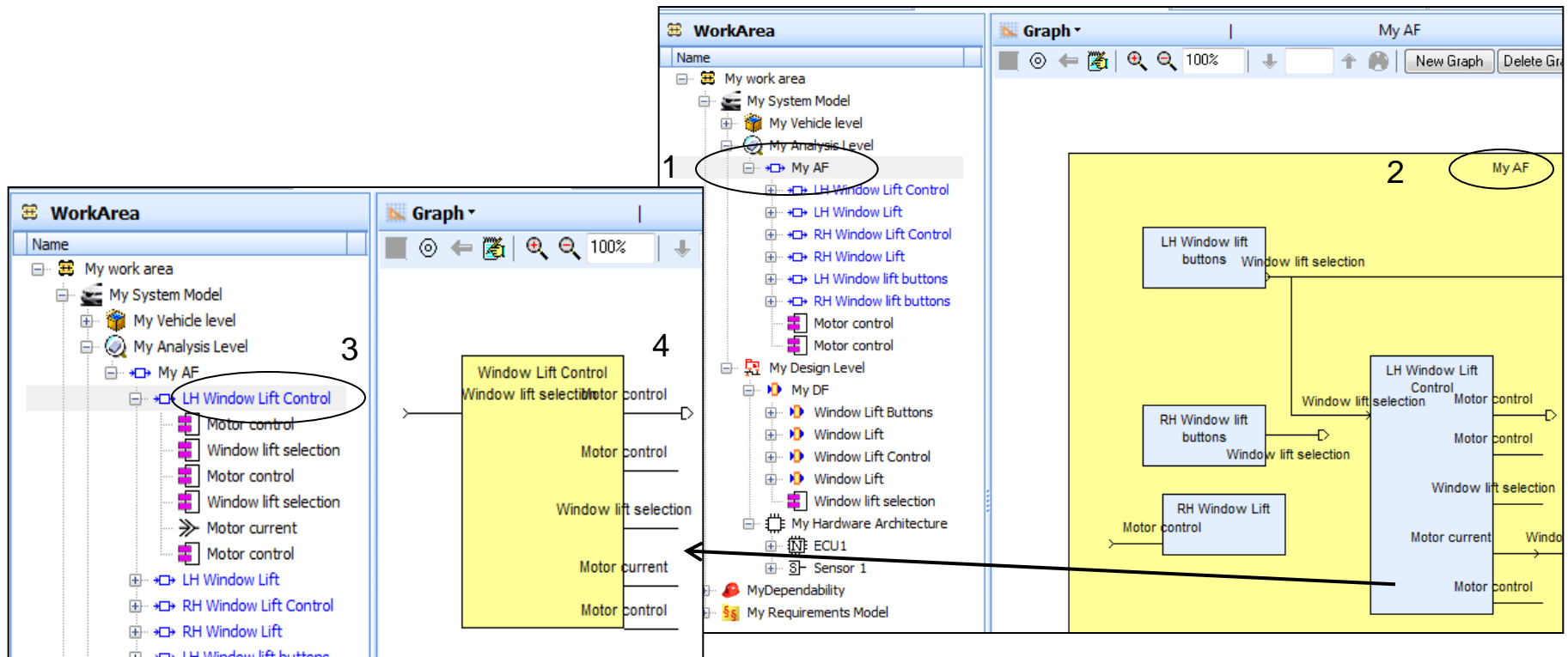
shortName

category

Description:

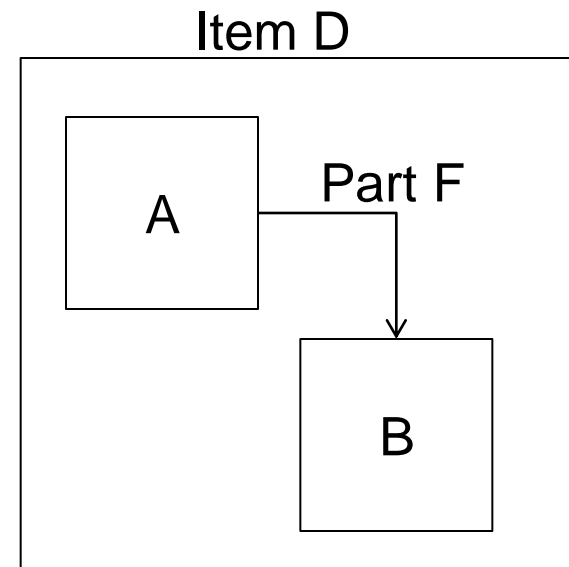
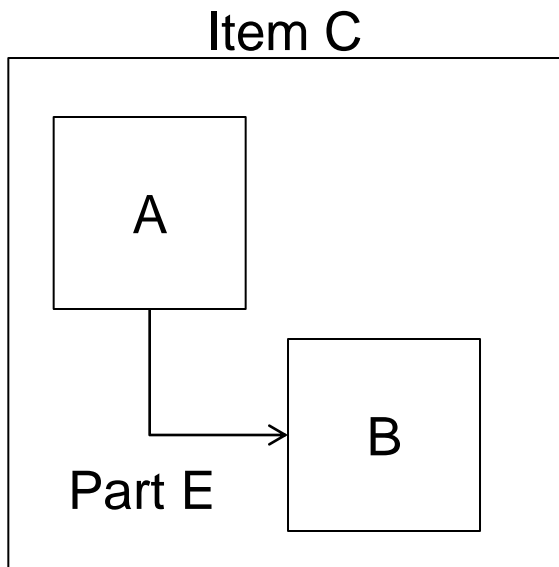
Specific and self-sufficient building blocks/contexts

- Every relation is defined within a specific context: the item that owns the relation (i.e. owns not just its prototypes, but all the relations between prototypes). Prototype/Type pattern is used for all instances in a model in SystemWeaver. Relations are defined only within a context. No relation can exist without a defined context owning the relation.



Every relation is defined within a specific context:

- As explained every relation is defined within a specific context.



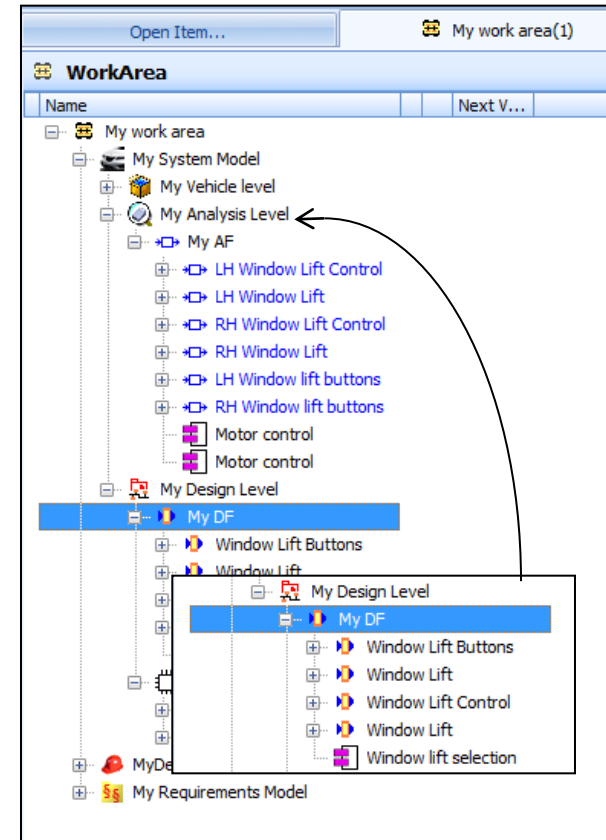
Concepts

Motivation of the SystemWeaver concepts:

- Methodology
 - Effective modelling abilities: Every model structure is defined in a specific context, creating hierarchical prototype/type structures
 - Strong meta model: cannot do things that should not be in the implemented meta model
 - Documentation on model and meta model available by just a “click”
 - Portconcept & graphical support

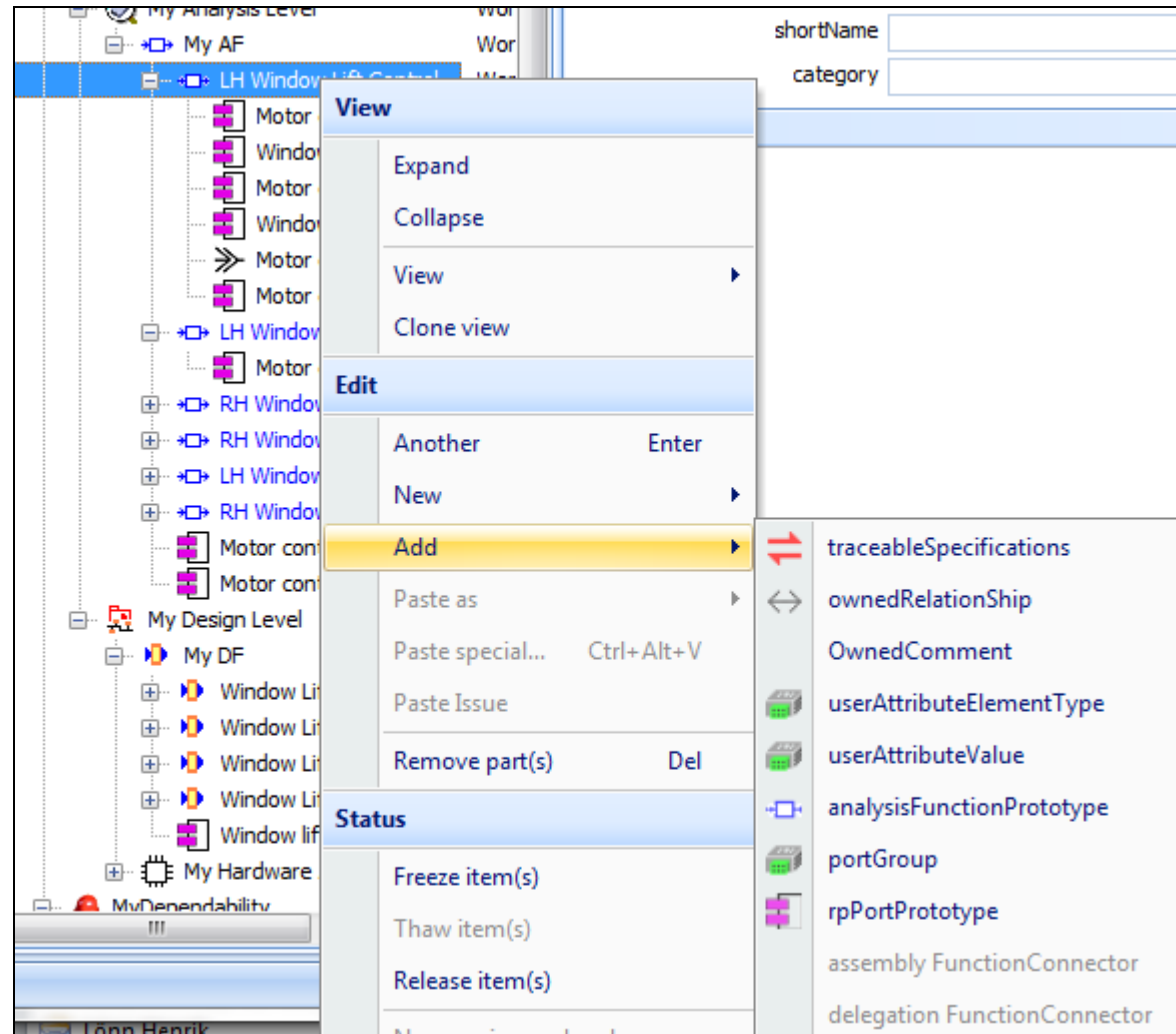
Effective modeling abilities

- Every model structure is defined in a specific context, creating hierarchical prototype/type structures.
- This gives ability to effective build
 - Top down = add prototypes and create types, then add prototypes to type or
 - Bottom up = add prototypes where the type in turn may have prototype structure. The type is always defined for a prototype: the user must select the type when a prototype is created

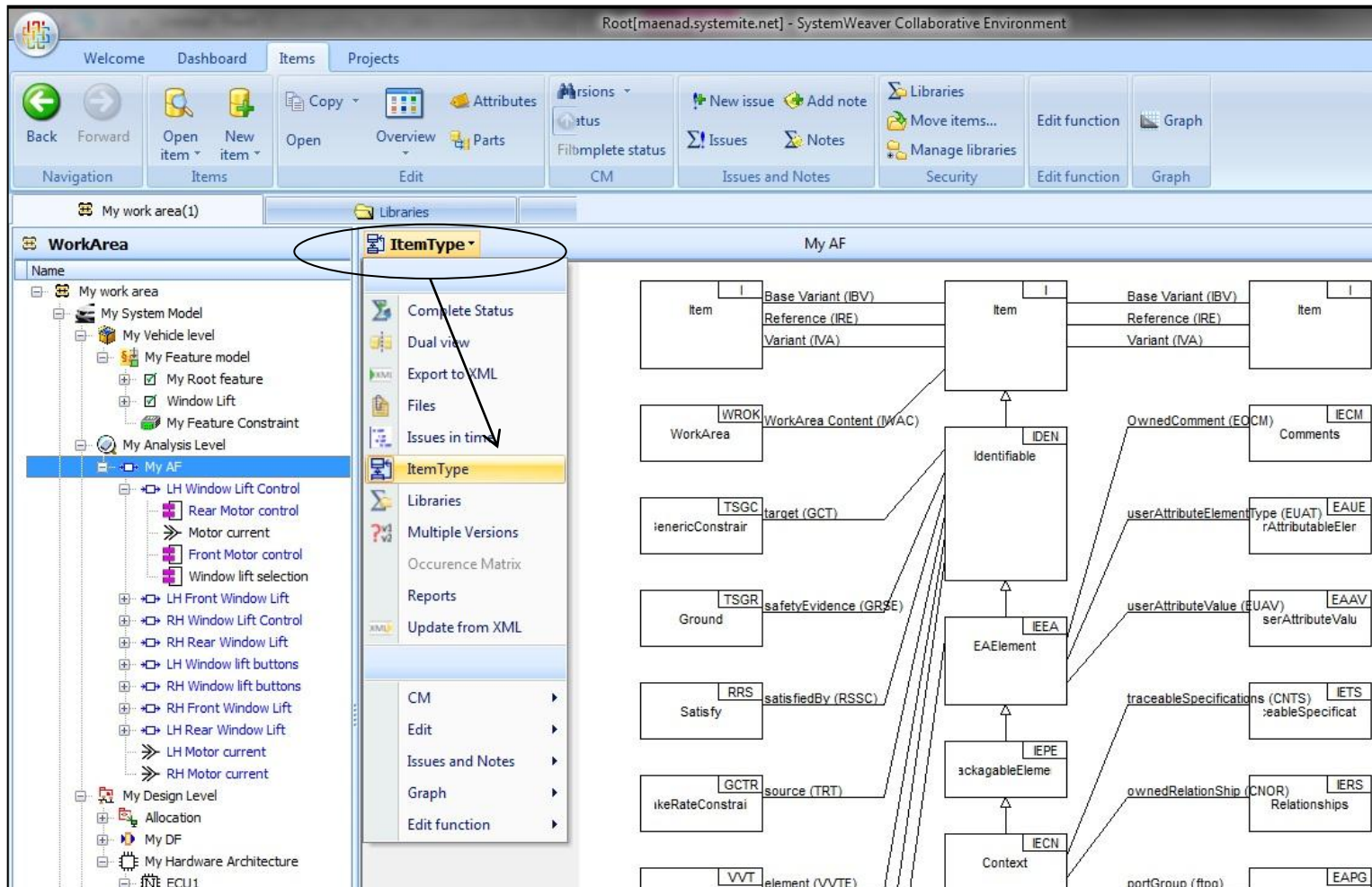


Motivation: Strong meta model

- Strong meta model: cannot do things that are not in the meta model. Each option is guided by the meta model. As an example: when you right click in the tree, the only options that the implemented EAST-ADL meta model accepts can be chosen.
- This example: LH Window Lift Control – is an EA AnalysFunctionType

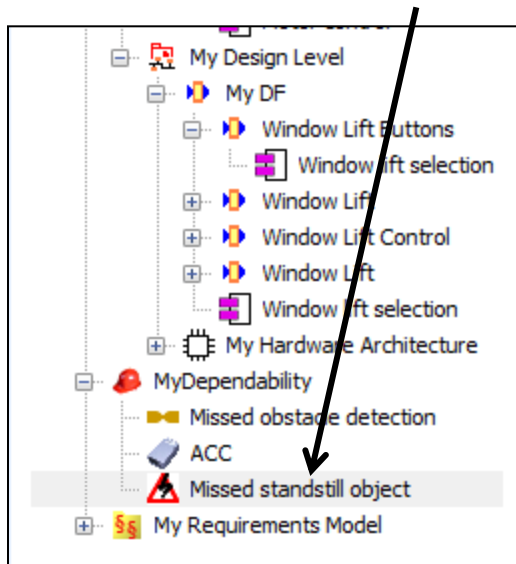


Documentation by “click” – Inheritance information



Documentation by “click” – Model information

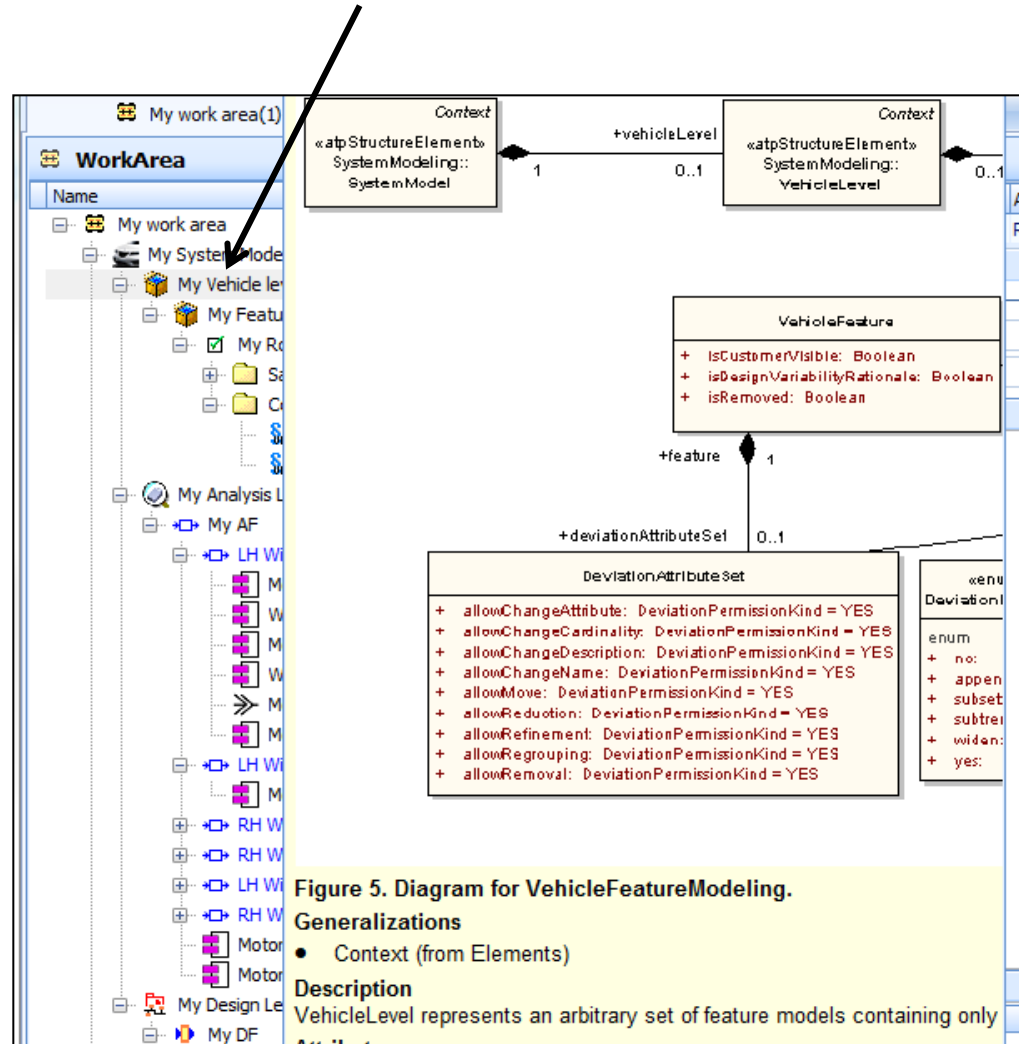
- Hold down alt and select by click on an interesting object. In this example; the Missed standstill object: then meta model information on the right appears.



Name	Last Changed	Last Changed By	Creation Date	Access	Own		
Missed standstill object	2011-05-19	admin	2011-04-18	Read/Write	admin		
classificatio	shortName	category	Contollability	ExposureCl.	ASILKind (n	SeverityCla:	ASILKind
			C3	E2	ASIL_B	S3	ASIL_B ...
<Generic description of the Hazardous event>							

Documentation by “click” – Meta model

- In the same way you can also get info on the (EAST-ADL) meta model by alt + shift + click on the interesting context, in this example "My Vehicle level"
- Here we have added info from the EAST-ADL2 specification



3. Portconcept & graphical support

EAST-ADL

- FunctionFlowPort has the attribut "direction" which decides the direction of the FlowPort.

SystemWeaver

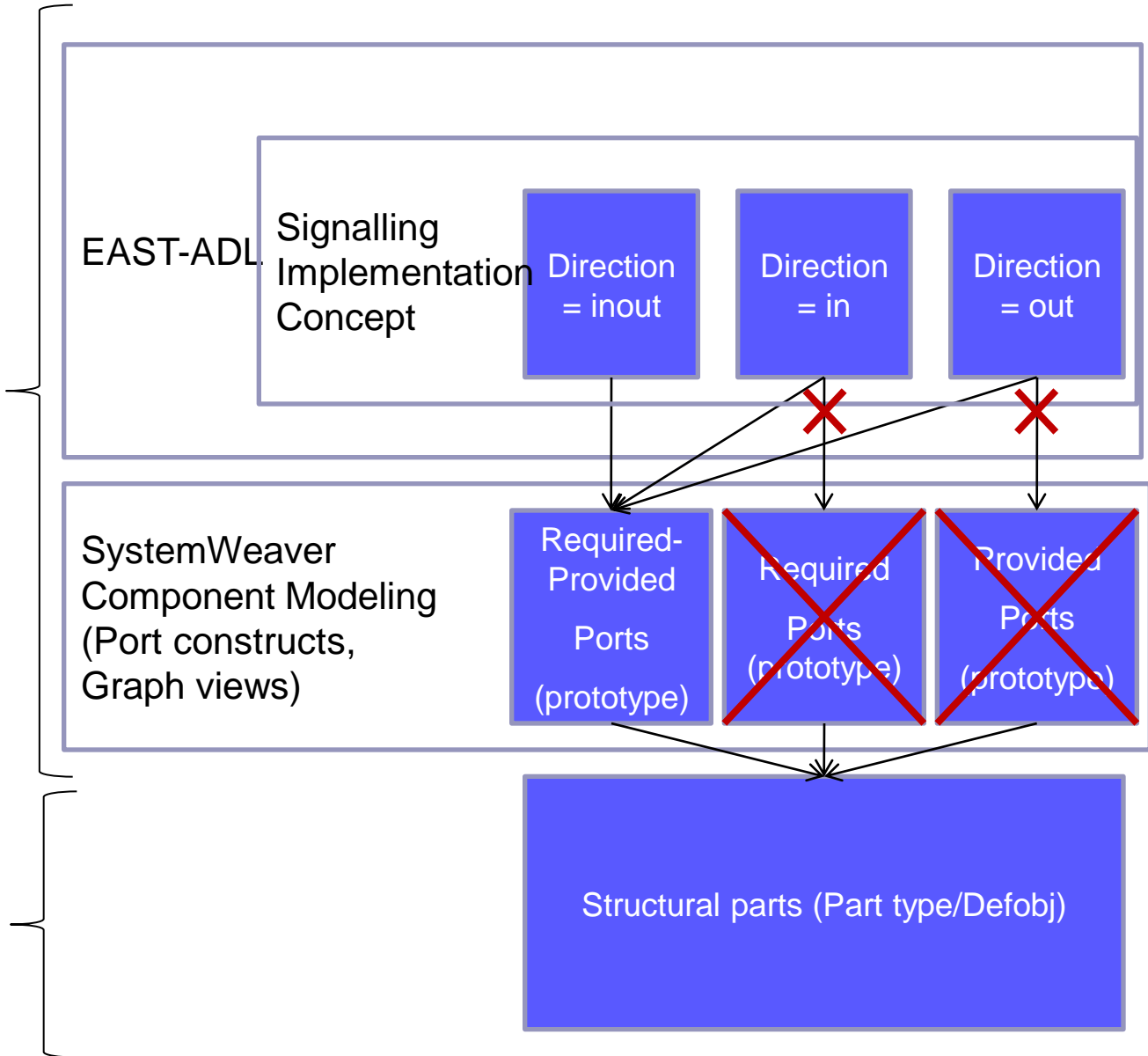
- Classical approach is that the portdirection is decided when the port is created ("part" in systemweaver)– and you cannot change direction on ports after the prototypes (parts in SystemWeaver) is created
 - Reason 1: No need for consistence control of port directions; changing direction means that the port must first be removed, in turn meaning that all connections must also be removed (external assembly, internal delegation)
 - Reason 2: Especially in large projects it is always problematic when ambiguous ports occur
- Current graph implementation where the attribute solution is used does not indicate direction, since a basic in/out construct is used for all port types.

Adaptation

- meta models
- plug in views

SystemWeaver Platform

- meta meta model
- platform concepts
- std views



But you can set an attribute that shows the direction in the attributes view.

The screenshot displays a software interface with two main panels. The left panel, titled 'WorkArea', shows a hierarchical tree structure. The right panel, titled 'Attributes', shows a table of attributes for a selected component.

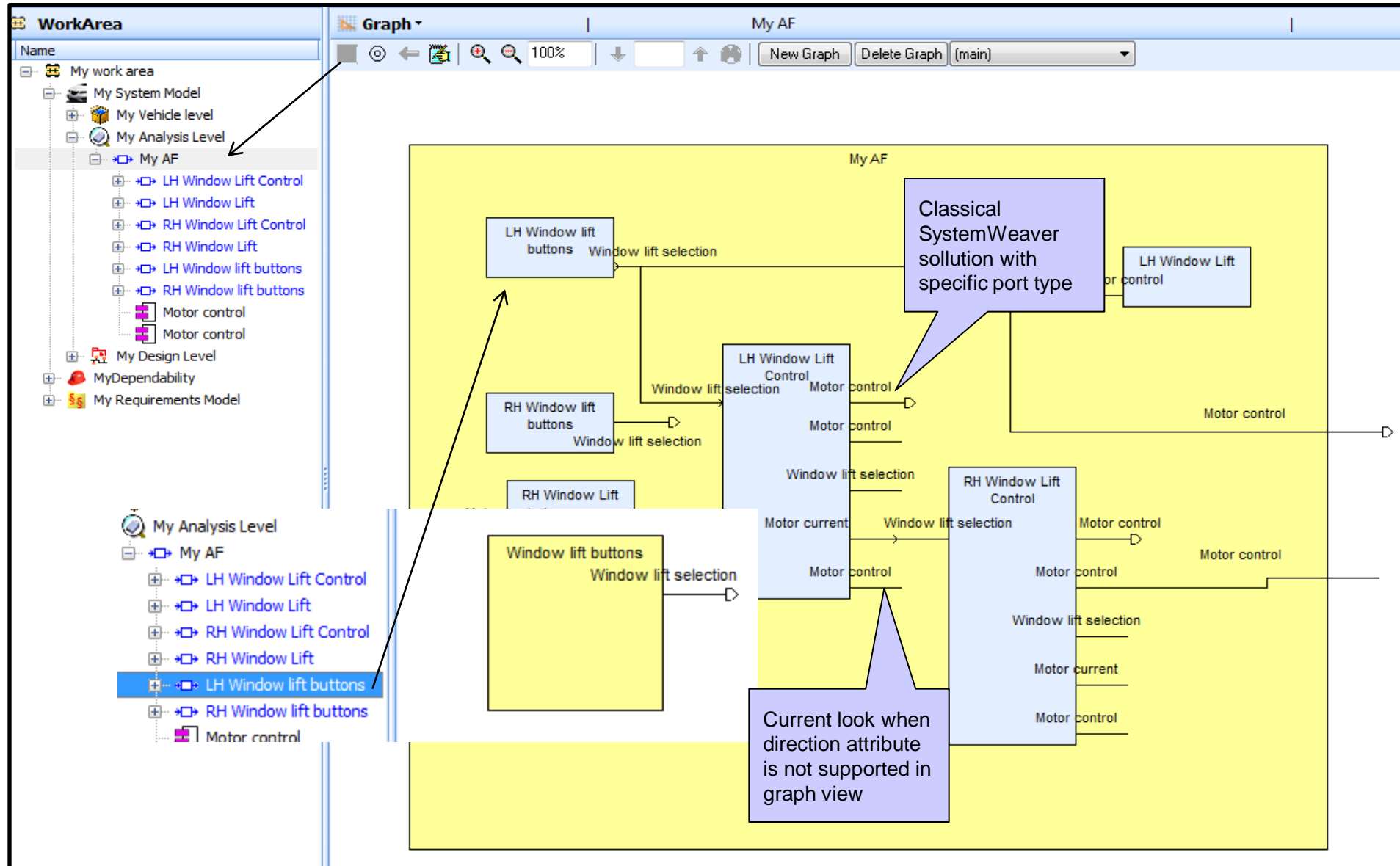
WorkArea Hierarchy:

- My work area(1)
 - My System Model
 - My Vehicle level
 - My Analysis Level
 - My AF
 - LH Window Lift Control
 - Motor control
 - Window lift selection
 - Motor current
 - Motor control
 - Motor control
 - Window lift selection
 - LH Window Lift
 - RH Window Lift Control
 - RH Window Lift
 - LH Window lift buttons
 - RH Window lift buttons

Attributes Table:

Name	EADirection
Motor control	out
Window lift selection	
Motor current	inout
Motor control	
Motor control	
Window lift selection	

System Weaver Graph Editor visualizing



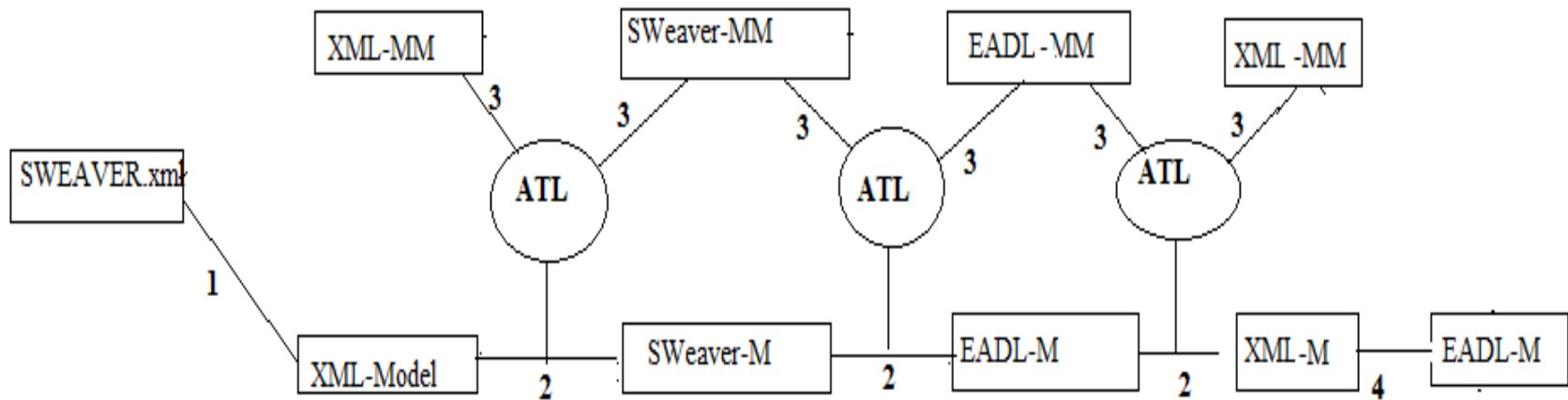
4 File exchange – choosing between 2 options

- File exchange – choosing between 2 options
 - A XML-transformation technique, with help from the ATL – doing model transformation.
 - Or an separate application/plugin based on only a DOM

Result of the research – the Model transformation approach.

- As a result of the researchs made, ATL (Atlas transformation language) is chosen as the suitable model transformation language
- Eclipse modeling framework is used for building the models and meta-models
- Eclipse is used as developing environemnt
- A total of 8 ATL transformations are needed in order to make the complete bi-directional transformation
- A roundtrip of three elements is possible. No yet decided to use.
- Roughly 1 manyear for implementating the whole MM.

General overview of SW2EA transformation



Legend:

1. transform from XML to XMI format
2. ATL transformation
3. Uses - the ATL uses meta models
4. XMI to XML transformation using different tool