

# MetaEdit+ for EAST-ADL

Concept Presentation

2013



# Concept presentation roadmap

- Short description:
  - MetaEdit+ Modeler – A modeling tool for EAST-ADL language
  - MetaEdit+ Workbench – A metamodeling tool to specify EAST-ADL
  - Industry-strength: multiple concurrent users, 4 billion elements per project
- EAST-ADL support
  - Native metamodel support: concepts, constraints, checkings, notation
  - Views: Feature, FAA, FDA, HDA, Requirements, Error, Environment, etc.
  - Representations: Diagram, Table, Matrix, Various trees
- MetaEdit+ integration capabilities
  - EAST-ADL XML interchange format
  - Tool specific formats (Simulink mdl, UPPAAL, SPIN etc. native formats)
  - Programmable API that other tools or plug-ins can use
    - Eclipse plug-in
    - Visual Studio extension

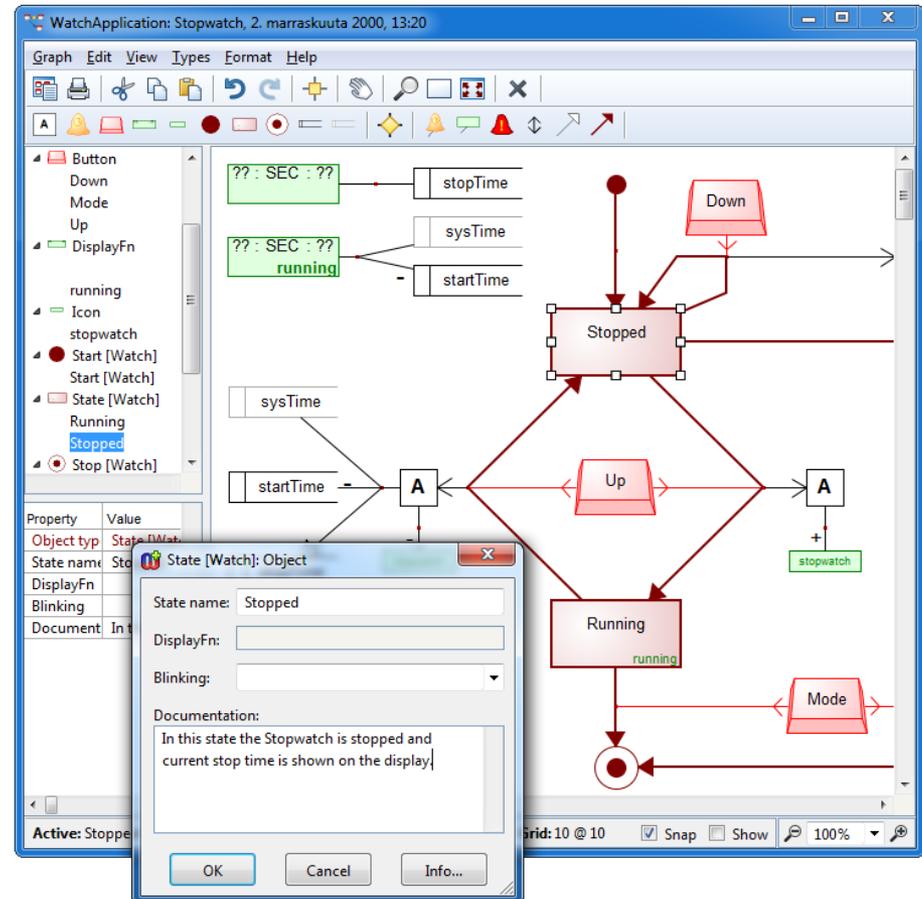
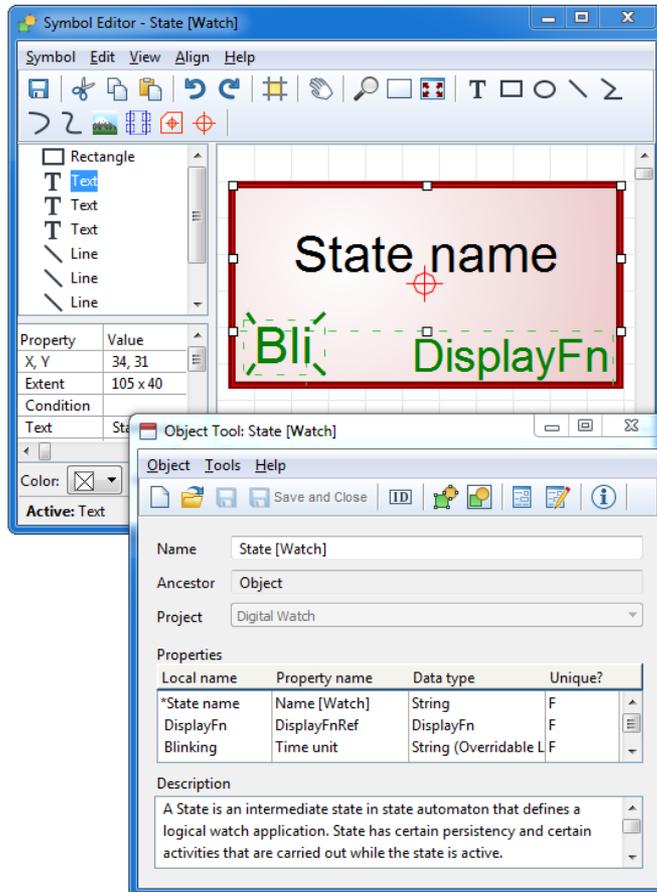
# MetaEdit+ environment

Workbench

Design your language  $\Rightarrow$

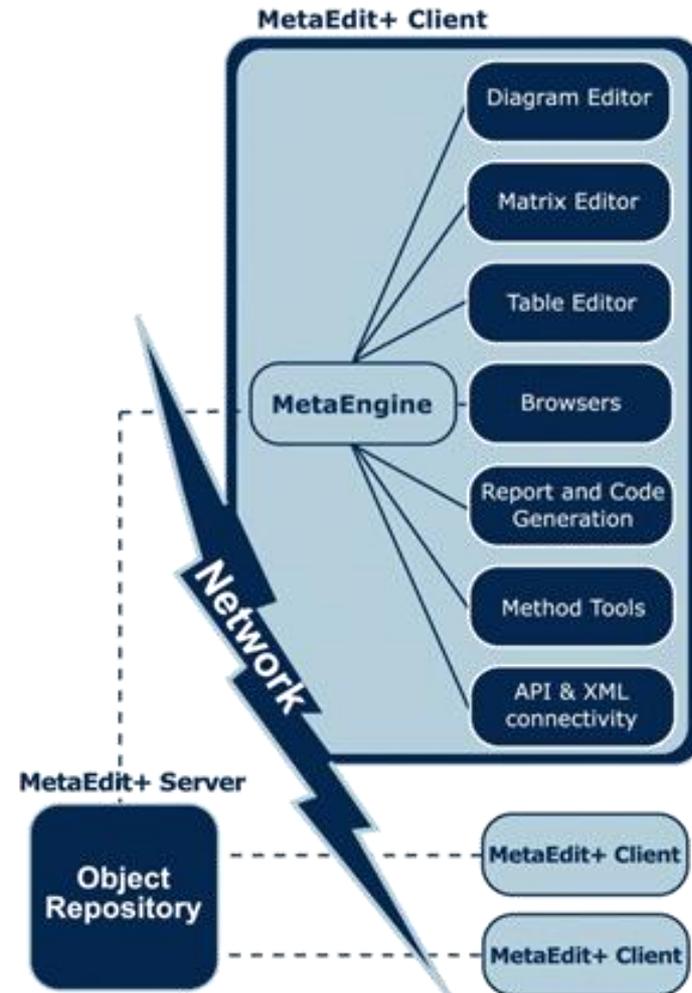
Modeler:

Use your language



# Architecture: Multi-user & multi-platform

- Windows
- Linux
- Mac OS X



# Tool support for EAST-ADL covers

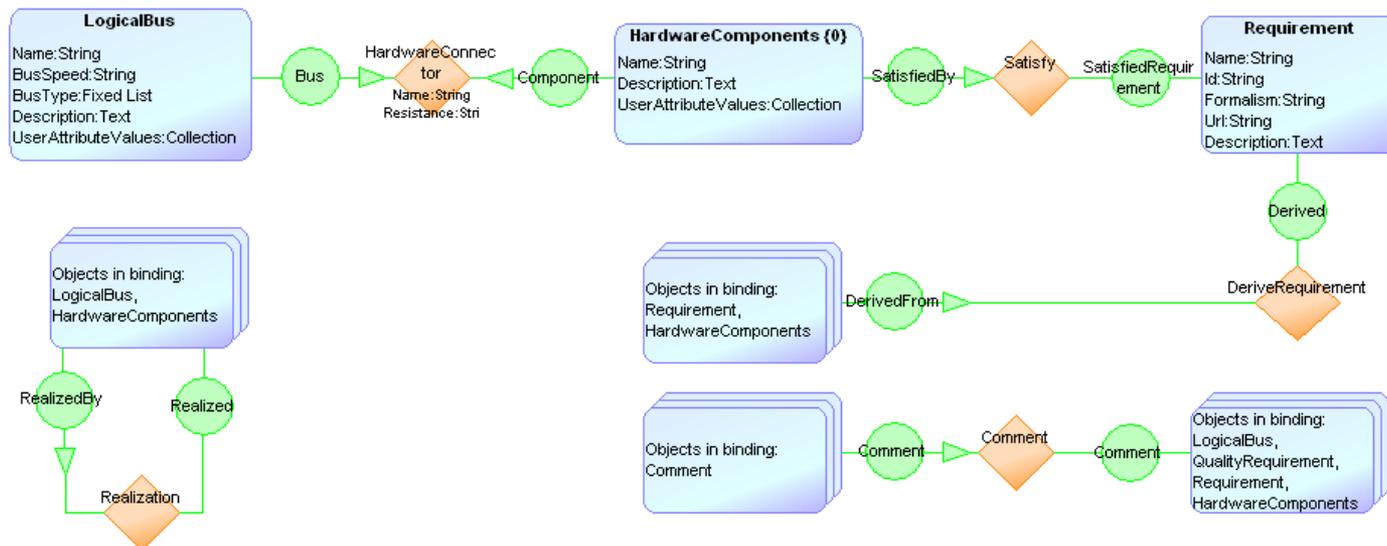
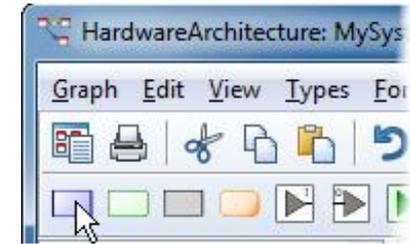
- Language concepts
- Constraints
- Checking rules
- Notation
- Generators
  - EAXML, docs, Simulink, requirements, etc.
- Dedicated tool behavior dedicated to
  - dialogs, toolbar, icons in toolbar etc

# Tool support for EAST-ADL covers

- Language concepts

- Are defined in the metamodel

- Become available in the toolbars, dialogs etc.



# Tool support for EAST-ADL covers

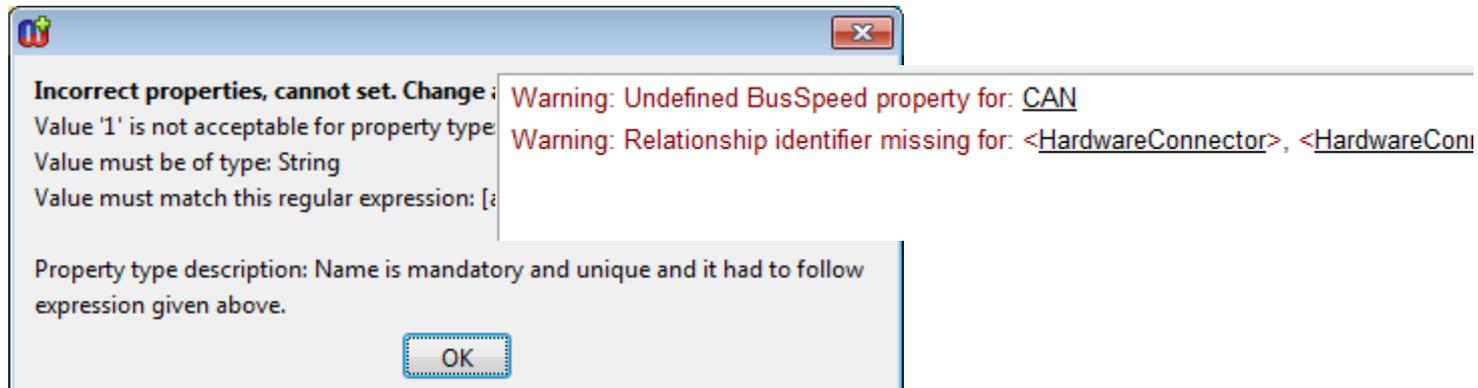
- Constraints

- E.g. IO can connect to IO only, In port to Out Port, etc



- Checking rules

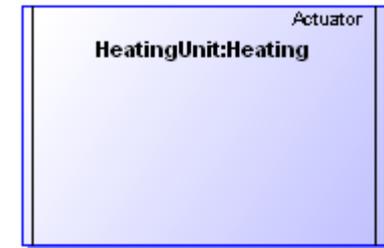
- Completeness, consistency, naming rules etc.



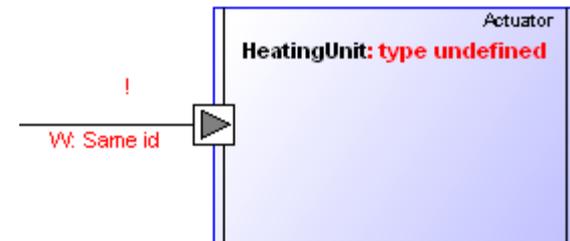
# Tool support for EAST-ADL covers

## ● Notation

- E.g. an Actuator has two vertical lines on the right and left borders of the rectangle etc



- Notation used to annotate errors, missing data etc.

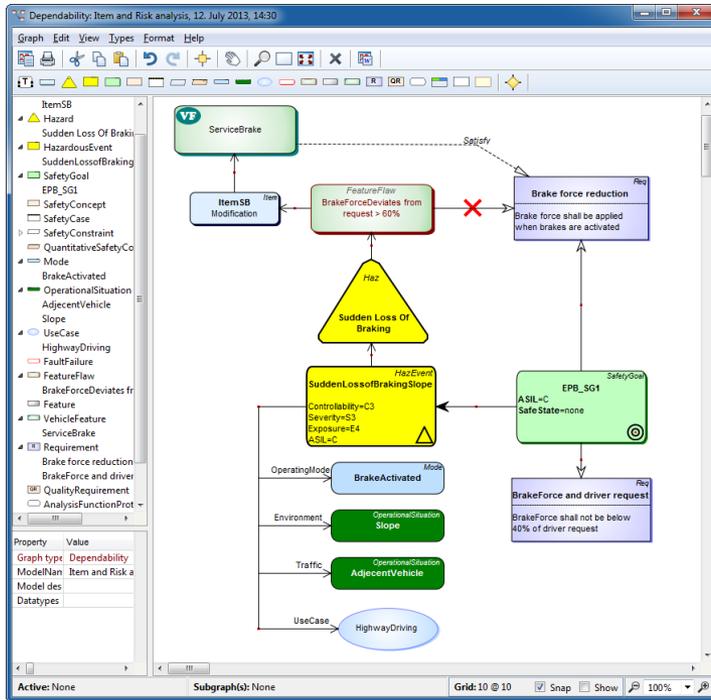


# MetaEdit+ & EAST-ADL

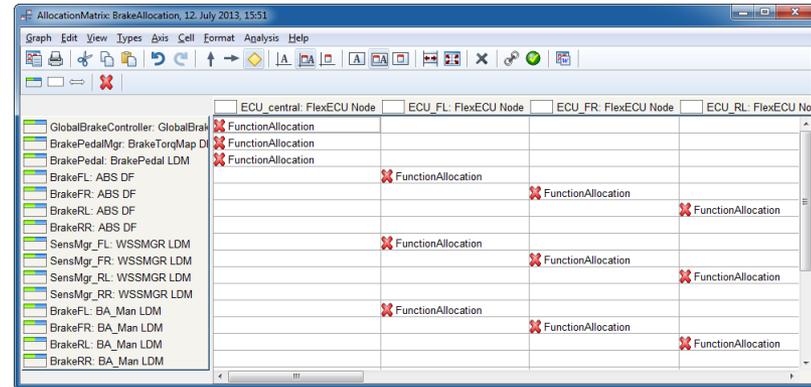
- Implemented views/languages
  - System Model
  - Vehicle Feature Model
  - Functional Analysis Architecture
  - Functional Design Architecture
  - Hardware Design Architecture
  - Requirements Model
  - Package structure
  - Dependability
  - Error modeling
  - Error Behaviors, etc.



# Sample models 2

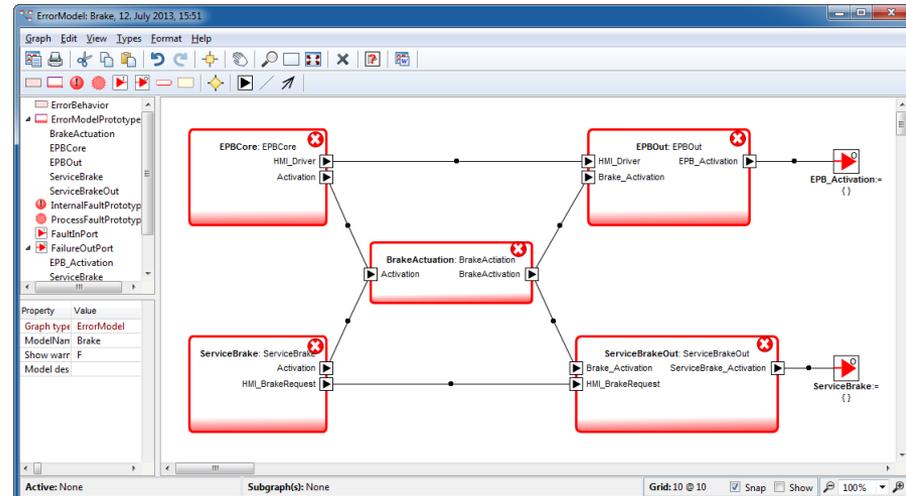


Dependability



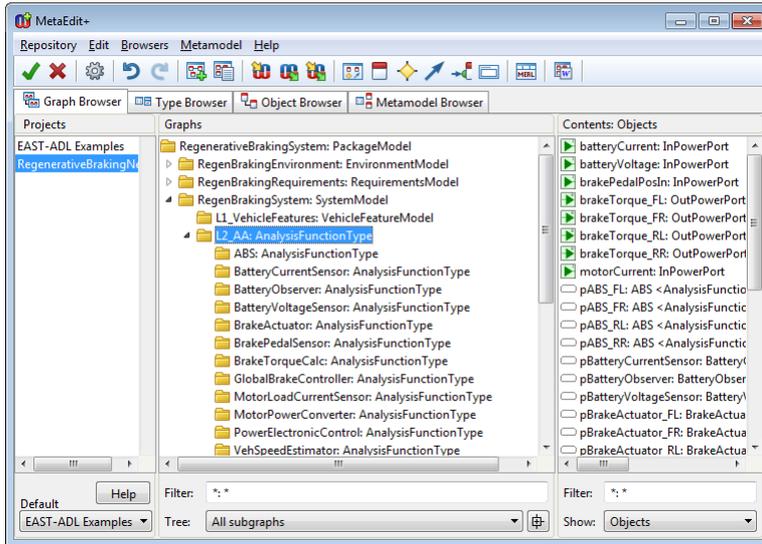
	ECU_central: FlexECU Node	ECU_FL: FlexECU Node	ECU_FR: FlexECU Node	ECU_RL: FlexECU Node
GlobalBrakeController: GlobalBrake	FunctionAllocation			
BrakePedalMgr: BrakeTorqMap D	FunctionAllocation			
BrakePedal: BrakePedal LDM	FunctionAllocation			
BrakeFL: ABS DF		FunctionAllocation		
BrakeFR: ABS DF			FunctionAllocation	
BrakeRL: ABS DF				FunctionAllocation
BrakeRR: ABS DF				FunctionAllocation
SensMgr_FL: WSSMGR LDM	FunctionAllocation			
SensMgr_FR: WSSMGR LDM		FunctionAllocation		
SensMgr_RL: WSSMGR LDM			FunctionAllocation	
SensMgr_RR: WSSMGR LDM				FunctionAllocation
BrakeFL_BA_Man LDM	FunctionAllocation			
BrakeFR_BA_Man LDM		FunctionAllocation		
BrakeRL_BA_Man LDM			FunctionAllocation	
BrakeRR_BA_Man LDM				FunctionAllocation

Function Allocation

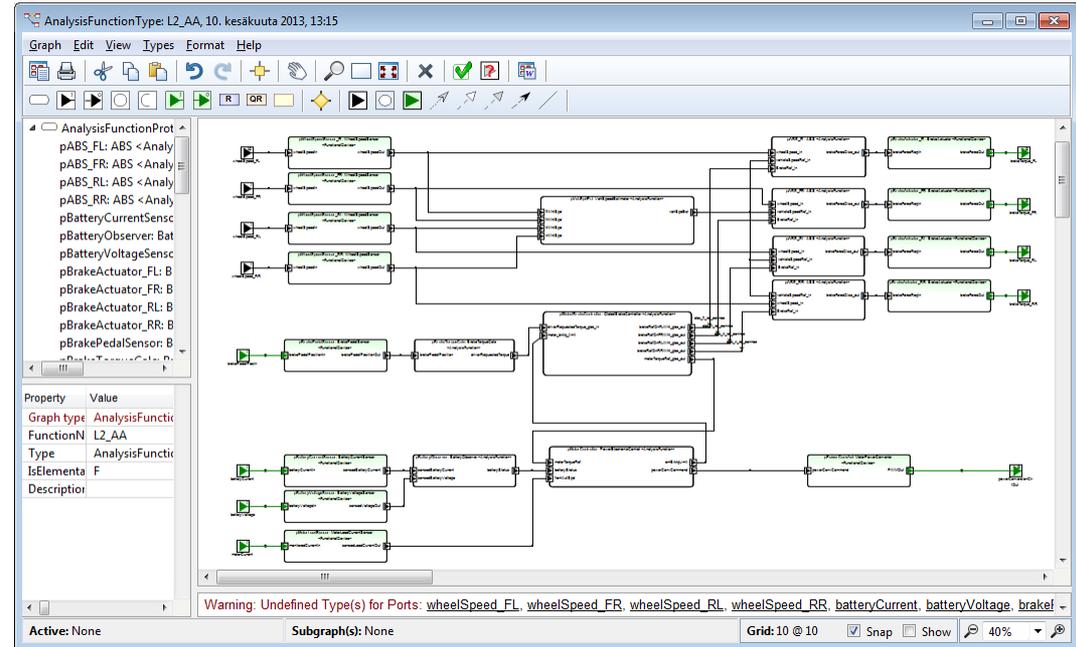


Error Models

# Representations

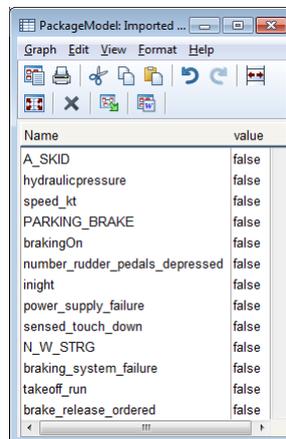


MetaEdit+ interface showing a tree view of a project structure. The left pane shows the project hierarchy, including 'RegenerativeBrakingSystem' and 'L2\_AA: AnalysisFunctionType'. The right pane shows the contents of the selected object, listing various ports and sensors.



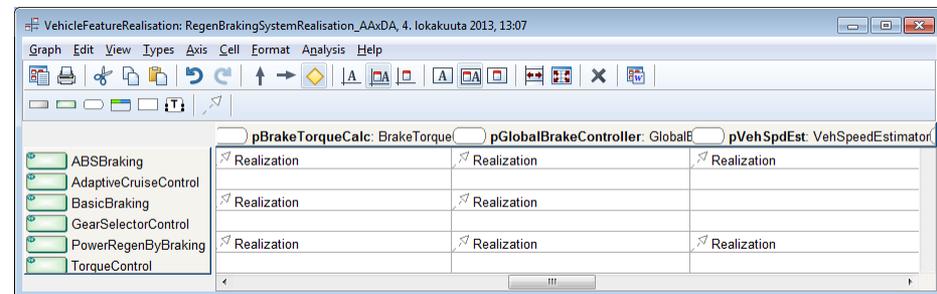
MetaEdit+ interface showing a diagram view of an analysis function type. The diagram illustrates the internal structure and connections of the function, including various ports and sensors. A warning message is visible at the bottom: 'Warning: Undefined Type(s) for Ports: wheelSpeed\_FL, wheelSpeed\_FR, wheelSpeed\_RL, wheelSpeed\_RR, batteryCurrent, batteryVoltage, brakeTorque...'.

- Tree views
- Diagrams
- Matrices
- Tables



MetaEdit+ interface showing a table view of a package model. The table lists various parameters and their values.

Name	value
A_SKID	false
hydraulicpressure	false
speed_kt	false
PARKING_BRAKE	false
brakingOn	false
number_rudder_pedals_depressed	false
inight	false
power_supply_failure	false
sensed_touch_down	false
N_W_STRG	false
braking_system_failure	false
takeoff_run	false
brake_release_ordered	false



MetaEdit+ interface showing a realization matrix for a vehicle feature realization. The matrix maps various realization types to specific components.

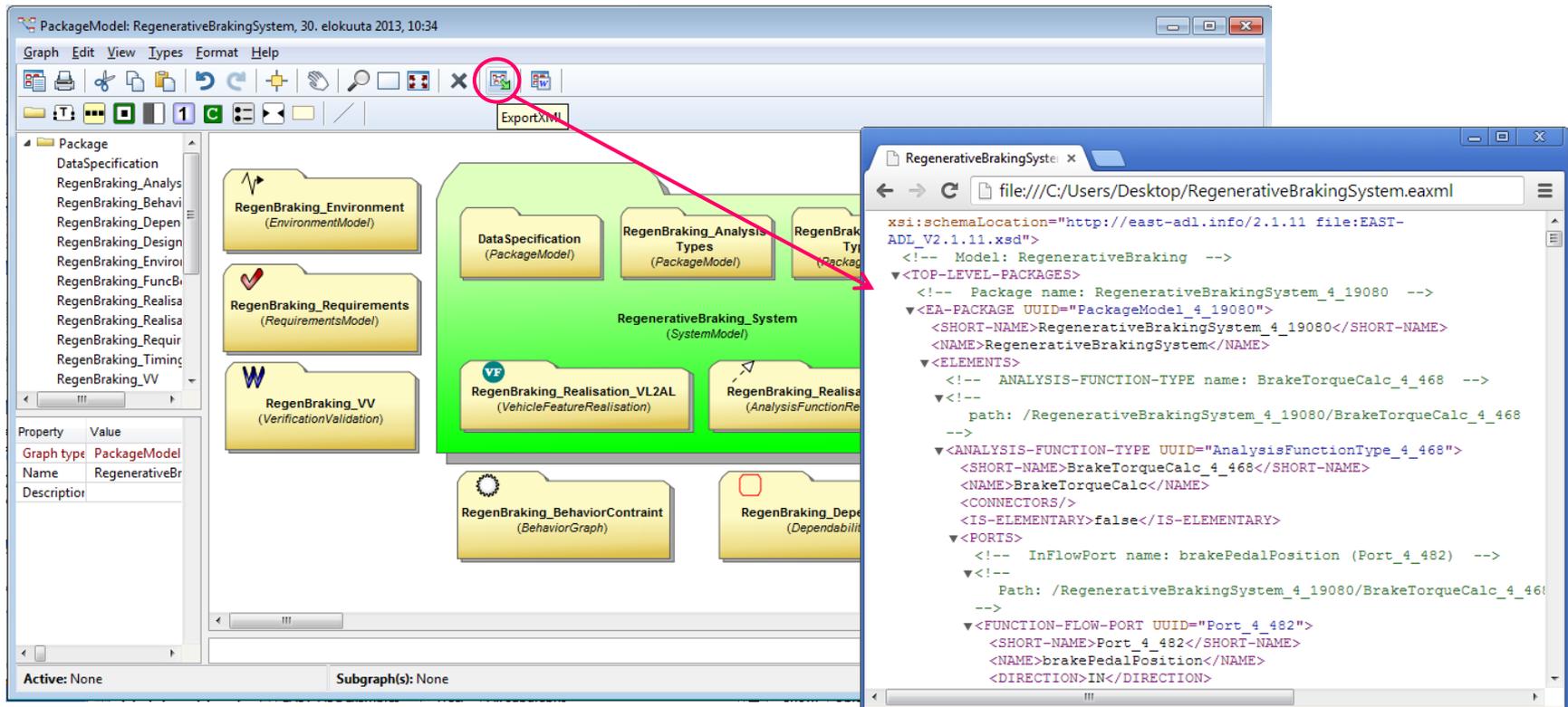
	pBrakeTorqueCalc: BrakeTorque	pGlobalBrakeController: GlobalE	pVehSpdEst: VehSpeedEstimator
ABSBraking	Realization	Realization	Realization
AdaptiveCruiseControl			
BasicBraking	Realization	Realization	
GearSelectorControl			
PowerRegenByBraking	Realization	Realization	Realization
TorqueControl			

# MetaEdit+ integration capabilities

- MetaEdit+ provides a modeling tool and integrated with other tools, namely analysis tools via:
  - EAST-ADL XML interchange format
  - Tool specific formats
    - Simulink, UPPAAL, SPIN, etc.
  - Programmable API that other tools or plug-ins can use, e.g. available for
    - Eclipse integration
    - Visual Studio integration

# EAXML support

## ● Generator for EAXML export



The screenshot displays a software application window titled "PackageModel: RegenerativeBrakingSystem, 30. elokuuta 2013, 10:34". The interface includes a menu bar (Graph, Edit, View, Types, Format, Help) and a toolbar with various icons. A red circle highlights the "Export XML" icon in the toolbar. Below the toolbar, a package model diagram is shown, featuring several sub-packages such as "RegenBraking\_Environment", "RegenBraking\_Requirements", "RegenBraking\_VV", "DataSpecification", "RegenBraking\_Analysis Types", "RegenerativeBraking\_System", "RegenBraking\_Realisation\_VL2AL", "RegenBraking\_Realisation", "RegenBraking\_BehaviorConstraint", and "RegenBraking\_Dependability". A red arrow points from the "Export XML" icon to the "RegenerativeBraking\_System" package.

Overlaid on the right side of the application window is a text editor window titled "RegenerativeBrakingSystem\_eaxml" showing the generated EAXML code. The code is as follows:

```

xsi:schemaLocation="http://east-adl.info/2.1.11 file:EAST-ADL_V2.1.11.xsd"
<!-- Model: RegenerativeBraking -->
<!-- TOP-LEVEL-PACKAGES -->
<!-- Package name: RegenerativeBrakingSystem_4_19080 -->
<EA-PACKAGE UUID="PackageModel_4_19080">
  <SHORT-NAME>RegenerativeBrakingSystem_4_19080</SHORT-NAME>
  <NAME>RegenerativeBrakingSystem</NAME>
  <ELEMENTS>
    <!-- ANALYSIS-FUNCTION-TYPE name: BrakeTorqueCalc_4_468 -->
    <!--
      path: /RegenerativeBrakingSystem_4_19080/BrakeTorqueCalc_4_468
    -->
    <ANALYSIS-FUNCTION-TYPE UUID="AnalysisFunctionType_4_468">
      <SHORT-NAME>BrakeTorqueCalc_4_468</SHORT-NAME>
      <NAME>BrakeTorqueCalc</NAME>
      <CONNECTORS/>
      <IS-ELEMENTARY>false</IS-ELEMENTARY>
      <PORTS>
        <!-- InFlowPort name: brakePedalPosition (Port_4_482) -->
        <!--
          Path: /RegenerativeBrakingSystem_4_19080/BrakeTorqueCalc_4_468
        -->
        <FUNCTION-FLOW-PORT UUID="Port_4_482">
          <SHORT-NAME>Port_4_482</SHORT-NAME>
          <NAME>brakePedalPosition</NAME>
          <DIRECTION>IN</DIRECTION>

```