

MetaEdit+ for EAST-ADL

Concept Presentation

2013

Concept presentation roadmap

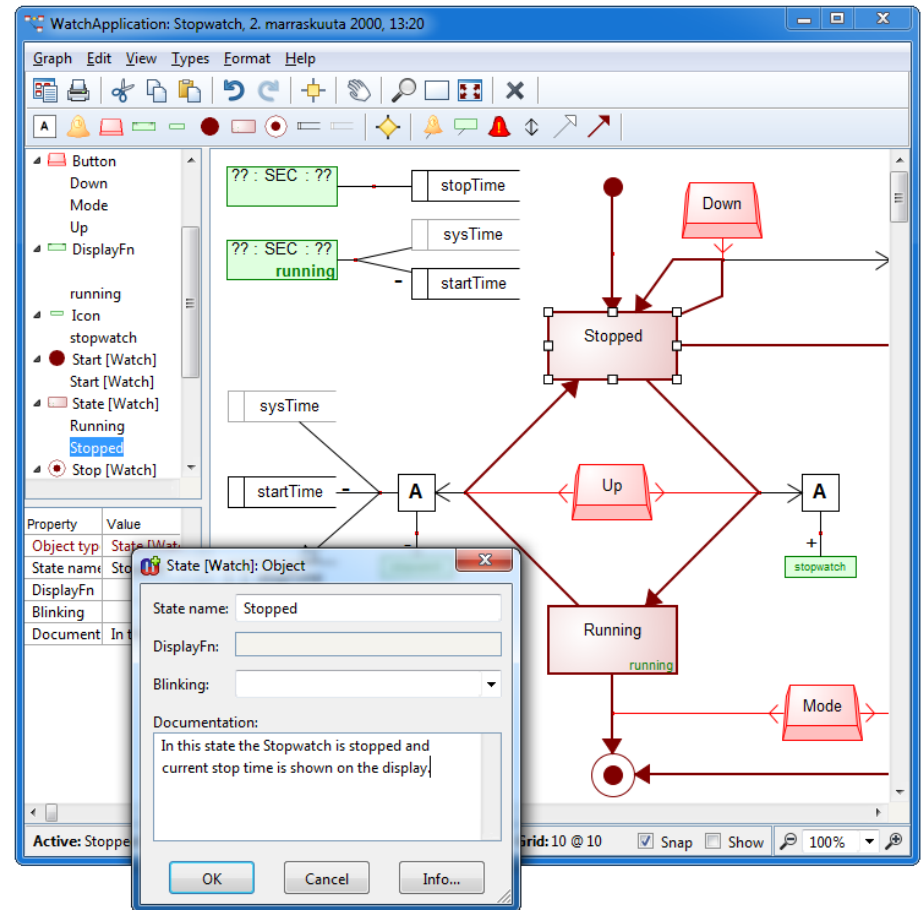
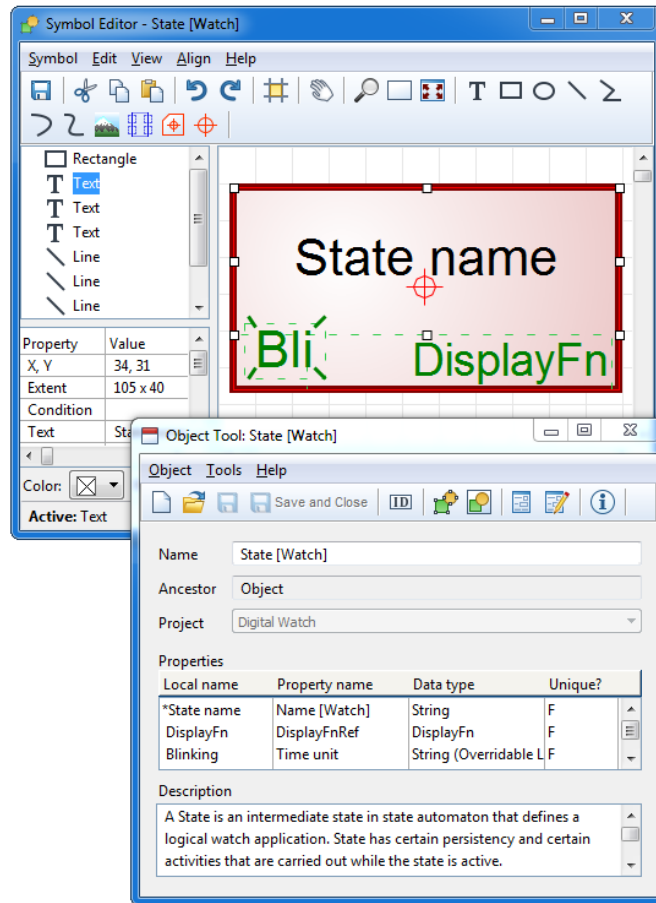
- Short description:
 - MetaEdit+ Modeler – A modeling tool for EAST-ADL language
 - MetaEdit+ Workbench – A metamodeling tool to specify EAST-ADL
 - Industry-strength: multiple concurrent users, 4 billion elements per project
- EAST-ADL support
 - Native metamodel support: concepts, constraints, checkings, 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

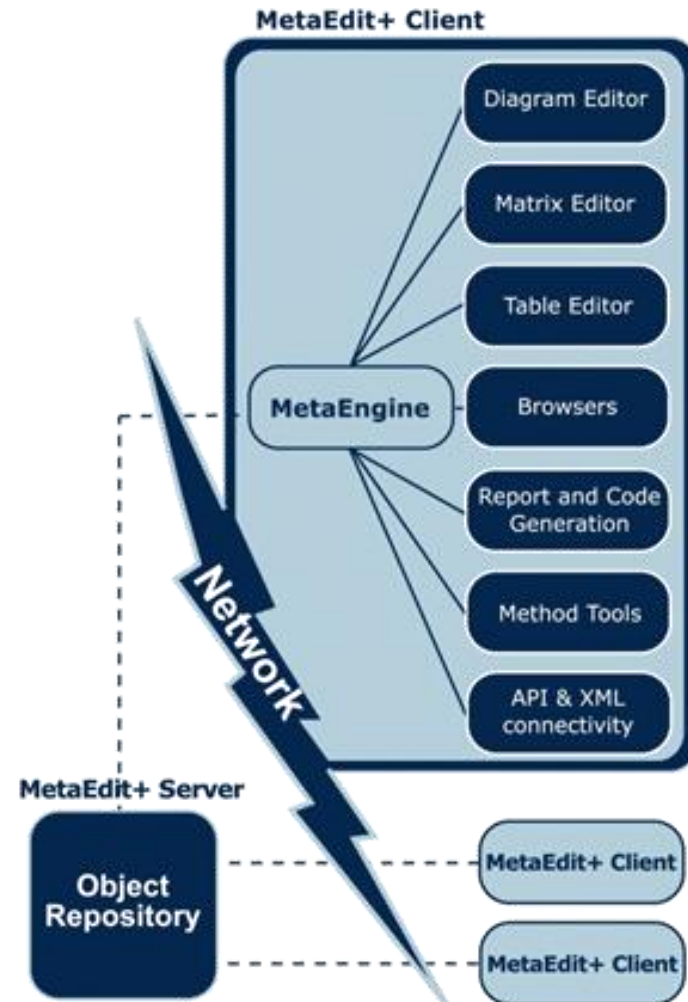
Design your language → Use your language

Modeler:



Architecture: Multi-user & multi-platform

- Windows
- Linux
- Mac OS X



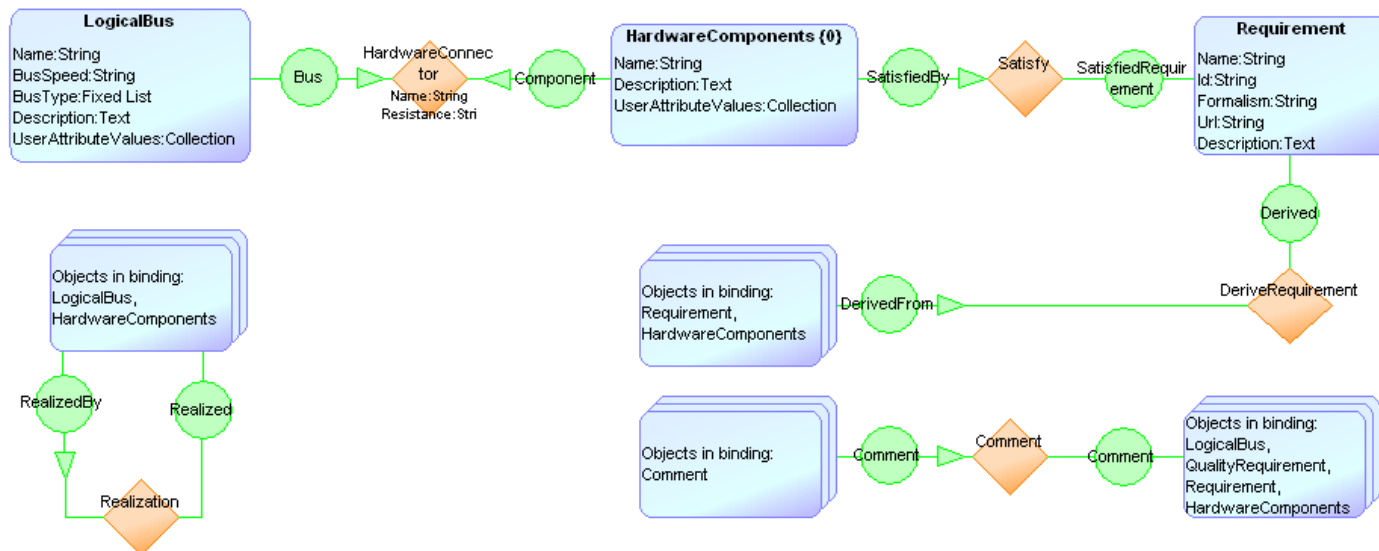
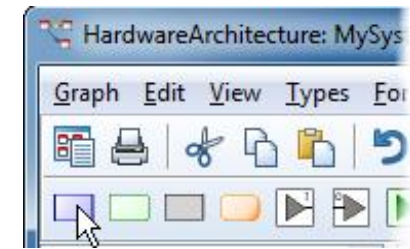
Tool support for EAST-ADL covers

- 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

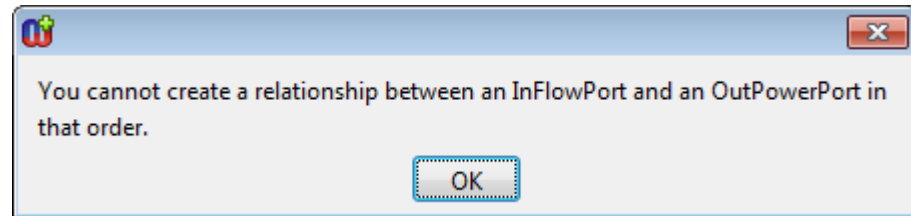
- Are defined in the metamodel
- Become available in the toolbars, dialogs etc.



Tool support for EAST-ADL covers

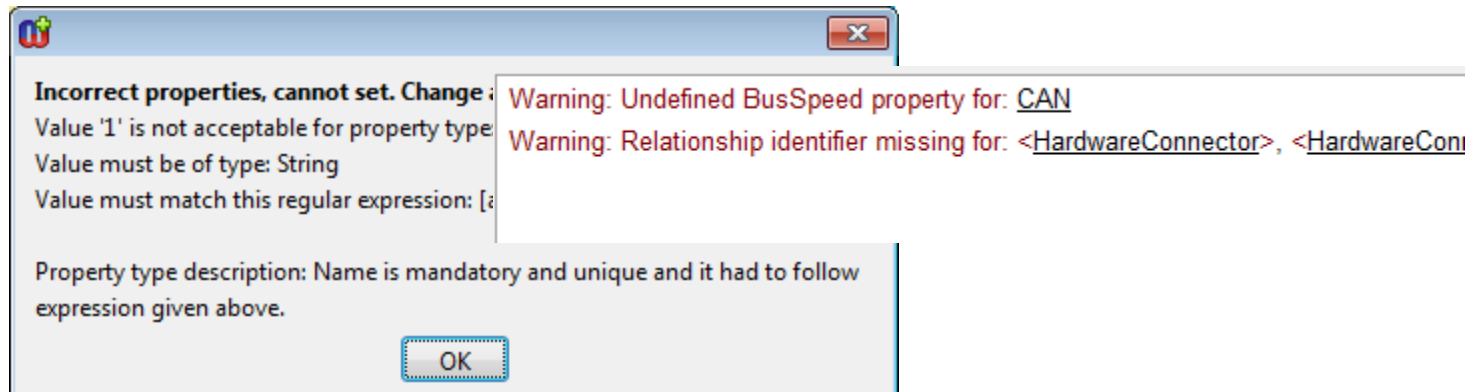
- Constraints

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



- Checking rules

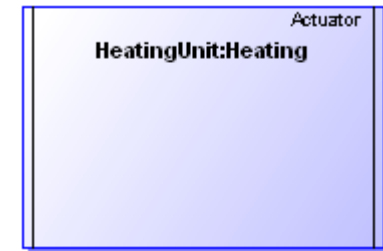
- Completeness, consistency, naming rules etc.



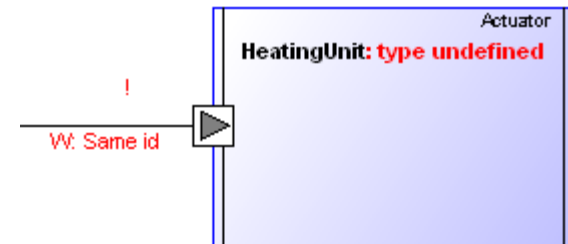
Tool support for EAST-ADL covers

● Notation

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



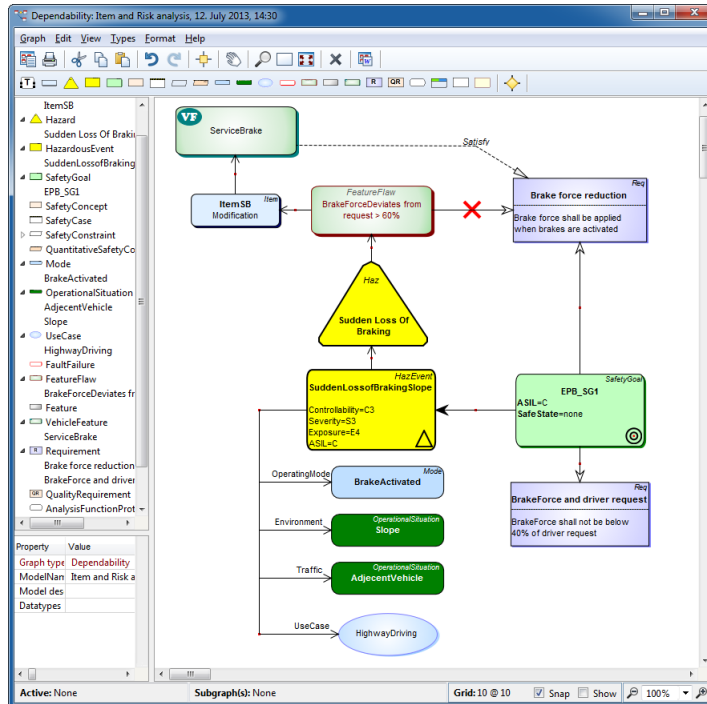
- Notation used to annotate errors, missing data etc.



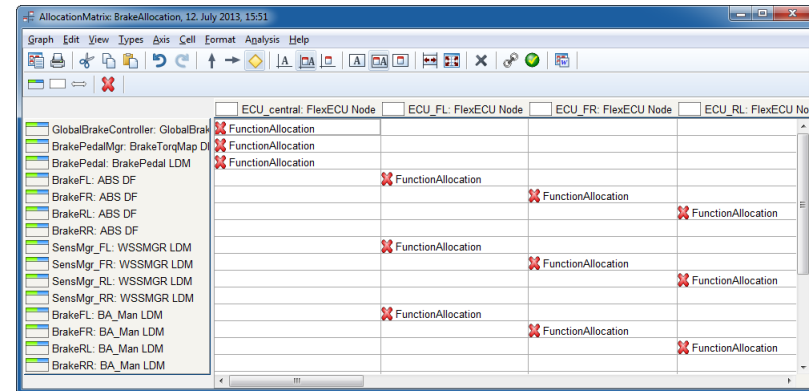
MetaEdit+ & EAST-ADL

- 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.

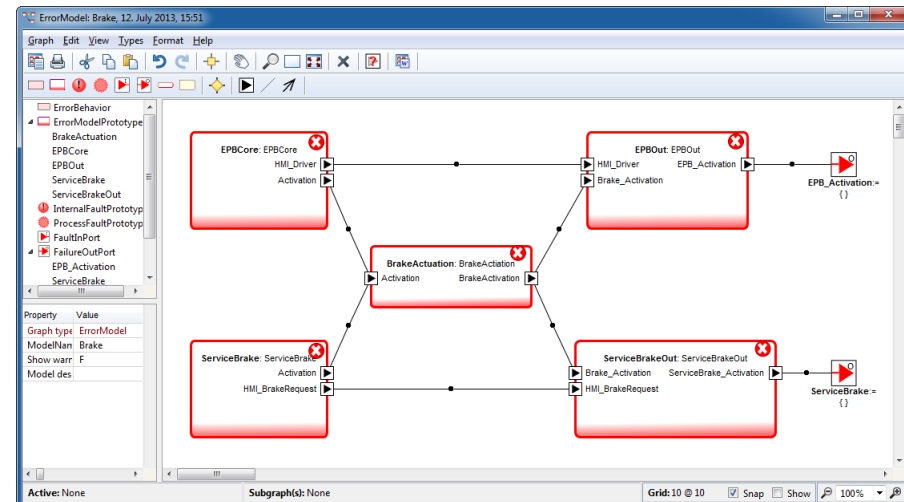
Sample models 2



Dependability

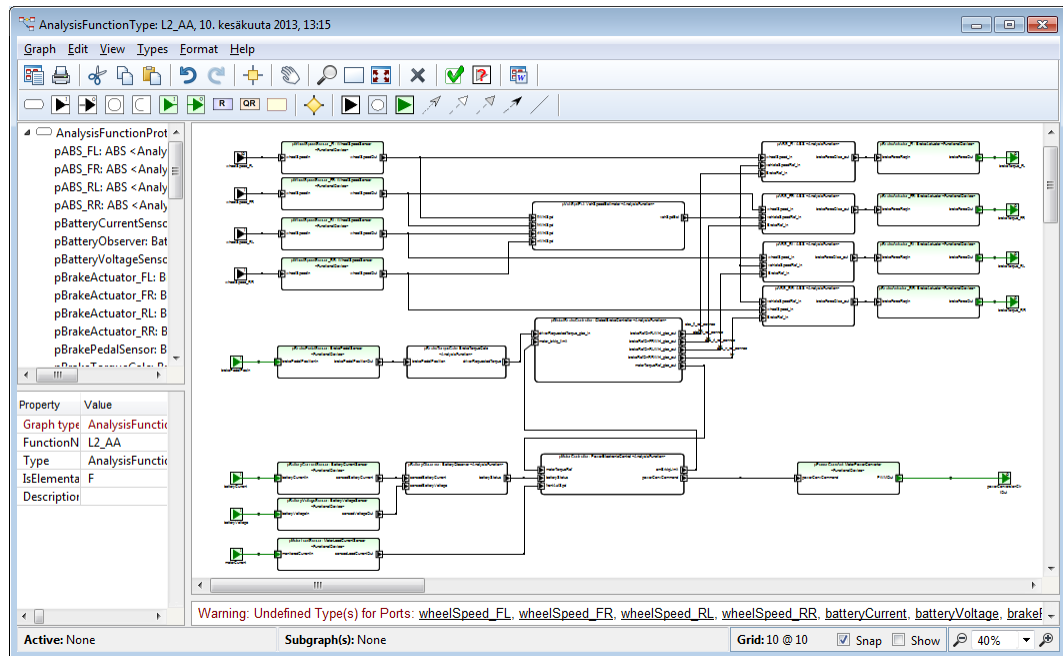
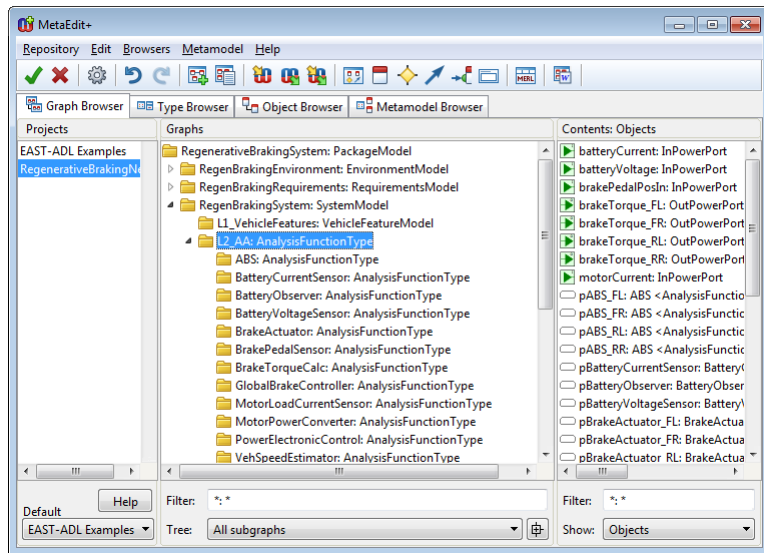


Function Allocation



Error Models

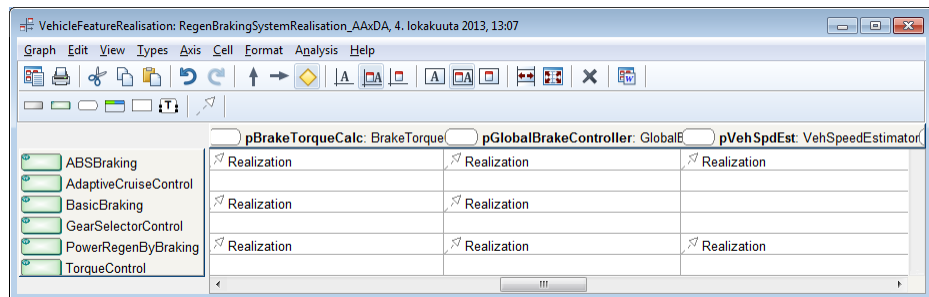
Representations



- Tree views
- Diagrams
- Matrices
- Tables

PackageModel: Imported ...

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

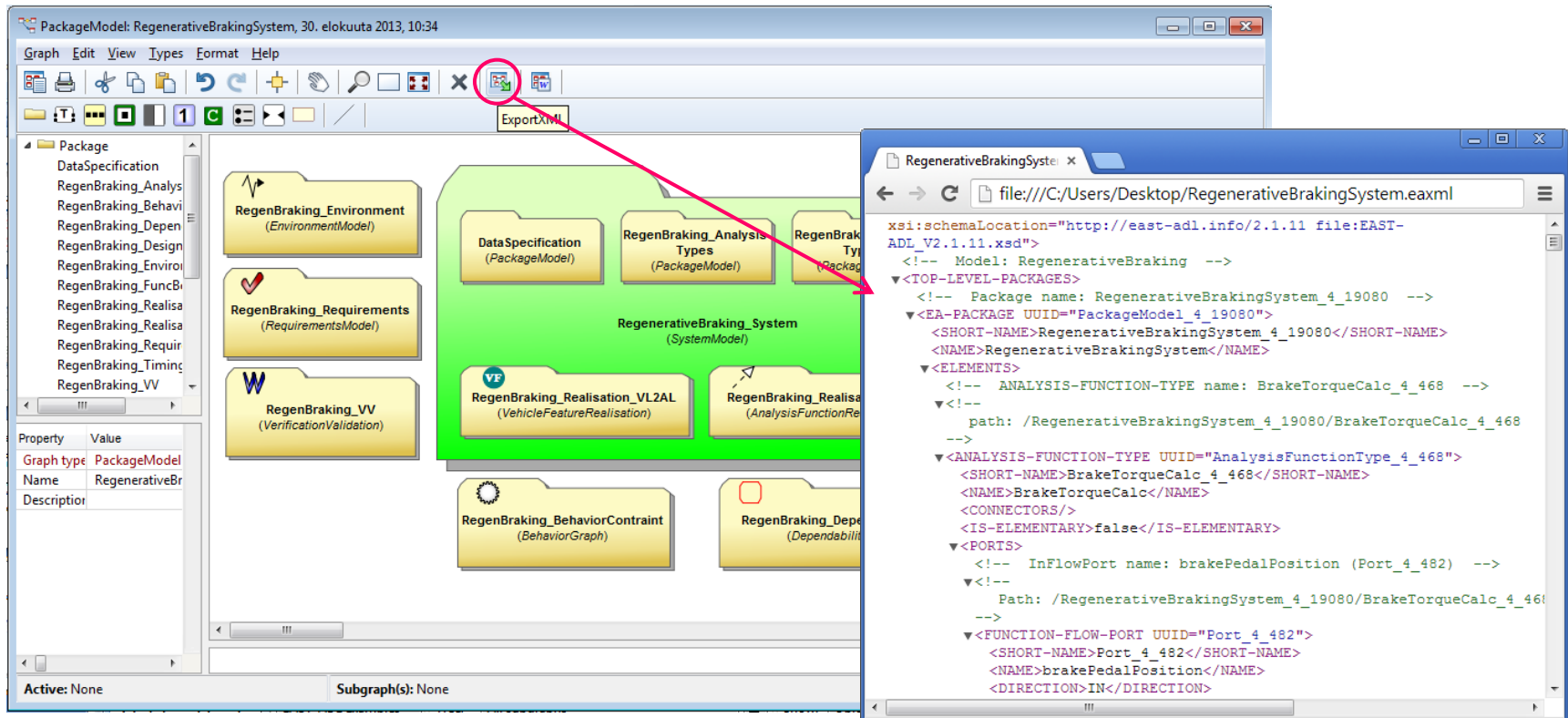


MetaEdit+ integration capabilities

- 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 the 'RegenerativeBrakingSystem' package model editor. The main workspace shows a hierarchical diagram of the system components, including 'RegenBraking_Environment', 'RegenBraking_Requirements', 'RegenBraking_VV', 'DataSpecification', 'RegenBraking_Analysis Types', 'RegenerativeBraking_System', 'RegenBraking_Realisation_VL2AL', 'RegenBraking_Realisation', 'RegenBraking_BehaviorConstraint', and 'RegenBraking_Dependability'. The 'Export XML' button in the toolbar is highlighted with a red circle, and a red arrow points from it to the 'RegenerativeBrakingSystem.eaxml' file in a text editor.

The text editor shows the EAXML file content, which is an XML document representing the system model. The XML structure includes a header with the schema location and a comment, followed by a root element 'RegenerativeBrakingSystem' with a 'PackageModel' type. The XML content is as follows:

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!-- 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>
        </FUNCTION-FLOW-PORT>
      </PORTS>
    </ANALYSIS-FUNCTION-TYPE>
  </ELEMENTS>
</EA-PACKAGE>
  
```