

# Integrating Model Checking and MIC with a QoS-aware Component Middleware Platform

**Gabriele A. Trombetti, Aniruddha Gokhale  
and Douglas C. Schmidt**

*Institute for Software Integrated Systems (ISIS), Vanderbilt University*

**John Hatcliff, Jesse Greenwald  
and Gurdip Singh**

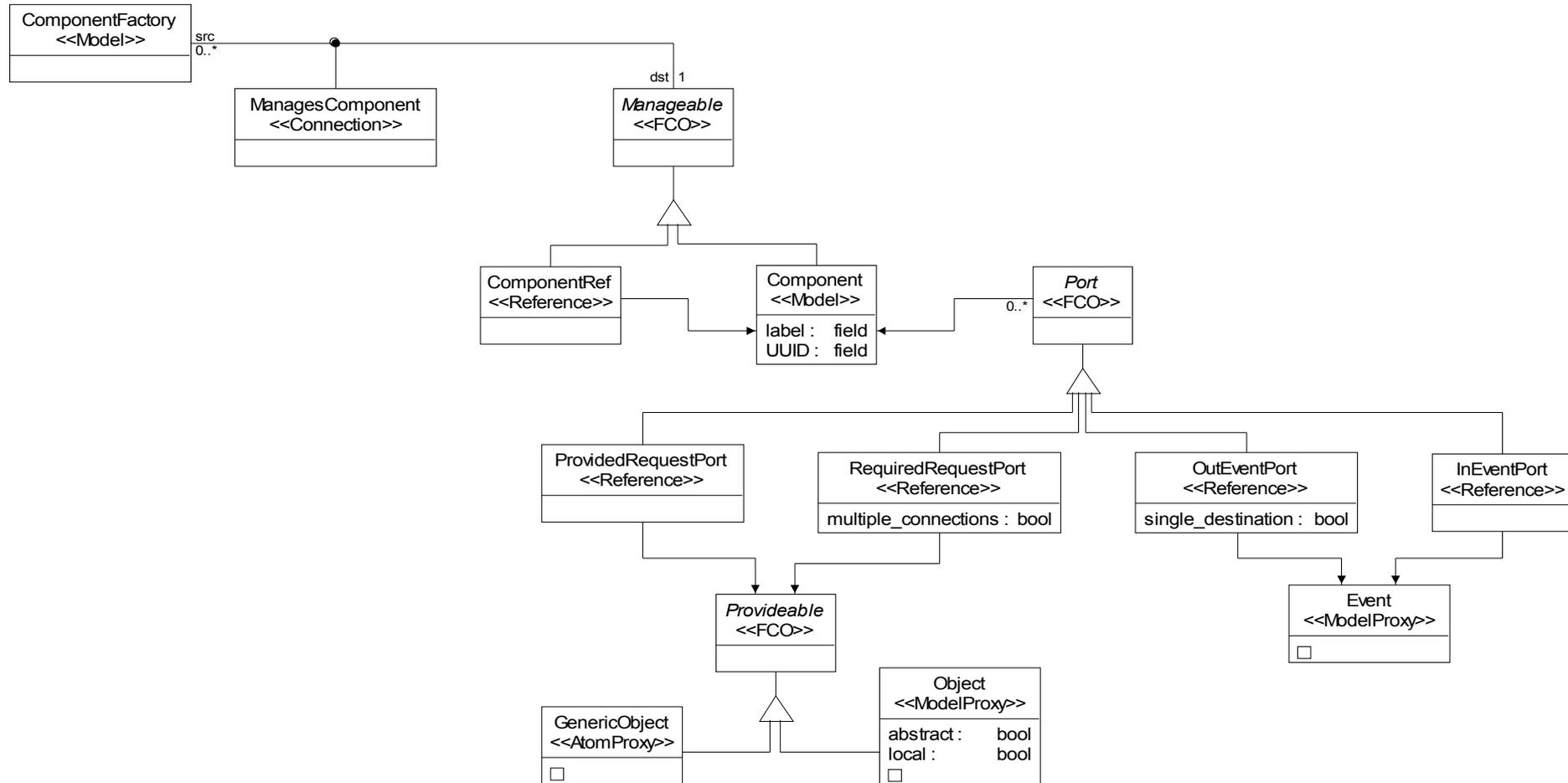
*Computer and Information Systems Department, Kansas State University*



# Platform Independent Component Modelling Language

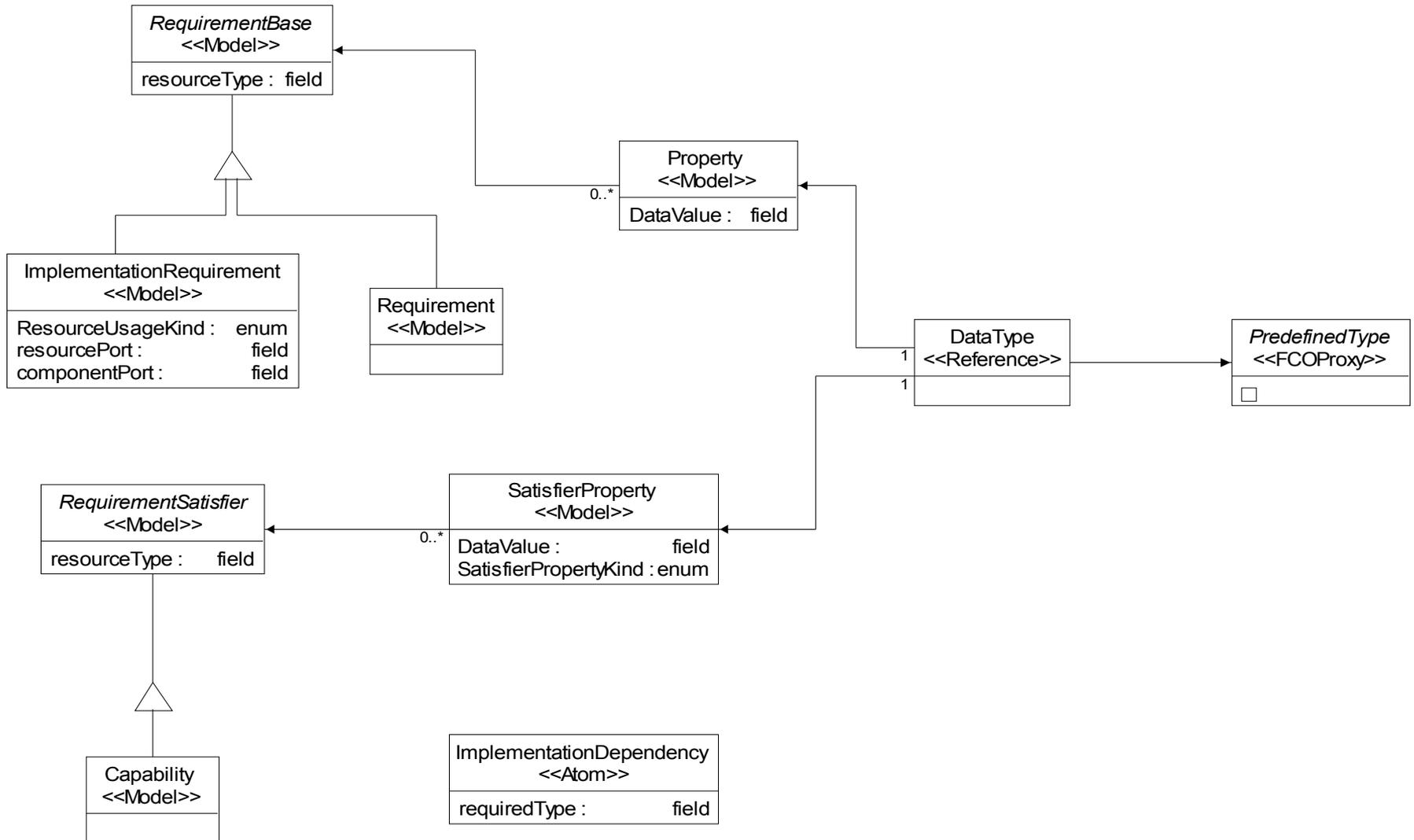


# PICML Meta: Components

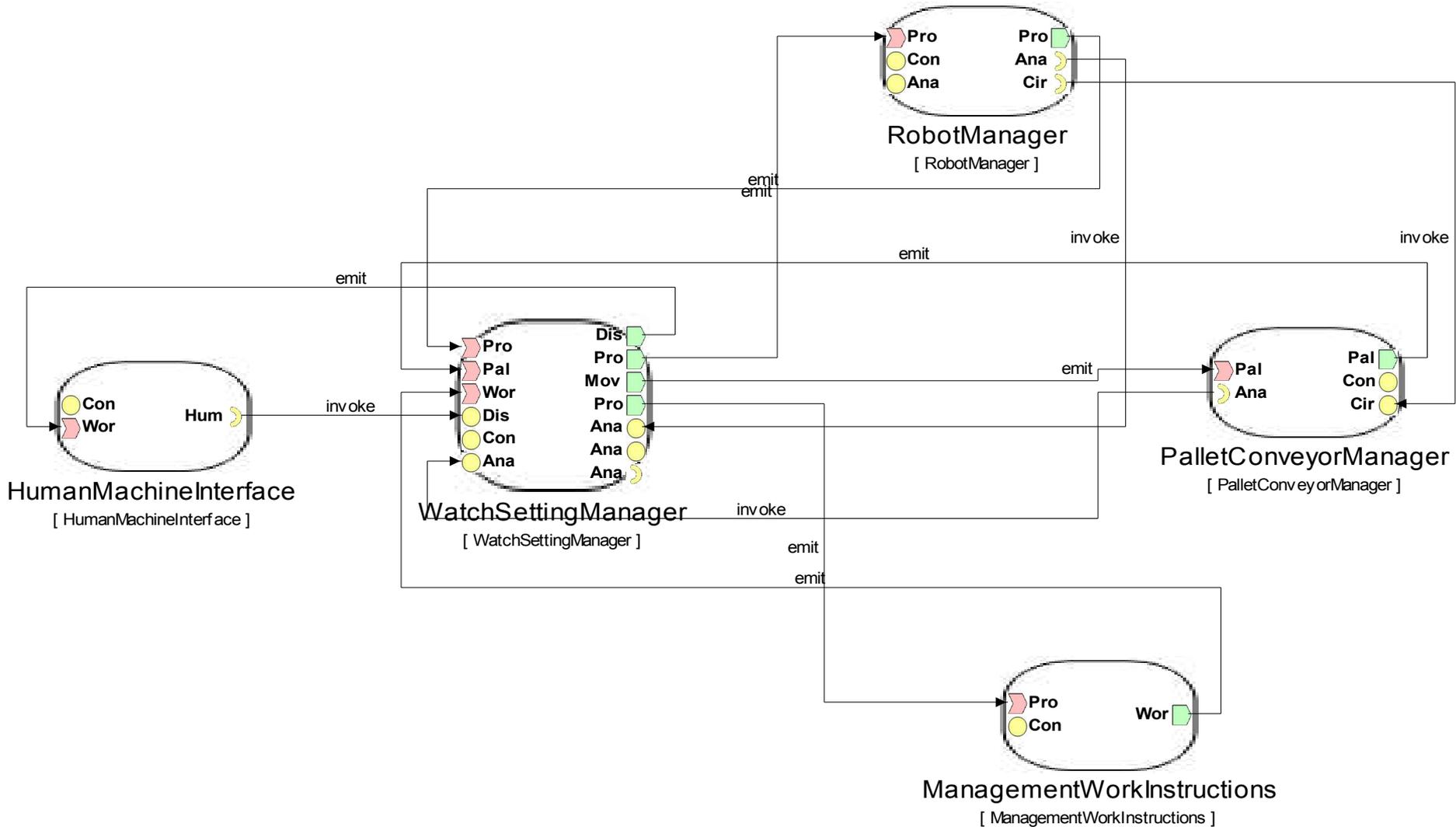




# PICML Meta: Properties & Requirements



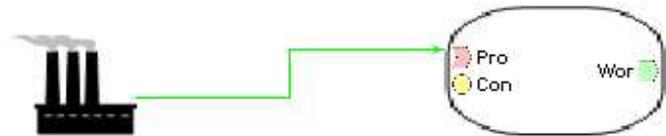
# A PICML Instance: Robot - Assembly



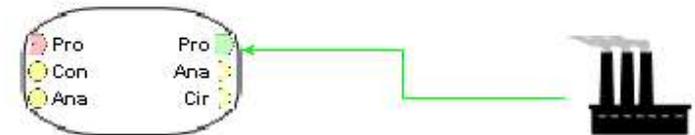
# PICML Instance: Robot-Component Def.

The screenshot shows the PICML software interface for defining a RobotAssemblyInterface. The main workspace contains four graphical elements: a yellow circle labeled 'Controller', a yellow crescent moon labeled 'HumanResponse', a pink arrow labeled 'WorkDisplayUpdate', and a blue square with a pencil labeled 'HumanLanguage'. A toolbar at the bottom provides icons for 'Attribute', 'InEventPort', 'Inherits', 'OutEventPort', 'ProvidedRequestPort', 'ReadOnlyAttribute', 'RequiredRequestPort', and 'Supports'. A right-hand pane shows a hierarchical tree view of the project structure, with 'HumanMachineInterface' selected. A bottom-right pane shows the 'Properties' tab for the selected interface, with fields for 'VersionTag', 'SpecifyIdTag', 'label', and 'UUID'.

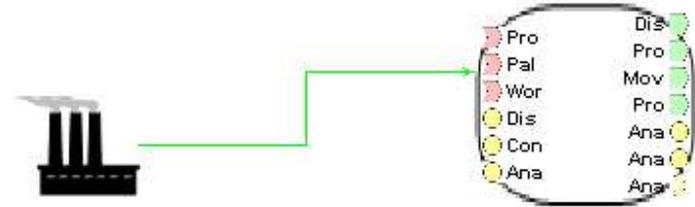
# PICML Instance: Robot-ComponentHomes



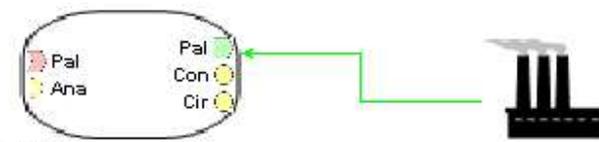
ManagementWorkInstructionsHome    ManagementWorkInstructions



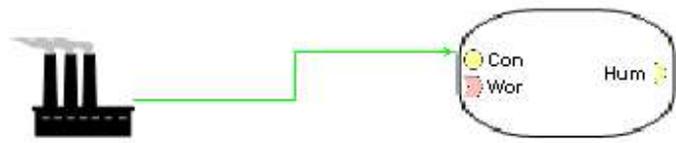
RobotManager    RobotManagerHome



WatchSettingManagerHome    WatchSettingManager



PalletConveyorManager    PalletConveyorManagerHome

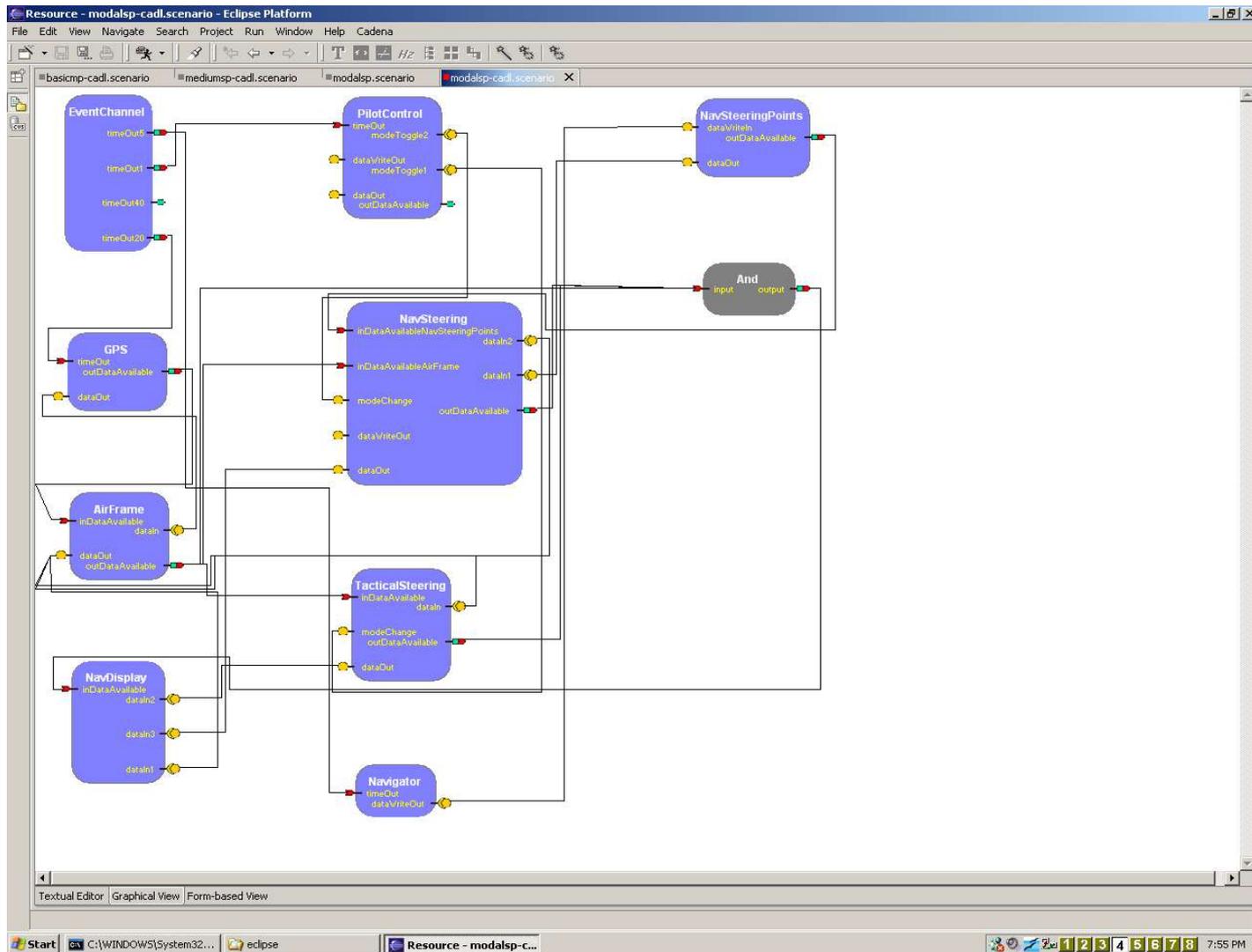


HumanMachineInterfaceHome    HumanMachineInterface

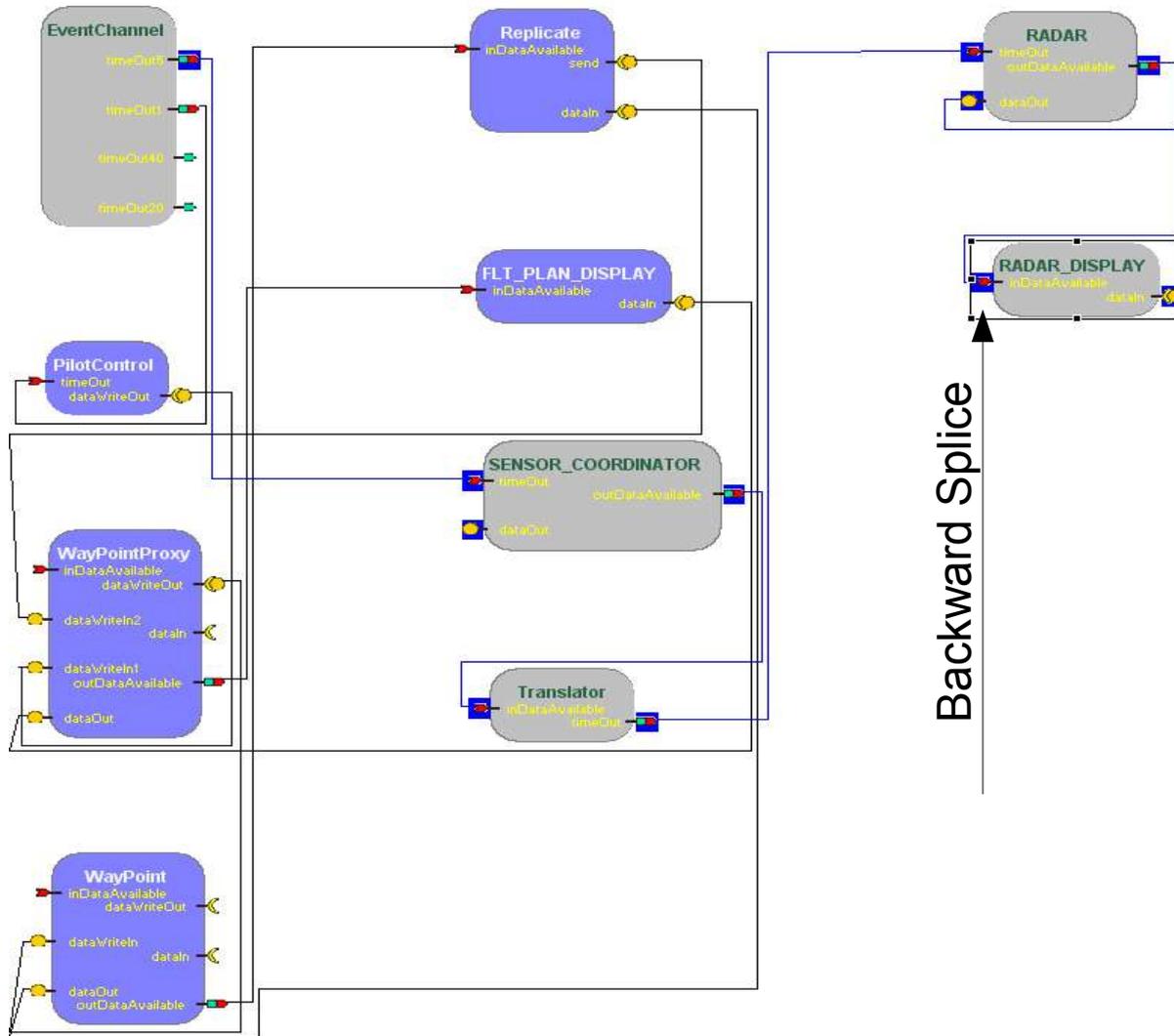
# Cadena



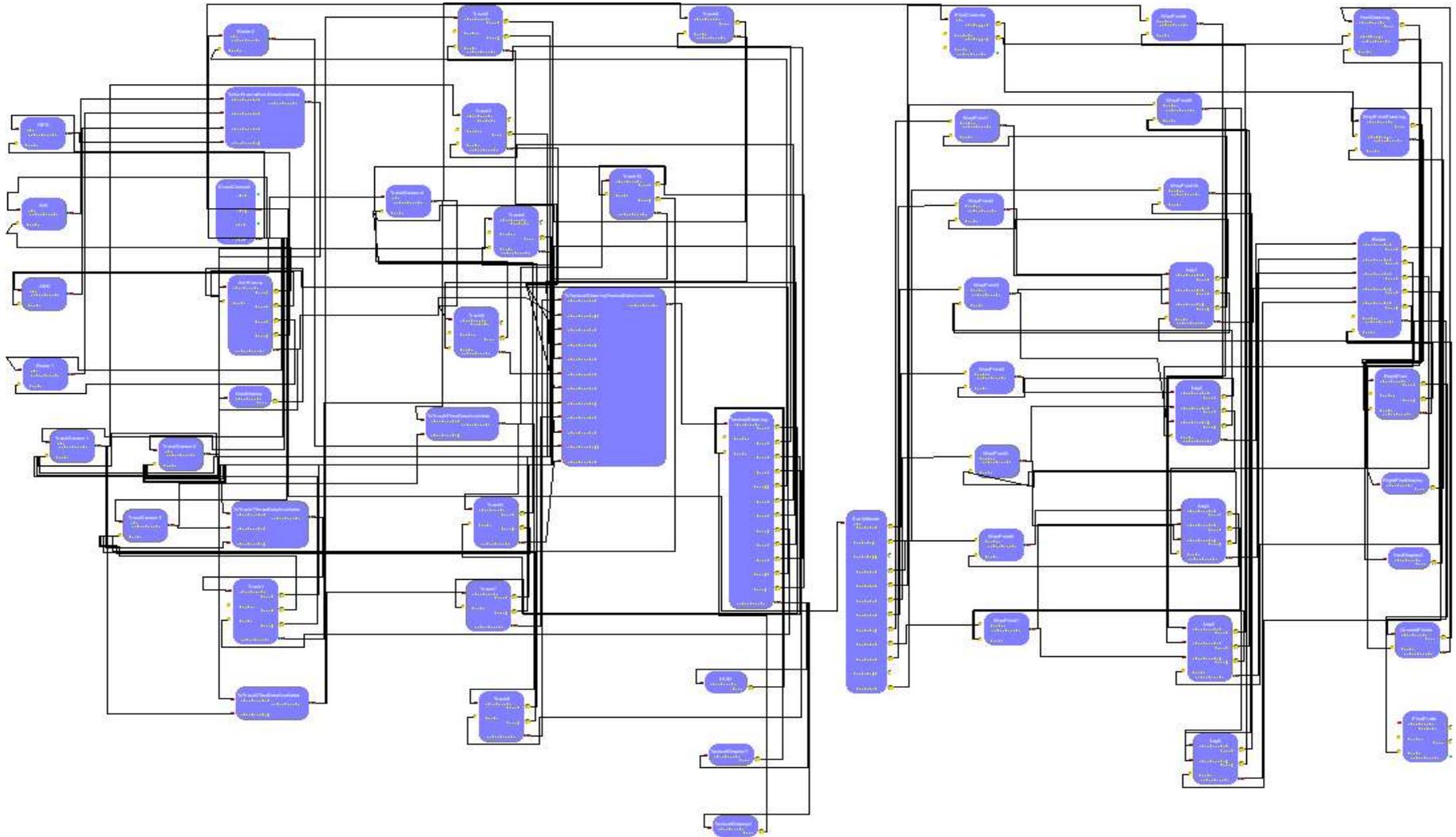
# Cadena examples - ModalSP



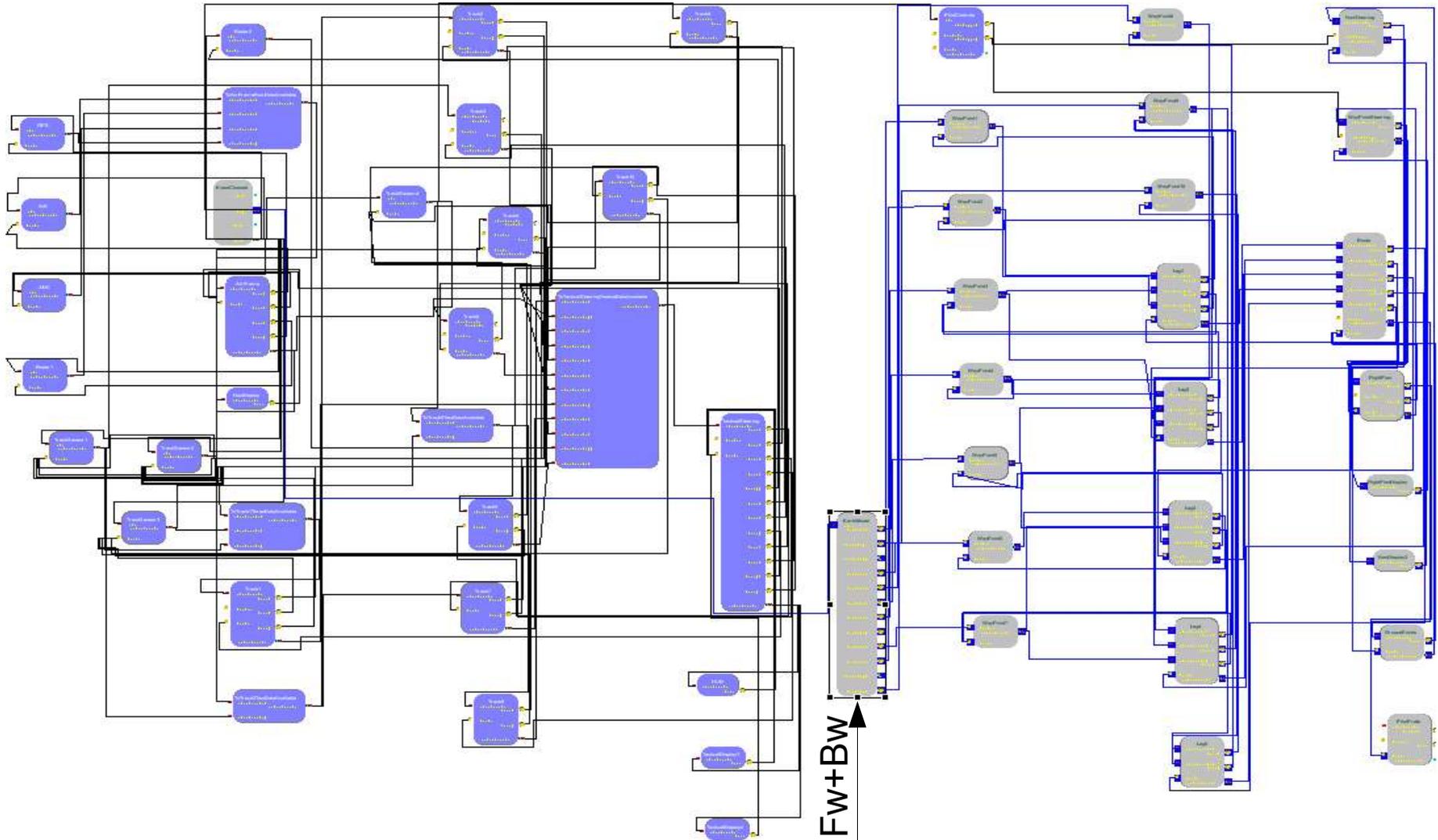
# [MultirateMP] Backward Splice



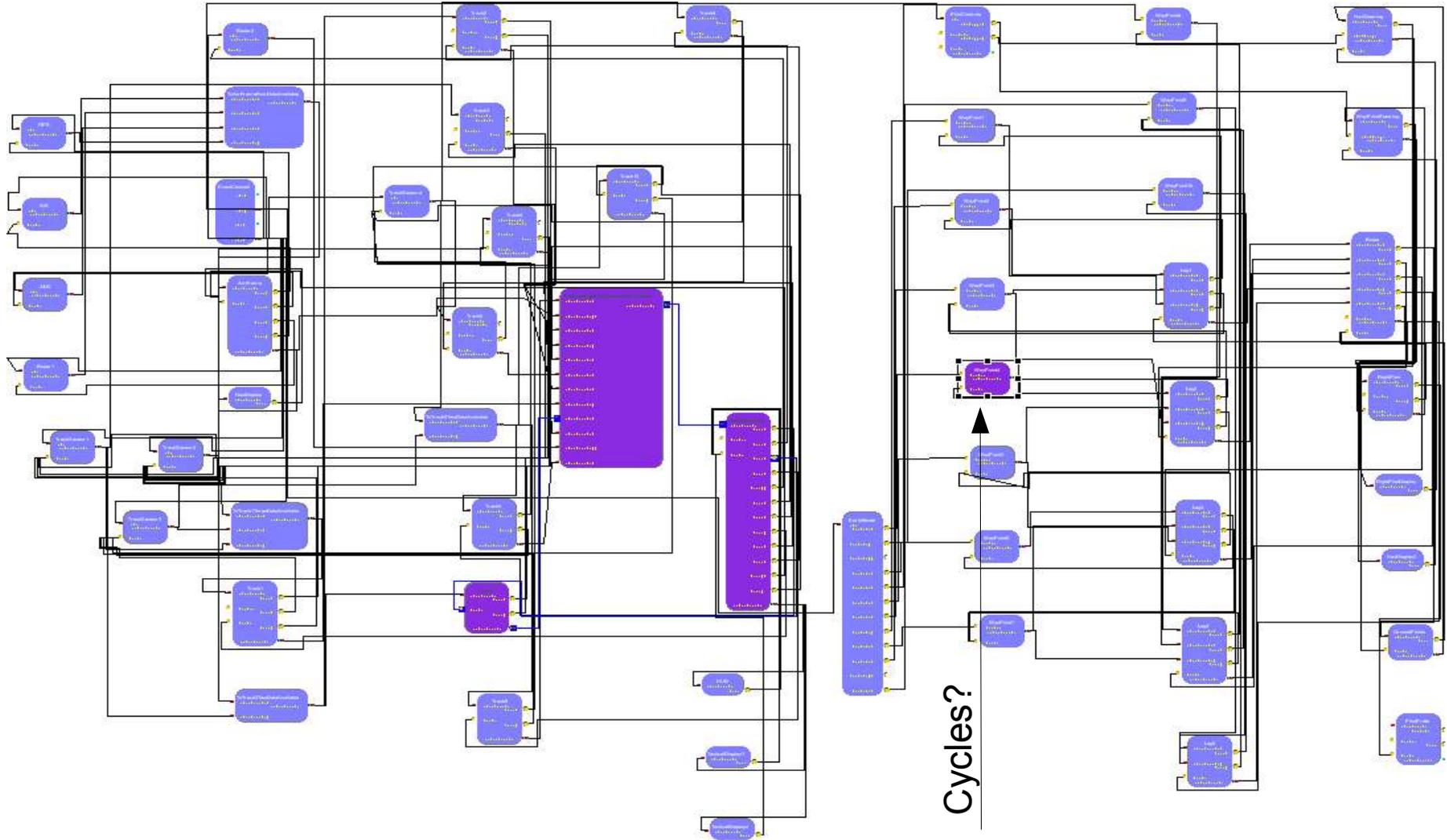
# MediumSP



# Forward + Backward splice



# Cycle Detection



# Incremental Specification

## Specifications

## Component Structure



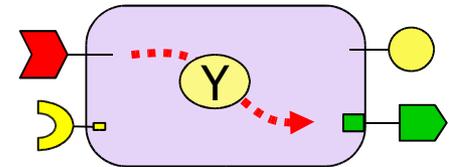
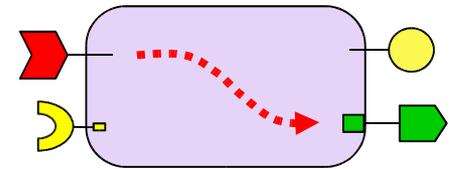
port action dependencies

refinement

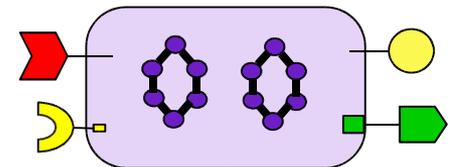
mode-based dependencies

refinement

component transition semantics



...only in mode Y



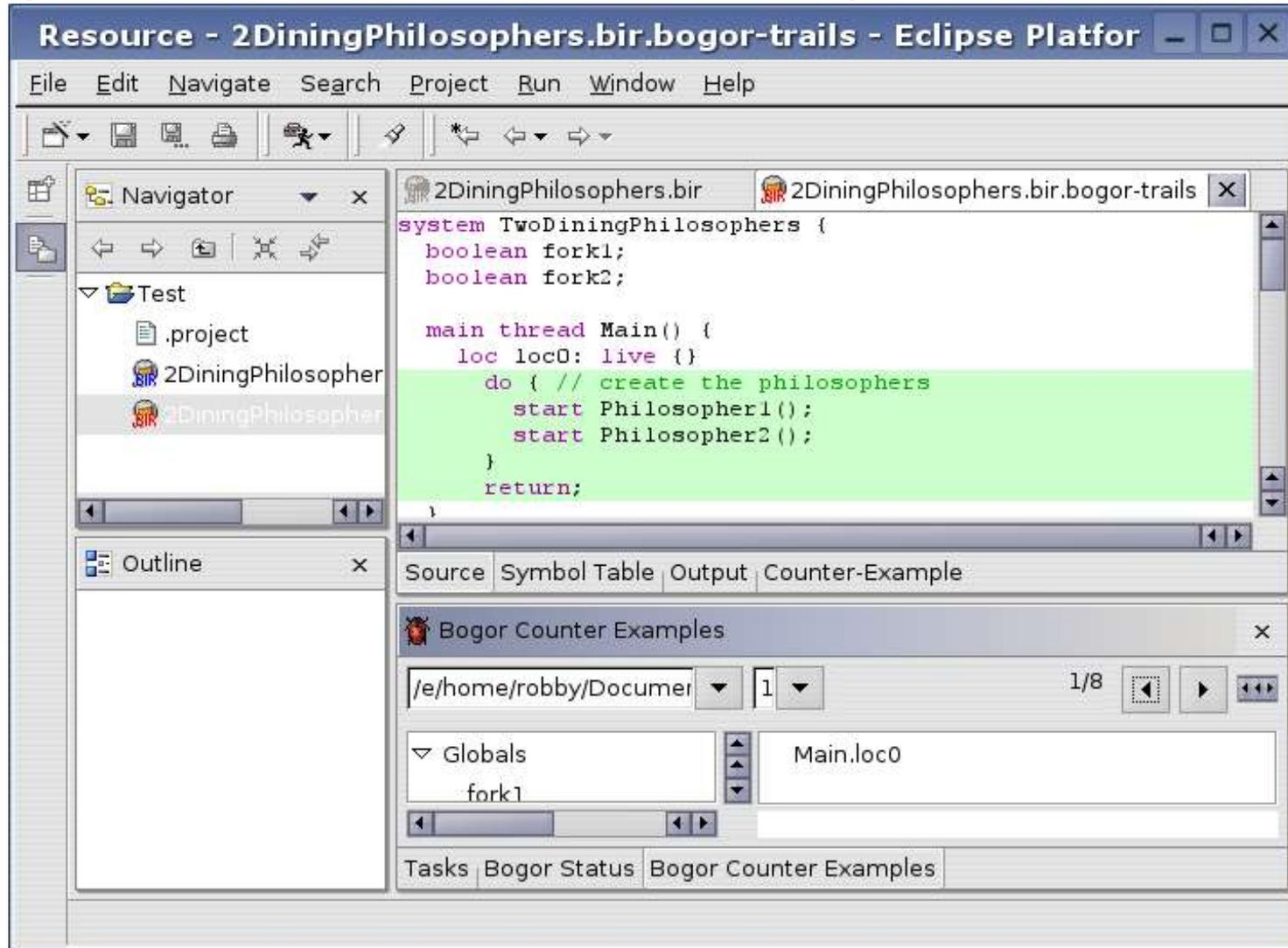
...state machines give abstract behavior

**Bogor**



*K-State*  
Kansas State University

# Bogor example – 2 Dining Philosophers



Resource - 2DiningPhilosophers.bir.bogor-trails - Eclipse Platfor

File Edit Navigate Search Project Run Window Help

Navigator

- Test
  - .project
  - 2DiningPhilosopher
  - 2DiningPhilosopher

```
system TwoDiningPhilosophers {
  boolean fork1;
  boolean fork2;

  main thread Main() {
    loc loc0: live {}
    do { // create the philosophers
      start Philosopher1();
      start Philosopher2();
    }
    return;
  }
}
```

Source Symbol Table Output Counter-Example

Bogor Counter Examples

/e/home/robby/Documer 1 1/8

Globals Main.loc0

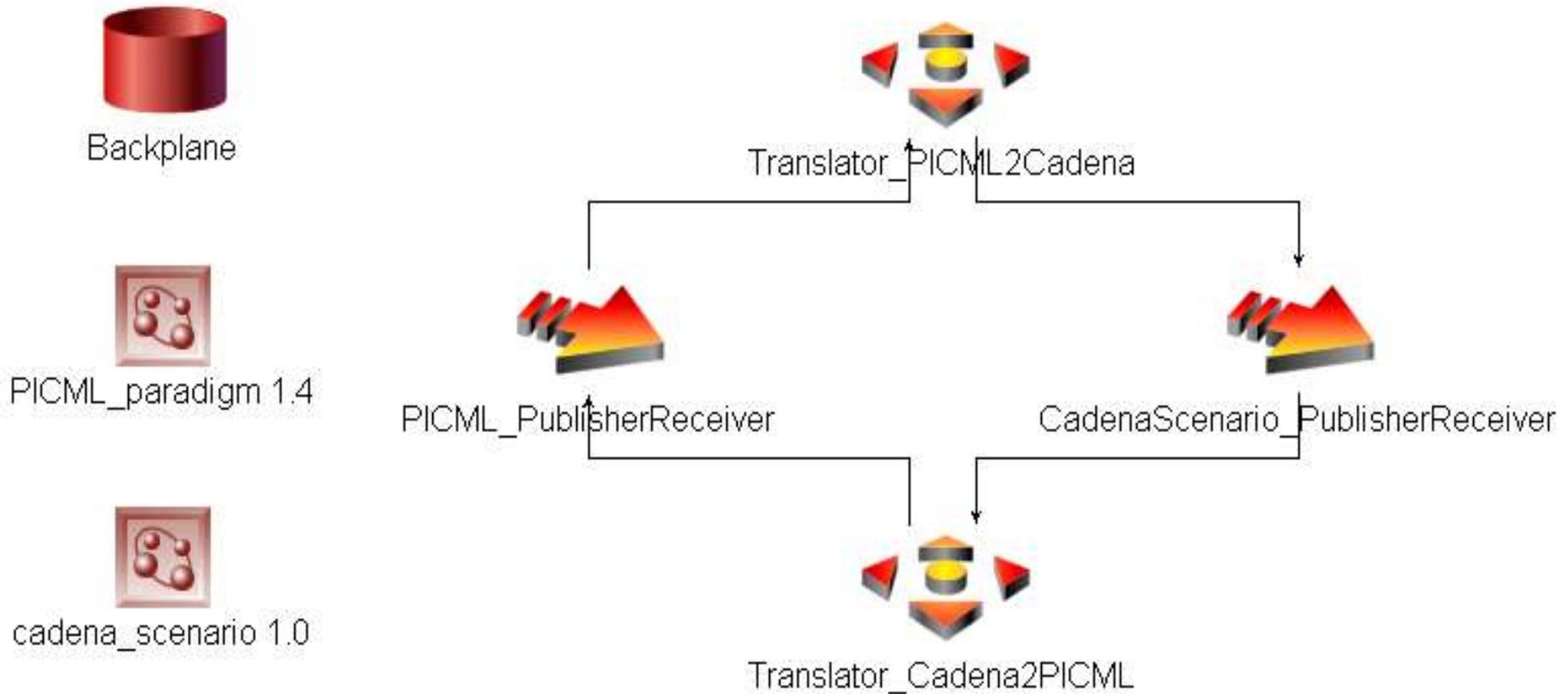
fork1

Tasks Bogor Status Bogor Counter Examples

# Open Tool Integration Framework



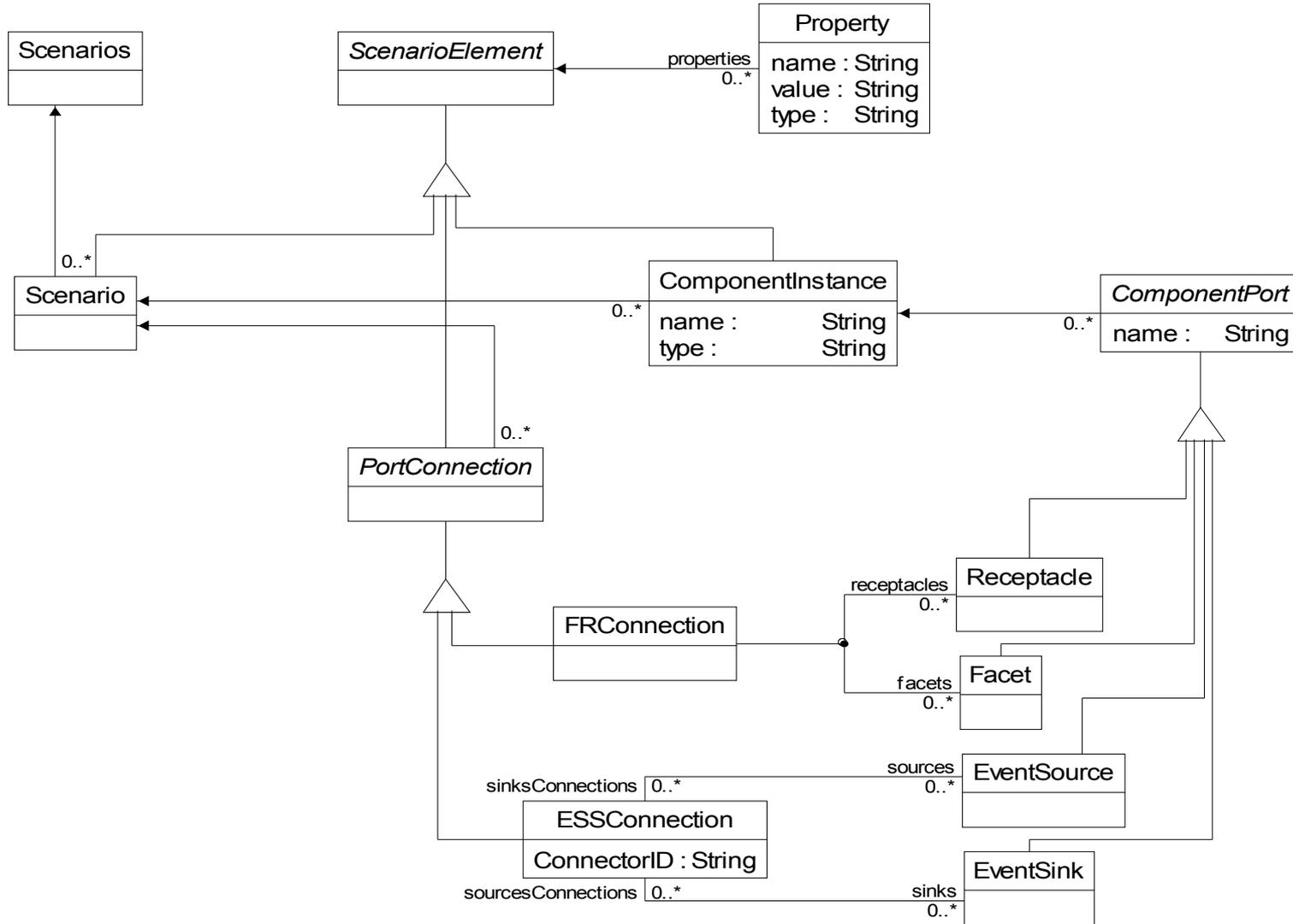
# OTIF - workflow



# Interchange Format



# CadenaScenario – UML meta



# Generated Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
<?udm interface="cadena_scenario" version="1.00"?>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <!-- generated on Mon Jun 14 01:27:17 2004
-->
- <xsd:complexType name="ESSConnectionType">
- <xsd:sequence>
  <xsd:element name="Property" type="PropertyType"
minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
  <xsd:attribute name="ConnectorID" type="xsd:string"
use="required" />
  <xsd:attribute name="sources" type="xsd:IDREFS" />
  <xsd:attribute name="sinks" type="xsd:IDREFS" />
  <xsd:attribute name="_id" type="xsd:ID" />
  <xsd:attribute name="_archetype" type="xsd:IDREF" />
  <xsd:attribute name="_derived" type="xsd:IDREFS" />
  <xsd:attribute name="_instances" type="xsd:IDREFS" />
  <xsd:attribute name="_desynched_atts" type="xsd:string" />
  <xsd:attribute name="_real_archetype" type="xsd:boolean" />
  <xsd:attribute name="_subtype" type="xsd:boolean" />
</xsd:complexType>
- <xsd:complexType name="ScenariosType">
- <xsd:sequence>
  <xsd:element name="Scenario" type="ScenarioType"
minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
  <xsd:attribute name="_id" type="xsd:ID" />
  <xsd:attribute name="_archetype" type="xsd:IDREF" />
  <xsd:attribute name="_derived" type="xsd:IDREFS" />
  <xsd:attribute name="_instances" type="xsd:IDREFS" />
  <xsd:attribute name="_desynched_atts" type="xsd:string" />
  <xsd:attribute name="_real_archetype" type="xsd:boolean" />
  <xsd:attribute name="_subtype" type="xsd:boolean" />
</xsd:complexType>

```

```

- <xsd:complexType name="FRConnectionType">
- <xsd:sequence>
  <xsd:element name="Property" type="PropertyType" minOccurs="0"
maxOccurs="unbounded" />
</xsd:sequence>
  <xsd:attribute name="facets_end_" type="xsd:IDREF" />
  <xsd:attribute name="receptacles_end_" type="xsd:IDREF" />
  <xsd:attribute name="_id" type="xsd:ID" />
  <xsd:attribute name="_archetype" type="xsd:IDREF" />
  <xsd:attribute name="_derived" type="xsd:IDREFS" />
  <xsd:attribute name="_instances" type="xsd:IDREFS" />
  <xsd:attribute name="_desynched_atts" type="xsd:string" />
  <xsd:attribute name="_real_archetype" type="xsd:boolean" />
  <xsd:attribute name="_subtype" type="xsd:boolean" />
</xsd:complexType>
- <xsd:complexType name="EventSinkType">
  <xsd:attribute name="name" type="xsd:string" use="required" />
  <xsd:attribute name="sourcesConnections" type="xsd:IDREFS" />
  <xsd:attribute name="_id" type="xsd:ID" />
  <xsd:attribute name="_archetype" type="xsd:IDREF" />
  <xsd:attribute name="_derived" type="xsd:IDREFS" />
  <xsd:attribute name="_instances" type="xsd:IDREFS" />
  <xsd:attribute name="_desynched_atts" type="xsd:string" />
  <xsd:attribute name="_real_archetype" type="xsd:boolean" />
  <xsd:attribute name="_subtype" type="xsd:boolean" />
</xsd:complexType>
- <xsd:complexType name="EventSourceType">
  <xsd:attribute name="name" type="xsd:string" use="required" />
  <xsd:attribute name="sinksConnections" type="xsd:IDREFS" />
  <xsd:attribute name="_id" type="xsd:ID" />
  <xsd:attribute name="_archetype" type="xsd:IDREF" />
  <xsd:attribute name="_derived" type="xsd:IDREFS" />
  <xsd:attribute name="_instances" type="xsd:IDREFS" />
  <xsd:attribute name="_desynched_atts" type="xsd:string" />

```

[.....]

# Robot Assembly – Equivalent XML

```

- <Scenarios xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\Udm\cadena_scenario.xsd">
- <Scenario>
- <ComponentInstance name="HumanMachineInterface" type="RobotAssembly/HumanMachineInterface">
  <EventSink _id="id18a4" name="workDisplayUpdate" sourcesConnections="id18a5" />
  <Facet name="AnalysisController" />
  <Facet name="Controller" />
  <Receptacle _id="id18a6" name="HumanResponse" facets="id18a7" />
</ComponentInstance>
- <ComponentInstance name="ManagementWorkInstructions"
type="RobotAssembly/ManagementWorkInstructions">
  <EventSink _id="id18aa" name="ProductionReport" sourcesConnections="id18ab" />
  <EventSource _id="id18a8" name="workOrder" sinksConnections="id18a9" />
  <Facet name="Controller" />
</ComponentInstance>
- <ComponentInstance name="WatchSettingManager" type="RobotAssembly/WatchSettingManager">
  <EventSink _id="id18b2" name="ProcessingStatus" sourcesConnections="id18b3" />
  <EventSink _id="id18b4" name="PalletStatus" sourcesConnections="id18b5" />
  <EventSink _id="id18b6" name="workOrder" sourcesConnections="id18a9" />
  <EventSource _id="id18ac" name="Display" sinksConnections="id18a5" />
  <EventSource _id="id18ad" name="ProductionControl" sinksConnections="id18ae" />
  <EventSource _id="id18af" name="MovePallet" sinksConnections="id18b0" />
  <EventSource _id="id18b1" name="ProductionReport" sinksConnections="id18ab" />
  <Facet name="AnalysisCalls" />
  <Facet _id="id18b7" name="DisplayResponse" receptacles="id18a7" />
  <Facet name="Controller" />
  <Facet _id="id18b8" name="AnalysisOne" receptacles="id18b9" />
  <Facet _id="id18ba" name="AnalysisTwo" receptacles="id18bb" />
</ComponentInstance>
[.....]

```

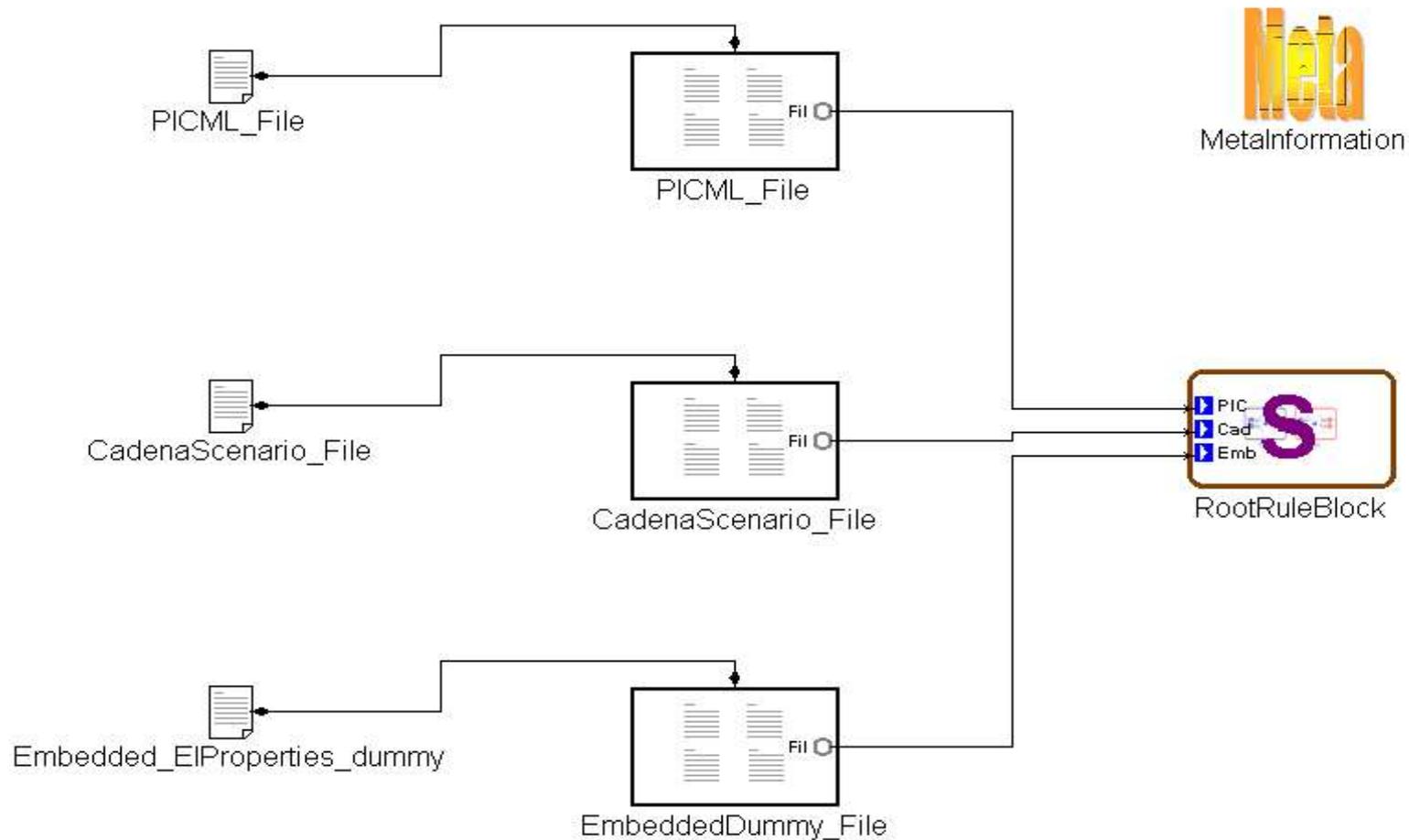
# Cross-Meta Graph Transformation

## A Model-To-Model driven approach

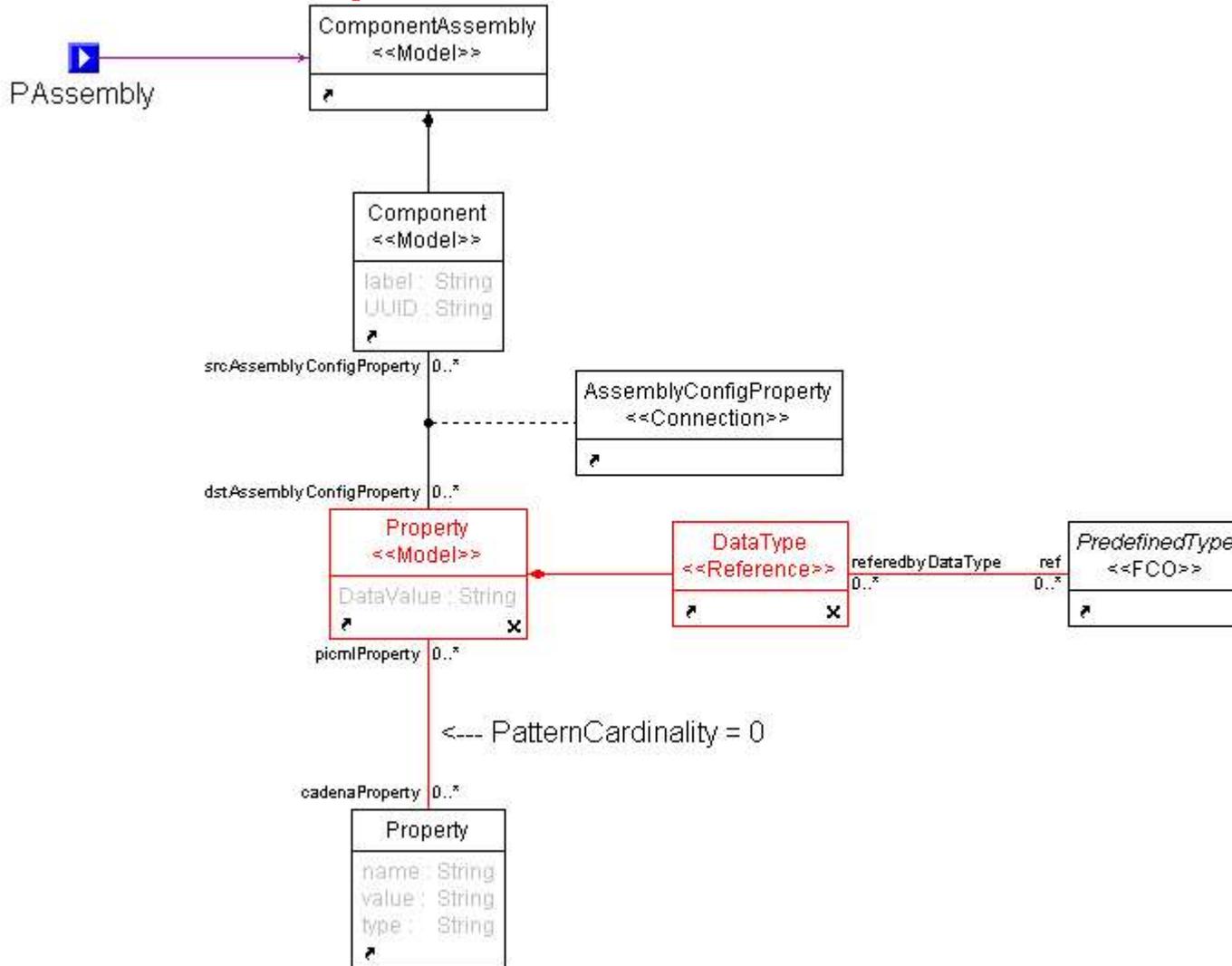


# Graph Transformation

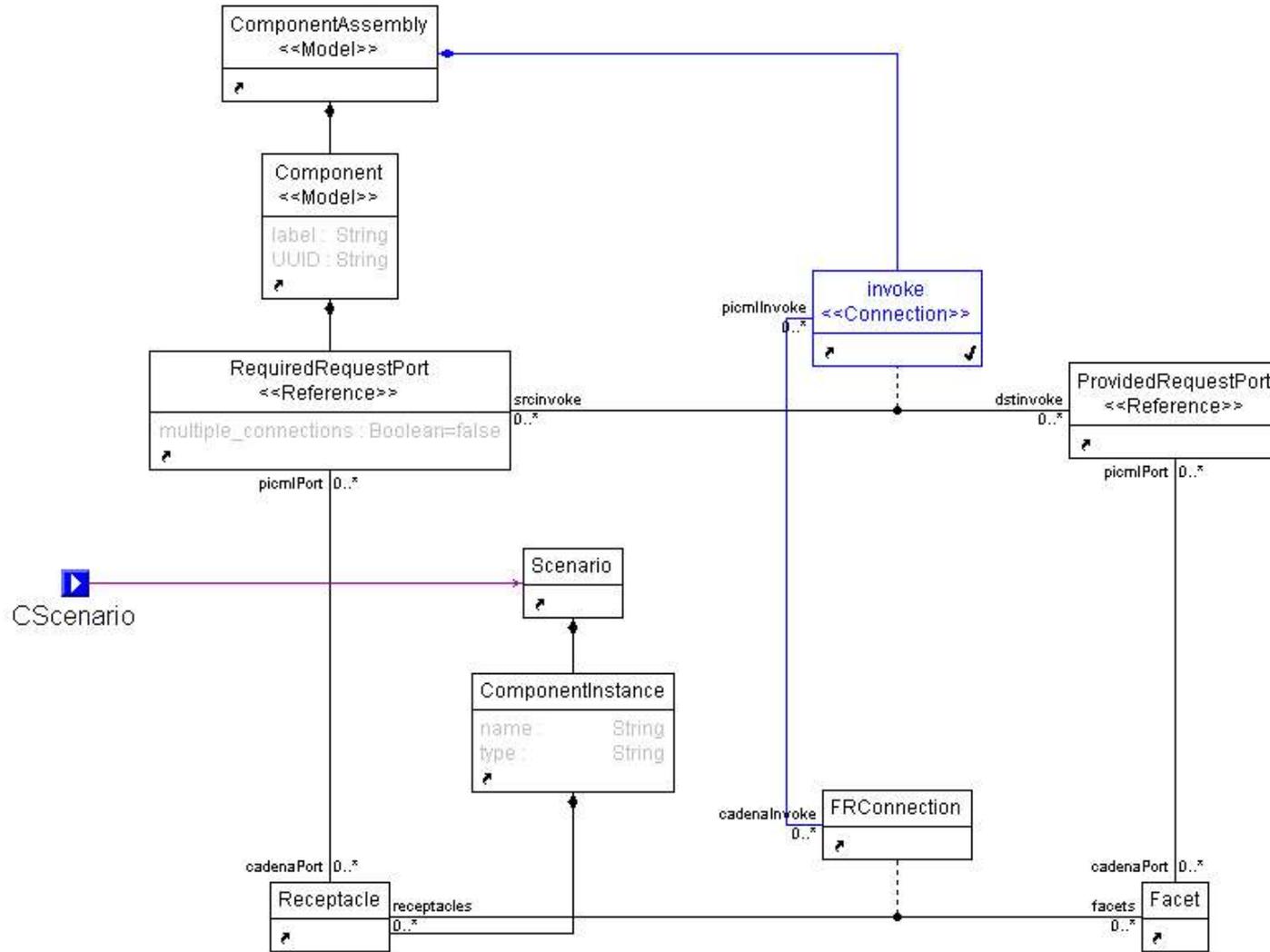
## A Model-To-Model driven approach



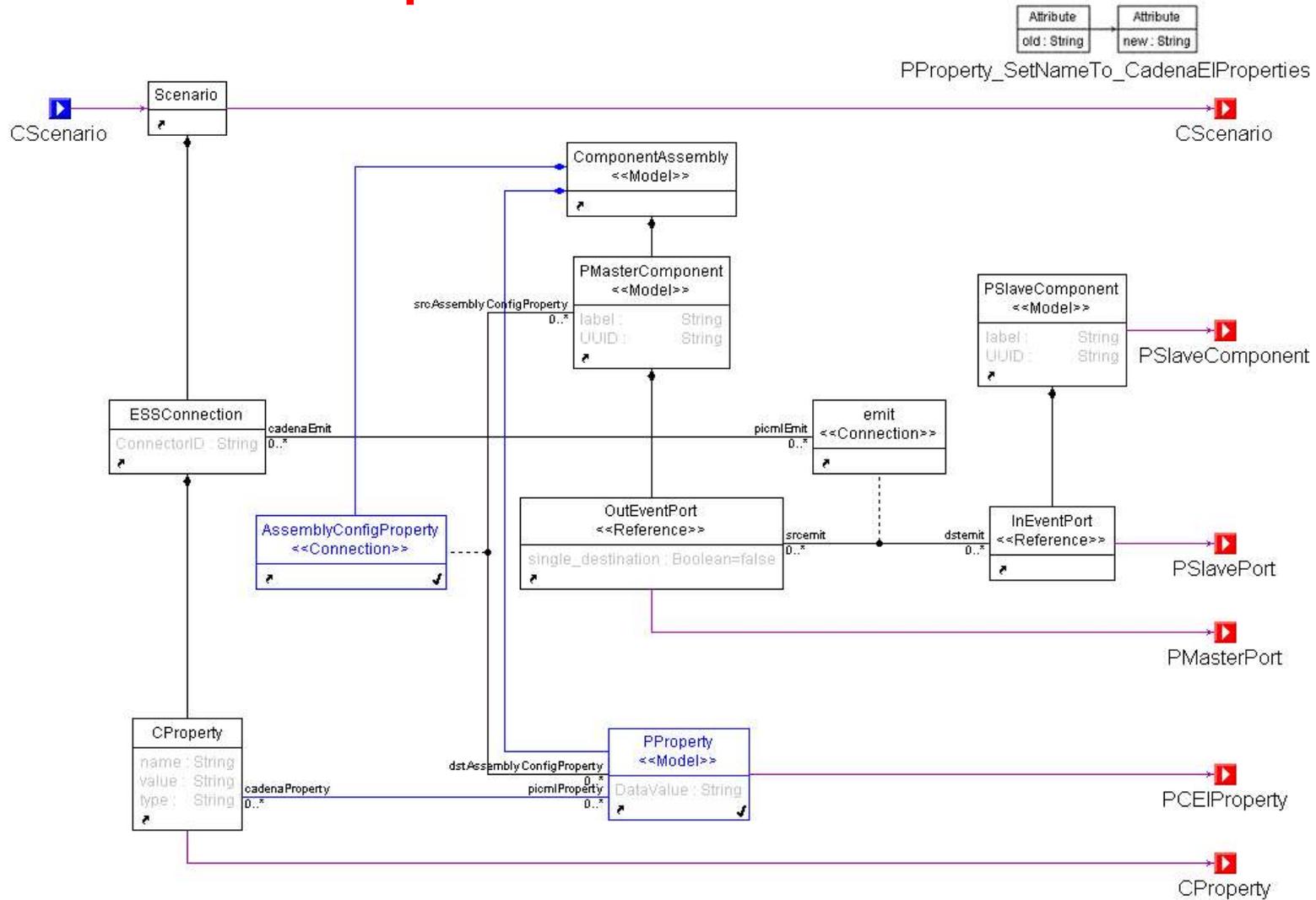
# Graph Transformation



# Graph Transformation



# Graph Transformation



# Graph Transformation

The screenshot shows the UML Model Transformer interface with a graph transformation diagram. The diagram consists of several nodes and edges:

- Root RuleBlock** (purple box) is the central node, connected to various other nodes.
- Assembly** nodes (red boxes) are connected to **Root RuleBlock** and other nodes.
- Component** nodes (blue boxes) are connected to **Assembly** nodes.
- Connector** nodes (green boxes) are connected to **Component** nodes.
- Scenario** nodes (blue boxes) are connected to **Assembly** nodes.

The diagram includes the following nodes and their connections:

- CreateComponents\_step1** (red box) connects to **Root RuleBlock** and **Assembly**.
- FindFirstPackage** (red box) connects to **Root RuleBlock** and **Assembly**.
- MatchAPackage** (red box) connects to **Root RuleBlock** and **Assembly**.
- FindSubPackage** (red box) connects to **Root RuleBlock** and **Assembly**.
- MatchAPackage\_ref** (red box) connects to **Root RuleBlock** and **Assembly**.
- FinalPackage\_instantiateComponent** (red box) connects to **Root RuleBlock** and **Assembly**.
- AssociateComponentsNow** (red box) connects to **Root RuleBlock** and **Assembly**.
- FindComponents2BDeleted** (purple box) connects to **Root RuleBlock** and **Assembly**.
- DestroyOrphanPropertiesAndDeleteComponent** (purple box) connects to **Root RuleBlock** and **Assembly**.
- CreatePublishConnectors\_step1** (purple box) connects to **Root RuleBlock** and **Assembly**.
- CreatePublishConnectors\_step2** (purple box) connects to **Root RuleBlock** and **Assembly**.
- CreatePublishConnectors\_step3** (purple box) connects to **Root RuleBlock** and **Assembly**.
- DestroyPublishConnectors** (red box) connects to **Root RuleBlock** and **Assembly**.

An algorithm for importing ConnectorIDs is provided in the diagram:

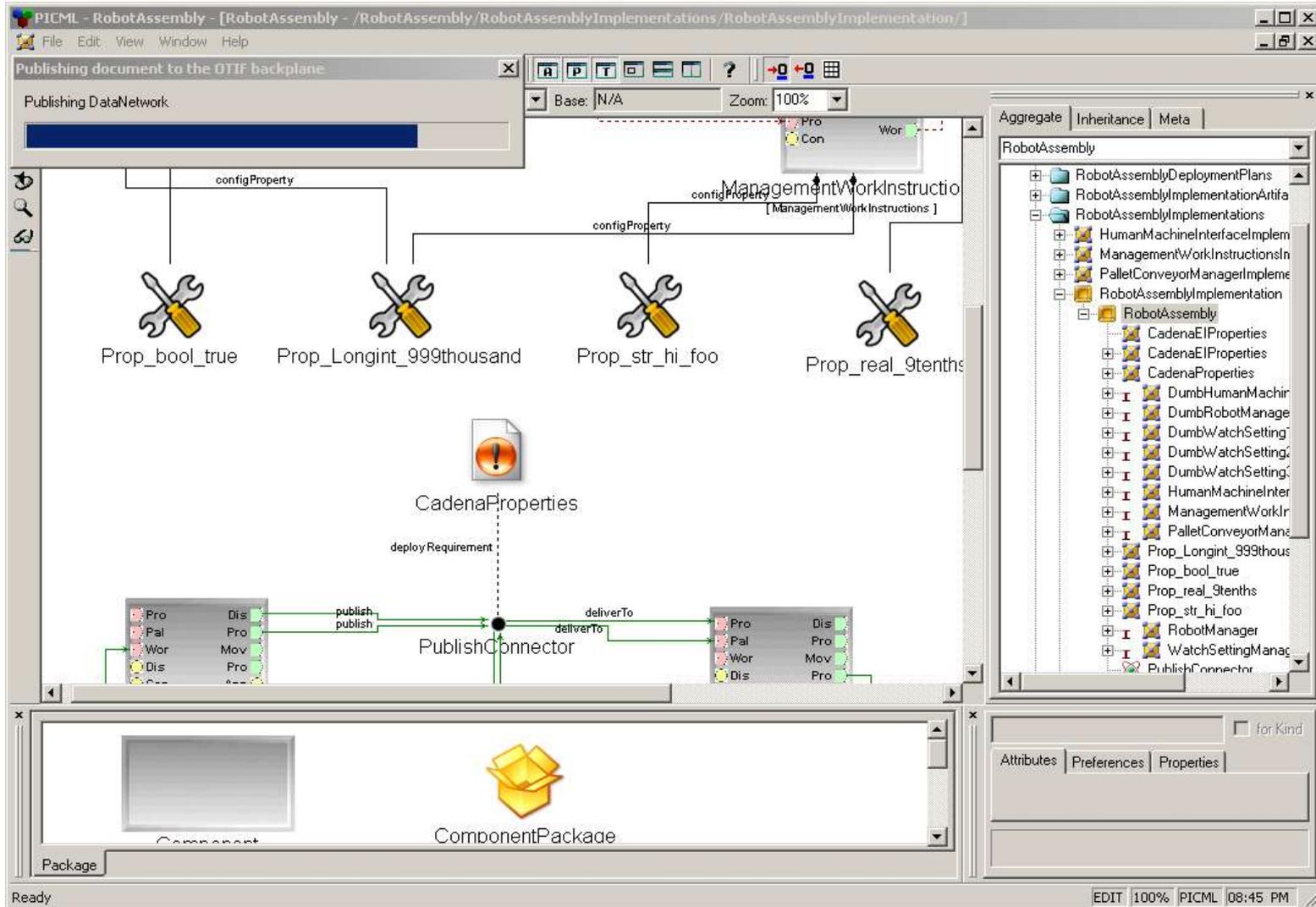
1. Those with empty ConnectorID -> emit.
2. Then, those flagged with emit and more than one participant -> empty.
3. Then, those unlinked which are not emit => make PublishConnector (if empty => don't make ConnectorID).

The interface also shows a toolbar with icons for Block, ExpressionRef, ForBlock, In, Out, Rule, and Test. The status bar at the bottom indicates 'Ready' and 'EDIT | 100% | UMLModelTransformer | 09:00 PM'.

# OTIF Tooladaptors



# OTIF Tooladaptors – GME Publisher



# OTIF Tooladaptors – GME Receiver

The screenshot displays a software interface with several components:

- Monitoring backplane for: paradigm "PICML" version "1.0"**: A window containing a table with the following data:
 

Name	Version	Remarks	DocID	KeepAlive
a_cadena_document	1.0	-> CS2PA-translated	8	276
- RobotAssembly Tree View**: A hierarchical tree structure showing the following elements:
  - RobotAssemblyImplementationArtifacts
  - RobotAssemblyImplementations
    - HumanMachineInterfaceImplementation
    - ManagementWorkInstructionsImplementation
    - PalletConveyorManagerImplementation
    - RobotAssemblyImplementation
      - RobotAssembly
        - CadenaEIProperties
        - CadenaEIProperties
        - CadenaEIProperties
        - CadenaEIProperties
        - CadenaProperties
        - CadenaProperties
        - DumbHumanMachineInterface1\_GabrieleTry
        - DumbRobotManager1\_GabrieleTry
        - DumbWatchSetting1\_GabrieleTry
        - DumbWatchSetting2\_GabrieleTry
        - DumbWatchSetting3\_GabrieleTry
        - ManagementWorkInstructions
        - PalletConveyorManager
        - Prop\_Longint\_999thousand
        - Prop\_bool\_true
        - Prop\_real\_9stenths
        - Prop\_str\_hi\_foo
        - RobotManager
        - WatchSettingManager\_MODIFIED\_NAME
        - PublishConnector
        - PublishConnector

- Palette**: A collection of model elements including Component, ComponentPackage, ComponentPackageReference, Property, PublishConnector, and Requirement.
- RobotAssembly Properties Panel**: A panel with tabs for Attributes, Preferences, and Properties. The Properties tab is active, showing fields for label and UUID.

# GME Receiver: receiving...

The screenshot displays the GME (Geometric Model Engineering) software interface. At the top left, a window titled "Monitoring backplane for paradigm 'PICML' version '1.0'" is open. Below it, a "Fetching file" dialog box is active, showing a progress bar and a table with the following data:

DocID	KeepAlive
8	202

The main workspace shows a tree view of the "RobotAssembly" model. The tree structure includes:

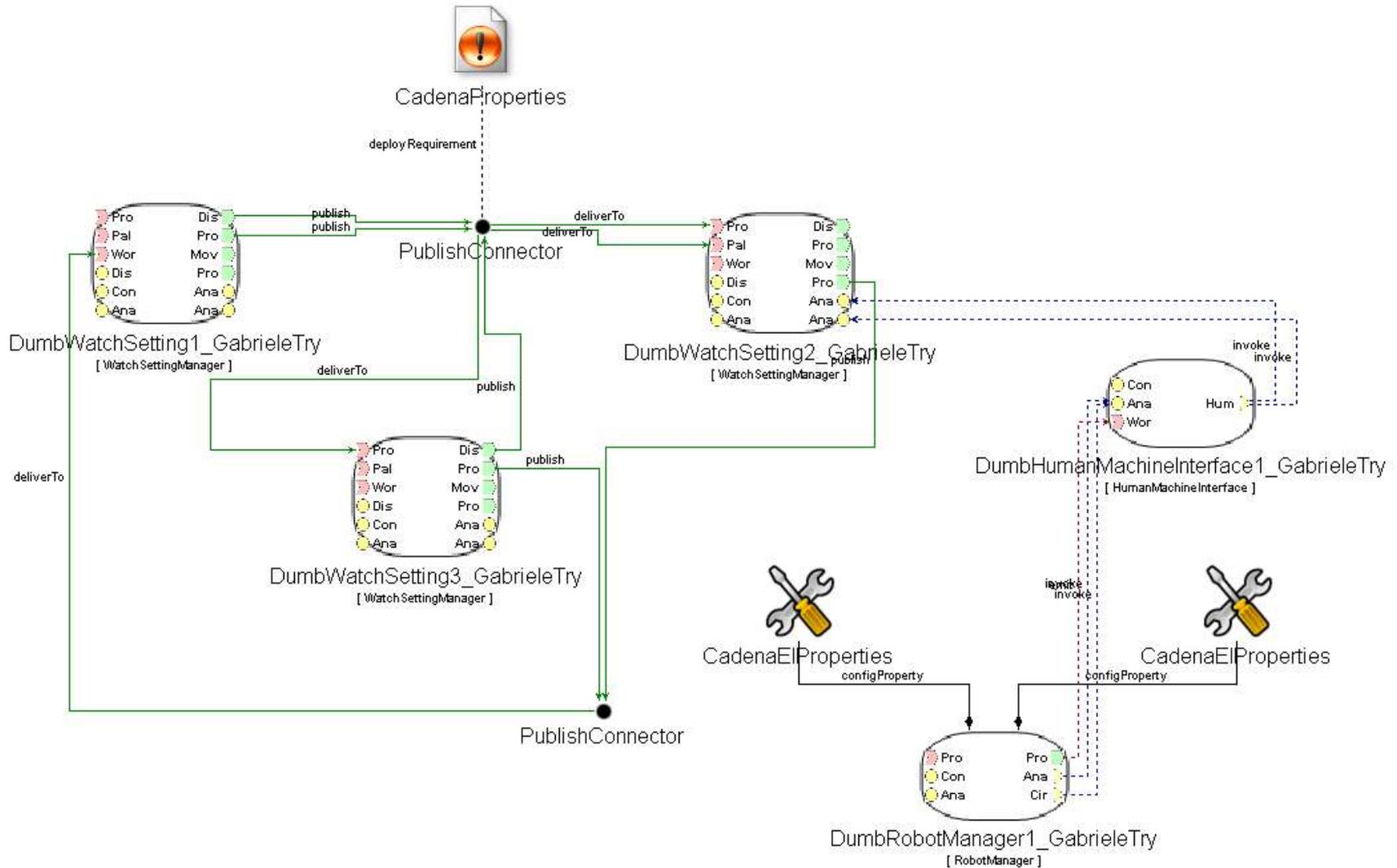
- RobotAssemblyImplementationArtifacts
- RobotAssemblyImplementations
  - HumanMachineInterfaceImplementation
  - ManagementWorkInstructionsImplementation
  - PalletConveyorManagerImplementation
  - RobotAssemblyImplementation
    - RobotAssembly
      - CadenaEIProperties
      - CadenaEIProperties
      - CadenaEIProperties
      - CadenaProperties
      - CadenaProperties
      - DumbHumanMachineInterface1\_GabrieleTry
      - DumbRobotManager1\_GabrieleTry
      - DumbWatchSetting1\_GabrieleTry
      - DumbWatchSetting2\_GabrieleTry
      - DumbWatchSetting3\_GabrieleTry
      - ManagementWorkInstructions
      - PalletConveyorManager
      - Prop\_Longint\_999thousand
      - Prop\_bool\_true
      - Prop\_real\_9tenths
      - Prop\_str\_hi\_foo
      - RobotManager
      - WatchSettingManager\_MODIFIED\_NAME
    - PublishConnector
    - PublishConnector

At the bottom, a "Package" palette is visible, containing icons for Component, ComponentPackage, ComponentPackageReference, Property, PublishConnector, and Requirement. The bottom right corner shows a status bar with "EDIT 100% PICML 11:19 PM".

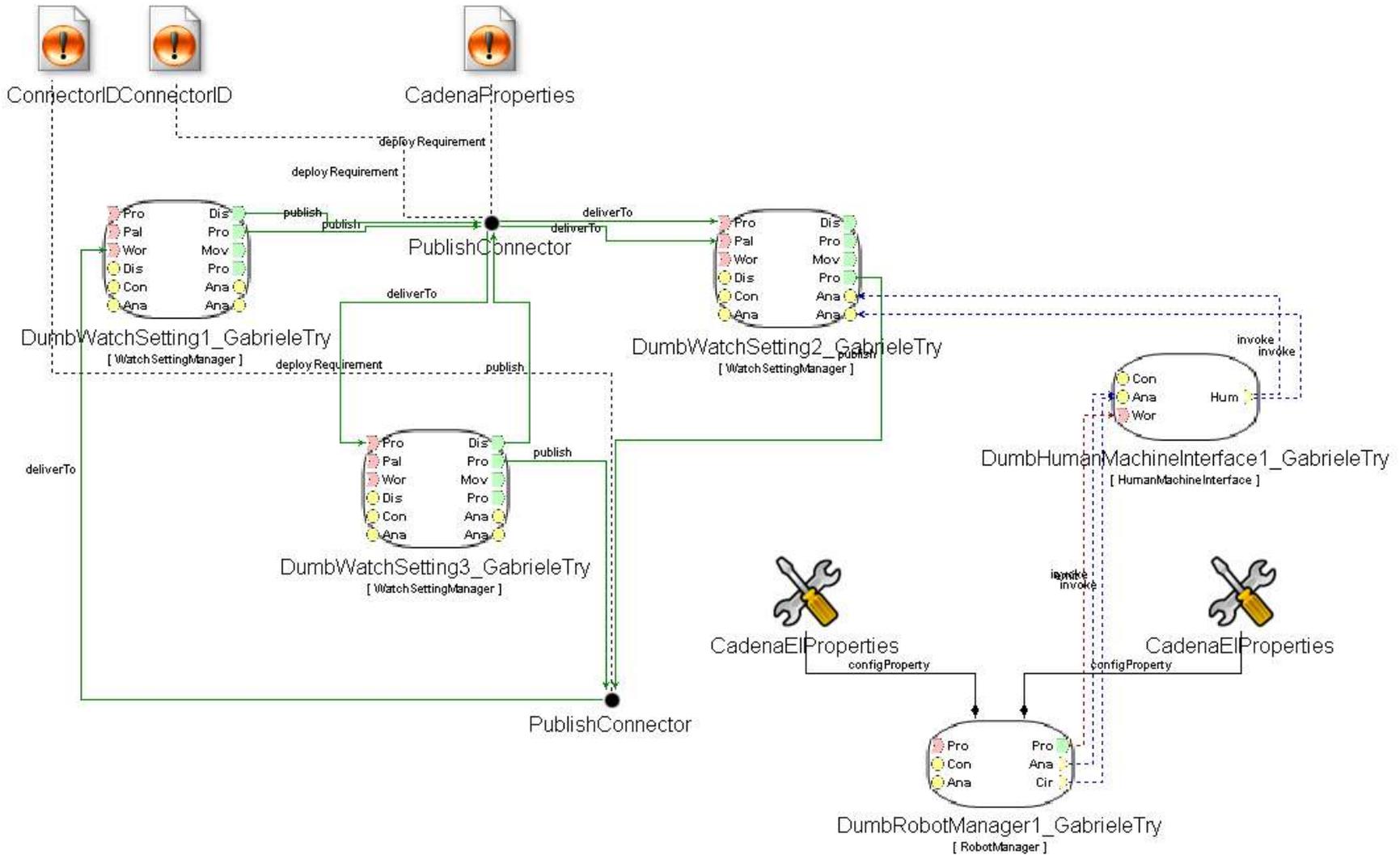
# Test Drive for Transformation Semantics...



# Tweaked-RobotAssembly: before export



# TweakedRobotAssembly: after export



# TweakedRobotAssembly: exported

```

- <Scenarios xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="..\Udm\cadena_scenario.xml">
- <Scenario>
- <ComponentInstance name="HumanMachineInterface" type="RobotAssembly/HumanMachineInterface">
  <EventSink _id="id18a4" name="workDisplayUpdate" sourcesConnections="id18a5" />
  <Facet name="AnalysisController" />
  <Facet name="Controller" />
  <Property name="Prop_Longint_999thousand" type="INT" value="999000" />
  <Receptacle _id="id18a6" name="HumanResponse" facets="id18a7" />
</ComponentInstance>
- <ComponentInstance name="ManagementWorkInstructions" type="RobotAssembly/ManagementWorkInstructions">
  <EventSink _id="id18aa" name="ProductionReport" sourcesConnections="id18ab" />
  <EventSource _id="id18a8" name="workOrder" sinksConnections="id18a9" />
  <Facet name="Controller" />
  <Property name="Prop_Longint_999thousand" type="INT" value="999000" />
  <Property name="Prop_str_hi_foo" type="STRING" value="Hi foo!!" />
</ComponentInstance>
- <ComponentInstance name="WatchSettingManager" type="RobotAssembly/WatchSettingManager">
  <EventSink _id="id18b2" name="ProcessingStatus" sourcesConnections="id18b3" />
  <EventSink _id="id18b4" name="PalletStatus" sourcesConnections="id18b5" />
  <EventSink _id="id18b6" name="workOrder" sourcesConnections="id18a9" />
  <EventSource _id="id18ac" name="Display" sinksConnections="id18a5" />
  <EventSource _id="id18ad" name="ProductionControl" sinksConnections="id18ae" />
  <EventSource _id="id18af" name="MovePallet" sinksConnections="id18b0" />
  <EventSource _id="id18b1" name="ProductionReport" sinksConnections="id18ab" />
  <Facet name="AnalysisCalls" />
  <Facet _id="id18b7" name="DisplayResponse" receptacles="id18a7" />
  <Facet name="Controller" />
  <Facet _id="id18b8" name="AnalysisOne" receptacles="id18b9" />
  <Facet _id="id18ba" name="AnalysisTwo" receptacles="id18bb" />
  <Property name="Prop_bool_true" type="BOOLEAN" value="TRUE" />
</ComponentInstance>
- <ComponentInstance name="PalletConveyorManager" type="RobotAssembly/PalletConveyorManager">
  <EventSink _id="id18bd" name="PalletRequests" sourcesConnections="id18b0" />
  <EventSource _id="id18bc" name="PalletStatus" sinksConnections="id18b5" />
  <Facet name="Controller" />
  <Facet _id="id18bf" name="CircleAnalysis" receptacles="id18c0" />
  <Receptacle _id="id18be" name="AnalysisTwo" facets="id18bb" />
</ComponentInstance>
[.....]

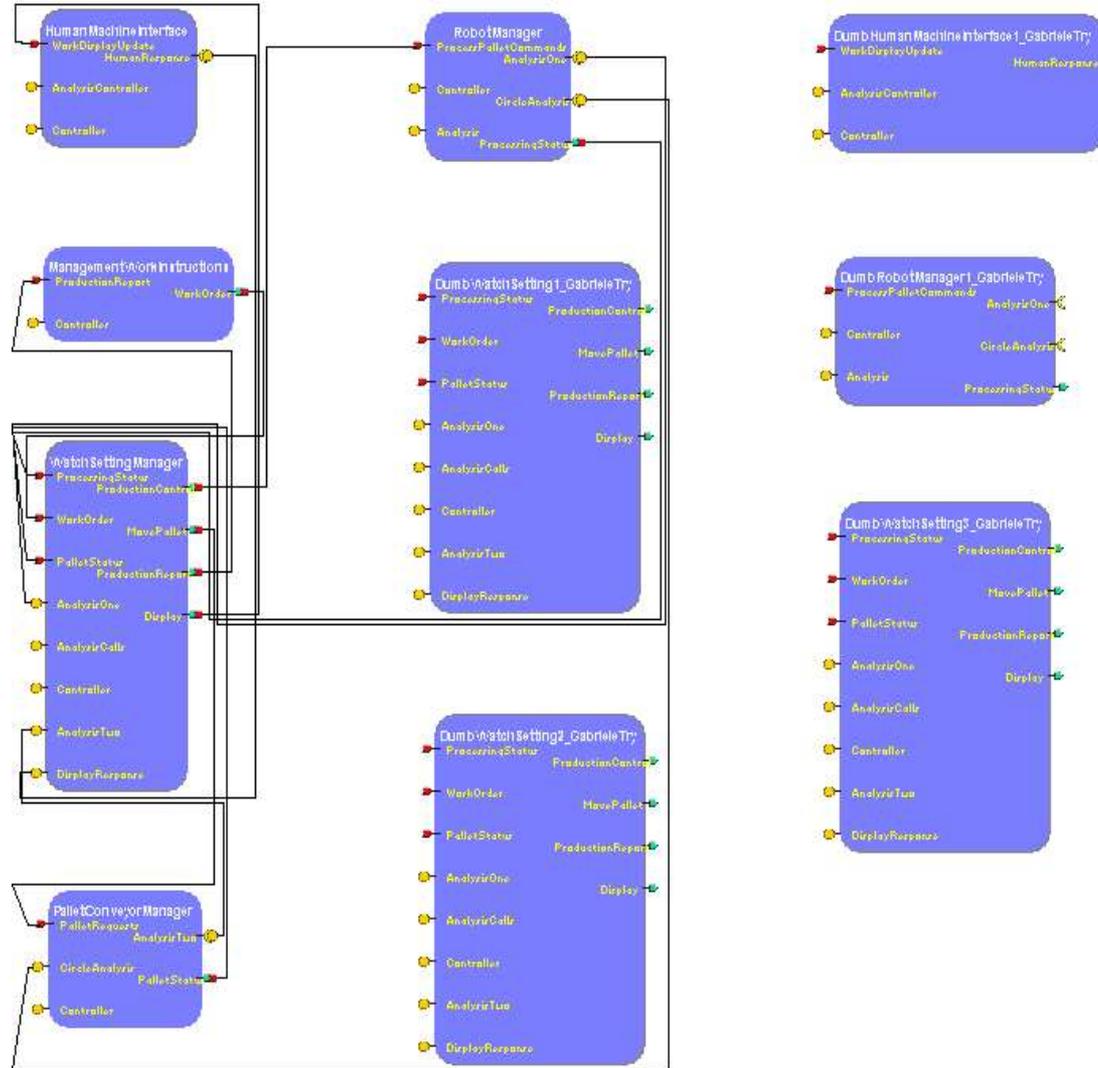
```

# Importing scenario in Cadena

Tasks (14 items)					
✓	!	Description	Resource	In Folder	Location
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 42
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 48
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 54
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 60
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 66
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 72
✗		Expecting instance_name.port for Display or DumbWatchSetting3_GabrieleTry.ProcessingStatus	RobotAss...	Integration_TestDrive/sys...	line 78
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 84
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 88
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 120
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 139
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 142
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 145
✗		Expecting same types for source and destination of connection	RobotAss...	Integration_TestDrive/sys...	line 149

**Cadena detects type inconsistencies for connections**

# Imported in Cadena



# Reimporting scenario in PICML

GR Engine v1.3.1.

Copyright (c) 2000-2004 ISIS, Vanderbilt University

Engine Started. Loading relevent files...

Running...

Executing Block RootRuleBlock

Executing Rule Assembly\_and\_Scenario\_Finder

-- InputPackets:1 Matches:1 OutputPackets:1

Executing Block Associate\_Components\_n\_PublishConnectors

Executing Rule AssociatePublishConnectors

-- InputPackets:1 Matches:0 OutputPackets:1

Executing Rule AssociateComponents

-- InputPackets:1 Matches:10 OutputPackets:10

Executing Block CreateDestroy\_Components\_n\_PublishConnectors

Executing Block CreatePublishConnectors\_step1

Executing Rule CreatePublishConnectors\_step1\_rule

-- InputPackets:1 Matches:0 OutputPackets:1

Executing Block CreatePublishConnectors\_step2

Executing Rule CreatePublishConnectors\_step2\_rule

-- InputPackets:1 Matches:0 OutputPackets:1

Executing Block CreatePublishConnectors\_step3

Executing Rule CreatePublishConnectors\_step3\_rule

-- InputPackets:1 Matches:0 OutputPackets:1

Executing Rule DestroyPublishConnectors

-- InputPackets:1 Matches:2 OutputPackets:2

Executing Rule FindComponents2BDeleted

-- InputPackets:1 Matches:0 OutputPackets:0

Executing Rule CreateComponents\_step1

-- InputPackets:1 Matches:0 OutputPackets:0

Executing Block TearDown\_Connections

Executing Rule Optimization\_FindComponents

-- InputPackets:1 Matches:10 OutputPackets:10

Executing Rule TD\_Emits

-- InputPackets:10 Matches:8 OutputPackets:13

Executing Rule TD\_Invokes

-- InputPackets:10 Matches:8 OutputPackets:13

Executing Rule TD\_DeliverTo

-- InputPackets:10 Matches:0 OutputPackets:10

Executing Rule TD\_SrcPublish

-- InputPackets:10 Matches:0 OutputPackets:10

Executing Block Associate\_Ports

Executing Rule Match\_Components

-- InputPackets:1 Matches:10 OutputPackets:10

Executing Rule Associate\_Ports\_rule

-- InputPackets:10 Matches:76 OutputPackets:76

Executing Block ReCreate\_Connections

Executing Rule Create\_DeliverTo

-- InputPackets:1 Matches:0 OutputPackets:1

Executing Rule Create\_SrcPublish

-- InputPackets:1 Matches:0 OutputPackets:1

Executing Rule Create\_Invokes

-- InputPackets:1 Matches:4 OutputPackets:4

Executing Rule Create\_Emits

-- InputPackets:1 Matches:7 OutputPackets:7

Executing Block Associate\_Properties

Executing Rule Associate\_CadenaProperties\_on\_PublishCon

-- InputPackets:1 Matches:0 OutputPackets:1

[.....]

The Transformation took 45.347seconds

Done!

# Reimporting scenario in PICML



DumbWatchSetting1\_GabrieleTry  
[ WatchSettingManager ]



DumbWatchSetting2\_GabrieleTry  
[ WatchSettingManager ]



DumbWatchSetting3\_GabrieleTry  
[ WatchSettingManager ]

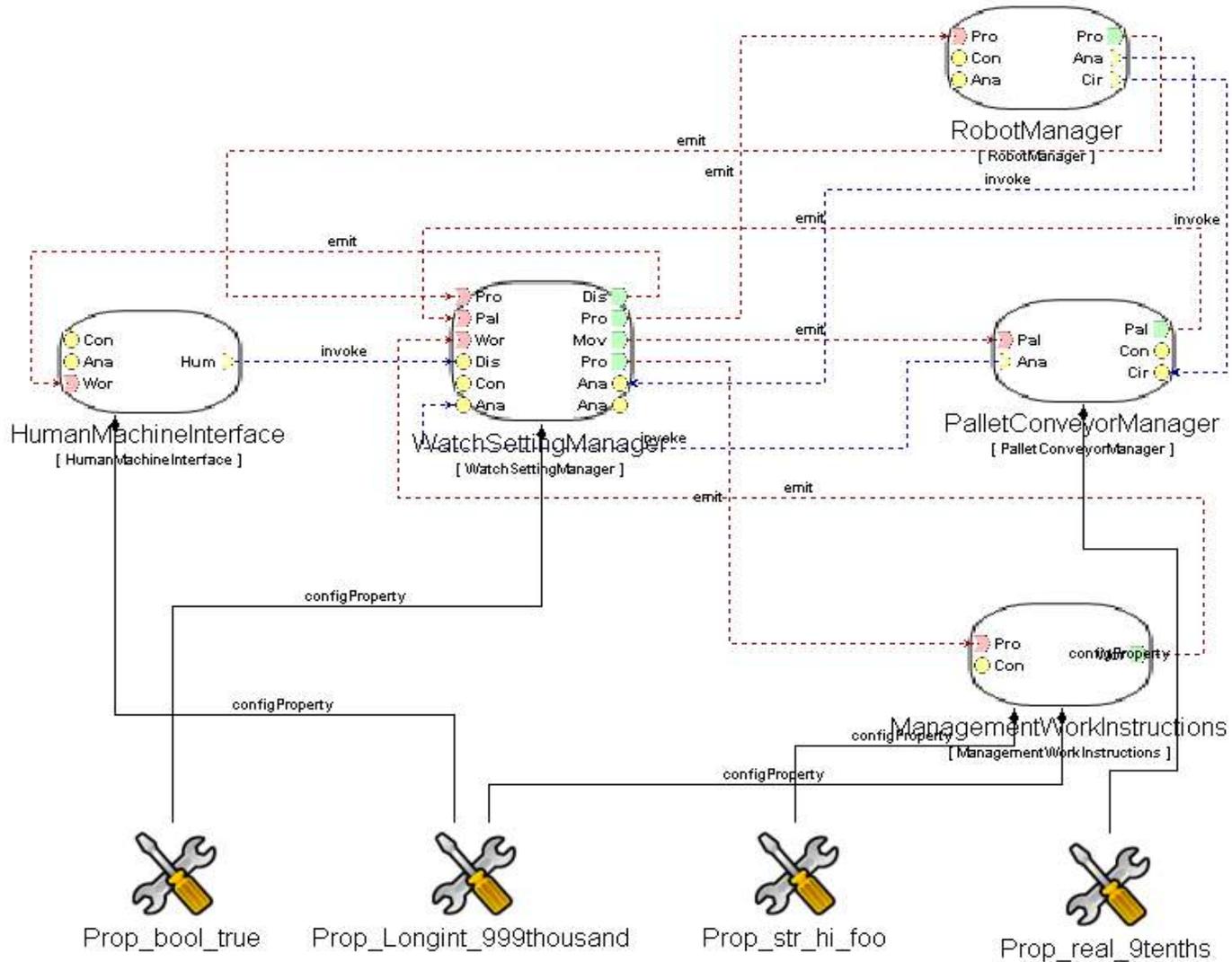


DumbHumanMachineInterface1\_GabrieleTry  
[ HumanMachineInterface ]



DumbRobotManager1\_GabrieleTry  
[ RobotManager ]

# Reimporting scenario in PICML (valid part)



# Special Thanks

Special Thanks to:



Institute for Software Integrated Systems - Vanderbilt University

Computer and Information Systems Department - Kansas State University

Object Management Group

Developers of: Cadena, GME, GReAT, OTIF, UDM, Python.org

More information about this opensource integration project and the latest version of the slides at:

WEBPAGE: <http://www.dre.vanderbilt.edu/~gtrombetti/>

E-MAIL: gtrombetti [at] dre [dot] vanderbilt [dot] edu

***Gabriele A. Trombetti***

# QUESTIONS

# ?