MetaModelAgent

for IBM Rational® Software Architect Designer and RealTime Edition
HCL RealTime Software Tooling and Eclipse Papyrus™

Model Analysis Manual

MetaModelAgent version 4.6.0

AO-MMA-023 • 2022-01-05

ADOCUS

Contents

1 Introduction

MetaModelAgent (MMA) is an Eclipse-based modeling tool extension that is installed on top of a host tool and accessed by extended menus and views within the Eclipse workbench. MetaModelAgent supports the following host tools:

- IBM Rational® Software Architect Designer (RSAD)
- IBM Rational® Software Architect RealTime Edition (RSARTE)
- HCL RealTime Software Tooling (RTist).
- Eclipse Papyrus™

This is the user manual for the static model analysis capabilities in MetaModelAgent using the built-in metamodels for standard UML or UML-RT or when using any user-defined domain-specific metamodel. There are two other manuals available in the Help-system after installation:

MetaModelAgent Modeling User Manual	Describes how to activate MetaModelAgent for your models and how to use it for domain-specific modeling using your own metamodel.
MetaModelAgent Metamodeling Manual	Described how to develop your own domain-specific metamodel for usage in MetaModelAgent.
MetaModelAgent License Management	Described how to require and install licenses to be able to use MetaModelAgent.

1.1 Reading instruction

To be able to understand this analysis manual, basic knowledge of the Unified Modeling Language (UML) and the host tool being used, is required.

This manual includes the following chapters:

Chapter 1: *Introduction*, this chapter.

Chapter 2: Terminology*,* explains terms that are used widely in this manual.

Chapter 3: *Activate and inactivate MetaModelAgent*, explains how to activate MetaModelAgent for your models to be able to perform model analysis.

Chapter 4: *Overview of Analysis Views*, gives a comprehensive overview of the Analysis Views provided by MetaModelAgent.

Chapter 5: *Property Table View*, describes how to use the Property Table View for displaying and editing element properties in a spreadsheet layout.

Chapter 6: *Trace Matrix View*, describes how to use the Trace Matrix View to achieve overviews of connected elements in a matrix layout.

Chapter 7*: Trace Tree View*, describes how to use the Trace Tree View to elaborate incoming or outgoing connection paths to and from an element.

Chapter 8: *Chart View* describes how to use the Chart View to display graphical overviews in terms of bar charts and scatter charts of different kind of aspects of your models.

Chapter 9: Static *Behavior Analysis*, describes how to use the different analysis view to perform static analysis of activities, state machines and interactions.

1.2 What´s New

From v4.5.0 to v4.5.1

- Property Table View: Support for metaclasses representing navigable associations and connectors.
- Trace Matrix View: Supports for navigable associations and connectors.
- Trace Matrix View: New *Referenced Classifier filter* for behavior details in Trace Matrix View. See chapter 6.4.
- Trace Matrix View: New *Specializations filter* for matrixes displaying properties (attributes or ports) and associations. See chapter 6.4.
- Trace Matrix View: Refined alternatives for displaying interconnections between selected elements in Property Table View. See chapter 6.1.2.

From v4.5.1 to v4.5.2

- Property Table View: Custom selection of multiple metaclasses to fill the view. See chapter 5.1.1.
- Property Table View: Support for displaying heterogenous list of elements of different types in the same table with a union or intersection of property columns.
- Property Table View: Improved column filter and sorting behavior.
- Trace Matrix View: Custom selection of multiple metaclasses representing connections, sources, and targets to be displayed. See chapter 6.1.1.This can also be applied as a filter afterwards on an already displayed matrix, see chapter 6.5.5.
- Trace Matrix View: Support for initiating the view with derived interconnections between elements in the Property Table View based on a custom selection of connection metaclass(es). See chapter 6.1.2.
- Trace Matrix View: Improved support for long target names by providing resizable and scrollable target labels area.
- Trace Matrix View: Added a CSV-export options that contains list of connections in each non-empty cell as an alternative to only display number of connections in each cell. See chapter 6.3,
- Trace Matrix View: Support for displaying additional kind of behavior element references such as Interaction Uses, Call Behavior Actions, Lifelines etc.
- Trace Matrix View: Hovering over a non-empty matrix cell will display source and target in the tooltip.
- All analysis views: Improved UI when using MacOS or Linux platforms.
- All analysis views: Improved font handling if platform default font is resized or scaled.
- All analysis views: Extended initial explanation on how to fill the view.
- All actions in a view’s toolbar have been made available in the view´s pull-down menu also, to conform to Eclipse user-interface guidelines.

From v4.5.2 to v4.6.0

- Property Table View: Support for listing all accessible features of a classifier with respect to inheritance, see chapter 5.1.1.
- Property Table View: Support for listing classifiers not visible in any diagram, see chapter 5.1.1.
- Property Table View: Support for listing metaclass instantiation overview, see chapter 5.1.1.
- Trace Matrix View: Support for displaying accessible properties (attributes and association ends) with customizable filter with respect to inheritance and abstract classifiers, see chapter 6.1.1.
- Trace Matrix View: Support for displaying element occurrences in diagrams, see chapter 6.1.1.
- Trace Matrix View: Support for displaying distant relationships, e.g., packaged owned relationships positioned away from their source element, see chapter 6.1.1.
- Trace Matrix View: Support for displaying stereotype applications, see chapter 6.1.1.
- Trace Matrix View: New behavior details filter to hide action pins, state entry/exit points and interaction lifelines, see chapter 6.5.3.
- Trace Tree View: New option to merge all references targeting the same element which may lead to a more compact tree, see chapter 7.3.
- Improved support for Eclipse Dark theme.

1.3 Support

To obtain support, please use the support request form on the Adocus Website (www.adocus.com) or send an email to support@adocus.com.

2 Terminology

The following terms are used frequently in this user manual.

2.1 Modeling terms

Term	Explanation
Item	Item is the generic term for an element, a diagram, or a relationship. Examples of items are classes, dependencies, and activity diagrams.
Property	A property is a predefined or user defined feature of an item. Examples of common properties are name, stereotype, multiplicity, documentation.
Model	A model is a UML-model holding a set of items which represents a model as defined in UML, e.g., a semantically complete abstraction of a system. Examples of common models are Use Case Models, Analysis Models and Design Models.
Metaclass	A metaclass is a classification of items sharing the same characteristics. Examples of standard UML metaclasses are Use Case, Actor and Class.
Metamodel	A metamodel is a UML-model that defines the model guidelines for a specific kind of models in a formal way using metaclasses. A model will therefore be an instance of its metamodel.

2.2 Eclipse terms

Explorer View	The Explorer View is the name used in this manual for the standard view that shows the model structure of loaded models. • *Project Explorer View* in RSAD/RSARTE and HCL RTist. • *Model Explorer View* in Papyrus.
Diagram Editor	The Diagram Editor is the graphical editor view for editing UML-diagrams.
Property View	The Property View is used for viewing and editing item properties.

3 Activate and inactivate MetaModelAgent

3.1 Activate MetaModelAgent

MetaModelAgent must first be activated for a model, or part of a model, to be able to use the analysis functionality. If the model is connected to a custom metamodel, that metamodel will be used, otherwise the built-in general metamodels for UML or UML RT will be used.

Activation can be performed as follows:

- Automatically when a model is opened in the workspace, this is controlled by a preference setting.
- Manually by selecting an open model or part of a model in the explorer view or diagram editor and then either select *MetaModelAgent→Activate* followed by the metamodel to be used, from the context menu, or by using the MetaModelAgent drop-down button in the workbench toolbar.
  You can even select several open models or parts of models and select *MetaModelAgent→Activate* to activate them all in one single operation.
- Manually by selecting the green Activate-button in the MMA Activation View. This brings up a popup-dialog where you can activate one or several open models in a single operation. With this alternative you will also be able to activate MetaModelAgent for library models not available in the Explorer View.

See *chapter 4 Start Using MetaModelAgent* in the ***MetaModelAgent Modeling User Manual*** for details.

3.2 Inactivate MetaModelAgent

MetaModelAgent will be automatically inactivated for a model when a model is closed or deleted.

If you would like to inactivate MetaModelAgent for some of the current activations, use one of the following alternatives:

- Select the root element of one or several activations in the Explorer View or in the Diagram Editor. Select *MetaModelAgent→Activate→Inactivate* in the context menu
- Select one or several activations in MMA Activation View and select *Inactivate* from the context menu.

If you would like to inactivate MetaModelAgent for all your current activations, use one of the following alternatives:

- Select any element in the Explorer View or Diagram Editor and select *MetaModelAgent→Activate→Inactivate All* in the context menu
- Press the red Inactivate All button in the MMA Activation View.
- Press the clear button in the MMA Problem View.

IMPORTANT: The content of the MMA Analysis View is not automatically updated to reflect any of the models being displayed being inactivated or even closed. The content of these views might therefor be obsolete after inactivation of a model.

4 Overview of Analysis Views

MetaModelAgent provides four different views that are used for model analysis:

- The **Property Table View** lets you inspect elements, relationships and diagrams and their significant properties in an editable table layout. The table can be filtered and sorted on any column and the content may also be exported to a CSV-file for further post-processing, for example in MS Excel.
- The **Trace Matrix View** lets you inspect relationships and other kinds of references in a matrix layout. It can also be used to detect hidden relationships and implicit relations between and within models based on underlying explicitly modeled relationships.
- The **Trace Tree View** lets you inspect relationship chains starting from or finishing at a specific element. You can easily detect unwanted relationship chains and circularities.
- The **Chart View** lets you visualize several aspects of your model in graphical bar charts and scatter charts.

4.1 Interconnected views

MetaModelAgent provides unique capabilities to navigate between the views to further elaborate the structures, contents, and behavior of your models.

The picture below, summarize the different ways each view can be invoked based on the content in other views.

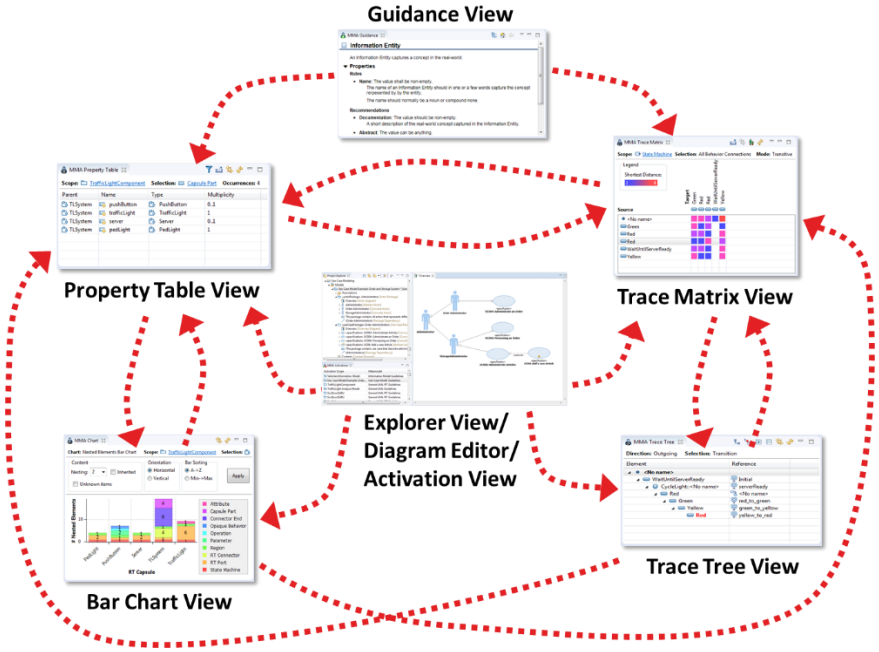

Figure 1: *Schematic picture on potential navigation between the different views using each view´s context menus*

The capabilities and features of each view are described in detail in the following chapters.

5 Property Table View

The *Property Table View* is a unique spread-sheet table view which primarily shows an overview of all significant properties for all items within a selected scope that belongs to the same metaclass.

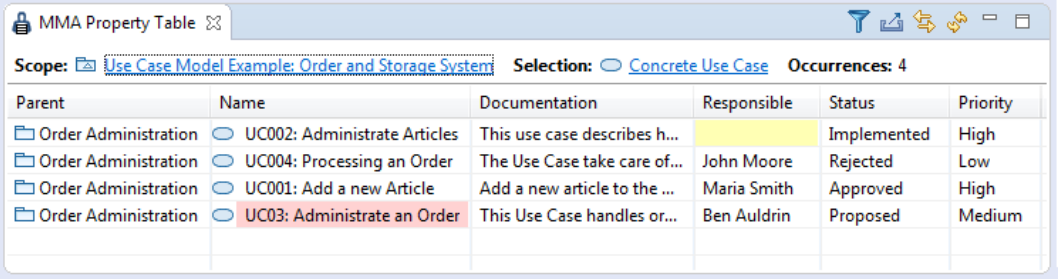

Figure 2: *Property Table View showing significant properties for all concrete use cases within the package "Order Administration".*

Any invalid property value, according to the current metamodel is displayed with a background color that indicates the severity of the violence.

All non-fixed properties are editable. Just click in the corresponding cell and edit the value. Depending on the kind of values that are valid, popup dialogs for selecting or editing a value may occur.

5.1 Invoking the View

5.1.1 From Explorer View, Diagram Editor and Activation View

Instances of a specific metaclass

To show all items that are instances of a specific metaclass:

1. Select one or several elements in your activated models to be the scope.
2. In the context menu, select *MetaModelAgent→Show in Property Table View*.
3. There will be one menu entry for each metaclass that is valid within the selected scope. If there are a lot of metaclasses to choose from, they will be organized in consecutive submenus according to the current metamodel.

All items that match the selected metaclass and that are nested to the selected scope will be displayed in the table view.

All instances of a specific metaclass

An alternative way to populate the Property Table View when using your own metamodel available in the workspace:

1. Select a metaclass for which there are at least one current instance
2. Bring up the context menu and select the context menu entry *MetaModelAgent→Show in Property Table View→Show all Occurrences*.

This will display all instances of the selected metaclass from al current activations.

Custom Selection

The menu entry *Custom Selection…* brings up a dialog where you can select several metaclasses. All items within the selected scope that matches any of the metaclasses will be listed in the view.

Only properties that are common for all items will be visible initially. Additional properties are made visible using the column filter dialog available in the view's button bar.

Pre-defined selections

There are currently three pre-defined menu entries in the *MetaModelAgent→Show in Property Table View* submenu.

All Accessible Features	[Only available when a single classifier is selected] Lists all features such as attributes, ports and operations that are owned or inherited by the selected classifier.
Invisible Classifiers	Lists all classifiers within and including the selected elements that are visible in at least one diagram within the models of the selected elements. E.g., if you select a package within a model, the Property Table View will display all classifiers within the package that do not show up in any diagram within the model.
Instantiated metaclasses	Lists all metaclasses for which there are instances within the selected scope. For each metaclass the following information are displayed in separate columns; UML element type, element category, and number of instances.

5.1.2 From Guidance View

The Property Table View can show all instances of the metaclass that is currently displayed in the Guidance View.

1. Make sure that the Guidance View shows the metaclass for which you are interested in, by selecting an item belonging to that metaclass in some other view or in the diagram editor.
2. Select *Show all occurrences in Property Table View* in the view´s pull-down menu in the view's header.

The Property Table View will then display all occurrences of the current metaclass in all activated models.

5.1.3 From other Views

The Property Table View can be populated in different ways from the other analysis views as well. Please refer to the description of those view's content menus for details.

5.2 View Pull-down menu and Toolbar

The following actions are available in the view´s pull-down menu and some of them also in the view´s toolbar:

Filter Columns — Brings up a dialog where you can sort and filter out the columns to be displayed in the table view. There are always three potential columns that are initially hidden and can be revealed using this dialog, those are:

- *Path*; complete path from the model root to the element.
- *Metaclass*; the item's metaclass. This column is useful if the view is used to display a heterogeneous list of items.
- *Metamodel*; the metamodel that the model holding the element is using. This column is useful if the view is used to display items in models where different metamodels are applied.

Export to File — Brings up a dialog where you can export the current view content to a comma-separated text file for post-processing in other tools, for

example MS Excel.

Link to Selection	Links the content of the view to the element selected in the Explorer View or in the Diagram Editor. This means that the view will be update with the new selection as scope, but the same selection of metaclass and applied column filter remains.
Refresh	Refreshes the content of the view. This is useful if the items being displayed have been changed in some other view.

5.3 Context Menu

Bringing up the context menu for a single selected cell in the table representing one or several elements makes it possible to navigate to the element(s) in the explorer view or to delete the element in the corresponding row.

Bringing up the context menu when two or more rows are selected reveals the following context menu entries:

Show in Trace Matrix View	Populates the Trace Matrix View in several different ways. Please refer to the chapter 6.1 for details.
Show in Chart View	Populates the Chart View in different ways. Please refer to chapter 8.2.2 for details. This menu entry is only available if all elements in the table belongs to the same metaclass.
Bulk Property Edit	Enables bulk editing of a specific property. All selected rows must represent the same kind of element for this submenu to be available. This menu entry is only available if all elements in the table belongs to the same metaclass.
Delete <n> selected <Metaclass>	Deletes the items that are represented by the selected rows from the model. This menu entry is not always available, depending on from where and how the Property Table View was invoked.

6 Trace Matrix View

The *Trace Matrix View* displays the existence of relationships, other connections as well as element references, hereafter only referred as *references*, in a grid style layout. There is one row for each source element and one column for each target element.

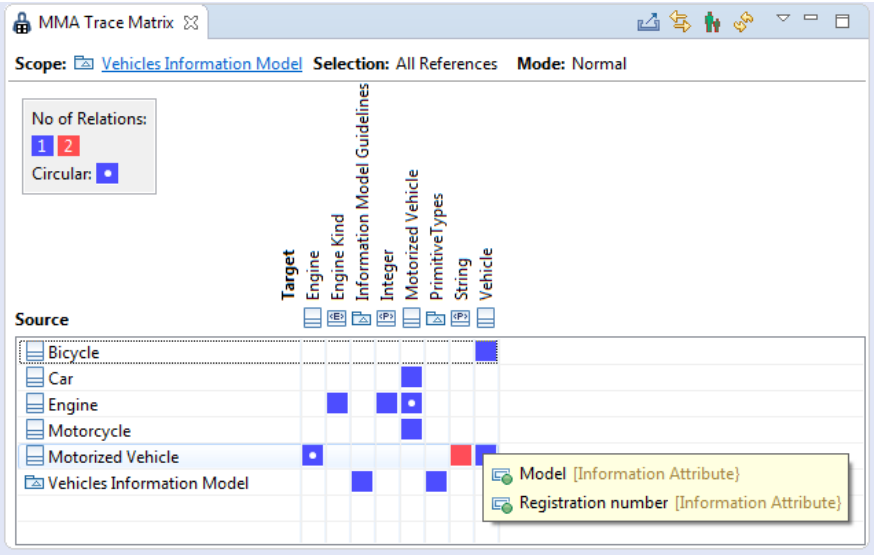

Figure 3: *Trace Matrix View showing all relations and attribute types.*

The intersection cell between a source and a target indicates if there are any direct or indirect references or references chains, depending how the Trace Matrix View was invoked, from the source to the target element.

- The **cell color** (gradual from blue to red) indicates the relative number of references between source and target or the distance between source and target in terms of consecutive references. Blue color represents the longest number/shortest distance. Red color represents the highest number/longest distance.
- A **white dot** in the middle of a colored cell indicates that the source and target element is involved in a circular reference chain.
- A **grey cell color** indicates that the cell represents an implicit reference between source and target that is relevant in the context of current selection. This is only used in static behavior analysis of state machines, activities, and capsule connectors. See chapter 9.
- An **empty cell** indicates that there are no references or reference chains that fulfills the chosen criteria between the source and target element.

The view does not support horizontal scrolling, if there are more target elements that can fit in the view, numbered “page” buttons will be displayed that let you switch between different “pages” of targets.

You may observe that this view is just a read-only snap-shot view of relationships or association ends. You may not edit the relationships in the view and any changes to the relationships from other parts of the UI will not be automatically reflected.

6.1 Invoking the View

6.1.1 From Explorer View, Diagram Editor and Activation View

Instances of a specific metaclass

To show all relations or attribute type references that are instances of a specific metaclass:

1. Select one or several elements in the Explorer View or Diagram Editor for which you are interested in their nested relationships or element references.
2. Bring up the context menu and select *MetaModelAgent→Show in Trace Matrix View*.
   There will be one menu entry for each metaclass representing a relationship, other connection or referring element (see *Appendix A*: *Referring elements* for details) that are valid within the selection and for which there are at least one matching occurrence within the selection.

IMPORTANT: It is possible to invoke the Trace Matrix View with activity flows, state machine transitions or capsule connectors but they will only display the immediate sources and targets in the matrix. To analyze those kinds of connections in a correct way with respect to UML and UML-RT semantics, you should use the static behavior analysis features, see chapter 9 for details.

All instances of a metaclass

An alternative way to populate the Trace Matrix View when the metamodel is available in the workspace:

1. Select a metaclass in the metamodel representing a relation or referencing element for which there are at least one current instance,
2. Bring up the context menu and select the context menu entry *MetaModelAgent→Show in Trace Matrix View→Show all Occurrences*.

This will display all instances of the selected metaclass from all current activations.

Custom selection of metaclasses

The menu entry *Custom Selection…* brings up a wizard where you on three successive pages select metaclasses representing connections & referencing elements, sources, and targets to be displayed in the matrix view.

The source metaclasses selectable on the second page will depend on the selected connection metaclasses on the first page and the selectable target metaclasses on the final page will depend on the selected connection and source metaclasses on the previous pages.

Pre-defined selections

There are several pre-defined menu entries in the *MetaModelAgent→Show in Trace Matrix View* submenu. Some of those will only appear if they are applicable according to the metamodel. These menu entries are only available when one single element is selected

 *All Invisible Relationships*	Shows all relationships within the selected scope that are not visible in any structural diagram within the current model.
 *All Unknown Relationships*	Shows all unidentified relationships within the select scope. An unidentified relationship is a relationship that does not match any metaclass in the metamodel. You may notice that nested relationships to unidentified elements are not reported.

All Incoming Relationships	Shows all incoming relations to the selected element and its nested elements from elements outside the selected element.
All Outgoing Relationships	Shows all outgoing relations from the selected element and its nested elements.to elements outside the selected element.
All Relationships	Shows all relationships owned by the selected element or any nested element.
All Navigable Associations	Shows all navigable associations that are navigable from the selected element or any nested element. Associations that are not navigable in any direction will not be included.
All Composition Properties	Shows all properties representing compositions owned by the selected element or any nested element. Matrix cells will display the property, the owner of the property will be the source and the type of the property will be displayed as target.
All Properties	Shows all properties owned by the selected element or any nested element. Matrix cells will display the property, the owner of the property will be the source and the type of the property will be displayed as target.
All References	Shows all relations, connectors and referring elements (see *Appendix A: Referring elements*) owned by the selected element or a nested element.
Interface Connections	Shows all interfaces within the selected element and nested elements. Matrix cells will display interfaces and each row will display an interface consumer (that has a usage relationship to the interface) and each column will display an interface provider (that has an realization relationship to the interface).
Accessible Properties…	Brings up a dialog where you in detail can control which kind of properties to include, if inheritance should be considered, and if abstract source and target classifiers should be excluded or not.
Distant Relationships	Displays all package-owned directed relationships where the source element is within the selected scope and the relationship is stored in another package than the source element.
Stereotype Appplications	Displays a row for each stereotyped element within the selected scope and a column for each stereotype. The intersection cell indicates if the element applies the stereotype.
Element Visiblity	Shows one row for each element within the selected scope and one column for each diagram in the selected scope. Each cell indicates if the element is present in the corresponding diagram.

Visible and invisible connections in a diagram

By selecting a diagram in the explorer view or the background canvas of a diagram in the diagram editor, you will have two menu entries within the *MetaModelAgent→Show in Trace Matrix View* submenu:

Visible Connections	Shows all navigable connections (edges) that are visible in the diagram.
Invisible Connections	Shows all connections between elements in the diagram that do not occur in the diagram. Associations where both the association ends are owned by the association itself will not be included.

6.1.2 From Property Table View

From the Property Table View it is possible to populate the Matrix View based on a more precise selection of scope and to display implicit derived relations, based on underlying explicit relationships.

In this context, an *Implicit Derived Relation* between two elements occurs if the two elements either has a direct relationship two each other or consists of at least one pair of nested elements that have a relationship between each other.

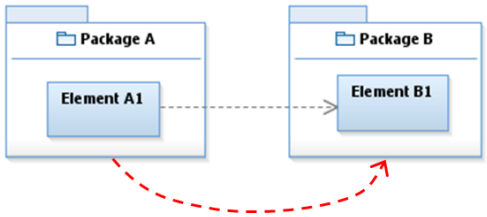

Figure 4: *There is an implicit derived relation between Package A and Package B, manifested by the explicit relationship between* A1 *and B1.*

Being able to display implicit derived relations is an especially useful feature to be able to find out how larger parts of a model is related to other parts, or how a complete model is related to other models.

Show in Trace Matrix submenu in the *Property Table View* context menu:

Outgoing Relations →<Metaclass>	Shows direct outgoing relationships of the selected kind of relationship, or all kinds of relations, from the selected set of elements.
Derived Outgoing Relations Towards →<Metaclass>	Shows all outgoing relationships from the selected elements or from any of their nested elements to elements of the selected metaclass.
Derived Incoming Relations From →<Metaclass>	Shows all incoming relationships to the selected metaclass to the selected elements or to any of their nested elements.
Derived Interconnections	Shows connections between the selected elements or their nested elements. You can choose to show only a specific kind of relations or from several predefined queries.

6.2 View Pull-down menu and Toolbar

The following actions are available in the view´s pull-down menu and some of them also in the view´s toolbar:

Show Context	Shows the context of the source and target elements. This feature may have no effect in some selections.
Refresh	Refreshes the content of the Trace Matrix View. If the view has been resized, the number of targets displayed are adjusted to fit the new view width. You may observe that the content of the view is normally not updated if it has been changed elsewhere. To make sure the view is updated with changed data, make a new invocation from the context menu you last used.
Link to Selection	Link the content of the Trace Matrix View to the element selected in the Explorer View or in the Diagram Editor. This means that the same selection criteria will be used but the scope of the selection

		will change.
	*Export to File*	Exports the content of the view to a CSV-file which can be post-processed in some other tool, for example MS Excel. When in normal model, you will be able to select if details of cell content (list of connections) will be exported or only the number of connections. In Transitive modes the shortest reference chain length between each source and targets will be exported.
	XXX Mode	Mode switches are described in detail in chapter 6.4.
	XXX Filter	Matrix filters are described in detail in chapter 6.5.
	*Show in Property Table View*	Initiates the Property Table View with all connections in the matrix.
	*Show in Chart View*	Initiates the Chart View with connected elements based on the context in the matrix view in a bar chart or scatter chart style.

6.3 Context Menu

There are context menus available for all source and targets elements, as well as for the filled cells in a matrix. Where the *Property Table View* and *Trace Tree View* can be populated based on the content in the matrix.

For a selected source element:

	*Show Outgoing(s) in Trace Tree View*	Displays in Trace Tree View a tree of all outgoing direct, and indirect, references started from the source element.
	*Show Targets in Property Table View*	Displays in Property Table View a list of all elements referenced directly from the selected source element. If the matrix is in Transitive Mode all directly and indirectly referenced elements will be included in the list.

For a selected target element:

	*Show Incoming(s) in Trace Tree View*	Displays in Trace Tree View a tree of all incoming direct, and indirect, references ending at the target element.
	*Show Sources in Property Table View*	Displays in Property Table View a list of all elements referencing the target element. If the matrix is in Transitive mode all directly and indirectly referencing elements will be included in the list.

For a filled intersection cell in *Normal Mode*:

	*Show in Property Table View*	List all references from the source to the target in the Property Table View.
	*Show in Model Explorer*	Highlights the reference in the *Explorer View*. If the cell represents several references, there will be a submenu with an entry for each reference.

For a filled intersection cell in *Transitive Mode*:

	*Show shortest path(s) in Trace Tree View*	Displays in Trace Tree View a tree of the shortest paths between the source and target in terms of intermediate elements.
	*Show all path(s) in Trace Tree View*	Displays in Trace Tree View all paths between source and target element.

6.4 Matrix Modes

MetaModelAgent provides three matrix modes which can be altered using the entries in the view´s pull-down menu. The modes are described in detail below.

6.4.1 Normal Mode

This is the default mode when the matrix view is initiated from menus in other views. The color of each cell indicates the number of relations and/or references there are between the source and target element according to operation selected to initiate the matrix.

6.4.2 Transitive Mode

Transitive mode shows if two elements are related to each either directly or indirectly by following one or several consecutive relationships. Each cell representing such an *implicit transitive relation*. The cell will have a color from blue to red that indicating the shortest distance between source and target.

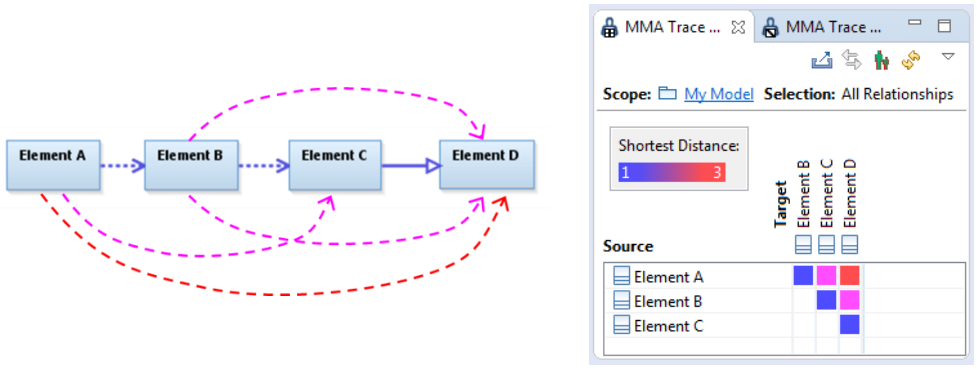

Figure 5: *The blue cell indicates a direct relationship.*
The magenta and red cells indicate implicit transitive relations between the element with the shortest distance of two and three.

Examine the path(s) of transitive relations

The relationship path(s) between source and target elements can be examined in detail in the *Trace Tree View* by selecting "*Show all path(s) in Trace Tree View*" from the cell's context menu.

IMPORTANT: If there are a lot of unique paths between source and target, it may take several seconds or even minutes to calculate all paths.

6.4.3 Transitive (Start-End) mode

Transitive (Start-End) Mode is a filtered variant of the *Transitive Mode* where only complete paths between elements are shown.

There will be a colored cell indicating the shortest distance from a source element to a target element only if there are no incoming relationships to the source element and no outgoing relationship to the target element.

Paths that contain loops and therefore do not have final element will not be displayed.

In the example above, only the red arrow between Element A and Element B would be displayed in the Transitive (Start-End) Mode.

6.5 Matrix Filters

The Matrix View provides several useful filters. All filters are available in normal mode and some of them in all modes. The filters are controlled from the view´s pull-down menu and are applied one after another. That means that the effect will differ dependent on the order in which they are applied.

IMPORTANT: un-applying filters in another order than applied may give some unwanted effect. You are recommended to un-apply all filters prior to apply another order of filters.

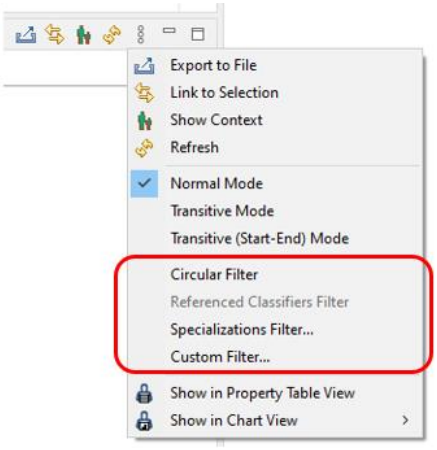

Figure 6: *Trace Matrix View dropdown menu for applying filters. Only filters applicable for the current content are enabled.*

6.5.1 Circular filter

This filter removes all matrix entries that do not represent at least one circular connection or connection chain. This also means that rows which represent source elements and columns which represent target elements that do not participate in any circular connection chains will be removed from the matrix.

6.5.2 Referenced Classifiers Filter

This filter only applies to matrixes displaying behavior details and/or instances. It will substitute source and target elements to classifiers that are referred from the original elements. The content of matrix will thereafter be recalculated to reflect the new source and target elements.

The substitution rules for source and target elements are as follow:

Element type	**Substituted to the following classifier**
Activity Partition	Value of the *Represents* property
Call Behavior Action	Value of *Behavior* property.
Data Store Node	Value of the *Type* property.
Central Buffer Node	Value of the *Type* property.
Activity Parameter Node	The owning element
Pin	The owning element. If the owning element is a *Call Behavior Action* then its value of the *Behavior property*, if not null.
Message	Value of the Signature property
Lifeline	Value of the *Type* property of the value of the *Represents* property. If null, the value of the *Represents* property is returned.
Interaction Use	Value of the *Refers to* property

Element type	Substituted to the following classifier
Instance Specification	First value of the *Classifier* property
Property	The owner of the property
Operation	The owner of the operation

If the result of a substitution is null, the original element remains.

6.5.3 Behavior Details Filter

This filter only applies to the normal mode when the matrix displays source or targets that are entry/exit points in states, pins in actions or lifelines in interactions.

The filter replaces:

- sources and targets that represent input/output points with the state that owns the input/output points.
- sources and targets that represent pins with the action that owns the pin.
- sources and targets that represent lifelines with the property that the lifeline represents. The lifeline will remain if it does not represent a property.

6.5.4 Specializations Filter

This filter only applies to the normal mode when the matrix displays navigable properties or interface connections.

In a matrix displaying navigable properties, the filter expands the matrix with specializations of source and target classifiers, hiding abstract classifiers, with respect to the semantics of UML generalizations.

- A source element which represents a classifier is substituted with all its sub-classifiers that inherits any of the properties in the matrix that are owned by the original source element. Abstract classifiers are omitted and that will include abstract original source elements.
- A target element which represents a classifier is substituted with all its sub-classifiers that inherits from the original source element any of the properties in the matrix that are owned by the original target element. Abstract classifiers are omitted and that will include abstract original target elements.

In a matrix displaying the realization and usage of interfaces, the filter will expand the matrix with new targets (e.g., realizators) for all non-abstract classifiers that inherits from a classifier realizing an interface and with all non-abstract interfaces that a displayed interface inherits from. All abstract realizators and abstract interfaces are omitted.

The lack of users of an interface or realizators of an interface will be denoted by a placeholder representing “no element” in the source row or target column, respectively.

6.5.5 Custom Filter

This filter makes it possible to select which kind of sources, targets and connections that should remain in an already displayed matrix. A wizard will appear where you select metaclasses for connections, sources, and targets on three successive pages for which connections should remain in the view. Candidate source metaclasses are filtered based on the selected connection metaclasses, and candidate target are filtered based on selected connection and source metaclasses on the previous pages.

7 Trace Tree View

The *Trace Tree View* is used to display reference chains starting from, and/or ending at, a specific element. Within the view, several features support the analysis of complex reference chains.

All kind of UML relationships can be traced in this view including navigable associations, object flows, control flows and transitions. The tree view also supports other kind of references such as attribute, call behavior actions in activities and messages between lifelines in interactions. The tree view will also support all kind of content in the *Trace Matrix View*.

The term “Reference” is used for all supported kind of relations and referencing elements throughout this chapter and in the user interface.

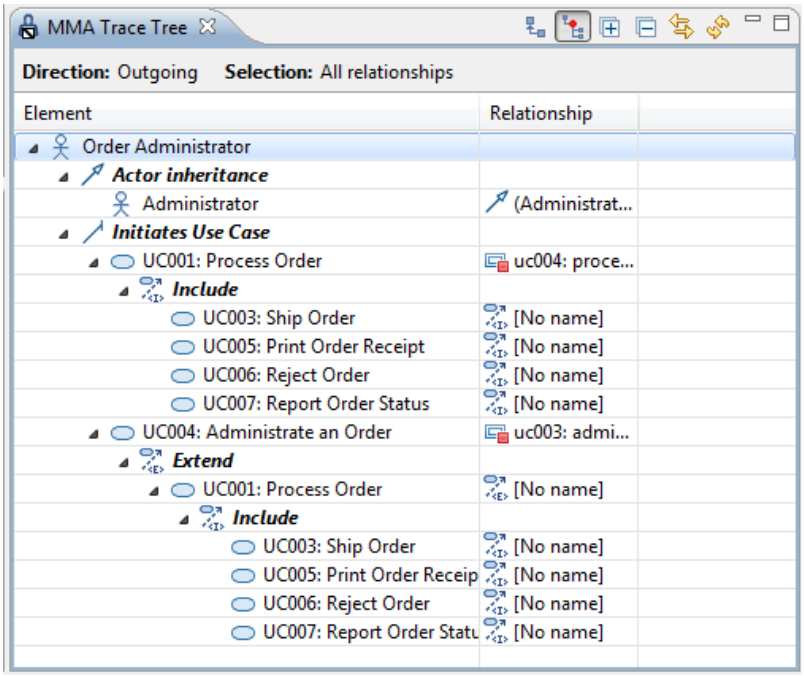

Figure 7: *Trace Tree View displaying all relationships starting from the actor Order Administrator.*

If an existing element already displayed occurs again then it will be highlighted in red, indicating a circular reference.

By selecting an element in the tree, all other occurrences of the same element in the tree are highlighted in blue and a popup-dialog will display the complete reference chain from the root element to the selected element. This makes it easy to inspect the details of a specific reference chain.

Double-clicking an element will highlight the element in the Explorer View.

7.1 Invoking the View

7.1.1 From Explorer View and Diagram Editor

To display the outgoing or incoming reference chain from a specific element in the Explorer View or Diagram Editor.

1. Select one element in the Explorer View or in the Diagram Editor for which you are interested in their nested references to other elements.

2. Bring up the context menu and select *MetaModelAgent→Show in Trace Tree View.*
3. Continue by selecting *Outgoing References* or *Incoming References* in the nested submenu.
4. Select the metaclass representing the kind of relationship or reference to be traced.
   You will also be able to select *All Relations* and/or *All References* which will include all kinds of supported relations or both relations and other kind of references.

IMPORTANT: It is possible to select metaclasses representing state machine transitions, control flows and object flows using the steps described above. But those selections will not respect UML semantics of state machines and activities. To analyze those kinds of relationships with respect to semantics, start by populating the *Trace Matrix View* using *Static Behavior Analysis*, see chapter 9, from which the Trace Tree View then could be populated as described below.

7.1.2 From Trace Matrix View

Based on a populated Trace Matrix View, the Trace Tree View can be used to further elaborate the content starting from a source element, a target element, or a filled matrix cell.

Selection	Context menu entry	Description
Source element	*Show outgoing(s) in Trace Tree View*	Displays a tree of all outgoing references represented in the Matrix View starting from the selected source element.
Target element	*Show incoming(s) in Trace Tree View*	Displays a tree of all incoming references represented in the Matrix View ending at the selected target element.
Filled matrix cell	*Show shortest path(s) in Trace Tree view*	Displays a tree of all shortest paths from source to target in terms of intermediate elements.
	Show all path(s) in Trace Tree view	Displays a tree of all potential paths from source to target.

7.2 View Pull-down menu and Toolbar

The following actions are available in the view´s pull-down menu and some of them also in the view´s toolbar:

Show Incoming	Switches the tree to show incoming references to the root element instead of outgoing references. This option is disabled if the tree is filtered, see context menu entry *Hide paths to other elements* in chapter 7.3.
Group References	Groups all references of the same kind and display them as nested nodes in the tree.
Merge references by element	Merge all references targeting the same element into one combined reference. This option is disabled if the tree has been filtered, see context menu entry *Hide paths to other elements* in chapter 7.3.

Hide duplicates	Only displays each identical subtree once where it occurs first. Other occurrences will have the label Duplicate and will not be expandable. This option is disabled if the tree has been filtered, see context menu entry *Hide paths to other elements* in chapter 7.3.
Turn the tree over	This option is only enabled if the tree has been filtered to only display paths to a specific final element, see *Hide paths to other elements* in chapter 7.3. Selecting this option will turn over the tree so the final leaf element will be the root and the root will be the leaf.
Expand All	Expand the tree to the bottom. E.g., all indirectly related elements will be exposed.
	IMPORTANT: To avoid long execution time. The expansion will stop after a certain number of paths have been expanded. Very deep paths will also be truncated.
Compress All	Compresses the tree, only the root will be visible.
Refresh	Refreshes the content of the view.
	IMPORTANT: As the content of the view in many cases are pre-calculated, refreshing the view does not always reflect changes in the underlying model made in other views and editors. To make sure that the view shows correct data after changes in the model, invoke the view once again.
Link to Selection	Link the content of the view to the element selected in the Explorer View or in the Diagram Editor. This means that the same selection criteria will be used but the scope of the selection will change. This will only work if the Trace Tree view was invoked from the Explorer View.

7.3 View Context Menu

The context menu of the trace tree has the following menu entries:

Show Path Details	Brings up a popup-dialog where the path from the root to the selected element is displayed in a table layout.
Expand All	Expands the tree starting from the selected element.
	IMPORTANT: To avoid long execution time. The expansion will stop after a certain number of paths have been expanded. Very deep paths will also be truncated.
Collapse All	Collapses the tree until the selected element.
Navigate to reference in Model Explorer	Selects the reference in the Model Explorer View.
Set as Root	Makes the selected element the new root of the tree.
Hide paths to other elements	Filters the tree so that only paths that leads from the root to the selected element will remain. The tree will be automatically expanded to show all paths.
Show in Property Table View	Shows all instances of a specific metaclass below the selected item in the reference chain in the Property Table View.

Show in Trace Matrix View	Shows all references beneath the selected element in the reference chain in the Trace Matrix View.
Show in Trace Diagram	Creates a new class diagram within the selected element that will visualize the complete reference chain starting from the selected element.

IMPORTANT: If the selected element is not an element that can hold a class diagram, no diagram is created.
If the elements and reference to be displayed cannot exist in a class diagram, the resulting diagram will be incomplete.

7.4 Labels in the tree

- The blue Duplicate label in the end of an element label indicates that the path below the element is a duplicate of an identical path already displayed. This is only displayed if the *Hide Duplicates* settings is checked.
- The red Circular label in the end of an element label indicates that it is part of a circular path, and that the same element occurs in an ancestor node in the tree.
- An underline label indicates that there is a continuation from the displayed element, but it has been truncated, typically because of an applied filter.
- If a node in the tree represents an element which of some reason is unloaded or unavailable, the element label is displayed in italic font.

When selecting a node in the tree representing av element, all nodes representing the same element will be highlighted in bold font.

8 Chart View

The *Chart View* will provide you with different kind of bar charts and scatter charts that will help you analyze your models.

Figure 8: *Chart View displaying an Enumerated Properties Bar Chart of Use Cases organized around status and priority.*

All bar charts can be sorted in alphabetic order or by size and you can switch from vertical bars to horizontal bars. Select a bar and open the context menu to populate the Property Table View or the Problem View with the content of the bar.

In the upper right area of a bar chart, you will see some statistics, Max and min height of the bars, the median bar height, and the average bar height.

By hovering the dots in a scatter chart, you can see the list of elements that each dot represents. In the context menu you will be able to populate the Property Table View with those elements.

8.1 Available Charts

The following kinds of charts are available:

Properties Bar Chart	Displays the distribution of the selected kind of items based on the value(s) of one or two selected properties. One of the properties values will be represented as bars and the other one as groups on the bars. Only properties with an upper multiplicity of 1 will be available for selection. If the number of property values to be displayed as bars exceeds 100 or the number property values to represent groups exceeds 30, the chart will not be displayed.
Nested Elements Bar Chart	Displays the total number of nested elements for the selected kind of elements, organized in groups based on their element kinds. The nesting depth to be included can be set in the settings area.

Related Elements Bar Chart	Displays the number of related elements for each element of the selected metaclass, grouped based on their metaclass. You can set the depth of relations, as well as whether incoming or outgoing relations should be displayed in the settings area. **IMPORTANT:** Associations, control flows, object flows, and transitions are not regarded as relations in this chart.
Problems Bar Chart	Displays the number of problems for each item of the selected metaclass and its nested items, grouped based on the severity of the problems. You can set the depth of nested items to be included in the settings area.
Problem Distribution Bar Chart	Displays the distribution of problems based on a selection of two of the following three criteria: *Severity*, *Problem Kind* and *Metaclass*. One of the selected criteria will make up the bars and the other one will be represented by groups on each bar. **IMPORTANT:** This chart is only available from the context menu in the Problem View.
Related Elements Scatter Chart	Displays a scatter chart of all nested elements of the specific kind, where one axis represents number of incoming relationships, and the other axis represents the number of outgoing relationships. The color (from blue to red) of the “dots” in chart indicates how many elements that the dot represent. You can set the depth of relationships to be included in the settings area.

8.2 Invoking the View

8.2.1 From Explorer View, Diagram Editor and Activation View

The bar charts and scatter chart are available from the context menu in Explorer View, Diagram Editor and Activation View.

To invoke one of the charts:

1. Select one or several elements as the scope.
2. Bring up the context menu and open submenu *MetaModelAgent→Show in Chart View.*
3. Select the kind of chart among the submenu entries. For each kind you will finally select the metaclass of elements to be displayed.

For each kind of chart, the available settings are displayed in the upper left area. After the settings have been set, click the ***Apply***-button to bring up the corresponding chart.

8.2.2 From Property Table View

The same context submenu with different charts as above is available when selecting one or several element rows in the Property Table View.

8.2.3 From Trace Matrix View

Related Elements Bar Chart and *Related Element Scatter Chart* are available from the Matric View’s view menu. These menu entries make it possible to get a graphical overview of the current content in the Trace Matrix View.

8.2.4 From Problem View

The *Problem Distribution Bar Chart* could be invoked from the Problem View.

1. Select a set of problems in the Problem View.
2. Bring up the context menu and select *Show in Chart View→Problem Distribution Bar Chart* to open the *MetaModelAgent Chart View*.
3. In the chart view header, you can switch between letting the bars and groups represents metaclasses, problem kinds and severities.

8.3 View Pull-down menu and Toolbar

The following actions are available in the view´s pull-down menu and toolbar: :

	*Show Context*	Shows the context of the elements.
	*Refresh*	Refreshes the content of the view. However, you will need to click Apply once again to display the chart.
	*Link to Selection*	Link the content of the view to the element selected in the Explorer View or in the Diagram Editor. This means that the same selection criteria will be used but the scope of the selection will change.

8.4 View Context Menu

By bringing up the context menu a in a bar chart or scatter chart you will have the following context menu entries:

Adjust Axis Range	Submenu with entries for adjusting the range of the X and Y axis.
Zoom In	Submenu with entries for zooming in on X and/or Y-axis.
Zoom Out	Submenu with entries for zooming out on X and/or Y-axis.
Save As…	Opens a dialog where you can save the chart as a picture in png or jpeg format.
Properties…	Provides a dialog where you can customize the layout of the chart in numerus ways.
Show in Property Table View	Only available when a dot in a scatter chart or a bar segment in a bar chart is selected. Populates the Property Table View with all elements that are represented by the selected dot or bar segment.

9 Static Behavior Analysis

Trace Matrix View in combination with Trace Tree View provides excellent capabilities for static analysis of activities, state machines and UML-RT capsules.

Those analysis are based on the default semantics in UML and UML-RT and are independent on which metamodel is applied.

This chapter describes how to perform static behavior analysis of State Machines (UML and UML-RT), Activity Flows (UML) and Capsule compositions (UML-RT) as well as analysis of protocol usage (UML-RT).

IMPORTANT: For correct behavior when using UML-RT (Capsule modeling) in RSARTE or HCL RTist, make sure that the MetaModelAgent preference setting *Enable real-time modeling support* is checked.

9.1 Potential State Machine Transitions

Static analysis of state machine transitions involves finding all potential transitions between states and their entry and exit points. As no simulation or evaluation is made, all transitions found are potential. E.g. they may occur depending on their guards.

Potential transitions between states that are not explicitly modeled are reported as implicit references in the analysis result. For example exiting from the CycleLight composite state in the example below, which is possible from all three nested states.

Implicit references are also used for denote how entry/exit points are connected to states.

The state machine is flatten in the analysis. That means that only the atomic states will be displayed in the analysis result.

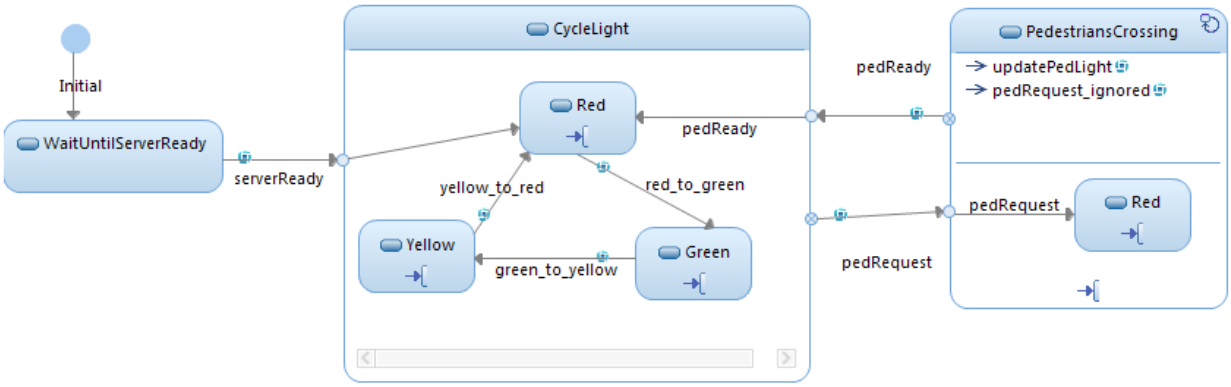

Figure 9: *State machine of Traffic Light capsule used as an example in the state machine transition analysis below.*

9.1.1 Transition Paths to/from a specific State, Entry or Exit point

To analyze all *potential forward transition paths* from a specific atomic state or entry/exit point in the Trace Tree view.

1. Select the atomic state or entry/exit point that should be the starting point in the Explorer View or in the Diagram Editor.
2. Bring up the context menu and select *MetaModelAgent→Show in Trace Tree View→Outgoing References→Potential Transition Paths.*

To analyze all potential transition paths backwards from a specific atomic state or entry/exit point in the Trace Tree View.

1. Select the atomic state or entry/exit point that should be the end point in the Explorer View or in the Diagram Editor.
2. Bring up the context menu and select *MetaModelAgent→Show in Trace Tree View→Incoming References→Potential Transition Paths.*

As the state machine is flatten, composite states containing nested states can´t be the starting point for these analysis.

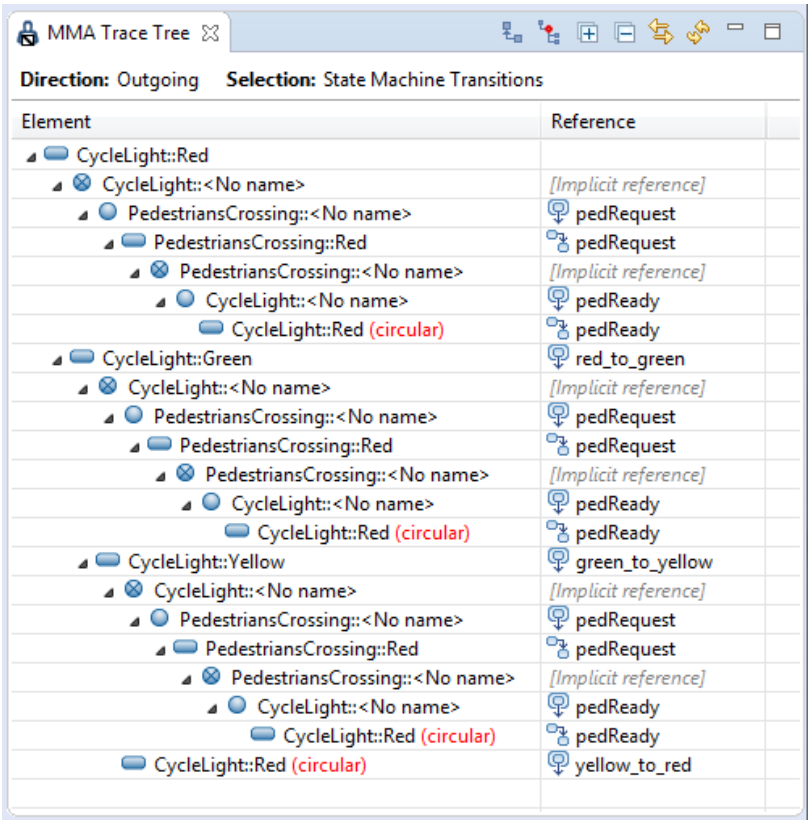

Figure 10: *Potential transition paths from the CycleLight::Red atomic state in the state machine example above.*

Only the state machine that owns the selected state, entry or exit point will form the scope for this analysis. Any state machine that inherits from the state machine will not be considered.

If you need to analyze the transition paths starting from or ending at an element that is reused in an inheriting state machine, please populate the Trace Matrix View with that inheriting state machine and select a state or entry/exit point from the matrix to populate the Trace Tree View. See 9.1.2 below.

9.1.2 All Transitions in a Hierarchical State Machine

All transitions in a state machine and its composite states can be displayed in the *Trace Matrix View.*

3. Select a state machine or a composite state in Explorer View or in the Diagram Editor.
   (In an UML-RT model you could also select a Capsule, as a capsule is supposed to have a companion state machine).
4. Bring up the context menu and open submenu *MetaModelAgent→Show in Trace Matrix View→State Machine Transitions.*

Nested state machines will be flatten, e.g. only the most nested states will be displayed. If a nested state machine contains a history state. This history state will be substituted with all states that potentially could be the next state.

Pseudo states such as entry/exit points, fork, join and junctions are excluded in the resulting matrix. Initial states are always displayed as well as entry/exit states if the selected scope is a composite state.

Transition paths that pass excluded pseudo states are displayed as transitive relations in the matrix cells.

The analysis does not evaluate any guards or other expressions that controls the transitions being traversed. The resulting matrix will therefore show “potential” transitions.

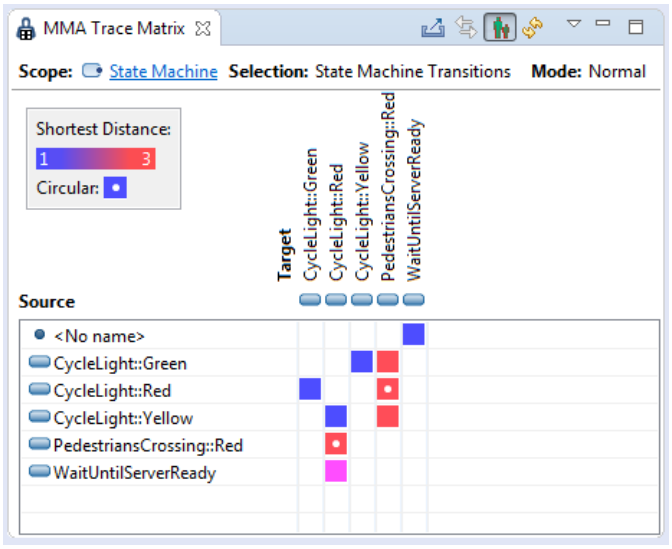

Figure 11: *All potential transitions in a state machine excluding entry/exit points. Transitive relationships in red indicates transition chains where entry/exit points are traversed.*

Switching to Transitive Mode in the view´s drop down menu highlights which states that can be reached from other states, directly or indirectly. The color indicates the number of consecutive transitions (explicit and implicit) from source to target state.

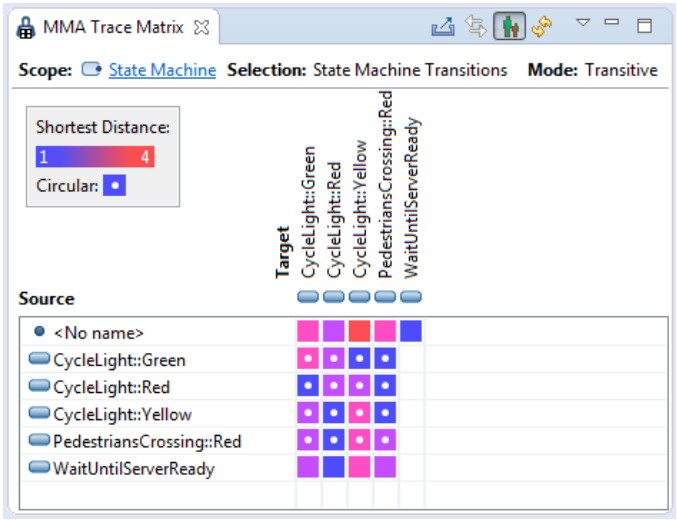

Figure 12: *Turning the matrix view into Transitive Mode to show potential transition paths between all states.*

The chain of transitions making up a transitive relation can be further examined in the Trace Tree View. Select a cell representing a transitive relation, bring up the context menu and select Show all path(s) in Trace Tree View, see next chapter for details about the Trace Tree View.

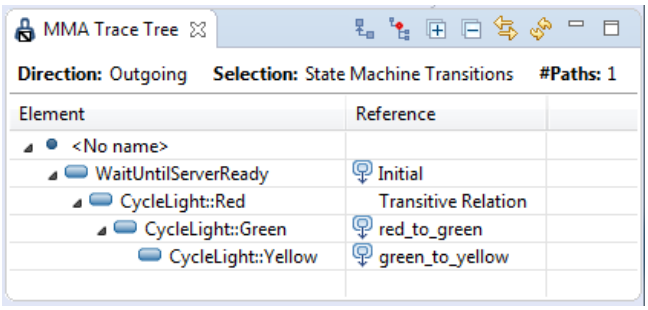

Figure 13: *The potential transition path from the initial state to the Yellow state by selecting the intersection cell (red) in the Transitive Model of the Trace Matrix View.*

9.2 Potential Activity Flows

Static analysis of activities involves finding all potential control and object flows between actions and other activity nodes. As no simulation or evaluation is made, all flows found are potential. E.g. they may occur depending on their guards.

Implicit references are used to denote how pins are connected to actions.

Analysis of an activity will include analysis of any activity referred from call behavior actions.

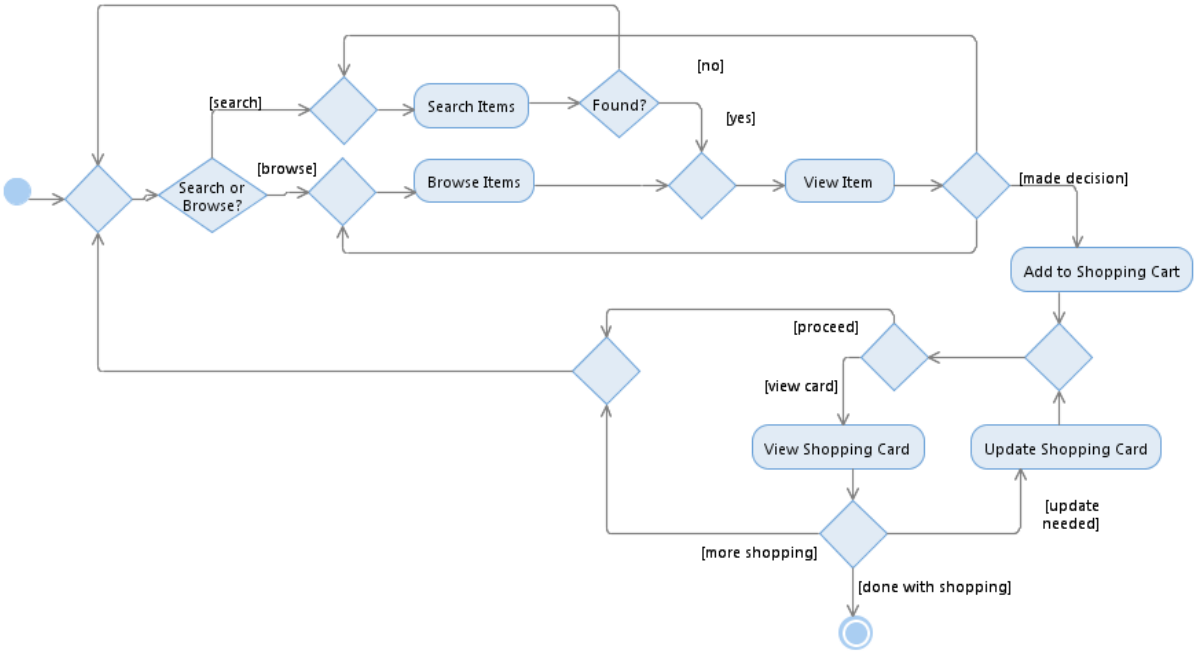

Figure 14: *Activity Flow of an Online Shopping process used as an example in the analysis below.*

9.2.1 Activity Flows to/from a specific Action

To analyze all *potential flow paths* from a specific activity node (e.g. actions, control nodes, object nodes or pins) in the Trace Tree view:

1. Select the activity node that should be the starting point in the Explorer View or in the Diagram Editor.
2. Bring up the context menu and select *MetaModelAgent→Show in Trace Tree View→Outgoing References→Potential Flow Paths.*

To analyze all *potential flow paths* towards a specific activity node in the Trace Tree view:

1. Select the activity node that should be the end point in the Explorer View or in the Diagram Editor.
2. Bring up the context menu and select *MetaModelAgent→Show in Trace Tree View→Incoming References→Potential Flow Paths.*

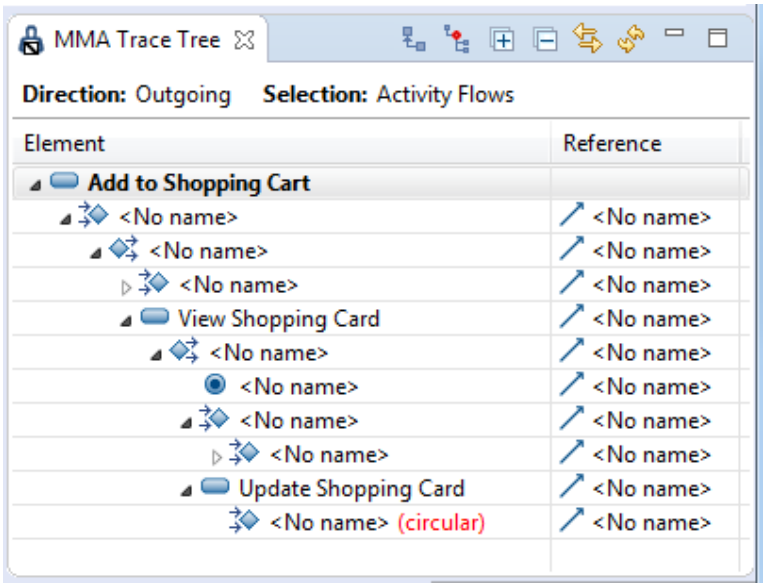

Figure 15: *Potential flow paths from the action Add to Shopping Cart in the example above. The paths are not fully expanded.*

9.2.2 All Activity flows in an Activity and referenced Activities

All Control and Object Flows within an Activity with nested structured activity nodes and referenced activities (through call behavior actions) can be analyzed at different levels of abstraction in Trace Matrix View.

1. Select an Activity in the Explorer View or in the Diagram Editor.
2. Bring up the context menu and open submenu *MetaModelAgent→Show in Trace Matrix View→Activity Flows.*
3. Select the level of details among the submenu items to bring up the *Trace Matrix View.*

The different submenu items represent different level of abstractions and details:

All Nodes	Shows all control and object flows between actions, control and object nodes including input/output pins. The “connection” between input/output pins and their actions are shown as implicit relations in the matrix.

Excluding Control Nodes	Same as above but control nodes, except initial and final node, are excluded.
Excluding Pins	Shows all control and object flows between actions, control and object nodes but not pins.
Excluding Control Nodes and Pins	Shows all control and object flows between actions, object nodes. Control Nodes, except initial and final node, and Pins are excluded.

For all about the “All nodes” selection, a filled cell may represent a flow path that passes one or several control nodes and/or pins that have been excluded. The color of the cell indicates the shortest path between the source and target action. The details of such a flow path can be examined in the Trace Tree View using the cell´s context menu.

The analysis does not evaluate any conditions or other expressions that controls the flows being traversed. The resulting matrix will therefore show *potential* flows.

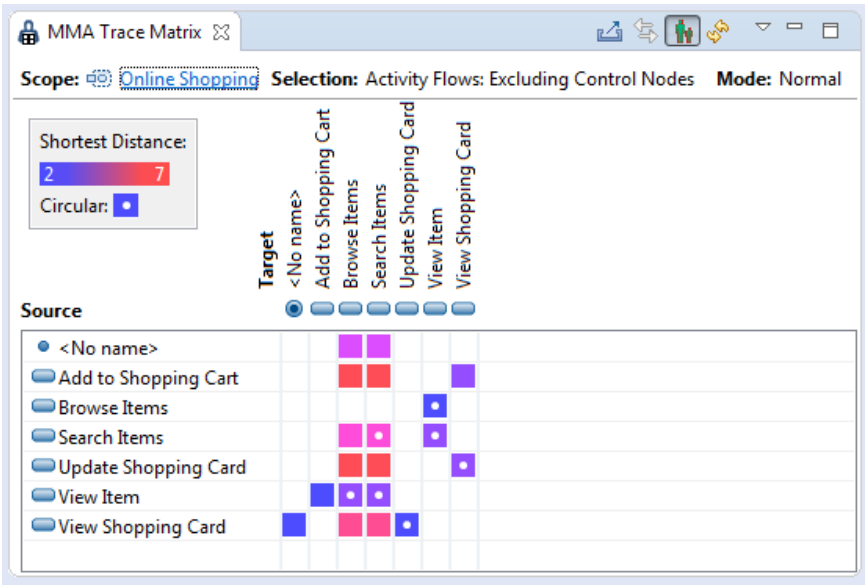

Figure 16: *Potential flows between actions in the Online Shopping process above where control nodes have been excluded.*

As with State Machine Transitions in Trace Matrix View, the Transitive mode can be used for Activity Flows to get an overview of which actions that can be reached from another action.

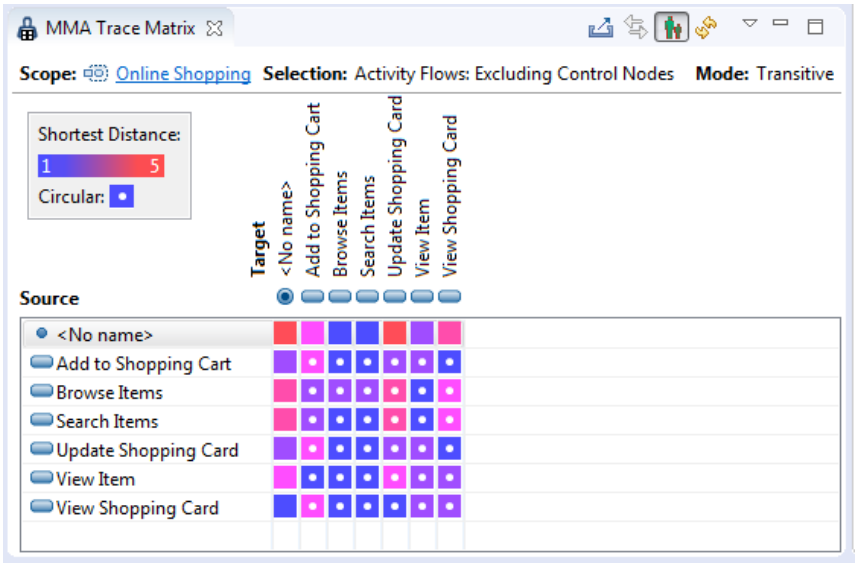

Figure 17: *Distance between all pair of actions in terms of intermediate actions.*

By bringing up a cell's context menu in Transitive Model, you can populate the Trace Tree View with all potential paths between the source and target node.

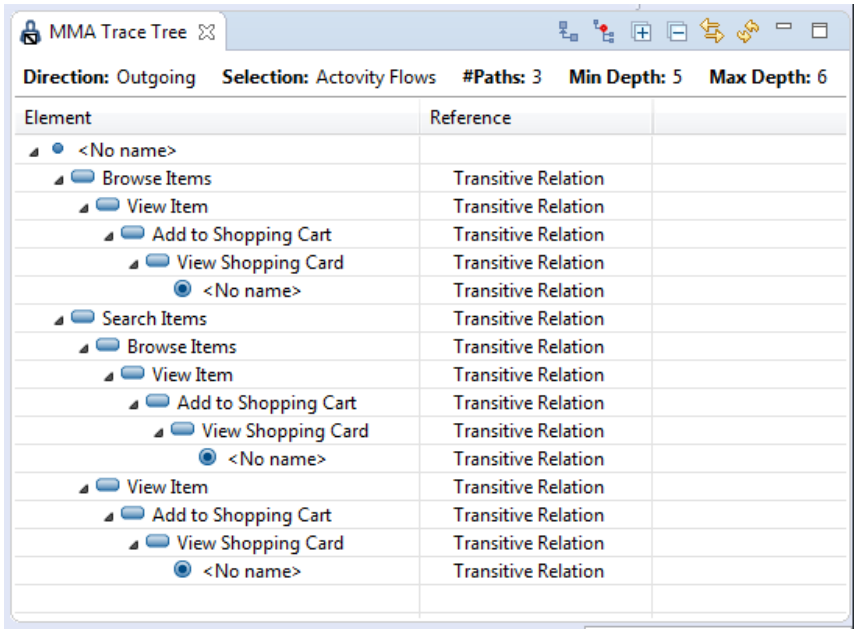

Figure 18: *All potential paths from initial node to activity final node in the Activity Flow above.*

IMPORTANT: Activity partitions will not be considered. All variants of Structure Activity Nodes (Expansion Region, Conditional Node, Loop Node and Sequence Node) will be reported as Structure Activity Nodes. The potential loop within a Loop Node will there for not be shown.

9.3 Capsule Connectors

Capsule is a concept within UML for Real-time (UML-RT) supported in RSARTE and HCL RTist.

Connectors within a Capsule and its composited Capsule Parts' capsules can be analyzed at different levels of abstraction in Trace Tree View and in Trace Matrix View.

Dynamic Connections for non-wired ports will however not be included as they are impossible to analyze without interpretation/simulation.

As no simulation or evaluation is made the following constraints holds:

- All connections are said to be potential. E.g., they may occur depending on the state machine behavior in the involved capsules.
- Dynamic Connections for non-wired ports will not be included.

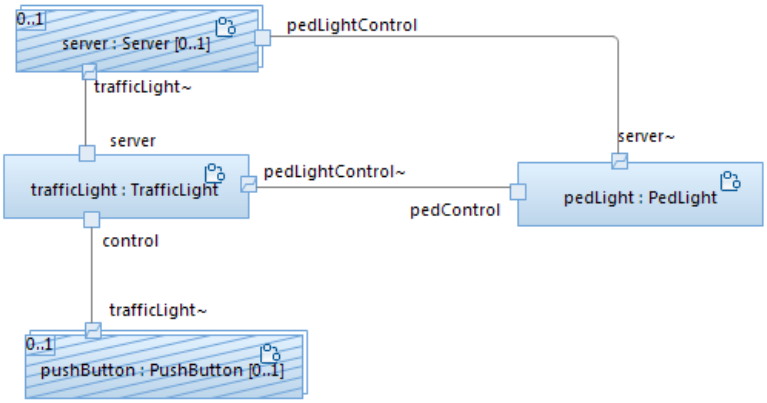

Figure 19: *Composite Structure of Capsule TLSystem capsule used as the example in the analysis below.*

9.3.1 Potential Connector paths to/from a Port in a Capsule Part

To analyze all potential connection paths starting from a specific port:

1. Select the starting port in a composite structure diagram
2. Bring up the context menu and select *MetaModelAgent→Show in Trace Tree View→Outgoing References→Potential Port Connection Paths*.

To analyze all potential connection paths towards a specific port:

3. Select the final port in a composite structure diagram
4. Bring up the context menu and select *MetaModelAgent→Show in Trace Tree View→Incoming References→Potential Port Connection Paths*

Both these operations will display a path of potential connected ports in the Trace Tree View.

Alternatively, you can select a staring or end capsule part and display the connection path on a capsule part level instead, leaving out the involved ports.

The context menu does also provide option to only display explicit connection paths.

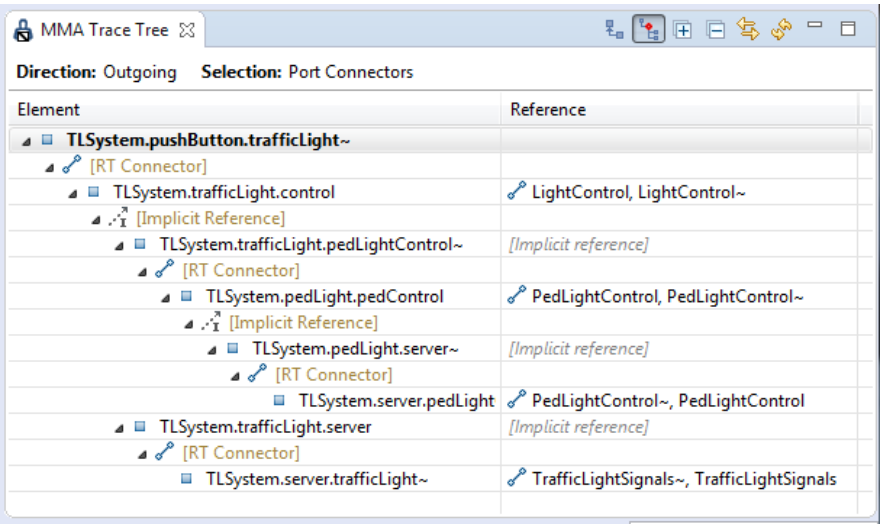

Figure 20: *Trace Tree View of potential communication paths between ports starting from the traffic light port in the capsule part pushButton.*

9.3.2 All Connectors in a Capsule Composition Structure

Invocation

1. Select a Capsule in the Explorer View or in the Diagram Editor.
2. Bring up the context menu and open the submenu *MetaModelAgent→Show in Trace Matrix View→Capsule Connectors.*
3. Select the level of abstraction among the submenu items to bring up the *Trace Matrix View.*

The different levels of abstraction available in the context submenu are:

All Explicitly Connected Ports	Shows all explicit connectors between ports within the selected Capsule and all its composited capsules.
All Potential Connected Ports	Same as above but also including potential implicit connections between behavior ports handled by a capsule's state machine.
All Potential Connected Parts	Shows all connectors between capsule parts, within the selected Capsule and all its composited capsules, including potential implicit connections handled by a capsule's state machine.

IMPORTANT: For correct behavior, make sure that the MetaModelAgent preference setting *Enable real-time modeling support* is checked.

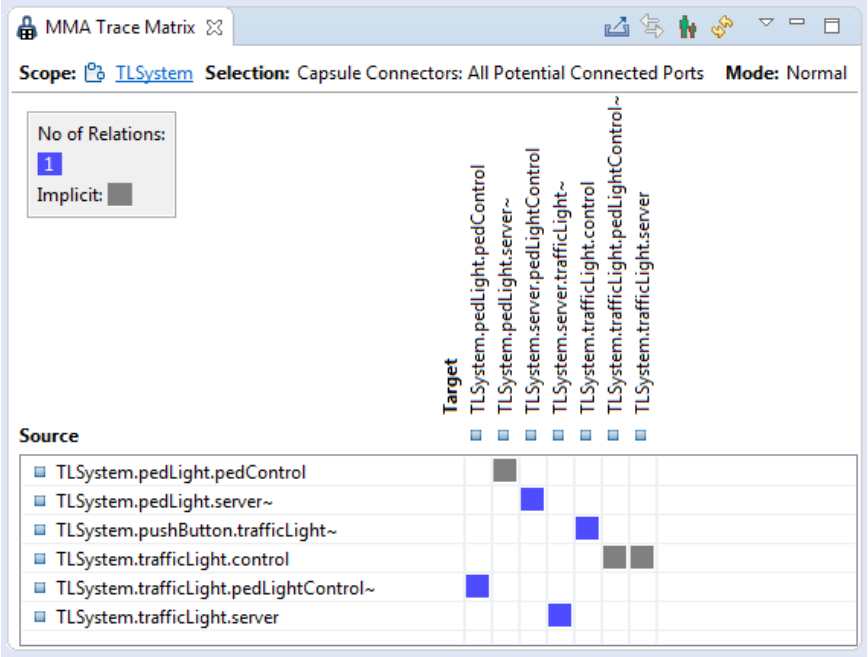

Figure 21: *Trace Matrix View of all potential connected ports. The grey cells indicate potential non-explicit connectors between behavior ports handled by the capsule's state machines.*

To further analyze the potential communication ways between ports, switch to Transitive Mode using the view's menu and to show all ports that potentially can communicate with each other.

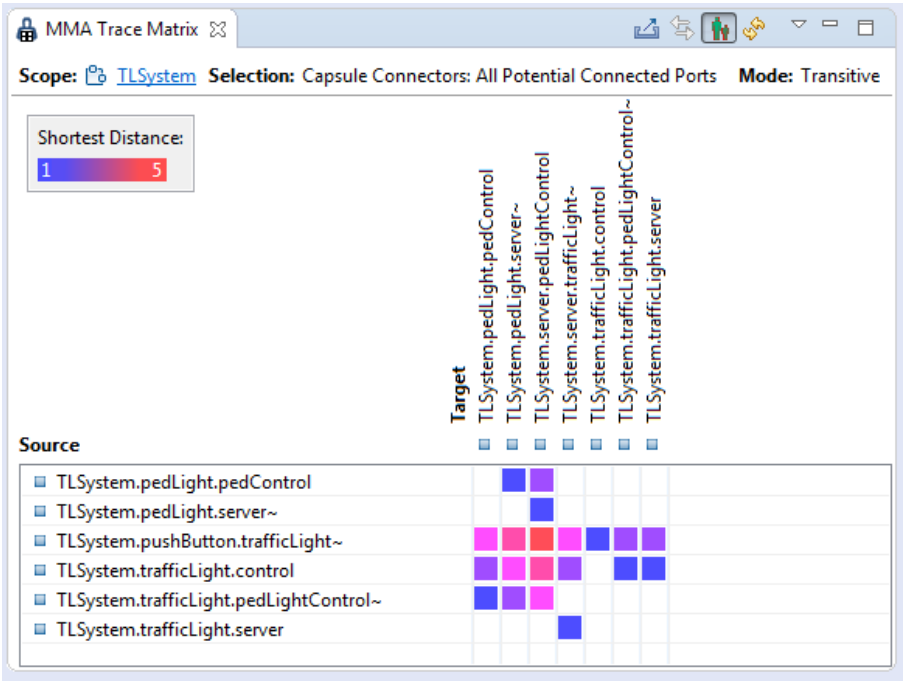

Figure 22: *The Trace Matrix View in transitive mode showing which ports could be connected directly or indirectly using explicit connectors and potentially through behavior ports according to the capsule´s state machines behavior.*

Select a filled cell representing potential communication path(s) and explore the path(s) further by invoking the Trace Tree View using the cell's context menu.

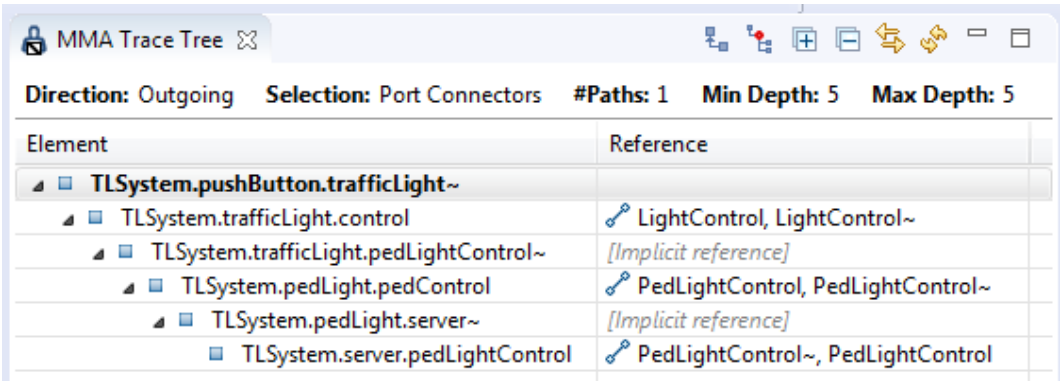

Figure 23: *The potential communication paths between two of the ports. Implicit reference represents potential communication dependent on state machine behavior.*

IMPORTANT: When analyzing capsule structures, the label of a port will consist of the path of parts starting from the initial capsule. If the composite structure of capsules are very deep, the corresponding path to the port will also be deep. In the Trace Matrix View, this can lead to the labels being truncated. Enlarge the view to minimize the problem with truncated labels.

9.4 Capsule Connectors for a specific Protocol

The Matrix view can also be filtered to only display connectors that connects ports typed by a specific protocol. This is useful for analyzing usage of a specific protocol within the composite structure of a capsule.

Invocation

1. Select a connector within a Capsule in the Explorer View or in the Diagram Editor.
2. Bring up the context menu and open the submenu *MetaModelAgent→Show in Trace Matrix View→Protocol <Protocol name> Connectors.*

The protocol usage analyzed will be the protocol used by the ports connected by the connector. If the two ports of the connector are typed by different protocols, the most specific of those protocols are used.

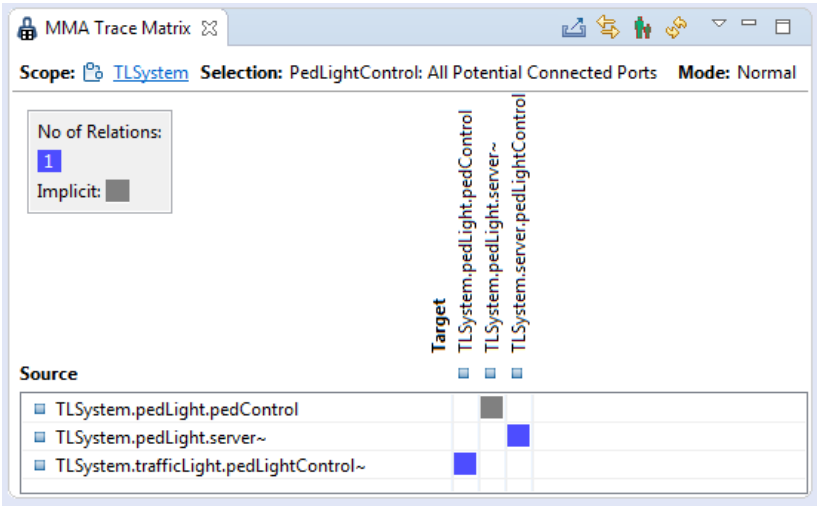

Figure 24: *Trace Matrix View of all potential connected ports using the protocol PedLightControl within the capsule TLSystems composite structure.*

Switch to Transitive (Start-End) Mode to see all complete paths of the specific protocol in the capsule structure.

Select a filled cell and select “Show all path(s) in Trace Tree View to display a specific complete connector path using that protocol.

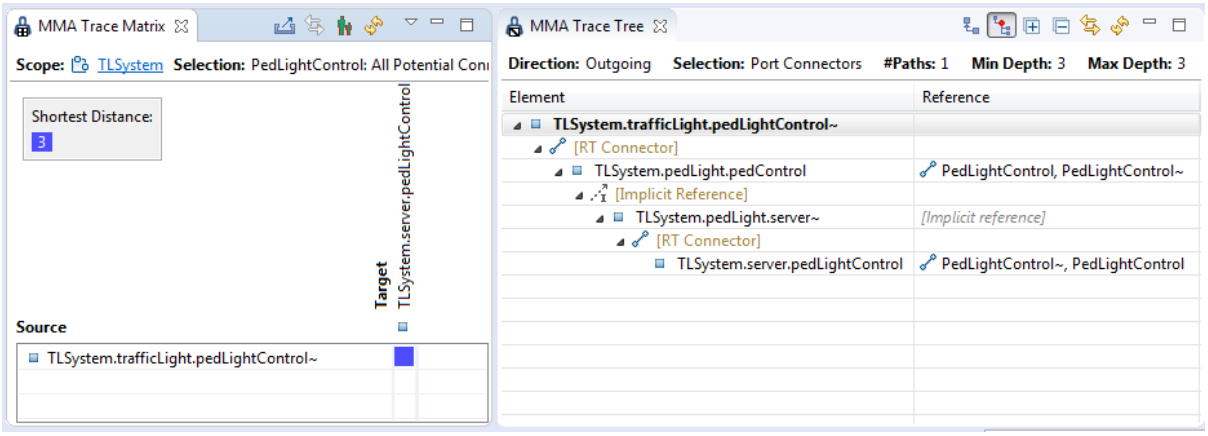

Figure 25: *Trace Matrix View of the only complete path of the protocol PedLightControl and the details of the same path expanded in the Trace Tree View.*

9.5 Overall Protocol Usage

All usage of a specific protocol can also be displayed in the Trace Matrix View

1. Select the protocol in Project Explorer or in the Diagram Editor.
2. Select *MetaModelAgent→Show in Trace Matrix View→All Connectors using Protocol* to bring up the *Trace Matrix View*.

The Matrix View will display all connectors in all activated models that are using the selected protocol or inherited protocols.

IMPORTANT: This gives an overview of the protocol usage, but there is no idea to turn to Transitive Mode for analyzing the potential communication paths. Instead, you should use analysis described in chapter 0 if you would like to analyze the potential communication path for a specific protocol.

Appendix A: Referring elements

The Trace Matrix View, see chapter 6, can besides relations and other connections display element references for several kinds of UML elements which refer to other elements,

The list below shows which kinds of element references (from UML metamodel) that can be tracked by selecting corresponding metaclasses in the *Show in Trace Matrix View* submenu in the Explorer view or Activation View. The owner of the referring element is always regarded as the source element in the resulting matrix

Referring Element type	**Referring element property**	**Referred element (target)**
Activity Partition	represents	Element
Call Behavior Action	behavior	Behavior
Call Operation Action	operation	Operation
Central Buffer	type	Type
Connector End	defining end	Property
Data Store Node	type	Type
Interaction Use	interaction	Interaction
Lifeline	represents	Property
Parameter	type	Type
Port	type	Type
Property	type	Type

For example, if you have a model activated using the built-in metamodel General UML Guidelines and contains at least one Activity with a Call Operation Action.

- Bring up the context menu and select MetaModelAgent->Show in a Trace Matrix View-> Call Behavior Action
- Each filled cell in the matrix view will represent a Call Behavior Action element. Its source element will be the owner of the Call Behavior Action and the target will be the value of its behavior property (e.g., a UML Behavior element).