

Henshin-Editor

Visual Development Environment for Complex EMF Transformations

User Manual

Enrico Biermann, Claudia Ermel, Johann Schmidt, Angeline Warning

Version 0.5
TU Berlin, January 2011

Preface

Henshin (Japanese for "transformation") is a development environment supporting the visual modeling and execution of rule-based EMF transformations. The Henshin Editor is based on the Henshin subproject of Eclipse EMFT (www.eclipse.org/modeling/emft/) that integrates the Henshin modeling language as Eclipse plug-in (www.eclipse.org/modeling/emft/henshin/).

The Henshin Editor was developed in a student project (visual languages and tools) at Technische Universität Berlin in 2010, and extended in the bachelor thesis of Johann Schmidt [Sch10] and the master thesis of Angeline Warning [War10], both supervised by Enrico Biermann and Claudia Ermel.

Henshin supports model transformation "in-place". This means that a transformation does not generate a new model but changes the instance model directly. In Henshin, we define EMF transformation systems that have two basic kinds of elements: *graphs* (EMF instance models) and *transformation units*. The simplest transformation unit is a *transformation rule* consisting of a precondition graph (left-hand side LHS) and a post-condition graph (right-hand side RHS). A transformation rule is applied to an EMF instance model by finding the pre-condition graph in the model and replacing it by the post-condition graph. Rule application may be restricted by defining additional *application conditions* for a rule. More complex transformation units like sequences, IF-THEN-ELSE blocks or counted units are composed from simpler transformation units. A transformation unit is applied by applying its subunits in the predefined order automatically. Using transformation units, complex transformations containing any number of rule applications can be performed.

This user manual explains the available means to define *EMF transformation systems*, i.e. graphs, transformation rules, application conditions and transformation units for complex EMF transformations using the Henshin-Editor. Moreover, we explain how transformation rules and units are applied to transform EMF instance models typed over arbitrary EMF models and how EMF graphs are checked for EMF consistency (objects are contained in exactly one container, and there are no containment cycles). Rules can be validated to ensure that their application will never destroy EMF consistency of consistent EMF instance models.

Table of Contents

| 0 Installation | 1 |
|--|----|
| 1 Create a Project Folder | 2 |
| 2 Create a Transformation System | 3 |
| 3 Import EMF Models | 5 |
| 4 Graph Editor | 7 |
| 4.1 Nodes | 8 |
| 4.2 Edges | 9 |
| 4.3 Attributes | 11 |
| 5 Transformation Rule Editor | 14 |
| 5.1 Mappings | 15 |
| 5.2 Parameters | 16 |
| 5.3 Attribute Conditions | 19 |
| 6 Application Conditions for Rules | 20 |
| 6.1 Create Condition Tree | 22 |
| 6.2 Create Application Conditions Stepwise | 25 |
| 6.3 Negate Application Conditions | 26 |
| 6.4 Switch between AND and OR | 26 |
| 6.5 Formula-Mappings | 26 |
| 6.6 Nested Application Conditions | 27 |
| 7 Transformation Units | 28 |
| 7.1 Create Transformation Units | 29 |

| | 7.2 Parameters | 30 |
|----|---|----|
| | 7.3 Open Sub-Unit | 31 |
| | 7.4 Mappings between Parameters | 32 |
| | 7.5 Changing the Order of Sub-Units | 32 |
| | 7.6 Create Transformation Unit with Content | 33 |
| | 7.7 Change Counter in Counted Units | 35 |
| | 7.8 Amalgamation-Units | 35 |
| 8 | Deletion of Elements | 38 |
| 9 | Application of Transformation Rules and Units | 39 |
| 10 | O Checking EMF Consistency of Graphs and Transformation Rules | 41 |
| | | |

0. Installation

The following software is required for installing the Henshin Editor:

- Java Version 1.6 (e.g. at http://www.oracle.com)
- Eclipse Version 3.6 ("Eclipse Classic" at http://www.eclipse.org)

• EMF and GEF:

In Eclipse, select $Help \rightarrow Install \ New \ Software \ and \ choose \ the \ Helios \ page.$ Under item Modeling, select EMF Eclipse Modeling $Framework \ SDK$ and $Graphical \ Editing$ Framework GEF SDK $Version \ 3.6.0$ (Alternatively, GEF may be installed from the update site http://download.eclipse.org/tools/gef/updates/releases/).

Henshin SDK

Install the latest Henshin SDK version from http://www.eclipse.org/modeling/emft/henshin/downloads.php

• Henshin Editor

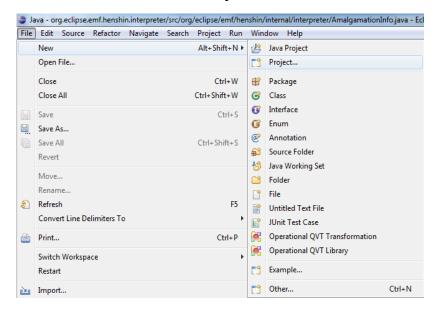
Install the Henshin Editor from the TFS software update site at http://tfs.cs.tu-berlin.de/software

Now, EMF transformation systems can be defined and applied in the Eclipse Runtime Workbench.

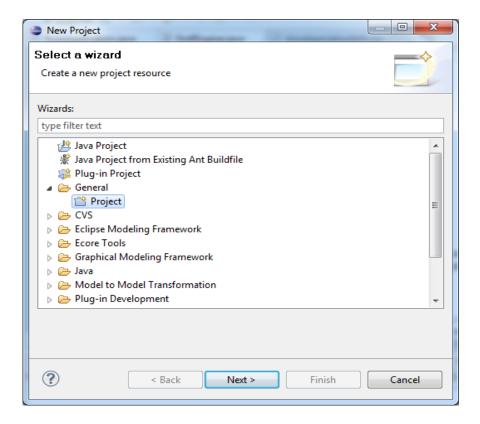
1. Create a Project Folder

For creating a transformation system, a new project folder is needed in the Eclipse-development environment.

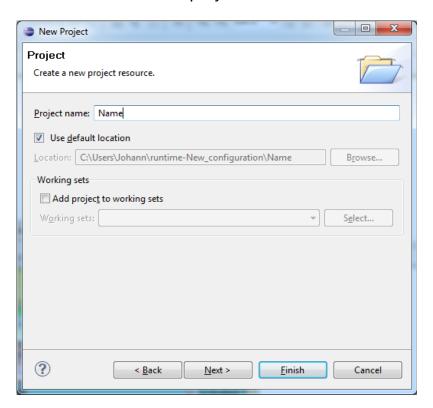
Click on $File \rightarrow New \rightarrow Project...$



Open Folder General and select Project. Click on Next.

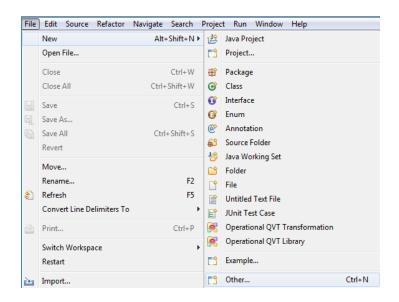


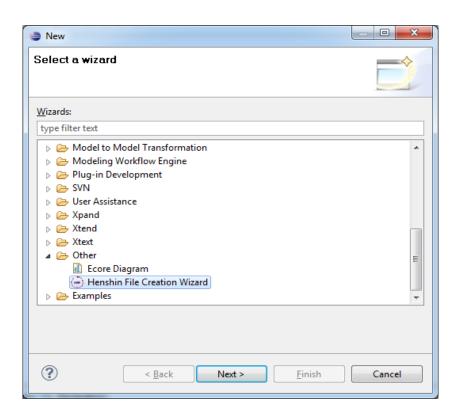
Define a name for the project. Click on Finish.



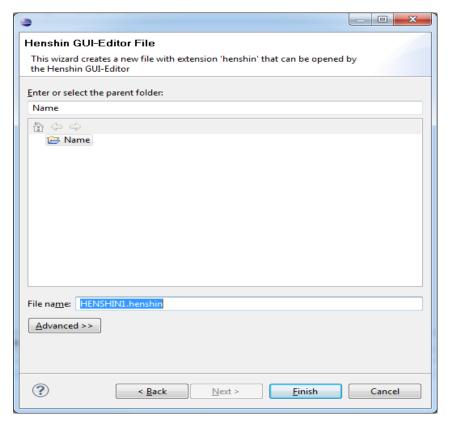
2. Create a Transformation System

Having created a project folder, a transformation system may be defined. Click on the project folder with the right mouse button. Select $New \rightarrow Other$. Open folder *Other* and select *Henshin File Creation Wizard*. Click on *Next*.

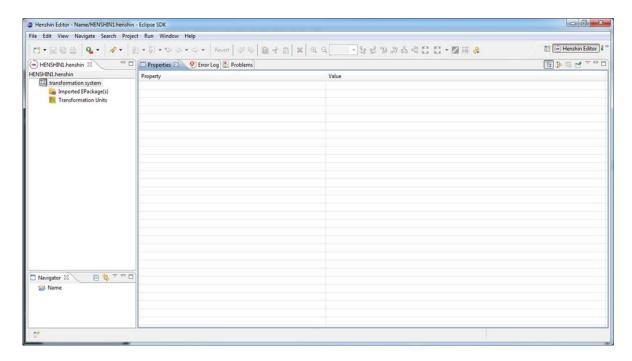




Define a name for the transformation system (or keep the pre-defined name). Clicking on *Finish*, the (empty) transformation system is created.



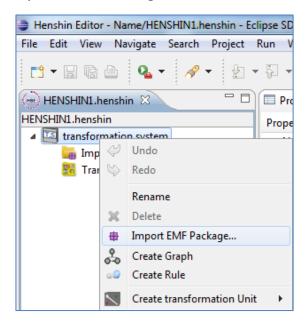
Now you can see the *Henshin Editor*-perspective:



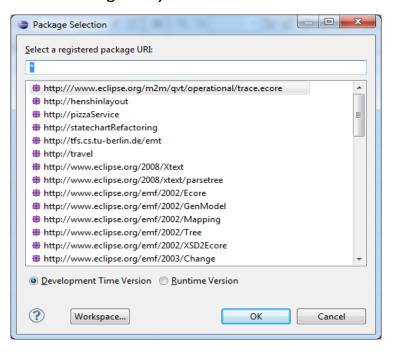
3. Import EMF-Models



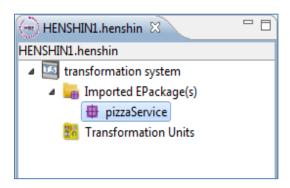
Having opened a transformation system, you may import one or more EMF models. Click with the right mouse button on *transformation system* and select *Import EMF Package*.



A list of all available EMF models in the current workspace is shown. Select an EMF model from the list or import an Ecore file by clicking on *Workspace...* and then selecting *File System*.



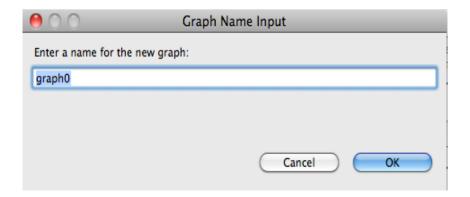
Example: We select the EMF model pizzaService and confirm with OK. Imported EMF models are shown in the tree view as ImportedEPackage(s).



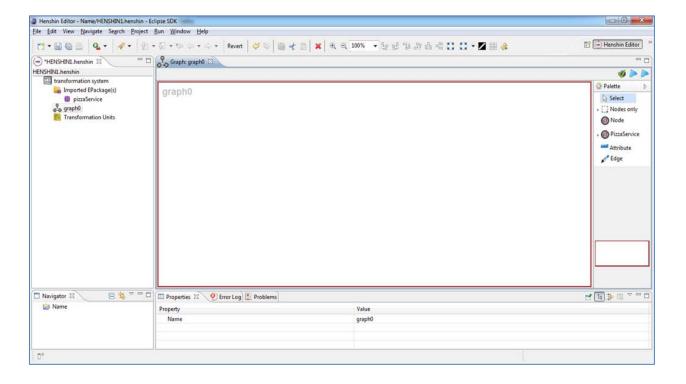
4. Graph Editor



To create a graph, select item *Create Graph* from the context menu of *transformation system*. Define a name for the graph or keep the pre-defined name and confirm with OK.



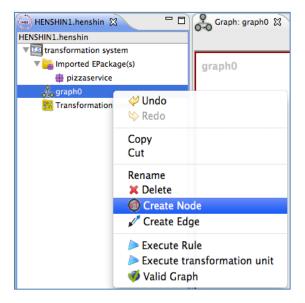
Names for graphs within the same transformation system must be unique. The Henshin-Editor ensures this by not accepting names that do exist already. After confirming the dialog, the graphical view for the new (empty) graph:

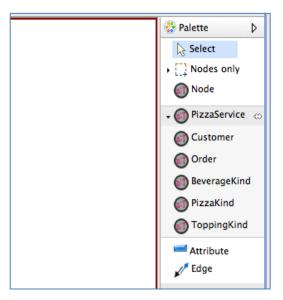


4.1. Nodes

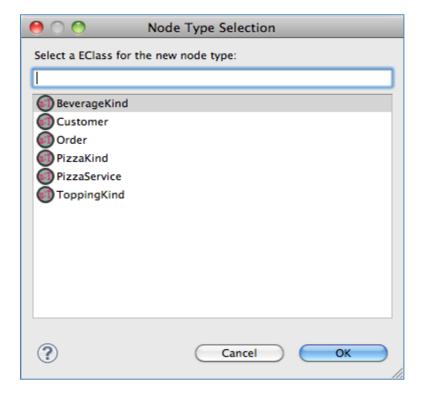


New nodes may be created in the tree view (left) or using the palette of the visual graph view (right):



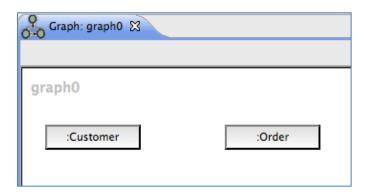


To create a node using the context menu, select item *Create Node*. A dialog appears to define the node's type (its EClass).



Using the palette, the node's type can be defined directly. Click on the node type in the palette and then into the graph editor panel at the desired node position. Alternatively, you may click on *Node* in the palette. In this case, the node type definition dialog appears after clicking into the graph editor panel at the node position.

Example: We create two nodes, one of type Customer and the other one of type Order



4.2. Edges

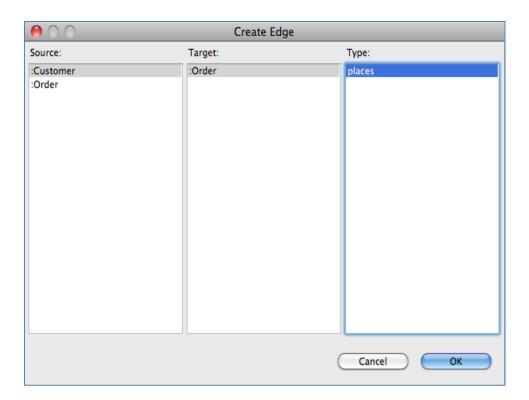


The creation of edges works similarly to the creation of nodes: either you select the item *Create Edge* of a graph's context menu, or you use the palette entry *Edge* in the graph editor palette.

When creating an edge with the context menu, the necessary data have to be inserted in a dialog. After selecting a *Source*-node for the new edge, all possible *Target*-nodes appear in the target node selection list. Having selected a *Target*-node, all available edge types are offered in a *Type* selection list. Only after *Source*-node, *Target*-node and *Type* have been defined for the edge, the dialog may be closed by confirming the selections with *OK*.

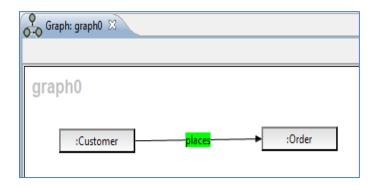
Example: There are two nodes in the graph, one of type Customer, and one of type Order. Select the item Create Edge of the graph's context menu. Select :Customer in the edge creation dialog. Node :Order appears in the target node

selection column and is selected automatically (being the only possible choice). Similarly, there is only one possible edge type "places" which is selected automatically.



It is much easier to create edges using the graphical editor view. Here, you click on the *Edge* item in the palette. Then you only have to click first on the source node and then of the target node in the editor panel. All possible edge types and their multiplicities are checked w.r.t. the EMF model. If more than one edge type exists between the types of the selected source and target nodes, a selection dialog is shown. Otherwise, the edge with the only possible type is created and drawn in the editor panel.

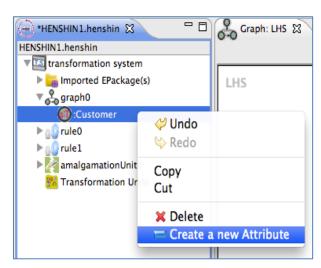
Example (without selection dialog): Draw an edge in the graphical editor panel. Click on item *Edge* in the palette. Then click first on node *Customer* and afterwards on node *Order*. Since there is only one edge type between the types *Customer* and *Order*, the correctly typed edge is drawn automatically.

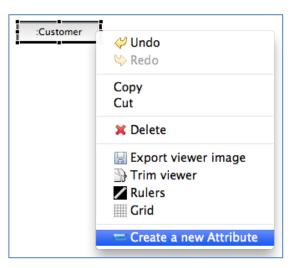


4.3. Attributes

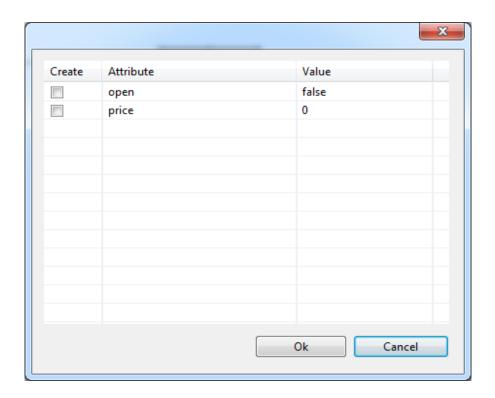


Nodes may have attributes. Attributes may be created selecting the item *Create* a new Attribute of the node's context menu in the tree view (left) or in the graphical view or in the palette (right):

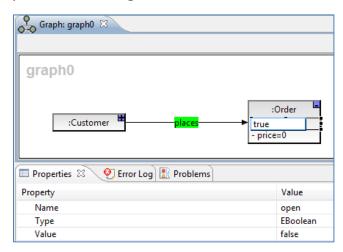




Clicking at first on the palette entry *Attribute* and afterwards on a node, a selection list of all possible attributes is offered in a dialog, together with default values. Tick those attributes you want to create and (optionally) change their values. Confirm your selection with *Ok*.



If a node has attributes, then this is indicated by a "+"-symbol in the upper right corner of the node figure in the graphical editor. Click on the "+"-symbol to expand the node view and show its attributes. Click on the "-"-symbol in an expanded node figure to hide the attributes.



There are two possibilities to change attribute values:

1. *Properties*-view

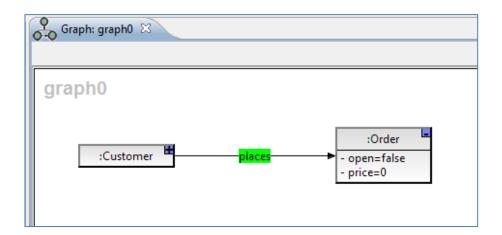
Select the node you want to change attribute values in the graphical or tree

view. In the view *Properties* you see all properties of the node. In the column *Value* the attribute values may be changed.

2. Graphical view

Double-click on the attribute name you want to change the value of. The attribute value now can be edited directly in the graphical editor panel.

Example: Change the value of attribute open from false to true.



5. Transformation Rule Editor



To create a new transformation rule, select item *Create Rule* in the context menu of *transformation system*. Define a name for the rule or keep the default name and confirm with *OK*.



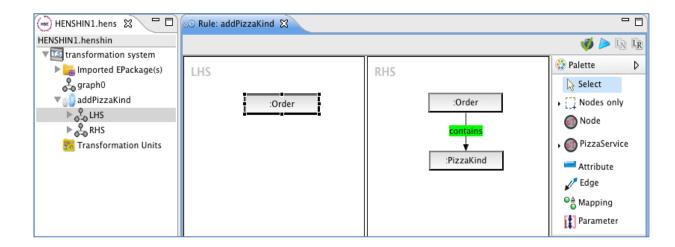
Rule names within the same transformation system must be unique. The Henshin-Editor ensures this by forbidding the definition of names that are already in use. After confirming the dialog, the rule view of the new rule is opened. A rule consists of two graphs *LHS* and *RHS*. It is possible to copy an edited graph *LHS* into *RHS* by clicking on symbol in the tool bar All elements from *LHS* are mapped identically to their copies in *RHS*.

Example: Define a rule addPizzaKind, that creates a new node PizzaKind within an existing Order.



Proceed as follows:

- 1. Create a node of type *Order* in *LHS*.
- 2. Create a node of type *Order* and one of type *PizzaKind* in *RHS*.
- 3. Create an edge from node *Order* to node *PizzaKind* in RHS.



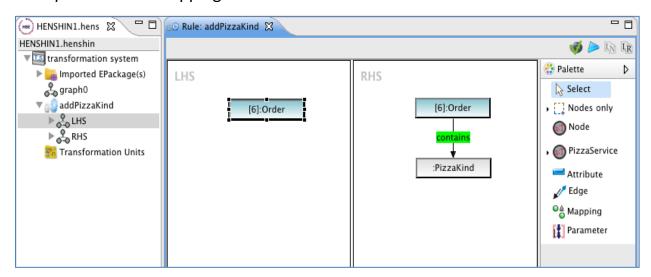
5.1. Mappings



If a node in *LHS* and a node of the same type in *RHS* represent the same node in the graph (i.e. the node is in the intersection of *LHS* and *RHS*), then you need to create a *Mapping* to map this node from *LHS* to *RHS*.

You create a mapping by selecting *Mapping* from the rule editor palette, clicking first on the source node in *LHS* and then on the corresponding target node in *RHS*. Mapped node pairs are visualized by the same color and the same number in front of their names.

Example: Create a mapping from node Order in LHS to node Order in RHS.

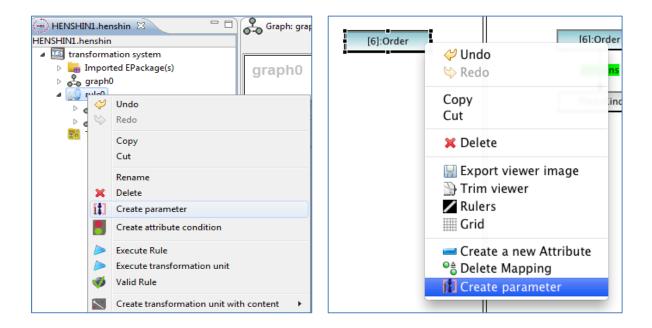


5.2. Parameters

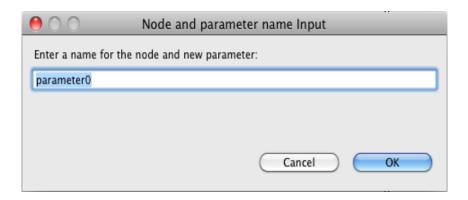


A Parameter can stand for a node or an attribute value. If a parameter stands for a node, it must have the same name as the node. Such parameters can be created by selecting item *Create Parameter* in the context menu of a node in the tree view or rule view, or by clicking on the entry *Parameter* in the rule palette.

Attention: The Option Create Parameter appears only in the context menu of a node in the rule view, not in the graph view.

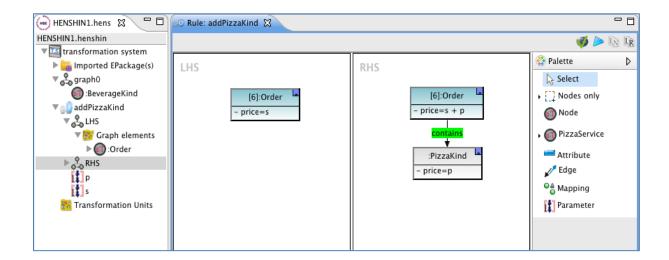


If the selected node has not yet a name, the user is asked to define a name which is then used for the node and the parameter.



Alternatively, parameters may defined using the context menu of the rule in the tree view. Again, the name for the new parameter is asked from the user.

Example: We create two parameters *s* and *p* using the context menu.



Parameters may be used as attribute values. In *LHS*, a parameter may be used for only one attribute. The parameter value may be defined by the user or, automatically, when the rule is applied. In *RHS*, an attribute value may be an arbitrary Java expression.

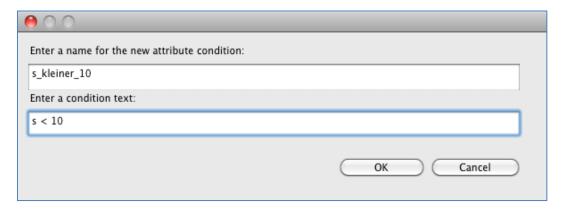
Example: In LHS, node Order has an attribute price with value s (a parameter). When the rule is applied, the parameter value is set to the value of the attribute of the node in the graph node Order is matched to. Node PizzaKind has an attribute price with value p. The parameter value of p is defined by the user and is set as price for the newly created PizzaKind. Node Order in RHS has an attribute price with the value s+p. When the rule is applied, the current price of Order is computed depending on the user-defined value of parameter p.





An attribute condition may be created by selecting item *Create attribute condition* in the rule's context menu in the tree view. The name and the condition (a boolean expression) have to be defined by the user in a dialog. Upon confirmation of the dialog, a new attribute condition is created.

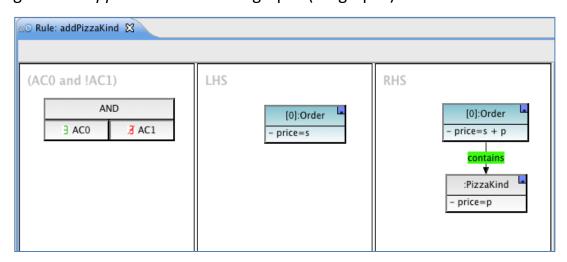
Example: Create the attribute condition s < 10 for the rule AddPizzaKind. Now the rule may be applied only if the value of parameter s is less than or equal to 10.



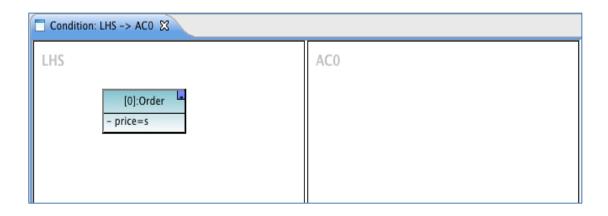
To change the name of an attribute condition or the condition itself, use the *Properties*-view. The condition may be edited also in the tree view, either by double-clicking on it or by clicking on it with the right mouse button and then selecting menu item *Rename*.

6. Application Conditions for Rules

For existing rules, application conditions can be defined. An application condition can be created by using either the context menu of a rule or the context menu of an *LHS* graph. If the rule has no application condition yet, the condition is added to the rule's *LHS* graph. If there is already an application condition, the new one and the existing one are automatically combined with AND. An application condition over *LHS* is shown left to the LHS in the rule view, both in textual form as formula and as block diagram nesting the logical expressions. The innermost elements of the block diagram are *Application Condition* graphs (AC graphs).



To create a condition using the context menu of a selected AC graph means that the new condition is a condition over the selected AC graph. A double click on *Application Condition* in the tree view or in its panel in the rule view opens the condition view. On its left-hand side, it shows the premise (e.g. the rule's LHS), on its right-hand side it shows the conclusion (the AC graph).



The following menu items are available for the creation of application conditions:

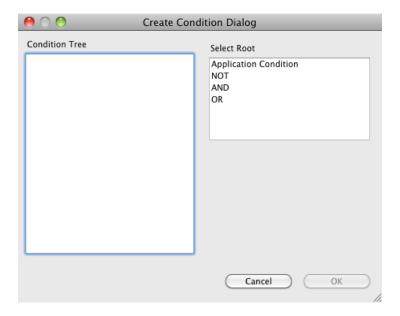
- Create Condition Tree...
- Create Application-Condition
- Create Not-Condition
- Create And-Condition
- Create Or-Condition

These menu items are visible only if it is possible to add a condition to the selected object (e.g. an LHS graph or a condition graph).

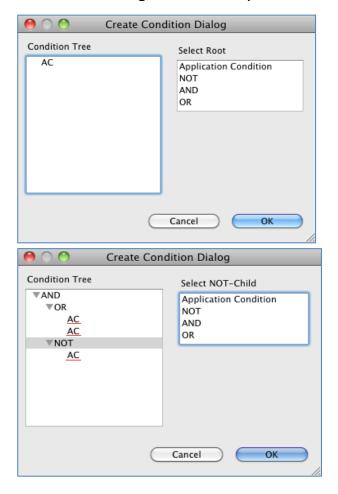
6.1. Create Condition Tree

An application condition may be created as a whole by selecting item *Create Condition Tree...* from the context menu. A dialog supports the creation of a well-structured condition tree (or formula).

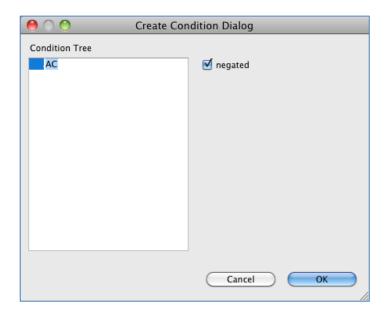
On the left-hand side of the dialog, the condition is shown as tree. On the right-hand side of the dialog, an *Application Condition*, a *NOT-*, *AND-* or *OR-Formula* may be inserted into the tree.



• In the beginning, the *Condition Tree* is empty and the *OK*-button is deactivated. This button will be activated only if the *Condition Tree* is syntactically correct, i.e. either an AC-graph (*Application Condition*) has been selected as root, or all leaves of the tree are AC-graphs. On the right-hand side of the dialog, the root may be selected.



- Pressing F2 you may rename an AC graph in the ConditionTree. To confirm this, press OK.
- If an *Application Condition* is selected in the tree view of the dialog, you can see a *negated*-box in the right-hand side, which serves to switch from a PAC (Positive Application Condition) to a *NAC* (*Negative Application Condition*). If the *negated*-box is not ticked, the application condition corresponds to a *PAC*.



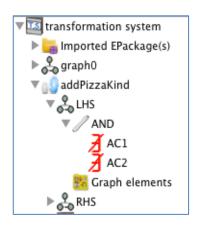
- To add a child node to a *NOT-, AND* or *OR-*Formula, select the parent node in the tree view of the dialog. On the right-hand side you see a list for *NOT-Child* or two lists for *AND-* or *OR-Children,* respectively. Choose one object from each list to define it as child node of the *NOT-, AND* or *OR-*Formula.
- If you want to define the *Condition Tree* anew starting from the root, the remove the tree in the dialog's tree view. On the right-hand side, you can start again by selecting the root element.

Example: Create the application condition AND (AC1, AC2).

Proceed as follows when the dialog is shown:

- 1. Select menu item *Create Condition Tree...* in the context menu of either a rule or an LHS graph.
- 2. Select *AND* as root in the right-hand side of the dialog. The *AND* node is shown in the tree view of the dialog as root node.
- 3. Select the *AND* node in the tree view and select *Application Condition* in both lists on the right-hand side. Two *ACs* are added to as child nodes to the *AND* node in the *Condition Tree*.
- 4. Select the first AC (the second AC), press F2 und rename them to AC1 (to AC2).

5. Confirming with *OK* closes the dialog. In the tree view of the Henshin Editor you find the new condition tree under the *LHS* entry of the corresponding rule.



6.2. Create Application Conditions Stepwise

To create an application condition stepwise, select the desired condition type as corresponding menu item in the context menu. The new condition type (AC / NOT / AND / OR) is added in the tree view to the selected element. Incomplete conditions are shown in red, e.g. NOT-Formulas without child node or AND- or OR-Formulas with less than two child nodes.

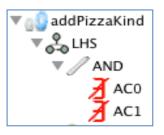
Condition panel of the rule

Example: Create an application condition AND (ACO, AC1).

view.



Select a rule or an LHS in the tree view and select menu item Create AND-Condition in the context menu.



Select the menu item *Create Application Condition* twice in the context menu of *AND*. Renaming the *Application Condition* can be done either in the tree / /*Properties*-view or in the



6.3. Negate Application Conditions

The negation value of an existing *Application Condition* may be changed via its context menu. Selecting the menu item Set negated = false, a *Negative Application Condition* ($\stackrel{>}{\longrightarrow}$) is switched to a *Positive Application Condition* ($\stackrel{>}{\longrightarrow}$). Vice versa, selectin the menu item *Set negated = true*, switches a PAC to a NAC. Alternatively, the negation value can be changed using the *Properties*-view of an *Application Condition*.

6.4. Switching between AND and OR ■

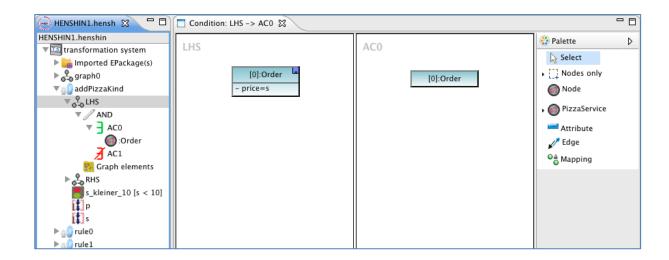
An AND-Formula can be switched with an OR-Formula. Select menu item $Swap\ AND \rightarrow OR$, to replace an AND- by an OR-Formula (keeping all child nodes in the formula). Vice versa, select menu item $Swap\ OR \rightarrow AND$ to replace an OR- by an AND-Formula.

6.5. Formula-Mappings

If a node in the premise and a node in the conclusion of an application condition represent the same node in the graph, a mapping has to created from the premise node to the conclusion node.

You create a mapping by selecting Mapping from the rule editor palette, clicking first on the source node in the premise graph and then on the corresponding target node in the conclusion graph. Mapped node pairs are visualized by the same color and the same number in front of their names.

Example: Create a mapping from node Order in LHS to node Order in ACO.



6.6. Nested Application Conditions

Application conditions need not always be defined over the *LHS* graph of a rule. They may be hierarchically nested (*Nested Application Conditions*). To define a nestest application condition, select an *Application Condition (AC)* in the tree view and define an application condition for this AC the same way you define application conditions for LHS graphs.

Example: Define an application condition AC2 over the application condi-



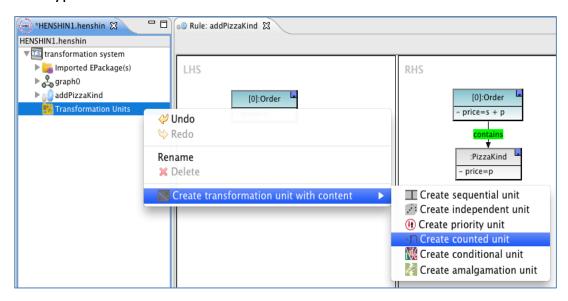
tion ACO. Select ACO in the tree view and select menu item Create Application Condition in it context menu.





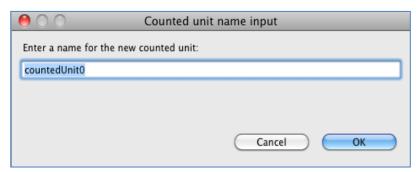
An empty Transformation Unit may be created in the tree view via the context menu of either the transformation system or the folder *Transformation Units*.

Select menu item *Create transformation unit with content* and the desired unit type.

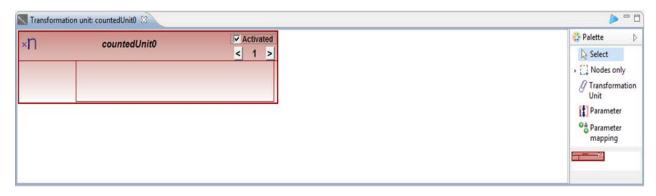


The name of the unit to be created will be requested from the user in a dialog.

Names of transformation units within the same transformation system must be unique. The Henshin Editor ensures this by forbidding names that are already in use. After confirming the dialog, the transformation unit view of the new unit is opened.



Example: Create a new Counted Unit. Select menu item Create counted unit. The transformation unit editor is opened and shows an empty Counted Unit figure.



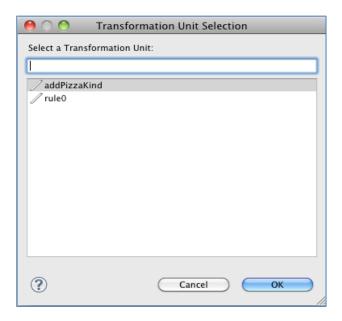
7.1. Create Transformation Unit @

A transformation unit may be created by selecting menu item *Create Transformation Unit* from the context menu of another arbitrary transformation unit or via the entry *Transformation Unit* in the palette.

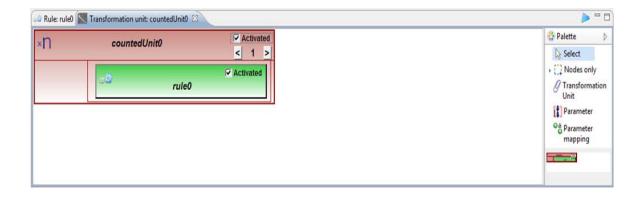
If a container transformation unit contains other sub-units already, then the position of the new sub-unit to be added via the palette may be defined as follows:

If you click on the container unit after selecting the palette entry, the new unit is added as last sub-unit to its container unit. If you prefer a different position, you click on the sub-unit that has the desired position in the container unit.

If there are more than one unit to be added, a dialog supports the selection.



Example: Insert transformation rule rule0 as sub-unit into the previously created Counted Unit via the context menu item Add Transformation Unit.



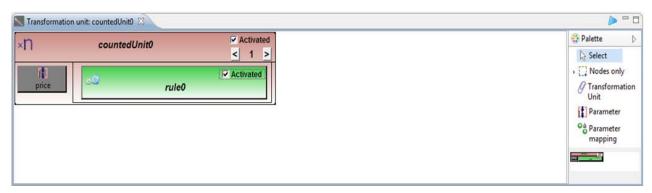
7.2. Parameters



Parameters for transformation rules may be defined by selecting menu item *Create Parameter* from the context menu or by clicking on the palette entry *Parameter*. If you create a parameter via the palette, you first click on the palette entry and then on the transformation unit the parame-

ter is supposed to belong to. When creating a new parameter, its name is requested by a user dialog.

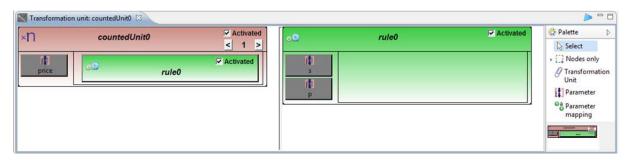
Example: Create a new parameter price.



7.3. Open Sub-Units

In the transformation unit view a container unit can be shown together with one selected sub-unit. A sub-unit is opened when double-clicking on it. Then, the container transformation unit, containing the selected sub-unit is shown on the left-hand side, and the sub-unit on the right-hand side of the view. In this way, a transformation unit can be viewed at arbitrary nesting depth. With a double click on the uppermost element of the unit on the left-hand side, you come back to the view showing the uppermost container unit.

Example: Open the view of sub-unit *rule0* by clicking on its figure as sub-unit of the Counted Unit.



7.4. Parameter Mappings

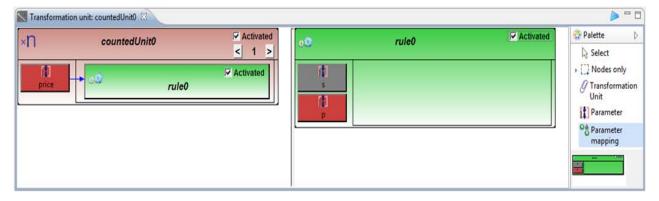


If a transformation unit and one of its sub-units are opened, a mapping may be created between their parameters as follows:

- 1. Select entry *Parameter mapping* from the palette.
- 2. Click on the parameter of the container unit you want to map to the sub-unit.
- 3. Click on the parameter of the sub-unit you want to map to.

In the container unit, the mapping is depicted as arrow from its parameter to its sub-unit that contains the second parameter of the mapping. The direction of the arrow indicates the direction of the parameter value passing. Mapped parameters have the same color.

Example: Create a parameter mapping from parameter price to parameter p.



7.5. Change the Order of Sub-Units

If a transformation unit contains more than one sub-units, their order can be changed by "Drag and Drop". Click on a sub-unit and keep the mouse button pressed. Drag the sub-unit to the desired position and release the mouse button.

7.6. Create Transformation Units with Content

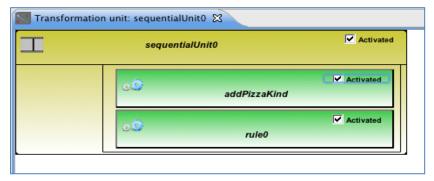
To create a transformation unit with content, proceed as follows:

- 1. Select one or more transformation units.
- 2. Click on the selected transformation units with the right mouse button. A context menu appears.
- 3. Select menu item *Create transformation unit with content*. A submenu opens with all transformation unit types, that may contain the currently selected transformation units.
- 4. Select one of the offered transformation units. A new transformation unit is created that contains the previously selected units as sub-units.

Attention: If you perform this procedure from another transformation unit, the selected units are replaced by the newly created unit and all existing parameter mappings are passed on to the newly created parameters.

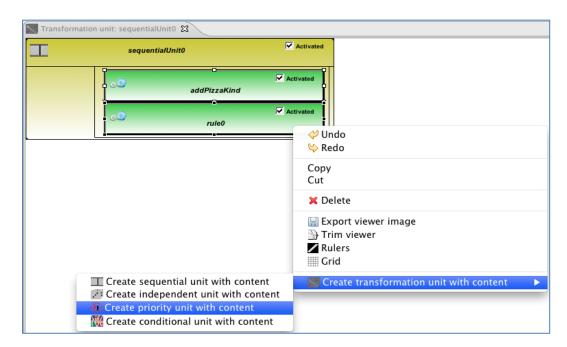
Example:

1. We start with transformation unit *sequentialUnit0* that contains two rules *addPizzaKind* und *rule0*.

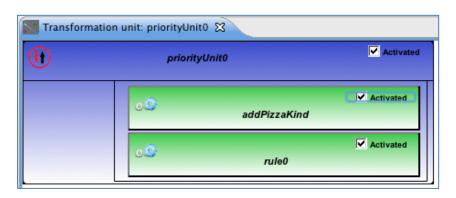


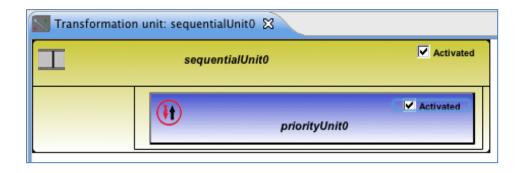
2. Select both rules.

3. Open the context menu with the right mouse button and select *Create priority unit with content*.



4. A *Priority Unit* with the sub-units *addPizzaKind* and *rule0* is created, and unit *sequentialUnit0* now contains this newly created Priority-Unit.

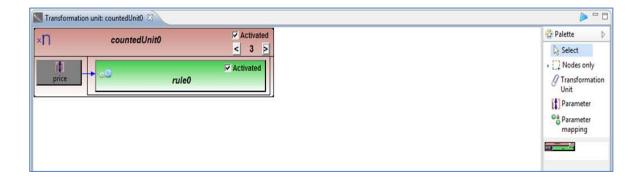




7.7. Change Counter in Counted Units

In a *Counted Unit* a counter may be defined denoting how often the subunit of this unit shall be applied. In the upper part of a *Counted Unit* figure, this number may be changed by clicking on the arrow symbols. By default, the counter is set to "1". A sub-unit of a Counted Unit may be applied without restricting the number (as long as possible) by setting the counter to "-1".

Example: Increment the counter of a Counted-Unit to 3.



7.8. Amalgamation-Units



An *Amalgamation Unit* is a special case of a transformation unit. This unit consists of exactly one kernel rule and at least one multi-rules.

Like all other units, amalgamation units are created by selectin the menu item *Create transformation unit with content* and the sub-menu item *Create amalgamation unit* in the context menu of either the transformation system or the container *Transformation Units*.

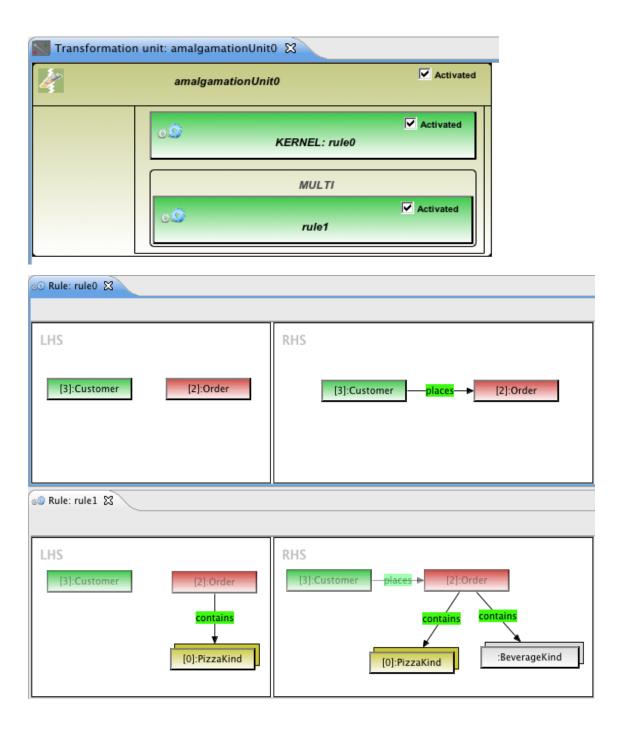
A special property of amalgamation units is that you have to define the kernel rule before defining the multi-rules. To define a kernel rule, you create either a new kernel rule via menu item *Create Kernel Rule* or you use an existing rule as kernel rule by selecting menu item *Add Defined Rule as Kernel* in the context menu of an Amalgamation-Unit.

As soon as an Amalgamation-Unit has a kernel rule, the menu item *Create Kernel Rule* disappears. As soon as the Amalgamation-Unit has one or more multi-rules, the menu item *Add Defined Rule as Kernel* is not shown anymore. This ensures that an amalgamation unit always has only one kernel rule.

When a multi-rule is created, all nodes and edges of the kernel rule are copied to the corresponding rule sides (LHS and RHS) of the multi-rule. All changes in the kernel rule, such as adding new nodes and edges, deleting elements, adding mappings or parameters etc. are also executed for all multi-rules. Note that changes concerning additional elements (that occur only in the multi-rule and not in the kernel rule) are executed in the multi-rule only.

In the multi-rule editor, you can distinguish kernel rule nodes from multi-rule node by their color (kernel rule nodes are a shaded grey) and by the different figures (kernel rule nodes are simple rectangles whereas multi-rule nodes (or multi-nodes) are depicted by rectangles with a double off-set frame to emphasize their semantics that a multi-rule node may be mapped to several graph nodes at once. Kernel rule nodes must not be changed in the multi-rule editor (only in the kernel rule editor). Multi nodes from the LHS may be copied to the RHS by clicking on symbol in the tool bar. An identic mapping is established for the copied nodes.

Example: Define an Amalgamation Unit amalgamationUnitO. Its kernel rule ruleO adds an open Order to a Customer. The amalgamation unit has a multi-rule rule1, which defines that for each ordered pizza (PizzaKind) a drink (BeverageKind) shall be ordered as well.



8. Deleting Elements

All objects that are shown and selectable in a tree view or graphical view may be deleted using the entry *Delete* in the tool bar or the menu item *Delete* in the respective context menu.

Mappings between nodes in a rule or application condition are an exception since they are visualized only via colors or corresponding node numbers. Mappings may be deleted only via the menu item *Delete Mapping* from the context menu of the node in the rule view.

Attention: Parameter mappings cannot be deleted by *Delete Mapping* as node mappings. To delete parameter mappings, click on the mapping arrow in the transformation unit figure and select the *Delete*-entry in the tool bar or the menu item *Delete*.

9. Applying Transformation Rules and Units

A transformation rule is applied to a graph either by selecting the menu item *Execute Rule* or *Execute transformation unit* in the context menu of a graph or a rule (1) or by selecting the button from the tool bar of a transformation rule, a transformation unit or the graph the rule or unit shall be applied to (2), (3).

There are two possibilities how to apply a transformation rule or unit:

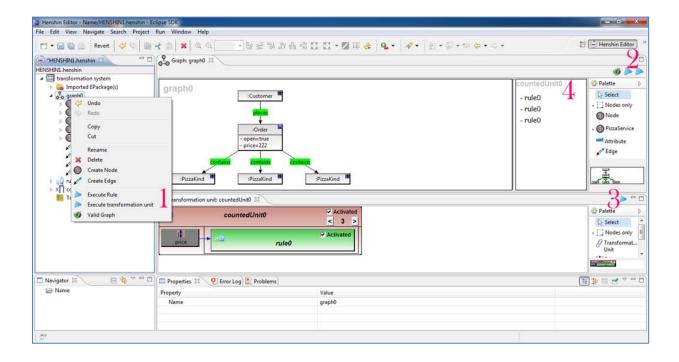
From the rule's (unit's) context menu (1):

If there is only one graph in the transformation system, the rule (unit) is applied to this graph. Otherwise, all existing graphs are shown in a user dialog, and the user is requested to select the graph the rule (unit) has to be applied to.

From a graph view (2):

If there is only one rule or unit in the transformation system, this rule (unit) is applied to the selected graph. Otherwise, all existing rules (units) are shown in a user dialog, and the user is requested to select the rule (unit) that shall be applied to the graph.

If the rule (unit) to be applied has parameters, the Henshin-Editor computes the parameter value binding from the match, as far as possible. If no unique parameter binding can be found, a dialog is opened requesting the user to define the parameter value binding.



After applying a transformation unit, the names of all applied transformation rules are displayed in their application order in a history view to the right-hand side of the graph (4). To undo one or more rule applications, double-click on a rule in the history view. Doing this, all rules before the selected rule and the selected rule itself remain in the list. All succeeding rules are undone. The graph is updated accordingly. All rule applications can be undone in one step by double-clicking on the name of the transformation unit in the history view.

10. Checking EMF Consistency of Graphs and Rules



A graph may be checked for EMF Consistency (each object (except the root) is contained in exactly one container, and the containment hierarchy is acyclic). The check is invoked by selecting menu item *Validate Graph* in a graph's context menu (1) or by pressing button in the tool bar of the graph view (2).

In the same way, a rule may be checked whether it preserves EMF consistency of a graph or not. Either the check is invoked by selecting menu item Validate Rule in a rule's context menu (1) or by pressing button in the tool bar of the rule view (3).

After the check, the validation results are shown in a dialog.

