Eclipse diagram

model used for a sequence diagram defines some elements and relationships that must be mapped to graphical elements in order to be represented on the sequence diagrams. For the UML2 modeler, the semantic model is defined in. The job of the architect is to map the UML2 interactions, life-lines, executions and messages. Even if the support for sequence diagrams in Sirius is not dedicated to UML2, these four kinds of elements or similar ones must be provided by the sequence meta-model in order to be represented as sequence diagrams in Sirius. In UML2, the instance role and the lifeline are represented by one element of type Lifeline. An execution is composed of three elements:. In UML2, the execution is represented by an element of type Execution Specification , the execution start and finish are defined by an abstract type Occurrence Specification. The message represents a kind of communication between lifelines of an interaction. A message is composed by three elements:. In UML2, the message is represented by an element of type Message Specification , the message send and receive are defined by one abstract type Occurrence Specification. All these occurrences are event ends contained in the EventEnds list. Otherwise, the message would be associated to the lifeline. Creation tools and reordering tool must manage the semantic orderings. More explanations are given in the next section. First, in an odesign file, from an existing viewpoint, you have to create a new kind of representation : Sequence Diagram Description. In a sequence diagram, graphical elements are ordered chronologically and this order is essential. Maintaining and updating the elements global order will be the main purpose of all tools that you will create later. Your tools must keep all the time the order of semantic elements and according to this, Sirius manages the graphical representation order. The ordered elements in a sequence diagram are defined with the Ends Ordering and Instance Roles Ordering fields. These properties will be used by Sirius to automatically order the graphical elements when you open a sequence diagram for an interaction. The Ends Ordering handles the vertical order of events. It specifies with an expression how semantic elements must be ordered. A specific variable exists for this expression: eventEnds. The variable eventEnds contains the list of all EventEnds existing for the current interaction. Pay attention: the evaluation of the Ends Ordering expression should returned only elements contained in the eventEnds list. These elements should be the semantic elements which will be represented as instance role. If we have a look at the UML2 meta-model, the fragment reference defined in an Interaction contains all the execution occurrences and the message occurrences. Execution occurrences and messages occurrences are EventEnd elements. But the fragment reference contains also some other types of elements as execution specifications. In order that the Ends Ordering property only references EventEnd elements we need to do an intersection of fragments elements and eventEnds using either Acceleo or delegating to a Java service using service Furthermore, the lifeline reference defined in an Interaction contains all the Lifeline representing both lifeline and instance role. And the Instance Roles Ordering expression will return for the diagram below, the ordered list: [producers, consumers]. We want to represent on a sequence diagram four different elements and then associate a mapping to each element:. Firstly, create the instance role mapping. It graphically corresponds to the box at the top of the lifeline. The Semantic Candidates Expression is an Acceleo expression returning the semantic elements for which the mapping will be evaluated and then a graphical element will represent the semantic element on the diagram. Execution mappings are used when you have an element which is composed by a start, a duration and a finish element. First, create the execution mapping for the lifeline execution. This represents the dashed line of lifeline. And create the execution mapping for the execution. This represents the execution square on lifeline or other execution. The end finder expressions are used by Sirius to graphically link the execution to its start and finish elements and to find during creation and reorder operations where to reattach the dragged element. Standard node mappings, direct children of a layer of the current sequence diagram description, can be used to represent the unknown message end. Lost and found messages should be created using a generic tool. Lifelines should be created using an Instance Role Creation Tool associated to the instance role mapping. The predecessor variable represents in the global instance role ordering, the element preceding the new instance role. As for many other tools, it is possible to define a precondition for message creation tools. Depending on the precondition expression, the tool allows the element creation only under certain conditions. The precondition is defined as an interpreted expression. This variable is the semantic element associated to graphical element that is currently hovered by the mouse. This tool is called when the user moves or changes the size of graphical elements on the diagram. A single unique event reorder tool can and must be specified for message and execution mappings. The purpose of the tool is to re-synchronize the graphical ordering with the semantic ordering. When the user reorders a graphical element, the global order of graphical elements changes and the tool must then reorder the semantic elements according to these changes. When we move the execution, the associated message

must be also moved. A single unique event reorder tool can and must be specified for instance role mappings. When the user reorders a graphical instance role, the global order of graphical instance roles changes and the tool must then reorder the semantic instance roles according to these changes. In this example, we want to move the consumers instance role after the producers instance role. Thus, the predecessorAfter variable will point to the producers execution occurrence. This variable represents the semantic elements producers associated to the graphical element preceding the consumers instance role after its move. The predecessorBefore variable will be null, because consumers was the first element of the ordering. Nothing specific for deletion tool, edit label tool, diagram creation and diagram navigation tool, have a look at the Sirius Specifier Guide. Sirius provides support for diagrammatic representations, which represent information in a graphical way. Diagrams are the main and most sophisticated type of representations supported by Sirius. TODO Add other sample screenshots showing various kinds of diagrams. When the current editor is a Sirius diagram, the Eclipse Outline view shows a graphical overview of the diagram which can be used to navigate inside it if the whole diagram is not visible in the main area. This document is structured in three parts. The first part gives an overview of all the main features, and is organized from a functional perspective. For example it contains a section which explains the different ways available to control which elements are hidden or not on a diagram, with links to the corresponding parts of the reference section. The second part is a reference and describes all the features available by looking at each of the main areas of the diagram editors the palette, the tab-bar etc. Finally, the last part gives a reference on all the preferences and configuration parameters which can affect the look and feel of diagrams. Note that as always with Sirius, some of the features described below may or may not be available with all specific kinds of diagrams. It all depends on which features the person who specified configured the diagram decided to support. For example, the names and effect of the tools available in the palette will be different for each graphical modeler. Refer to the documentation of the specific modeler if available for more precise details. Creating new diagrams works in the same as for any other kinds of representation in Sirius. There are three main ways:. Note that the set of diagrams you can create will depend both on the type of the selected element and on the set of viewpoints which are currently enabled. When you create a new diagram, unless you have unchecked the Synchronized mode for new diagrams preference, it will be initially populated with graphical elements according to its configuration and the current state of the semantic model s it represent. If the preference is checked, the diagram will be created empty, and only the elements you ask explicitly to be represented through tools, which will be specific to each kind of diagram will appear on the diagram. Opening existing diagrams and renaming or deleting them works exactly as for any other kind of representation. Refer to the general documentation about the Modeling Project UI for details. This grid can be in front of all figures if the property Grid In Front is enabled. Capability to allow edit parts to snap to the grid when editing during creation, moving or resizing. It is also possible to have connection bendpoints snap to it. Grid snapping and visibility are two distinct properties, and it is possible to enable one without the other. The Snap to shapes feature allows you to quickly align parts being dragged or resized to other parts in the diagram or that share the same parent e. The snap is effective on top, bottom, right, left and center of the figure. Feedback is shown in the form of a gray line when a part is being attached to another part. This drag part of this feature is also available on border nodes since Sirius 4. Note: For border nodes, the snap is effective only on the center of the figure. This is why the resize part is not handled for border nodes. By pressing the F4 shortcut key, it is possible to temporarily enable the snap to all shapes currently visible on the diagram. Some screenshots of this feature are shown below. When the diagram is larger than the editor area, you can move it in all directions by pressing the middle-button and dragging the mouse keeping the button pressed. The zoom on a diagram can be done by using the tools in the palette like explained in Chapter Standard Tools. Or by using the combination of the keyboard key CTRL and the mouse wheel. Since Sirius 4. If the specifier has authorized it, it is possible to resize a shape by dragging them until they are the size that you want. The resize of compartments might evolve in future versions. Contrary to containers with other layout List or Free Form , the size of the compartment container is always computed from the size of the contained compartments. There is an exception when the compartment container contains no compartment. In this case, the compartment label is fully displayed and the container resized accordingly. When resizing a compartmented container or a compartment from the top or from the left, the F3 shortcut configures the same behaviors than on standard containers see previous section. On experimental structures of containers with compartmented compartments, the F3 shortcut also allows to disable the resize propagation to manually correct the computed layout and remove empty space experimental. This might occur when there are several levels of compartments

without the same number of compartments on each level. The corrective resize can be done on the last compartments of each compartmented container from bottom for vertical stacks and from right for horizontal stacks. In the Appearance tab, you can change some graphical properties defined in the Sirius modeler; font size, font color, edge color, routing styles, You can also change graphical properties not defined in the modeler but global to Sirius. This diagram will be used as example to illustrate the effects of each property. All visible edge labels can be snapped back to their default position by using the action Snap back label s. The result of this action is visible here:. This action is also available individually for each visible label of an edge. It is possible to snap the bend-points to all shapes by pressing F4 shortcut key during the move. This feature is only available if the Snap to shapes is already activated for the current diagram. As for Snap to grid , and unlike to snap for node, there is no guide drawn during the move. You can define some bend-points or inflection points on an edge. It is possible to remove bend-points to retrieve simple edge. For oblique edges, all bend-points will be removed to retrieve the original Straight edge. The edge state just after its creation:. For rectilinear edges, bend-points will be removed to retrieve a rectilinear edge with source and target connection points centered on the middle of the source or target node appropriate side. One additional bend-point is inserted if source side and target side are in opposite direction. Two intermediates bend-points are inserted if source side and target side are in the same direction. If number of bend-points can not be reduced, remove action will be inefficient for example, case when the rectilinear edge is straight. The rectilinear edge after defining some bend-points:. There are eight actions to straighten an edge, I. If the edge is connected to a border node, the border node is moved too. There are two distinct kinds of straightening actions. The following examples show how they can be used. Example of initial state of vertical cases:. The action is available menu is displayed if the selection contains at least one edge note, labels and text attachments are not considered as edges. The menu will show all actions available for all edges in the selection. So if a straighten to top is available on one edge and straighten to left on another, both action will be available. Their execution will only affects the edges compatible with it. The straighten actions can be triggered on one edge but also in case of a multi-selection containing more than one edges. In this case the straighten actions available will be the union of actions available for each edges of the selection. When executing a straighten action, only edges from the selection that can be affected by this action will be impacted if a constraint prevent an edge to be straighten to the direction of the executed action, then it is untouched. In specific cases, we can find edges connected to border nodes. For example in a diagram representing data flows, exchanges are connected to port in and port out. When the user wants to move an exchange on a layouted diagram, he has to move the source port, then the target port and possibly move some bendpoints of the edge. The F3 key has no effect if you attempt a reconnect by moving the first or last edge end. Example when activating the tool and moving the second edge toward the top:. Furthermore, since Sirius 4. However, the selection needs to validate the following rules:. Note that an edge group new location corresponding to the former location of another edge group in the selection is valid. It is possible to display an attachment link between an edge and its labels when edge or label is selected. Only the top level shapes of the selection are retained for these actions: if inner shapes, border labels or edges are selected, they are ignored. These actions are enabled only if the selected shapes have the same direct parent. At least 3 shapes should be selected to enable distribute actions. For border nodes, they will be enabled only if all selected border nodes have the same parent and are on the same axis top and bottom sides for horizontal actions, left and right for vertical actions. The overlap is forbidden for border nodes, so in some conditions location already used , these actions may not have accurate results. For all distribute actions, the first and the last shapes do not move. The first and last shapes do not depend on the selection order. They depend on the location of each selected shapes and the chosen action. The diagram or container bounds the rectangle formed by the highest, the leftmost, the lowest and the rightmost elements can have a negative origin or can be shifted toward the bottom-right with a blank zone at the top-left. This action aims to move all diagram or container elements so that the diagram or container retrieves its origin while keeping the element layout. This is an example of a diagram with a blank zone between the origin location and the figures bounds:. After applying the action, the diagram figures are moved toward the top-left origin location:. After having scrolled the container, one of the edge end is masked so the edge is hidden. This one will be considered to compute the diagram bounds:. Every graphical element on a diagram except compartments can be hidden explicitly. To do that, simply right click on the graphical element or elements if you want to hide several elements at once you want to hide. The graphical element is now hidden from view. It is also possible to hide an element from the outline view. Choose outline mode to see the semantic model in a tree viewer. You can now right click on an element and choose Hide

element. This button opens a dialog to manage the shown and hidden elements on the diagram with a tree view, using various selection buttons Check all, Uncheck all, Expand All and Collapse all and various filters All element, only checked elements and only unchecked elements. Some elements might be grayed in this tree compartments, Sequence diagram elements,.. Their grayed status indicates that they can not be hidden from the wizard but their children might support it. It is also possible to hide the label of graphical elements see the node labels section for nodes supported case. It is possible to call this menu directly on the label or on its element. It is also possible to hide or reveal a label from the outline view, in outline mode. The elements whose label is hidden have their names displayed in italic style and their icon is decorated with a L in the bottom right corner. To reveal a label, simply right click on it and choose Show label. Labels of containers, lists and compartment might be hidden by default. It is possible to hide the node label only if it is on the border of its node. Those nodes can have their label hidden by default. Note that if a label of an edge is hidden center, begin or end , all the labels of the edge will be hidden. Same behavior for the reveal action, if a label of an edge is revealed center, begin or end , all the labels of the edge will be revealed. When working on big diagrams, you may want to hide the icons of the labels on all shapes or connectors, in order to improve the readability of your representations. The options "Hide label icons on shapes" and "Hide label icons on connectors" will allow you to do so. If you check both " Hide label icons on shapes " and " Hide label icons on connectors ", next time you will open your diagram, all label icons of shapes and connectors will be hidden :. When working with compartmented containers, you may want to collapse the compartments. On a diagram, you can have validation rules. As a user of the modeler, you can validate these rules on your model. To validate the rules open the context menu of the diagram itself by right-clicking on its background and choose the Validate action. TODO fix the family. To apply a filter, open the Filters menu in the tab-bar to select which filters should be enabled or disabled. To see more easily if one or more filters are activated, the icon in the tab-bar changes, showing a small check mark if at least one filter is activated. You can export a diagram to an image file. To do that right-click on the diagram and choose the menu Export diagram as image. After selecting the file path, you can choose the image format and if you want to overwrite an existing file without warning. On the first export of diagram as image the name is set by default with the representation name. After there is an history to select the last export. If you do not specify the file extension, the extension is determined from the selected image format. You can choose the directory in which to create the file using the Browse button. If you enter a path with an unsupported extension the file path is created with the wrong extension with selected extension in image format append to. Besides, an image size control is available to determine the size level of exported diagrams. If Image Size is set to Max , diagram will be exported with the maximum possible size limited by system-specific limitations while keeping its aspect ratio. This can produce higher resolution images but also larger files and so long export time. If Image Size is set to level Nominal , diagram will be exported with nominal size quality will be lower but export time will be shorter. For example, on this Family diagram: You can observe these different quality, file size and time generation with nominal size, max size and between both of them:. Note: Due to some technical limitations, not all export format support all the features of diagrams. For example, exporting to PNG files will not show gradients on diagram elements. The JPG format is the one which currently produces the best result in terms of fidelity to what you see in the diagram editors. It is possible to export all diagram of a representations file at once. To do that, select your representations file aird file and right-click on it. Then choose the menu Export diagrams as image On the opened pop-up select the path of the folder where the images will be exported. The same functionality is available on every semantic elements which has representations from the Model Explorer view. In this case, only the diagrams of the selected element will be exported. Drag and drop is a feature which can be enabled by the specifier in the Viewpoint Specification Model. It is turned off by default. If the resize of the port is disabled by the specifier in the Viewpoint Specification Model, there is 4 selection handles to indicate the selection, one at each corner of the figure. Only the shifting is authorized. If the resize of the port is enabled by the specifier in the Viewpoint Specification Model, there is more than 4 selection handles to indicate the selection. To shift the port, you should not click on one of this handles. Ports are special shapes because they are linked to a parent container, but are sometimes laid out over two containers. When you try to shift a port, if you click over the port at a location where it is not over the parent container, a virtual area is built around the parent container. This virtual area is not visible and its size is calculated to cover the container and the port itself. The port shifting is protected against unexpected drag and drop in the container above mouse location. If you want to drag and drop the port in a new container, just drag the port with the mouse beyond this virtual area e. It is possible to duplicate format of diagram

elements, that is to say to replicate mutual organization and style of diagram elements from one diagram to another. This replication only applies to the same semantic elements between diagrams. But it is also possible to only Paste Layout or only Paste Style. The initial copy is always a Copy Format in all cases. The next picture shows a partial view of a diagram. In this picture, format of John Doe and Earvin Johnson persons will be replicated to another diagram. The next picture shows the diagram where the layout will be replicated. The layout applies to John Doe and Earvin Johnson and is exactly the same as the original one compare John Doe and Earvin Johnson each other positions with first screenshot. When using low zoom level, some diagram elements can be twisted and the paste operation will duplicate theses twistings on the other diagram. Please, zoom in to reduce twistings. It is possible to mark specific elements in a diagram as having a fixed location so that they are not moved by the Arrange All action. Diagram element pinning allows you to combine both methods: manually arranging the position and size of elements when it is important while still being able to call Arrange All for the rest, without the risk of destroying your manual layouts. As we can see on the next picture, all diagram elements are moved. But we would like to have a fixed location for John Doe and Jane Smith. So theses elements must be marked as pinned thanks to the contextual menu. The notes can be pinned if they are attached to a pinned diagram element. If a note is attached to several elements of which at least one is pinned then the note is pinned. By default, this preference is disabled, in this case the unlinked notes are pinned. This button opens a dialog to manage the pinned and unpinned elements of the diagram with a tree view, using various selection buttons Check all, Uncheck all, Expand All and Collapse all and various filters All element, only checked elements and only unchecked elements. Edges will not be shown as they will never have children in this tree. Different diagram edit modes are available. Each mode focus on a specific way of editing diagrams. Modes are accessible from a dropdown menu in the tabbar when no diagram element is selected:. This is the default mode used to create diagrams and populate them. In this mode, there is no restriction regarding available tools and semantic changes. To go back to this mode after changing it, you just have to select it from the tabbar:. When you only want to do layout operations e. For example, when organizing bordered nodes, you may change the container of the semantic element, without wanting do so. Note that other operations which can change the semantic model are still possible. Layouting mode can be activated on a diagram through the tab bar when there is no selected element :. In this example, as Layouting mode is activated, reconnecting the Edge from Abstrat Customer to Company is forbidden. You can arrange this edge change start location, add bending points This mode allows to visualize all diagram elements either visible or invisible because of a filter or a previous hiding action execution on the element. Invisible elements will be shown with transparency:. This mode also allows to change visibility of diagram elements. Additional features of the double click are the following:. It is easy to do semantics changes whereas you only want to change diagram element visibility. So when this mode is enabled, the following operations are changed:. Visibility mode can be activated on a diagram through the tabbar when there is no selected element :. When the action is checked, the mode is Unsynchronized. Otherwise, the mode is Synchronized. With Synchronized mode, the displayed diagram represents what is provided by the modeler. However, with Unsynchronized mode, the diagram may not display all elements provided by the modeler, but only the elements that the user selected. For instance, we will consider having a semantic model with a root element that contains a child element. In the Viewpoint Specification Model, the feature Synchronization See the Common Attributes section of the specifier guide of the mapping for the elements is defined as Not synchronized and the edge mapping specifies an Unsynchronizable behavior default feature. To start we only have the root displayed. Then, we will drag and drop the child element from Model Explorer view. With Synchronized mode, the edges related to this dropped element will be displayed. Note: if the edge mapping was defined as Synchronized , the result will be the same. But if it is Not synchronized , the edges will not be displayed. Note: if the edge mapping was defined as Not Synchronized , the result will be the same. But if it is Synchronized , the edges will be displayed. The property is Synchronized mode for new diagrams. If you want your new diagrams be unsynchronized , uncheck this property. Elements matching the regular expression "m" are all elements that starts with the letter 'm' any case :. This allows you to rapidly search text in the displayed name or the contained String attributes of your diagram elements elements. Although the style of the diagram elements is defined in the VSM, it can be customized for each diagram elements. You have the possibility to customize the routing style of all new edges from the preferences. All Sirius diagrams have a palette of tools, which by default is docked on the right hand side of the main graphical area. The top row of the palette contains some general tools which are available on all Sirius diagrams, while the rest contains tools specific to each graphical modeler. You can resize it to take more or less horizontal space

by dragging the vertical separator between the palette and the diagram with your mouse. You can also fold the palette to hide it almost completely by clicking on the triangular icon in the corner of the palette header. When folded, you can restore it by clicking again on the same triangular icon. You can also keep it folded except when needed: when it is folded, a single click in the vertical folded area will reveal the palette temporarily so that you can select a tool. It will automatically hide again if you select a tool or click elsewhere. A few general tools are available in standard on all Sirius diagrams. They appear in the top row of the palette, just below the header. The selection tool is the default one initially selected when you open a diagram. To select an element on a diagram while this tool is active, simply click on it. To select several elements at the same time, you can either draw a rectangle on the diagram, or click on each element individually while keeping the Ctrl key pressed. When selecting elements which are already selected using this method, by clicking on them or drawing a rectangle around them, they are removed from the selection. The selection of several elements by drawing a rectangle has 2 modes:. Selected elements have an outline and anchors drawn on their border. Note that when a selection contains multiple elements, exactly one of them has black selection anchors; the rest have white anchors. The element with the black anchors is called the primary selection, and some tools treat it differently than the others for example alignment tools. Next in the palette come two buttons to control the zoom level of the main diagram area. When the Zoom in resp. Zoom out button is active, clicking anywhere on the diagram will increment resp. The current zoom level is visible in the tabbar when no diagram element is selected see the section on the tabbar buttons for more ways to control the zoom level. Notes, Representation Links and Note attachments. All Sirius diagrams support the creation of notes and text elements, which can be attached to diagram elements. These elements are purely graphical annotations, and have no effect on the semantic model. By default, the Note tool represented by a yellow sticky note is selected. If you click on the small arrow next to the sticky note, a menu appears where you can select one of the available tools: Note , Representation Link , Text or Note attachment. Notes , Representation Links and Text elements are created in a similar way: either a single click somewhere on the diagram which creates an element with a default size , or a click-drag to create the element with a custom initial size. The only difference between notes and text zones is the visual presentation; notes have a yellow background by default and a border which represents a sticky not with a folded top-right corner. A Representation Link is a special kind of note which references any existing representation in the project. When creating a representation link, a the target representation must be selected from a dialog. The representation link displays the name of the target representation and its icon as the note header. It is possible to navigate to the target representation by double clicking on a representation link. When the targeted representation is deleted, representation links that reference it are not deleted. Just like normal notes, representation links have a text field for free text and can be attached to other elements with the note attachment tool. It is also possible to set a new target representation for an existing representation link by selecting Set target representation Note attachments can be created to link either notes or text zone to diagram elements including other notes and text attachments. To create an attachment, activate the tool and then click once on the source element and once on the target element. You can also click and drag from the source to the target elements. Generic Edge Creation Tool allows to create an edge by selecting first the source and the target before choosing the concrete edge creation tool. That allows to restrict the list of possible edge creation tools between two elements. The tool is allowed when the mouse is hovering the first element if at least one edge creation tool is applicable from this element. If only one edge creation tool is available, the tool will be applied once the source and target have been selected. If several edge creation tools are applicable, a menu displaying the list of possible tools makes possible to select which one to apply. The final standard tools which are available in the top row of all the palette allow to mark or unmark diagram elements as pinned. Pinning an element on the diagram means that when an automatic layout of the diagram is requested see the Arrange All action , the element will not be moved or resized: it will stay at the exact same position as you placed it. Pinned element can still be moved manually. To pin or unpin an element, simply select the appropriate tool using the arrow right next to the icon to make the menu appear , and when the tool is active, click on the element to mark as pinned or un-pinned. Note that depending on which Viewpoints and Layers are currently enabled, some tools may or may not be visible for a given modeler. Tool Drawers. Tools in the palette may be organized in expandable drawers to group them by category. Click again to fold it back and hide its tools. When you expand a drawer, some others may be folded automatically if there is not enough space to show all of them. This will force it to be kept opened, and its tools available, at all times. Tool Groups. Most entries in the palette correspond to a single tool, but some of them represent tool groups. They behave like combo-boxes: click

on the arrow to expand the group and reveal all the tools available in this group. Click on the tool you want to activate to make it the default and close the group. You can then use the selected tool as any other, but you have to re-open the popup menu if you want to enable a different one. There are different interaction modes possible for the tools available in the palette. The basic pattern is to select the tool in the palette with a simple left-click on in it the palette, and then apply it once on the diagram. Normally, when you have used a tool once, it does not stay selected; if you want to reuse it a second time, you have to re-select it in the palette. To deselect a tool without executing it, simply press the Esc key, or select another one for example the default Selection tool. For such a tool, if you put the cursor on top of another diagram element, the mouse cursor will change to indicate the action is not allowed. Note that the exact shape of the mouse cursor depends on the operating system. Direct Action Tools. Direct action tools are the simplest and most common one. They require a single click somewhere on the diagram as long as it is allowed. The most common direct action tools are element creation tools. Edge Creation Tools. Edge creation tools are a little more complex, in that they require the specification of both the source and target element of the new edge. Once the tool is selected, there are two possible usage modes:. Tools With Selection Dialog. Some complex tools require additional information to perform their jobs. In particular, tools with selection dialogs will open a dialog box when you use them on a diagram element to ask for more information. Other Tools. Finally, because Sirius is an extremely extensible system, graphical modelers may be configured with tools which have more complex interaction patterns. These will usually be custom tools very specific to a particular domain or modeler. Refer to their documentation for details. The top area of all Sirius diagram editors is filled with the tab-bar , which provides access to many operations on diagrams and their elements. The content of the tab-bar will depend on whether the current selection is the diagram itself i. When the diagram itself and not a specific element is selected, the tab-bar contains the following buttons:. Automatic Layout Tools. The first group of tools are used to trigger an automatic layout of the elements on the diagram. Automatic layout uses a generic algorithm which tries to arrange the position and sizes of the elements on the diagram in a nice, readable way. In particular it makes sure, unless specific constraints prevent it, that no elements overlap each other, and that elements which contain others i. It also tries to minimize the crossing of edges. The two ways to invoke the automatic layout algorithms are available as a drop-down menu on the left-most icon in the tab-bar. Selection Tools. The next group of tab-bar tools, also organized in a drop-down menu, can be used to select groups of diagram elements in a single operation. When in manual refresh mode i. Automatic Refresh is unchecked , you must manually use the Refresh operation whenever you want the diagram to take into account changes in the model. Even in automatic refresh mode, it may be sometimes necessary to invoke an explicit, manual refresh using this operation if Sirius got confused. Layers Selection. The various kinds of elements which can appear on a diagram, and the tools which are available to manipulate them, are organized in Layers. Each diagram has a default layer which is always enabled, but may also have additional layers. Additional layers can be optional or not. You can enable or disable the optional layers at will, to hide or reveal new kinds of elements. Note that the list of layers available to you on a diagram may depend on which Viewpoints are currently enabled in the Modeling Project , as some viewpoints can contribute additional layers to diagrams. The set of layers currently enabled can be controlled using the drop-down menu in the tab-bar. Note that mandatory layers can not be disabled and are not displayed in this menu. Simply click on the button to reveal the menu, and check or un-check the optional layers you want to have enabled or disabled. A tick-mark is visible on the tab-bar button if at least one optional layer is currently enabled. They are separated with an horizontal separator. Filters Selection. Diagram configurations may define filters , than you can enable or disable to hide or show diagram elements depending on semantic conditions. For example on a UML class diagram, a filter may be defined to hide all abstract classes and interfaces, so that only concrete classes are visible. The set of filters currently enabled can be controlled using the drop-down menu in the tab-bar. Simply click on the button to reveal the menu, and check or un-check the filters you want to have enabled or disabled. A tick-mark is visible on the tab-bar button if at least one filter is currently enabled. You can chose to hide or reveal specific diagram elements or just their labels independently of the conditions in pre-defined filters. This can be useful if you want to concentrate on specific parts of your model, or if you want hide parts which are not relevant to simplify communication with other people. This button opens a dialog box which presents all the elements visible on the diagram, organized in a hierarchical way which corresponds to their graphical organization. Note that because a single semantic element may have multiple representations on the diagram, they may appear multiple times in the dialog box. Making one representation hidden will not hide the other graphical representations of the same element.

You can chose which elements should be visible resp. For some elements e. The top area of the dialog box contains various buttons to help you find the elements you want, by show only the checked i. You can also search for elements by their name using the search box. When this is possible, the label is presented as a sub-item of the element in the dialog box, with its own check-box. You can chose to pin or unpin elements on the diagram. Pinned elements are kept in their position by the automatic layout algorithms see Arrange All , while unpinned elements can be moved and resized. This button opens a dialog box which presents all the elements on the diagram, organised in a hierarchical way which corresponds to their graphical organization. Making one representation pinned will not pin the other graphical representations of the same element. You can chose which elements should be pinned resp. Note that pinning has currently no effect on edges. Paste Layout. This button is only enabled if you have previously copied the layout of some elements in the clipboard, using the Copy layout action only visible in the tab-bar when elements are select. Paste layout will apply the same size and positions which were copied in the clipboard to the corresponding elements in the current diagram. Zoom Controls. Next in the tab-bar is a set of buttons to control the zoom level. The Zoom in and Zoom out buttons behave similarly to their equivalents in the palette. The combo-box shows the current zoom level and allows you to choose among some pre-defined levels. You can also manually enter a specific zoom level e. Export As Image. This button can be used to export the current diagram as an image file stored on disk. Note that not all formats are currently equally supported; they may not all produce the same result. For diagrams which are tool large to export as a single image, the Export to HTML check-box can be used: when it is checked, instead of exporting a single image Sirius will split it in a matrix of smaller images, and produce an HTML file while loads them all arranged in a table to reproduce the original image. Layout Mode. It can be used when you want to tweak the layout of a diagram and want to be sure to only make graphical changes, and not semantic ones by mistake. When this mode is enabled, the following operations are changed:. Arrange Selection. This action launches the layout algorithm only for the diagram elements which are selected and not pinned. All the other elements on the diagram, whether pinned or not, will not be moved. Note that edges may be re-arranged if one of their ends is part of the selection. Alignment Control. This menu contains several operations which can be used to align several graphical elements in various ways. The actions in this menu are only enabled when several elements are selected. Distribute Elements. The actions in this menu are enabled only for selected nodes with same direct parent. These two buttons can be used to mark all the selected elements as pinned or not. Copy Layout. This tool can be used to duplicate the layout of some diagram elements from this diagram into another. First, select the set of elements of interest, and use this Copy layout action. Then go to the target diagram, and choose the Paste layout action from the tab-bar there. This button hides all the selected elements from view. This button hides the label of the selected elements. Delete from Diagram. This action removes the selected graphical element from the diagram, but does not delete the corresponding semantic elements. It is only available on un-synchronized diagrams or element types. Delete from Model. This action removes both the selected graphical element and the corresponding semantic elements. The exact effect of deletion on the semantic model depends on how the diagram was configured, and may have other consequences i. Font Controls. The next set of buttons can be used to control the font attributes bold or italic , the font color, and the font to use itself, for the labels of the select elements. Note that all font-related attributes. Color and Visual Style Controls. The next group of buttons can be used to control some graphical attributes of the selected elements: fill color, line color and line style. The Workspace image button can be used to replace the graphical representation of an element by an image that you can select from anyware in your Eclipse workspace. Cancel Custom Style. The Cancel Custom Style button resets all the style attributes of an element to its default values and un-marks it as customized. Apply Style. Use this button to reproduce the visual style of an element onto others. This action will apply the style of the last selected element to the others. When you click on this button, all the visual attributes from the last element which can are compatible with the other elements will be applied to them. Make Same Size. When multiple elements are selected, clicking on this tool will resize all of them to have the same size both width and height. The element used for reference to decide the size to apply is the last selected. For region containers, if the selected region containers have the same number of regions, the regions will have the same size. If the region container used for reference has more regions than the other s , the last region of others will have the size of all remaining regions in the one used for reference. If the region container used for reference has less region than the other s , the action has no effect. This button marks the selected elements as auto-sized. Auto-sized elements will adjust their width an height dynamically according to their contents. Due to a current limitation, on Eclipse 4. The tab-bar content enablement changes when at least one

element is selected:. In the Properties view, you will find several tabs which will not be the same if an element is selected on the diagram or not. If an element is selected on the diagram, you will have on the Properties view the following tabs:. If you do not select an element, you will have in the Properties view the properties of the diagram itself :. The hidden elements have their label displayed in italic style and their icon is decorated with a yellow dot in the top left corner. The elements whose label is hidden have their label displayed in italic style and their icon is decorated with a L in the bottom right corner. Some preferences and configuration parameters are available for you to customize your experience. They can affect either the look or the behavior of diagrams. Global preferences are global to a Sirius installation, or more precisely to an Eclipse workspace. If you change them, they will affect all the modeling projects and all their diagrams for the current workspace. The Appearance preferences page see below gives you two preferences to control whether or not to show the labels on shapes nodes, containers, bordered nodes and connectors edges. It also allows you to customize the default font and colors used for some basic diagram elements. Diagram configurations already specify default values for these style attributes, so in practice these preferences are mostly useful for notes. The Display viewpoint colors preference allows, if checked, to display, in the color Palette, the fixed colors that are defined in the activated viewpoints. The viewpoint colors are displayed after the grayed colors and before the colors of the rainbow. The Authorize decoration overlapping preference lets, if checked, the decorations of different positions overlapping each other or displaying outside the diagram element bounding. If unchecked, decorations may be grouped in a list decorator to avoid overlapping. The Connections preferences page see below gives you various preferences related to connections. The first preference Show link between edge and its labels on selection is documented in the Manage edges section of this documentation. The second block of preferences allows to override default values of some edges characteristics. These preferences do not affect existing elements. This preference is documented in the Style customizations chapter of this documentation. The second group of preferences concerns the override of jump links properties. First, you need to enable Jump links to override the default values. The Printing page see below gives you various preferences related to page layout for layout printing. The Print decoration preference allows to print decorations displayed on diagram graphical elements. The preferences on this page are only taken into account as starting values for new diagrams. Diagrams support several kinds of graphical constructs: simple nodes, containers which can contain other nodes or containers , lists, bordered nodes nodes which appear on the side of other elements, typically to represent ports , and edges between all of these. Each kind of element supports a rich set of possible styles shape and color. As the specifier of Sirius diagrams, you have total control on how the elements in your semantic models are represented using the constructs provided by Sirius, and how they behave. The two are completely de-correlated, and you can choose to represent your semantic models in ways which make sense from various point of views. Diagrams are configured by creating a Diagram Description element inside a Viewpoint and its sub-elements which describe the layers, graphical elements and tools. Like many elements inside a VSM , Diagram Description have a mandatory Id , which should be unique and stable across revisions of the diagram specification, and an optional Label which is used for presentation purpose i. If no Label is specified, the Id is used for presentation. Note that changing the Id will break existing diagrams which were created using the previous version; changing the Label is safe. The mandatory Domain Class attribute is the type of semantic element which are represented by the diagram. In the Model Explorer , end-users will be able to create new instances of this diagram on semantic elements of this type assuming the corresponding viewpoint is enabled in the Modeling Project. By default, new diagrams can be created on any instance of the Domain Class. You can use the Precondition Expression to change this. If such an expression is specified, it will be evaluated in the context of the semantic element the user has selected, and only if the expression returns true will the user be able to create a new diagram on this element. The default title for newly created diagram can be specified using the Title expression Advanced category , which is evaluated in the context of the semantic element on which the diagram is created, and should return a string. If the expression is not specified, the default title is the label of the diagram description of its Id if no Label is set prefixed with the string "new " e. It is recommended that the Diagram Description be explicitly associated with the meta-model s of the semantic elements it will represent. You can add referenced meta-models from different sources in the Metamodels property section of the Diagram Description. Sirius will work even without this association, but setting it explicitly will give you better feedback when validating your VSM. If Initialization is set, then when the viewpoint which contains this diagram description is enabled, instances of this diagram description will be automatically created on all compatible semantic elements. If Show on startup is set, then when a modeling project is

opened which contains instances of this diagram description, they will be opened: if only one such representation exist, it will be opened automatically; if there are more than one, a dialog box will appear to allow the user to select which one s to open. WARNING : Then Initialization flag can have a negative impact on performance: if it is set on a given diagram, then when the user enables the viewpoint which defines this diagram Sirius will attempt to find all possible elements in the semantic model on which it may be possible to create such a diagram. This can take a long time on big semantic models. If Enable Popup Bars flag is set, then when the end-user leaves the mouse pointer still on a diagram or diagram element for a small time, a popup bar will appear with shortcut buttons to all the tools which can be applied in this context. It can be useful to create elements without going to the palette to select tools, but it can also become visually distracting if too many tools are available in the popup bar. By default, diagrams are displayed with white background. In the Background property section of the Diagram Description , you can add either one of the pre-defined system colors or one which you have defined yourself in a Users Color Palette. See the general section on colors on colors for more details on dynamic and computed color of the Users Color Palette. The content of the Diagram Description is mostly made of graphical elements mappings , organized in layers , and their associated tools. In addition, it can also contain validation rules , filters and layout configuration information. Note that the Window Title label and Message label are Translatable Messages and can therefore be localized. Sirius provides a default algorithm to perform an automatic layout of all the graphical elements on a diagram excluding pinned elements. If the default algorithm does not fit your needs, you can configure some parameters or request for alternate algorithms currently only one directly inside the VSM. The Composite Layout enables one to specify the padding between elements and the direction for the default algorithm. The Ordered Tree Layout enables one to lay out nodes as an ordered tree. The manner to retrieve children for each node should be specified. NOTE: these layout algorithms manage only nodes connected by edges, other nodes are layouted from left to right as usual. Additional layout algorithms can be available if you have installed the ELK integration feature with Sirius. Please refers to the ELK documentation if you want details about these algorithms. This feature is considered as experimental because of remaining bugs concerning the ELK integration. If the default layout algorithm configuration does not fit the need, then clicking on the layout element will give you access to its configuration from the properties view. Using the button Add option override will open a dialog allowing you to override supported options of the layout algorithm:. Also you can filter the options by using the text field. The filtering applies to all columns except for the values. To get an overview of what is possible with ELK you can read this article. If you need to go beyond what already exists, you can provide your own layout algorithm programmatically see the developer documentation for details. This section describes how to specify the graphical elements which should appear on your diagram, and how to organize them. All the graphical elements and tools on a Sirius diagram are part of a Layer. Every diagram must have a default layer, which is always enabled, and zero or more additional layers that the end-user can enable or disable following the optional flag value. Note that the default layer is created at diagram creation. When a layer is enabled, all the graphical elements and tools it defines are also enabled. The elements which are defined in different layers can interact and reference themselves. For example, it is possible to define a type of edge connection between elements which are defined in different layers; the edges will only be visible if all the concerned layers are enabled: the ones which define the source graphical element, the one for the target graphical element, and the one for the edge itself. Layers are defined directly inside the Diagram Description element. You must have exactly one Default Layer whose name does not matter , and as many Additional Layers as you need. Mandatory layers are not visible in this menu. The Layers can contain graphical elements and tools definitions as described below , but they can also reuse tools and mappings graphical elements definitions which are defined in other layers using the Reused Tools and Reused Mappings in the Advanced category. If a layer L1 reuses elements defined in another layer L2 , these elements will be enabled whenever L1 is enabled, even if L2 is not. However if both L1 and L2 are enabled, the elements are enabled only once i. When applying a transient layer, the diagram is never modified. Tools are displayed in the palette and decorations are refreshed in opened editors. Note that the Active by default flag have a different meaning for transient layer. If Active by default is set, the transient layer will be automatically enabled when the user opens a diagram and not specially when he creates a new diagram. This section describes the different kinds of graphical elements supported on Sirius diagrams and the general principle of how they are specified inside the VSM, through the notion of mapping. Graphical Vocabulary. Sirius diagrams can contain several different types of graphical elements, which are described inside the Layer elements. These are:. All these graphical elements are described using the concept of a mapping for example a Node mapping

to describe a kind of nodes. A mapping is an element defined inside the VSM which identifies a sub-set of the elements in the semantic model and associates a graphical representation to them: it maps semantic elements onto some graphical notation. At runtime, each active mapping i. In the default functioning mode, whenever the semantic model changes Sirius will automatically re-compute which graphical elements should appear on the diagram according to the active mappings, and create or remove the necessary elements. You can think of the set of mappings defined in a diagram as a description of how to project the concepts in your semantic model onto the graphical language provided by Sirius nodes, containers, edges Mapping Definition. The mapping descriptions for all the different types of graphical elements share the same common principles:. Each mapping is defined in a context in the VSM: either directly inside a layer or inside another mapping. At runtime, the corresponding context element would be respectively either the target semantic element of the diagram itself, or the target of an instance of the parent mapping. This context element is used as a starting point to determine which instances of a mapping should be created on the diagram:. This order of evaluation should be kept in mind when specifying diagrams. In particular, to avoid performance issues you should:. Associated Semantic Elements. To associate more semantic elements to a graphical element than just its semantic target, you can specify the Associated Elements Expression in the Advanced category. It will be evaluated for each instance of the mapping in the context of the semantic target, and should return a set of semantic elements. These will be visible to the end-user in the Properties view when the graphical element is selected, and any change on one of these elements will trigger a refresh of the graphical element. All mappings have a Synchronization parameter in the Advanced category which controls when instances of the mapping are created on the diagram. The possible values are:. Mapping Imports. A mapping import is a feature which can be used to specialize modify a mapping defined somewhere else in an other layer, an other graphical representation or an other viewpoint. It is available for Nodes, Containers and Edges. Although it can be used in normal diagram descriptions, its main use case is when defining diagram extensions , so mapping imports are describe in their own section. A Node mapping is used to represent model elements graphically as atomic elements, i. Node mappings can appear directly inside a layer description, in which case the node will appear directly on the diagram, or inside another element for example a container. A node mapping which is defined inside another element can be either a sub-node mapping , in which case it will appear inside its parent which must be a container , or a bordered node mapping , in which case it will appear on the border of its parent which may be a container or a node, including another bordered node. Bordered node mappings are useful to represent ports. The definition of a node mapping follows the general rules described above to determine the semantic elements for which a node should be created. All the properties available in the Behavior category of the Node mapping configuration allow you to see all the various tools, defined elsewhere in the VSM, which apply to this mapping. The effect at runtime is the same as if you had created an equivalent Bordered Node mapping inside the node mapping. See the styles section for a general presentation of how styles are defined, including conditional styles. This section only presents the aspects which are specific to nodes. Most of the properties used to configure the various styles are self-explanatory. See the styles section below for a general discussion. Gauge Style. The gauge style is more complex than the others and deserves a specific description. It is used to represent nodes as a set of gauges, which can be used for example as progress indicators. In addition to the standard style properties for the label, size and colors, gauges have an Alignment attribute in the Advanced category which indicates how the individual gauges are layed out:. Gauge composites need one or more Gauge Section , which represent an individual gauge. The size of the gauge section depends on where the Value Expression sits between the Min the section is invisible and the Max the section occupies all the available space. Interpolated colors are particularly suited for gauge sections. See the tools section for a general presentation of how tools are defined. This section only presents tools which are specific to nodes. Node Creation. The only type of tool which is really specific to node mappings is the Node Creation tool, used to create new nodes. These work like all the other creation tools. You simply specify the Node Mappings that this tool can create and specify the behavior of the tool inside the Begin element using all the standard model operations. You can specify a Precondition to forbid the application of the tool on some of the elements where it would normally be allowed. The precondition is evaluated in the context of the view element on which the user has put the mouse. Symmetrically, you can specify Extra Mappings Advanced section on which the tool should be applicable, even though these mapping can not contain nodes of the type you will create. In this case, the node created by the tool will actually appear elsewhere on the diagram, but it can often help usability to allow users to click on related elements. A Container mapping is used to represent model elements graphically as containers,

which can contain sub-elements including other containers, recursively. A container can also have bordered nodes , which are like normal nodes but appear on the border of the container, for example to represent ports. Container mappings can appear directly inside a layer description, in which case the container will appear directly on the diagram, or inside another container mapping. The definition of a container mapping follows the general rules described above to determine the semantic elements for which a container should be created. All the properties available in the Behavior category of the Container mapping configuration allow you to see all the various tools, defined elsewhere in the VSM, which apply to this mapping. You can also reuse node and container mappings defined elsewhere in the VSM in any of these three roles using the corresponding Reused X Mappings properties in the Import category. The effect at runtime is the same as if you had created an equivalent mappings inside the container mapping. List containers are represented in a special way, in which only the labels of their direct sub-elements nodes only are taken into account and presented in a vertical list. When represented as a list, only some of the style attributes label text, tool-tip, colors of the container are taken into account. For the list items themselves, only the label text is used. Region containers presents their children containers and lists as compartments, vertically or horizontally organized. They are sorted following the mapping order and the semantic candidates expression evaluation result of each mapping. The other style properties of Region container or Region mapping are used as usual. Warning: The compartment feature has been introduced in Sirius 3. While working reliably, it has some limitations but you can rely on the improved VSM validation to ensure you are using it in supported modes. Known limitations:. This section only presents the aspects which are specific to containers. You can specify the default size of Gradient and Parallelogram styles at creation time by specifying the width and height computation expressions available in the Advanced tab. This default size will also be used to determine the container minimum size when its bounds are auto-sized. The result would be the same if the end-user drawn a rectangle of x pixels at the creation time. The compartment keeps its size. The compartment is increased in width. The first and second steps are the same. But the third is different. The second region has a width of pixels and also a height of pixels highest than the first one. The compartment is increased in both directions. The compartment is increased in height. This section only presents tools which are specific to containers. Container Creation. These tools are used to create new containers. They work like all the other creation tools. You simply specify the Container Mappings that this tool can create and specify the behavior of the tool inside the Begin element using all the standard model operations. Symmetrically, you can specify Extra Mappings Advanced section on which the tool should be applicable, even though these mapping can not contain containers of the type you will create. In this case, the container created by the tool will actually appear elsewhere on the diagram, but it can often help usability to allow users to click on related elements. The Drag Source property indicates if you want to react to graphical elements from a Sirius representation, to semantic model elements dragged from another Eclipse view for example from the Model Explorer , or both. The Containers property indicates for which drop kind of container mappings this tool applies. The Mappings property indicates what kinds of graphical elements can be dropped into the specified containers, when the Drag Source is Representation or Both. The body of the tool the model operations specify how to react when an element is dropped into a container. WARNING : if you choose Model or Both as source, your tool must be prepared to accept any type of source element which can be present in the Model Explorer view, including elements which have no relation to the source mappings you specified for the tool. Use a precondition to check the nature of the element being dragged available through the element variable to disable the tool source elements your tool does not support. When drag and dropping workspace resources files, folders Depending of the dropped resource, the element variable content in the root operation will be different:. A DResource instance has a name and a path. The path is the absolute location of the resource. Bordered nodes are specified using normal node mappings. They can appear inside node mappings include other node mappings which are bordered nodes and container mappings. They can have the same styles and tools associated to them as normal node mappings although they should generally be kept small to avoid weird graphical effects. Edges are used to define connections between diagram elements. There are two kinds of edges, which are defined using different mappings:. They indicate from what kind of graphical element the edges will start and to what kind of element they go. Both properties can contain several mappings, and some mappings can appear as both source and targets. Note that it is possible to have edges whose source or target is another edge, but not both. You must also specify the Target Finder Expression. It will be evaluated in the context of the semantic target element of a potential source as defined by Source Mappings and should return the semantic element s of the targets elements to connect with the source. The expression can return zero elements to

indicate that no edge starting from this element. An edge instance of this mapping will appear between each pair of graphical elements E1 and E2 on the diagram where:. Edges only exist between one source and one target element. Sometimes such an edge is just one segment of a more complex path which traverses many elements. You can highlight such a complex path using the properties in the Path category. Path Mappings should be the list of mappings considered by the path, and the Path Expression should return the list of semantic elements the edge path should go through, in the order the should be traversed. Element-Based Edges are configured much like nodes and containers. You must specify the Domain Class , Semantic Candidate Expression and optionally Precondition Expression to select the semantic elements the edges will represent. In addition, you must specify:. Both types of edges relation-based and element-based share the same style type. It supports conditional style like for other mappings. You can specify the line style solid , dash , dot or dash-dot , color, width using the Size Computation Expression in the Advanced Category , and optional decorations like arrowheads to be put on the end-points of the edge. Note that end-users can change the routing style of the edges individually, independently of the default style you set in the VSM. Edge Labels. Edges can have from zero to three labels. By default they have a single which will be placed at its center. In all cases, the configuration is the same as for all labels. Note that currently only the center label if present is editable if a Direct Edit tool is associated to the edge mapping. Edge Folding. It can often be useful, from one end of the edge, to be able to hide the elements at the other end. You can allow end-users to do this be setting a Folding Style on an edge mapping in the Advanced category. If the user clicks on it, the edges instance of this mapping for which the element is a source resp. The behavior is recursive: if the elements hidden were connected to other elements using this kind edges, those too will be hidden. This is useful when representing trees of elements: if you enable folding, users will be able with a single click to hide or reveal complete sub-hierarchies of elements. Edge Centering. The edge source and edge target are not systematically oriented toward the center of the node, depending on where the edge has been created or moved by the end user. You can specify that an edge end the source, target or both will always be centered. The Centered Source Mappings and the Centered Target Mappings within the edge style description advanced properties tab lets you select for which mapping the source or the target will be centered. Note that this behavior can be customized by the end user via the edge style centered attribute. The bracket edge figure is designed to be drawn from the middle of a node side to the middle of another node side. This section only presents tools which are specific to edges. Edge Creation. This tool is used to create new edges. You simply specify the Edge Mappings that this tool can create and specify the behavior of the tool inside the Begin element using all the standard model operations. The body of the tool will be executed in the context of the semantic element of the source of the new edge. Because edges have two ends, their creation requires users to specify both the source and target elements. By default, the possible sources are all the instances of one of the Source Mappings specified for this kind of edge and similarly for the possible targets. You can restrict this using the Connection Start Precondition resp. Connection Complete Precondition to forbid the use of the tool from resp. Reconnect Edge. In addition to the Mappings property, which indicate the types of edges supported by the tool, you must specify one Reconnection Kind : a tool can be written to only support reconnection of the source end, the target end, or of both. Tools that handle both cases are complex to write, so it is recommended if possible to write two separate tools: one which handle the source re-connections, and one for the target re-connections. The body of the tool is evaluated on the target semantic element of the edge that the user wants to reconnect, with the following variables available to describe the reconnection:. Here are illustration to show the values of this variables in a concrete case. Before the reconnection:. Decorations can be used to add small graphical annotations to the elements of a diagram. Mapping Based Decorations have a list of Mappings to which they may apply. Semantic Based Decoration have a Domain Class and will apply to all graphical elements which have an instance of this class as their semantic element, whatever are the mapping s used to represent them. Generic Decoration is only based on the Precondition Expression to know if it applies on the current graphical Element. The Precondition Expression is evaluated in the context of a candidate semantic element which is the semantic element of the current graphical element that is DSemanticDecorator. If the expression returns true, the decorator will be shown. The Decoration Description has a Position which indicates where the decorator should be placed. Many decorations can be specified at the same Position. To manage that, the decorations are displayed from their defined position along the border of the bounding box according to the Distribution Direction. By default the margin between the decoration and the border of the figure is 2 pixels. When a node contains border nodes, the node decorations are shifted from the border to avoid overlap with border node Decorations example with border nodes:. The

decoration icon is defined with Image Expression which is an InterpretedExpression. Image Expression may be:. Decoration tooltip example:. Sometimes, according to the graphical element size, the decorations are represented with a list decoration which is a 16x16 pixels frame containing three dots. The tooltip of a list decoration displays as lines, the contained decorations with its tooltip. Sometimes, the graphical element size can be too small to display all decorations without overlapping. In this condition, decorations will be gathered into one decoration called list decoration. This list decoration is represented with a 16x16 pixels frame containing three dots. The tooltip of a list decoration displays, as lines, the contained decorations with its tooltip. List decoration example:. Tooltip example on list decoration:. Styles define the graphical appearance of diagram elements. Sirius supports a rich set of possible styles, including conditional styles which can dynamically change depending on the current state of the underlying model elements. Each mapping must define at least one style do be visible on the diagram. Conditional styles make it possible to support different graphical aspects for a single mapping. The actual style used to render an element is determined dynamically depending on the current state of the models elements. To use conditional style, you must first create a non-conditional style, which will be the default. You can then add one or several conditional styles as siblings to the default style. Each conditional style is associated to a condition specified as a predicate. If there are conditional styles on a mapping, their conditions are tested in their order of appearance in the mapping description. The first one for which the condition is true is selected. If none of the conditional styles condition is true, the default style is used instead. One of the responsibility of styles is to describe labels which will be displayed and how they will be formated. The following properties are available :. While not strictly part of the label properties, a Tooltip Expression can be set for most style, and is used to compute the text of the optional tool-tip shown when the user leaves the mouse on an element. It can be seen as an alternate Label Expression which is not visible by default, and can be used to show additional details on an element only when the user requests them, to avoid visual clutter on the diagram. Most styles give you some control on the size of the graphical elements. The properties described below are not available for all styles, but when they are they have the same meaning unless specified for special cases :. Whenever you have to specify colors for a style, you can use either one of the pre-defined system colors or one which you have defined yourself in a Users Color Palette. The palette is created directly inside the Viewpoint element, and its colors can be used by any representation, not just diagram. See the general section on colors for more details. By default, border node can be located on any side of its parent. It is possible to restrict a border node authorized sides. Thus the end user will not have the possibility to create or to move a border node on a forbidden side. Authorized sides can be specified in the advanced tab of the border node style:. Tools are used to define the behavior of your modeler. Tools are defined in Tool Sections , which appear directly inside layers. If a tool is defined inside an additional layer, it will be enabled only when the corresponding layer is enabled. The top-level Tool Sections inside a layer are simply used to separate tools definition from mapping definitions and have no impact on how the tools are presented. However, if you create sub-sections inside those top-level section New Tool You can also create Groups New Tools Tool sections can have Reused Tools , which are defined elsewhere including in another layer, even from a different representation. They will appear and behave as if they were defined inside the section which reuses them. The Group Extension elements defined inside a tool section allow you to add tools in a existing Group defined elsewhere. It can be used for example to merge tools defined in an additional layer into a group of tools defined in the default layer. Note that the Tool Section label and Tool label are Translatable Messages and can therefore be localized. The tool sections created before are still present but you should move it into the default layer since it is likely to be deleted in the future. TODO Study the exact semantic of this and complete the description. The paste tool aims to define operations to paste the given copied semantic element, accessible through the copiedElement variable, into the semantic paste receiver, accessible through the container variable. The copiedView and containerView variables reference the corresponding graphical element. The Containers property indicates the mappings to which this tool apply. Its semantic target is the copied semantic element. If the copied element was a non domain based element e. When the semantic target is not copied, the tool receive a null copied element. Warning : The generic behavior is used only when there is no tool defined on the paste receiver. Otherwise the first tool with a precondition evaluated as true will be chosen and if all defined tools have a false precondition, paste action will be disabled. TODO Description not clear at all. If you do not associate a Delete tool to a mapping, the default delete behavior when the user deletes a graphical element is to remove the target semantic element and all the associated semantic elements from the semantic model. If you want a different behavior or if you want to forbid the

deletion of some elements, you must set an explicit Delete tool. Delete tools are invoked when the user select the Delete from model action from the context menu, from the tab-bar or using the Delete key. The element variable will be set to the main semantic target element being deleted excluding any associated semantic elements while the elementView and containerView tell you which graphical view is being deleted and in which context it is. Setting a precondition is usually preferred as it will mark the Delete action as disabled in the UI tab-bar and context menu while a tool which simply does nothing can still be invoked by users, but with no effect. Note however that the Precondition will be evaluated each time the user selects and element to decide whether or not the delete action should be enabled in the UI. A precondition which is complex to evaluate can have a negative impact on the perceived performance of the modeler. The Direct Edit Label tool is used to allow users to directly edit the label of a representation element. It specifies how to interpret the new label value, and which changes to apply to the model element. Users can invoke it using the F2 shortcut, through a slow double-click, or directly by starting to type some text when a label is selected. If you want to display a different label during the edition, you can use the input label expression. This expression is evaluated during the invocation of the direct edit and displayed to the end-user instead of the standard label defined in the style of the mapping. Note: Edges can have up to three different labels one in the center and one at each extremity , but currently if a Direct Edit tool is associated to an Edge mapping, only the center label if it exists can be edited. This restriction will probably be lifted in future versions. This tool can be used to open a wizard dialog asking the user to select one or several model elements. The body of the tool is then executed in the context of the element selected by the user. Then a dialog box opens with a list of candidate elements to select. By default these candidates are presented as a list. You can also enable the Tree mode to display elements as a tree. In that case, you must specify a Root Expression that will provide elements to be displayed as first level of the tree item in the dialog and a Children Expression , which will be evaluated recursively on each candidate, from the root elements, to build a tree of possible values. Note that Root Expression and Children Expression provides elements that must be included in Candidates Expression elements. Finally, you can set the Multiple flag to allow users to select several of the candidates instead of a single one. It differs only in the presentation of the selection window. The pane-based wizard offers two panels: on the left are all the candidates, which corresponds to the contents of the basic Selection Wizard. On the right is the current selection. The user has buttons to add or remove candidates from the left panel into the right one. The Pre Selected Candidates Expression can be used to define the initial content of the right-side panel. Each dialect supported by Sirius defines a tool which can be used to create a new representation using this dialect. They are using the default supported dialects : Diagram Creation description, Table Creation description and Tree Creation description. These tools one for each kind of representation supported by Sirius can be used to create and open a new representation from an existing diagram element. They will be available to end-users in the Navigate context menu on compatible diagram elements. To configure the tool, simply select in the Mappings property which kinds of elements the tool should appear on you can be more precise using the Precondition expression if necessary , and select the representation which should be created using the Representation Description property in practice, one of Diagram Description , Table Description or Tree Description depending on the tool. Normally, the new representation will be created on the semantic element represented by the diagram element on which the user invoked the tool. Sometimes you want the tool to appear on one element, but create a representation on another one. In this case, use the Browse expression in the Advanced category to navigate from the element selected to the one one which the new representation should actually be created. The Title Expression Advanced category can be used to set the title of the newly created representation. Finally, any operation you specify in the body of the tool will be executed when the tool is invoked; you can use it to initialize the content of the model represented. Each dialect supported by Sirius defines a tool which can be used to navigate to existing representation of this dialect. They are using the default supported dialects : Diagram Navigation description, Table Navigation description and Tree Navigation description. These tools are very similar to the representation creation tools. The only differences is that they allow users to navigate to existing representation instead of creating new ones. If such a tool exists, the Navigate context menu on an element will contains an entry for each corresponding representation which already exists on the element selected or reachable from the selected element using the Browse expression. You can specify a Navigation name expression Advanced category to be used in the menu entry instead of using the title of the existing representation. This can be useful to make it more explicit what the relationship is between the current element and the target tree. This tool is used to trigger operations when the user double-clicks on an element. The Mappings property indicate for what

kinds of element this tool is applicable. Inside the body of the tool, the elementView variable refers to the graphical element the user double-clicked on, while the element variable represents its semantic target element. This generic tool is used to provide a palette entry which can execute any code describe in the internal tool language, including for example invoking external Java actions. Inside the body of the tool, the elementView variable refers to the graphical element the user invoked the tool on, while the element variable represents its semantic target element. This tool can be used to define an entry in the context menu which can contain Operation Actions and External Java Action tools. Which ones are visible when the menu is opened depend on the preconditions of these actions, which can thus be activated or not depending on the currently selected element. Actions can be grouped using the Group Menu. The group has an id. This id can be an id of an existing group. In this case, all menus and actions defined in this group will be added in the existing group. The group also has a location URI. This is an URI specification that defines the insertion point at which the group will be added. The format for the URI is comprised of two basic parts:. Currently, it is not possible to add a group after or before a specific existing group. It is added at the end of the menu having the corresponding id. For menu scheme, the following chapter details the ids of existing group and menu to add new actions or groups in existing menus or groups. For tabbar sheme, only Select menu can be completed with new groups. Note: If your VSM is in the runtime environment workspace to do dynamic development of your VSM, the changes are not systematically reflected in the menu. A selection change may be necessary to rebuild the menu. This chapter lists the ids of existing menus and groups in a Sirius diagram representation. This tool can be used to define an operation which will appear as an entry in the context menu or in a sub-menu if defined inside a Popup Menu element. When invoked, the body of the action is executed with the views variable referencing all the representation elements which were selected on the diagram. The External Java Action tool simply creates a menu entry in the context menu which directly invokes any External Java Action you can define. This tool is for advanced uses only. It requires understanding and extending GMF, the graphical framework used by Sirius to display diagrams. A Request description tool will contribute an item in the palette, and when invoked on a graphical element it will send a GMF request to the corresponding edit part. The type of the request to send is specified as a fixed string in the Type property. The target edit part should have an edit policy capable of understanding and reacting to the request. Document it if we keep it. A diagram can define any number of filters, which can be enabled or disabled dynamically by the user to show or hide some elements of a diagram. Filters differ from layers on three main points:. To add a filter to a diagram, create a Composite Filter inside the Diagram Description element, specify its Name a fixed string , and add one or several Mapping Filter or Variable Filter elements inside of it. A mapping filter applies to all the elements instance of some mappings, and can filter some of them according to configurable conditions. The collapse mode is useful because the edges which use the filtered element as a source or target are still visible when the element is collapsed, while they disappear when the element is hidden. The Mappings property indicates the list of mappings to which the filter will apply. The Semantic Condition Expression will be evaluated in the context of the target semantic elements of each instances of the specified mapping. The View Condition Expression is similar to the Semantic Condition Expression , but is evaluated on the context of the view element of each instance of the specified mapping. The view element is the model element used by Sirius to represent the graphical element. Using these expressions require some knowledge of the structure of the view model used by Sirius. Both expressions are optional. If none is specified, the filter will have no effect. The expressions of all active filters are evaluated whenever the diagram is refreshed in the default Auto-refresh mode, whenever a change occurs in the semantic model , not just when the user enables the filter. This means that the active filters can have a big impact on performance if they perform complex tests, so keep this in mind when writing the condition expressions. A Variable Filter is similar to a Mapping Filter , except that instead of using a list of mappings to select the candidate elements to filter, all represented elements are candidates but can be filtered with a Semantic Condition Expression. In that expression, the specifier can use variables that are defined in the variable filter. A Typed Variable is a variable which value type is a EDataType as chosen by the specifier and which value is filled by the user when applying the variable filter. An interpreted expression allows defining the default value. When the user applies a variable filter with typed variables, a dialog box is displayed to allow user entering the values. This dialog box is prompted after all Select Model Element wizard have been prompted. It contains as many line as number of typed variables under the variable filter. The user documentation is displayed as tooltip on question mark. The default value is the result of the interpreted expression defined on VSM. In order that the EDataType class is instantiable, the value given by the user should follow rules which

depend on how it is triggered by EMF mechanism. At instantiation, the string argument will be the string input by the user. This string argument is translated into the right data in the class constructor for example. What is done in the constructor described above can also be done overloading the EDataType class EFactory. Example: If your class represents a point coordinates with two fields x and y of type int, the string input by the user could be „[integer]. This kind of information could be given to the user through the User documentation. A diagram description can define custom validation rules which will only be applied when launching a validation on an instance of this diagram using the Validate operation in the context menu. To define validation rules you must first create a Validation element inside the diagram, and then add one or more Semantic Validation Rule or View Validation Rule. Both kinds of rules are similar, but semantic rules check the structure of the underlying semantic model which view validation rules can check the structure of the representation itself. When a rule is violated, a marker will appear on the diagram on the problematic elements and in the Problems view. Each rule has a Level , which represents its severity when it is violated. It can be Information , Warning or Error. It also has a computed Message expression evaluated in the context of the checked element to provide the user an explanation of the problem detected. Semantic validation rules must specify Target Class , and will be checked on all the instances of that class in the semantic model. View validation rules have instead of a list of Targets which are the mappings it applies to. Inside a rule you must define one or more Audit , which perform the actual checks using an Audit Expression. The expression will be evaluated on each semantic or graphical element to check, and must return false if the rule is violated. If a rule contains several audits, it is triggered if at least one of them detects an issue. You may also define one or more Fixes inside a rule. When a rule is violated, if it defines fixes they will be available to the end-users in the context menu of the corresponding marker. Each fix has a name and a body. The body is defined in the same way as the body of a tool, using any of the available model operations. It should perform any required change in the model to fix the issue detected by the validation rule, if possible. Sirius has specific support for extending and refining diagram descriptions and mappings without modifying the original. The extensions are applied transparently when the viewpoint they are defined in are enabled, and removed when the viewpoint is disabled. For example, if a base viewpoint V1 defines a diagram description D , you can define a separate viewpoint V2 maybe even from inside a different plug-in, you do not need access to the source of V1 and in V2 define an extension DE to D. When users enable only V1 , they see D as defined in V1. Note that a given diagram type can be extended by multiple extensions at the same time. However if several extensions try to redefine elements defined in the base diagram instead of just adding new elements , the result is not specified. A Diagram Extension is defined directly inside a Viewpoint element, and it must specify precisely which diagram description it extends using the Viewpoint URI and Representation Name properties the Name property is the name of the extension itself. The Viewpoint URI indicates in which viewpoint the extended diagram is defined. Note that the name must be unique among all the viewpoints defined in all the VSMs contributed by that plug-in. The Representation Name is simply the identifier Id field of the diagram to extend, which is defined inside the specified viewpoint. A Diagram Extension can specify the Metamodels of the semantic elements it represents if it supports types of elements which were not represented in the extended diagram. Inside a Diagram Extension , you can define validation rules , filters and additional layers. When the extension is enabled i. The simplest use case of a diagram extension is to simply augment the base diagram with new graphical elements, tools, filters and validation rules. Mapping Imports are actually independent of diagram extensions, and can be used inside a normal diagram description as their semantic is defined in terms of layers activation, but their main use case is when using diagram extensions. This allows to extend several diagrams with one single Diagram Extension. You should use regular expressions only to define a Diagram Extension that contains only style customization. A mapping import is used to specialize modify a mapping defined somewhere else in an other layer, an other graphical representation or an other viewpoint. This feature is only available for diagram kind representations. When only L1 is activated the normal definition of M1 is applied. If L2 is also activated, M2 is applied instead of M1 , the overloaded features in M2 replace the ones of M1 and the new defined features or sub-elements are added. They all behave in the same way. For each one you must specify the Imported Mapping typically from another layer defined in a diagram you are extending or in the same diagram. Mapping imports have all the properties of normal mappings, but when you set an explicit value for a property in a mapping import: Diagram elements support conditional styles, which can be used to completely change the appearance of an element depending on some arbitrary condition. A more fine grained customization of styles is possible using Style Customization elements. To use this feature, you must first create a Style Customizations

element inside a layer from the New Customization Only one such element can exist in a layer, and it serves as a container for all style customizations the layer defines. Inside a Style Customizations you can define any number of individual Style Customization elements, each of which re-groups a set of actual customizations which can be either an Property Customization by expression or a Property Customization by selection. Both kinds work on the same principle, and you must define:. A Style Customization can

bag of air ebay
2002 nissan sentra fuel pump
side impact sensor

define a predicate expression to control whether or not it should apply. If defined, the expression is evaluated in the context of a semantic element to determine whether or not the customizations should be applied. If no expression is defined, the customizations will apply. If a Style Customization applies to a given element, the base style that the element would normally have will be modified by applying all the Property Customization by expression and by reference defined in the Style Customization , if they are compatible with that base style. These can be used to reuse customizations defined elsewhere in a different context specified using the Reuse property , and apply them to other elements specified using the Applied On property. They will behave as if they had been defined locally inside a Style Customization with no precondition. To customize a property of type Enumerator, the value expression must return an instance of Enumerator. Here is some examples of value expression:.