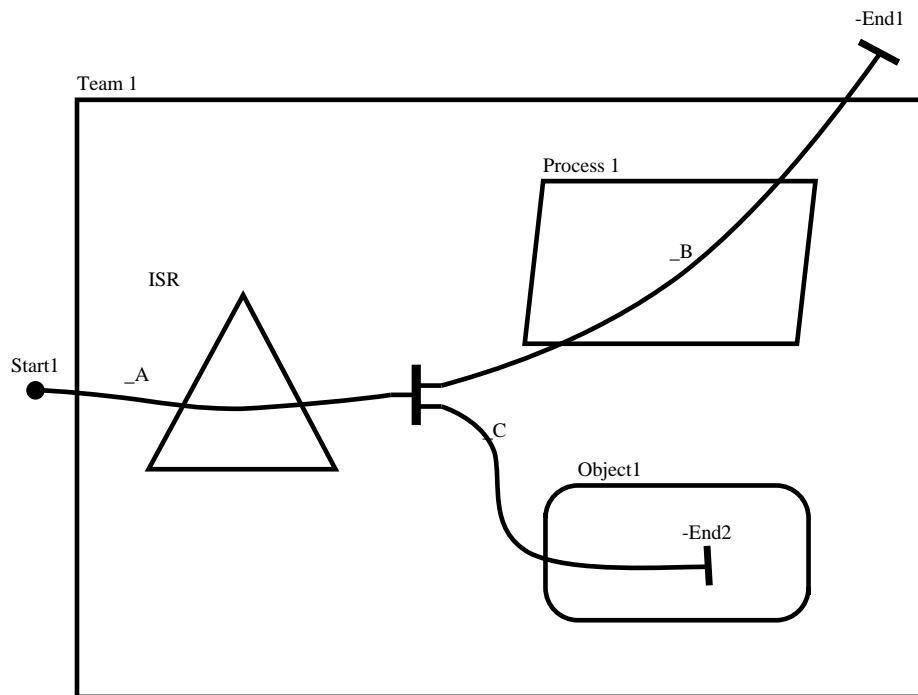


Use Case Map Navigator

version 1.10

User's Manual



Department of Systems & Computer Engineering
Carleton University, June 1999

Table of Contents

Table of Contents

I - UCM Navigator — User's Manual

1.1 Introduction	1
1.2 Parts of the UCM Navigator Main Screen	2
1.3 Navigating the Workspace	3
1.4 Creating a Component	3
1.5 Creating a Path	3
1.6 Selecting a Component or Path Element	3
1.7 Moving or Resizing a Path Element or Component	4
1.8 Accessing the Transformations Menu of a Component or a Path Element	4
1.9 Saving and Loading UCMs	4
1.10 Printing UCMs	4
1.11 Extensions for Performance Prediction	5
1.12 Extensions for Goal Modelling	5

II - Tutorial

2.1 Single-Point Transformations	6
2.2 Double-Point Transformations	7

III - Reference Section

3.1 Menu Bar Commands	8
<i>File Menu</i>	8
<i>Components Menu</i>	9
<i>Options Menu</i>	9
<i>Performance Menu</i>	10
<i>Maps Menu</i>	11
<i>About Menu</i>	11
3.2 Preconditions, Postconditions, Triggers and Resulting Events Windows	11
3.3 Workspace Mouse Functions	12
3.4 Shortcut Keys	13
3.5 Path Labels	13
3.6 Attributes and Single Element Transformations	14
<i>Components</i>	14
<i>Start Points</i>	15
<i>Empty Points</i>	16
<i>Waiting Places</i>	19
<i>Stubs</i>	19
<i>Timestamp Points</i>	20
<i>Goal Tags</i>	21

<i>End Points</i>	21
<i>AND Joins/Forks</i>	21
<i>OR Joins/Forks</i>	21
3.7 Double-Element Transformations	22
<i>Empty Point - Starting Point</i>	22
<i>Empty Point - End Point</i>	22
<i>Starting Point - End Point</i>	22
<i>End Point - Stub</i>	22
<i>Starting Point - Stub</i>	22
<i>Timer - End Point</i>	22
<i>Waiting Place - End Point</i>	22
<i>Waiting Place - Empty Point</i>	22
<i>Timer - Empty Point</i>	23
<i>AND join/fork - End Point</i>	23
<i>AND Join - Empty Point</i>	23
<i>Timestamp Point - Timestamp Point</i>	23
<i>Goal Tag - Goal Tag</i>	23
<i>End Point - End Point</i>	24

List of References

List of Figures

Glossary

I - UCM Navigator — User's Manual

Version: 1.10, dated June 2, 1999

1.1 Introduction

The Use Case Map (UCM) Navigator is a tool for creating and modifying UCMs. The UCM Navigator tool offers the following features:

- Creation and manipulation of Use Case Maps that are always syntactically correct.
- Path transformations and path connections based on internal hypergraph-based semantics.
- Nested levels of stubs and plug-ins (sub-maps), with explicit binding of plug-ins to stubs.
- Exporting and importing of root and plug-in maps to/from files.
- Separation of responsibility definitions from inclusion along paths, allowing reuse.
- Binding of path elements to Buhr's architectural notation:
 - Component attributes (type, formal, anchored, fixed, replicated, color, etc.) are automatically reflected visually.
 - Components can be nested inside other components (warnings can be displayed for invalid bindings).
 - Pools and information on dynamic responsibilities are supported.
 - Moving a component moves all its sub-components, elements, and path segments.
 - Resizing a component can automatically reshape the paths bound to it.
- Support of extensions for agent systems and performance modeling.
- Comment fields for almost any part of a map (path elements, components, goals, etc.).
- Pre/post conditions which can be attached to many path elements.
- XForms-based GUI with scalable maps, large drawing area, zoom, and scroll bars.
- Generation of XML files valid with respect to the UCM Document Type Definition.
- Flexible report generation in PostScript:
 - Selection of sections to include: map description, responsibilities, path elements, components, stub descriptions, and goals.
 - Ready for PDF generation: index, map stubs, and plug-in names are hyperlinks. (Very useful for on-line presentations.)
 - Option for maps on separate pages.
- Exports maps in Encapsulated PostScript (EPS) and Maker Interchange Format (MIF)
- Easy-to-install binaries available for three platforms: Solaris, Linux (Intel and SPARC), and HP/UX

The UCM Navigator is designed to handle any valid UCM as well as software components. It is capable of binding these UCM elements to components. UCM Navigator is also capable of creating multi-level maps in which submaps of a lower level are expressed as stubs in a higher-level map. The editor currently supports nested levels up to 10 deep. UCM Navigator generates industry-standard XML files which can be used as input to other tools, including Agent generation tools and performance simulators.

1.2 Parts of the UCM Navigator Main Screen

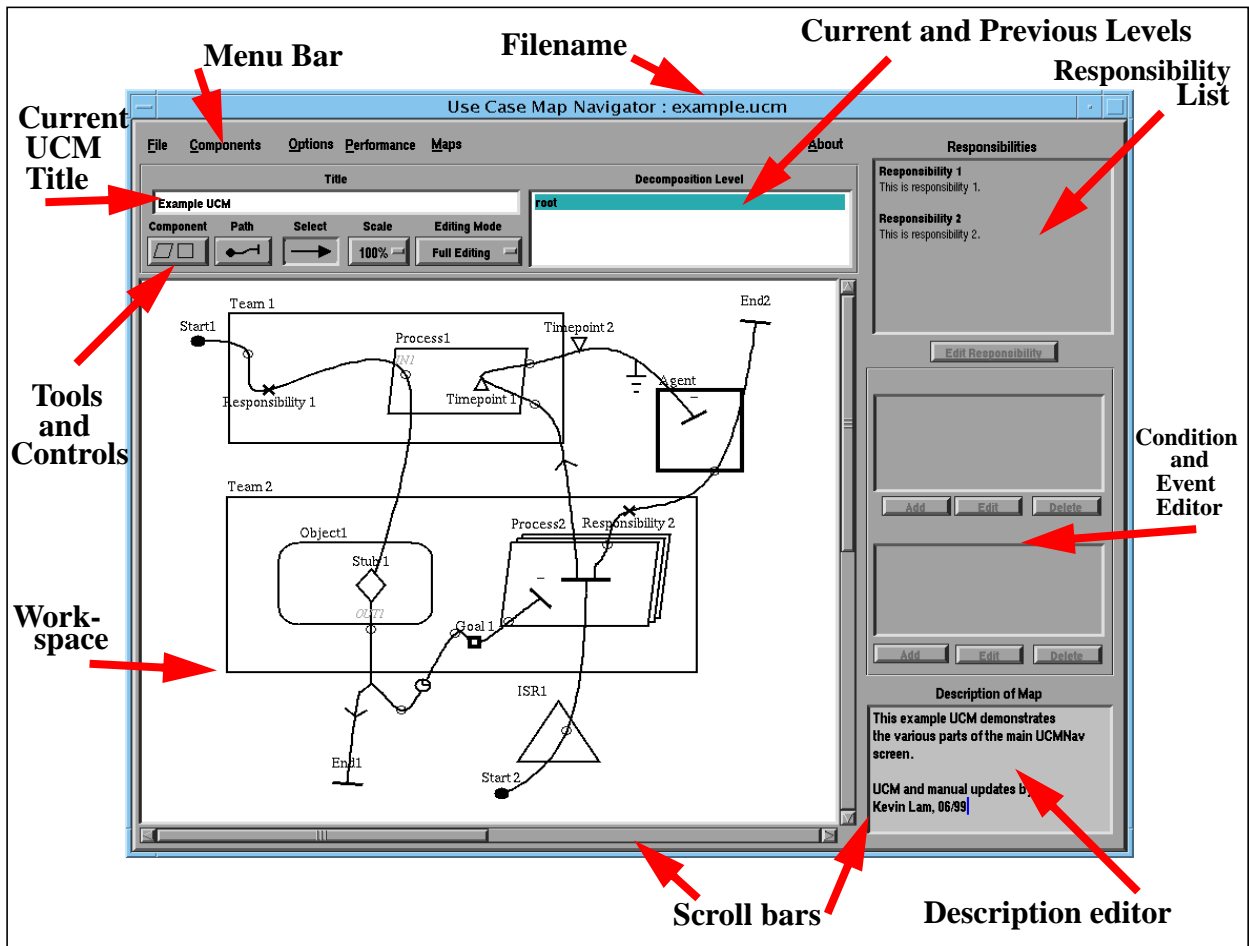


Figure 1. The UCM Navigator Main Screen.

Description of the UCM Navigator main screen:

- **Menu Bar:** This is where most of the UCM Navigator commands are found. See the Reference section for detailed information on individual menu commands.
- **Workspace:** This is where UCMs are drawn. The vertical and horizontal scroll bars allow the user to scroll to different areas, and resizing the UCM Navigator window changes the size of the visible workspace area.
- **Tools and Controls:** Select one of three tools: Component, Path, Select. Controls are also available to change the current scale and editing mode.
- **Current UCM Title:** The title of the current UCM. The current UCM filename is also displayed in the application title bar, with a * beside it to indicate unsaved changes.
- **Current and Previous Levels:** This shows the level being shown, and all levels below it.
- **Description:** Shows the description of the currently selected object or map. All objects, including components, path elements, and maps, can have a description.
- **Conditions and Events:** These boxes show conditions and/or events for the currently selected path element. See **Preconditions, Postconditions, Triggers and Resulting Events** in the Reference section for more information.
- **Responsibility List:** A listing of all responsibilities in the UCM.

1.3 Navigating the Workspace

The UCM Navigator allows UCMs to be constructed anywhere in the workspace. The virtual area is several times larger than what is visible on the screen at any given time using 100% magnification. When the workspace cannot be shown in its entirety, scroll bars appear on the right and bottom sides. Scrolling with these bars shows different areas of the workspace. To show more or less of the workspace at once, click on the Scale button and choose a magnification level. Several useful magnification levels are provided. “Fit Map” automatically calculates a zoom level to fit all of the current UCM on screen. “Fit Virtual Area” zooms out to show the entire workspace on the screen at once. Resizing the UCM Navigator window also changes the amount of the workspace visible at any given time.

1.4 Creating a Component

To create a component, first click the Component button to select the *Component* tool. Then click and drag the left button in the workspace to anchor one corner of the component. Proceed to drag the mouse to the desired location of the opposite corner. By default, new components are created as teams, but they can be easily changed by modifying the attributes of each component as necessary (see section 1.7 for more details).

1.5 Creating a Path

To create a path, select the *Path* tool by clicking on the Path button with either the left or right mouse button. Right-clicking the Path button will always start a new path; left-clicking will start a new path if no path is currently selected, or continue the currently selected path if one exists.

To begin, left-click once in the workspace to place the start point of the path. Further clicks in the work area will extend the current path, placing an empty point at the location of each click and connecting the new point to the last point. If a new point falls within the boundaries of a component, it is automatically bound to that component.

To begin a new path, first de-select the current path by clicking the *Select* button and clicking on a blank area of the workspace, or right-click on the Path button. Continue as noted above.

1.6 Selecting a Component or Path Element

To select a component or a path element (including empty points, start points and end points), click on the *Select* tool and left-click once inside the boundary of the desired component or element. To select two path elements at a time (a requirement for some double-selection transformations; see section 3.6 for more details), select the first element of interest and then hold down the Shift key while clicking the second.

1.7 Moving or Resizing a Path Element or Component

To move or resize any component or path element, the Editing Mode must be set to Full Editing by clicking on the Editing Mode tool.

To move a path element or component, first select it and then drag it to the new location. (When moving a path element, any other path elements bound to it will also move; similarly, moving a component by default moves all enclosed components and path elements bound to it.)

To resize a component, first select it. Handles will appear around the perimeter of the component and at each corner of rectangular components. Left-click and drag any handle to resize the component in that direction.

1.8 Accessing the Transformations Menu of a Component or a Path Element

In the UCM Navigator, each component and path element has its own set of attributes and transformations. To access these, first select the component or path element, then right-click to bring up the pop-up menu of available attributes and transformations. For transformations that involve two items, select both applicable items and then right-click. A detailed listing of all attributes and transformations is provided in the Reference section.

1.9 Saving and Loading UCMs

The File menu contains options for loading UCMs from disk. Note that loading a UCM into memory will overwrite any existing UCMs. If there are unsaved changes, the UCM Navigator will present a dialog box asking if the current map should be saved.

UCMs may be saved to disk using either the Save or Save As commands in the File menu. The files are saved by default with a .ucm extension and are stored in XML format. Options are also available to export the active UCM to .mif files (Maker Interchange Format, for import into FrameMaker documents) or .eps graphics (Encapsulated Postscript, for use with many publishing and graphics packages.)

1.10 Printing UCMs

UCMs may be printed by accessing the Print command in the File menu. UCMs may be printed to an installed printer or to a PostScript (.ps) file. When printing, options exist to specify what additional elements should be included in the report, including the current date and time, descriptions of elements, etc. In addition, the PostScript output is PDF-enabled (i.e. hyperlinks on UCM stubs and plug-in names are automatically created, and an index is automatically generated.)

1.11 Extensions for Performance Prediction

The UCM Navigator supports extended UCM notation, with timestamps, device accesses and other performance features. Timestamps are covered in Section 3.6 (**Attributes and Single Element Transformations**). Response Time Requirements are covered in Section 3.7 (**Double-Element Transformations**).

1.12 Extensions for Goal Modelling

The UCM Navigator supports goal modelling along paths. Goal tags can be created at any points along the path, and pairs of goal tags can be coupled to create goals. Goal tags are covered in more detail in Section 3.6 (**Attributes and Single Element Transformations**), and goals are covered in Section 3.7 (**Double-Element Transformations**).

II - Tutorial

2.1 Single-Point Transformations

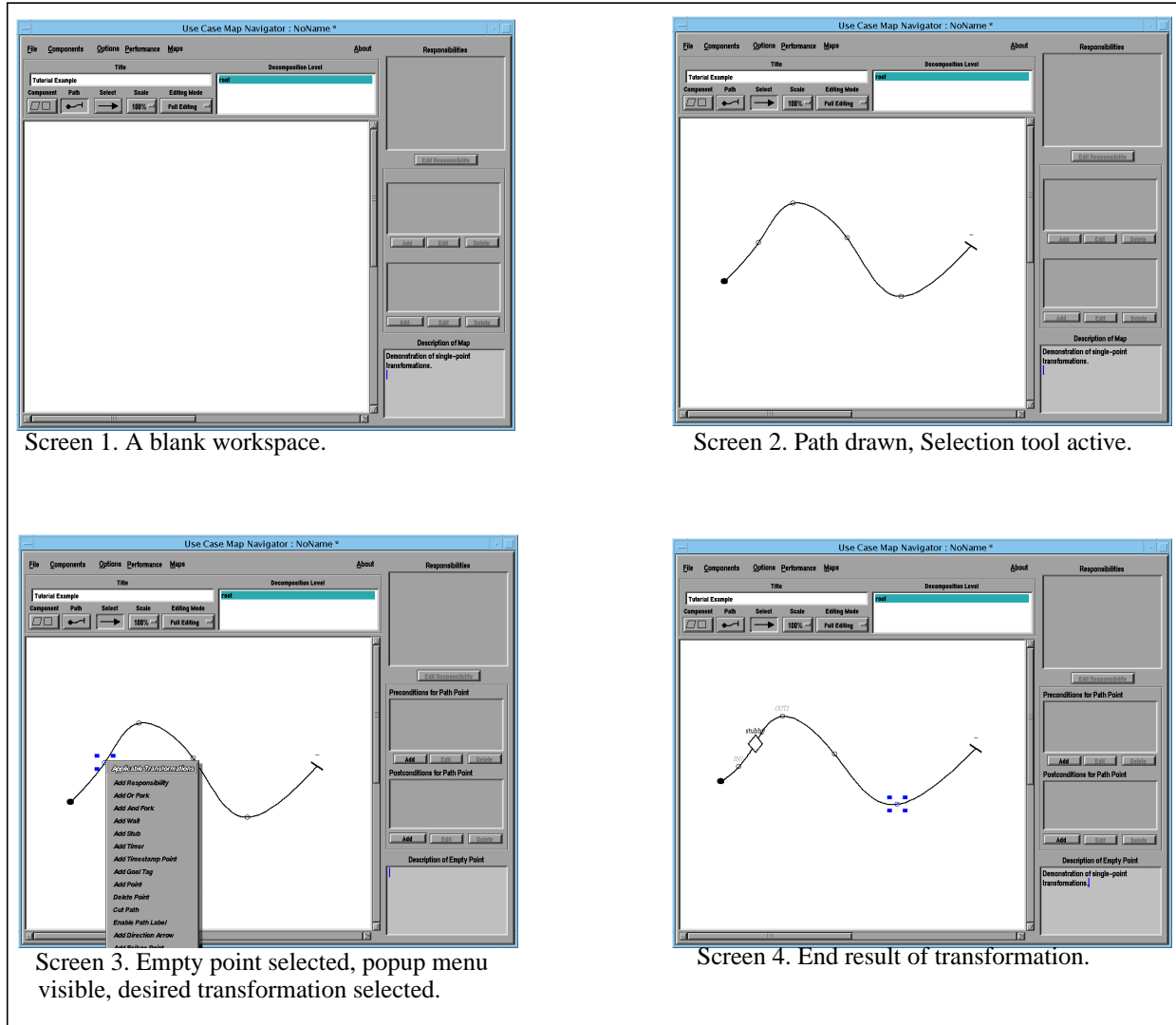
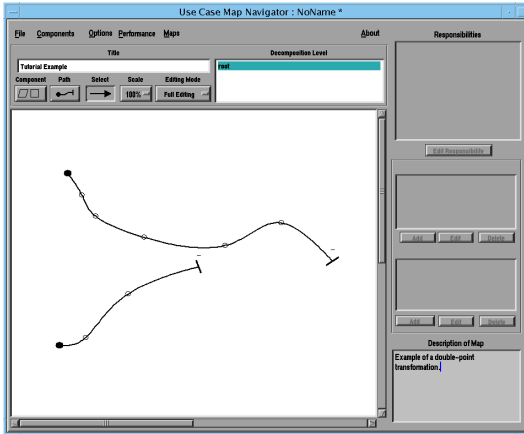


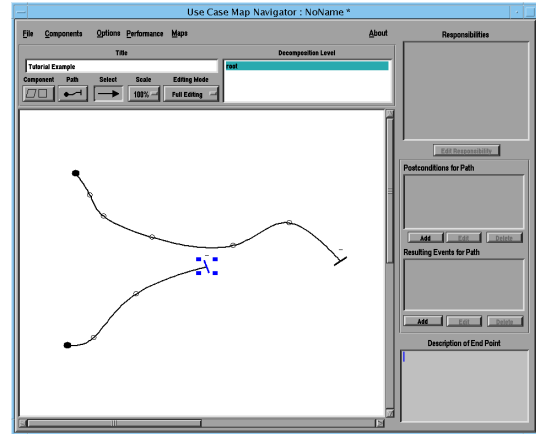
Figure 2. Single point transformation.

This is an example of a simple single-point transformation. In this example, a path is drawn from scratch and one point on that path is transformed into a stub. First, the path is drawn, using the Path tool. Then the desired empty point is selected using the Selection tool. The transformation menu is invoked (by right- or middle-clicking), and the transformation *Add Stub* is chosen.

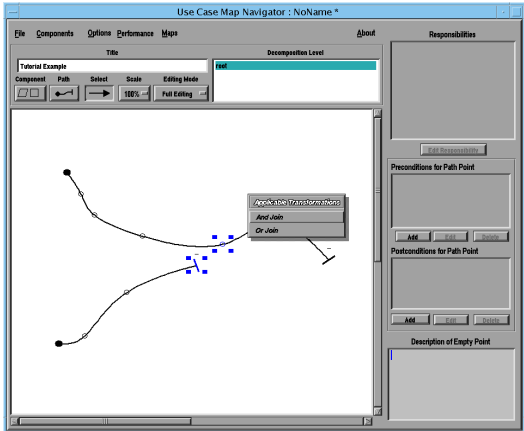
2.2 Double-Point Transformations



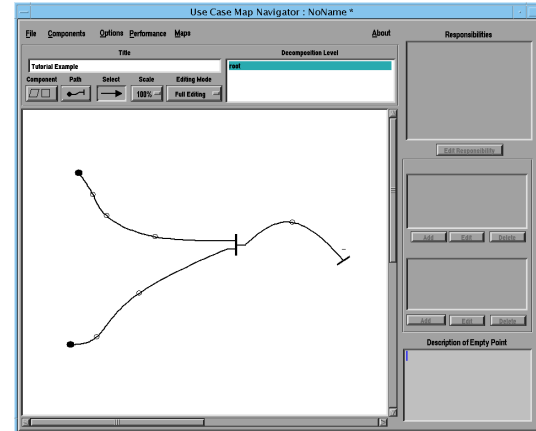
Screen 1. Two paths drawn, Selection tool active.



Screen 2. First point selected.



Screen 3. Second point selected, popup menu visible, desired transformation highlighted.



Screen 4. End result of transformation.

Figure 3. Double point transformation.

This is an example of a double-point transformation. In this example, two existing paths will be synchronized by an AND join. At first, the two paths are visible and separate, and the Selection tool is active. The first point is selected by clicking on it, then the end point on the second path is also selected by holding Shift and clicking on it. Next, the popup menu is activated (right- or middle-click) and the *AND join* transformation is chosen.

III - Reference Section

3.1 Menu Bar Commands

File Menu

- *New*: Clears the workspace and starts a new UCM. If the current UCM has been modified but not saved, the user will be prompted to save the file before continuing.
- *Load*: Loads a saved UCM from disk. If the current UCM has been modified but not saved, the user will be prompted to save the file before continuing.
- *Save*: Saves the current UCM to a file with a *.ucm* file extension. The file is saved in XML format.
- *Save As*: Saves the current UCM under a different name.
- *Print*: Prints the workspace to a printer or to a file. A dialog box is presented, allowing a selection of a number of useful print options:

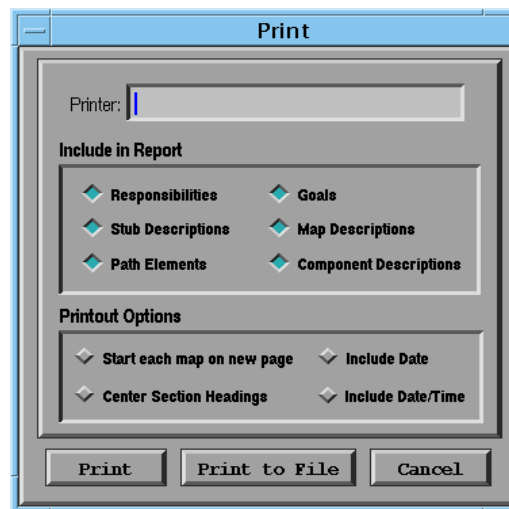


Figure 4. Print dialog box

The Print dialog box allows printing to a printer or to a Postscript (.ps) file. In addition, a number of reports (such as detailed descriptions of path elements) can be automatically generated from the Description boxes in the workspace and included with the printout. These options are selectable from the Print dialog box.

- *Export Current Map*: Save the UCM in the current workspace and all of its submaps as a separate *.ucm* file.
- *Import Root Map*: Imports a UCM from a file into the current design as the root map.
- *Create MIF File*: Exports all the maps in the current design to *.MIF* format (Maker Interchange Format) for inclusion into FrameMaker documents.

- *Create EPS Graphics*: Exports the all the maps in the current design to .EPS (Encapsulated Postscript) graphics files for inclusion into publishing programs. Filenames are automatically generated from the UCM file name.
- *Specify Design Name*: Allows the user to specify the name of the current design. This is used when printing and is also stored in internal structure of the UCM file.
- *Quit*: Quits UCM Navigator.

Components Menu

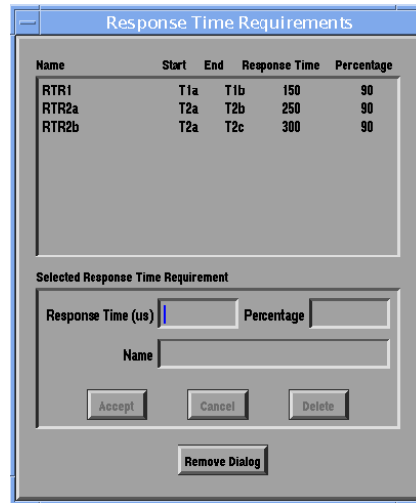
- *Decouple Resizing*: When turned off, resizing a component resizes and repositions all components within it to keep them proportional to the parent components. When turned on, resizing a component has no effect on the placement or size of any components within it. This option is off by default.
- *Fix All Components*: When turned on, all components are fixed such that they cannot be resized or moved in the workspace. Individual components can be toggled by the *Fixed* option in the Component Attributes dialog.
- *Cut*: Places the selected component on the clipboard and deletes it from the workspace.
- *Copy*: Copies the selected component to the clipboard but does not delete it.
- *Paste*: Pastes a component from the clipboard into the workspace.

Options Menu

- *Show Data Points*: Toggles whether the circular handles around empty points are visible.
- *Show Path Labels*: Toggles whether a path's name appears over every empty point or not. (See the **Path Labels** section for more information.)
- *Show Enclosing Warnings*: If turned on, warnings will appear for theoretically invalid component containments (e.g. placing a team inside a pool, placing a process inside an object).
- *Show Unavailable Transformations*: When this option is off, transformations which are not applicable to the current selection are not shown in the Transformations popup menu. When this option is turned on, unavailable transformations are shown but “grayed out” to indicate that they are not available.
- *Show File Pathname*: Toggles the display of complete directory paths when showing the current UCM filename in the title bar.
- *Large Selection Radius*: Toggles the use of large- or small-radius selection handles when selecting a path point. Large handles allow easier selection of small objects on-screen, but they can make it difficult to select objects which are closely spaced together.
- *Show Stub I/O*: Toggles the display of “IN” and “OUT” labels on path points immediately before and after a stub in the path.

Performance Menu

- *View Response Time Requirements*: Displays a dialog with all time requirements in it.



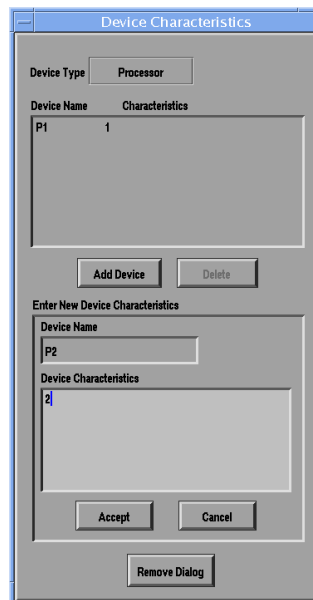
The dialog box titled "Response Time Requirements" contains a table with the following data:

Name	Start	End	Response Time	Percentage
RTR1	T1a	T1b	150	90
RTR2a	T2a	T2b	250	90
RTR2b	T2a	T2c	300	90

Below the table is a section titled "Selected Response Time Requirement" with input fields for "Response Time (us)" (containing 150), "Percentage" (containing 90), and "Name" (containing RTR1). At the bottom are buttons for "Accept", "Cancel", "Delete", and "Remove Dialog".

Figure 5. Response time requirements dialog box

- *Edit Device Characteristics*: Brings up a dialog with all devices and their characteristics. Available device types are: *Processor*, *Disk*, *DSP*, and *Other*. Device characteristics can be referenced in Responsibilities to specify service requests.



The dialog box titled "Device Characteristics" shows a "Device Type" dropdown set to "Processor". Below is a table with the following data:

Device Name	Characteristics
P1	1

Buttons for "Add Device" and "Delete" are below the table. A section titled "Enter New Device Characteristics" contains input fields for "Device Name" (containing P2) and "Device Characteristics" (containing 2). At the bottom are buttons for "Accept", "Cancel", and "Remove Dialog".

Figure 6. Device Characteristics dialog box

- *Edit Data Store Characteristics*: Allows editing of all data stores and their attributes. Available displays are *Data Stores* and *Access Modes*. Data stores can be referenced by responsibility points.

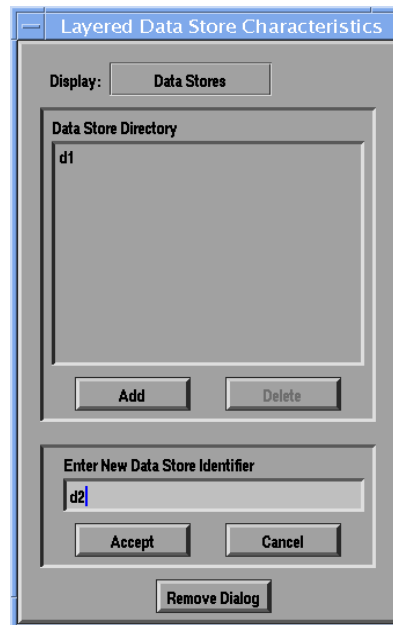


Figure 7. Data Store Characteristics dialog box

Maps Menu

- *Choose Root Map*: When more than one root map is defined in a UCM, this command presents a list of them and allows the user to select which one to work with.
- *Create New Root Map*: Creates a new empty root map for editing. All root maps are still available for editing and are stored in the same UCM file.

About Menu

- *Help*: Displays a box containing a help summary and keyboard shortcut reference.
- *Version*: Displays a dialog box with the current program version and release date.

3.2 Preconditions, Postconditions, Triggers and Resulting Events Windows

To the right of the workspace, a large pane of windows shows the conditions and/or events that are applicable to the currently selected path element. The window also shows a list of all the responsibilities defined in the current map. When the selected path element is a start point, wait point, or timer, it shows Preconditions and Triggering Events for that point; for a selected empty point, or stub, it shows Preconditions and Postconditions; for an end point, it shows Postconditions and Resulting Events. These settings are only used for reference and performance analysis.

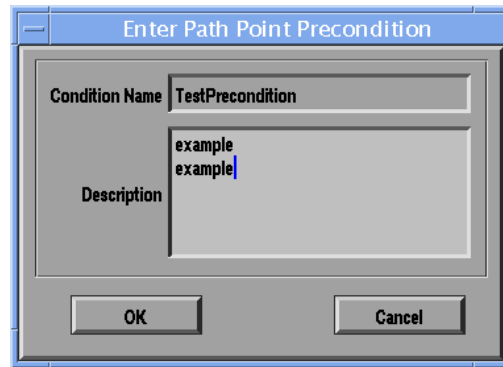


Figure 8. Add/Edit Condition/Events dialog box

Conditions may be added to any of these windows by using the Add button. This will bring up a dialog box, as shown above, to enter descriptions of the condition. Conditions may be similarly edited.

3.3 Workspace Mouse Functions

- **Left-click:** Under the *Path* tool, continues the currently selected path by placing an Empty Point at the mouse position. If no path is selected, a Start Point is placed at the mouse position and a new path is created. Under the *Selection* tool, it selects whatever component or path element is under the cursor. Under the *Component* tool, creates a small team object at the location of the mouse click.
- **Left Double-click:** Only used with the *Selection* tool. Double-clicking on a component brings up its attributes; double-clicking on a stub will bring up the stub's current plug-in or a list of available plug-ins; double-clicking on a point with a responsibility brings up that responsibility in the Add/Edit Responsibility Dialog box.
- **Left Drag:** When used under the *Component* tool, the upper left corner of a new component is anchored where the cursor was located when the button was first clicked, and the bottom right-hand corner of the component follows the cursor until the button is released. When used with the *Selection* tool, dragging any component or path element moves it to a new position in the workspace. Dragging one of the handles found around a selected component will resize the component in that plane.
- **Right-click or Middle-click:** Used only with the *Selection* tool. Brings up the list of transformations and attributes for the selected component or path element(s). Currently unavailable transformations / attributes are grayed out or not shown at all, depending on the *Show Unavailable Transformations* menu setting.

3.4 Shortcut Keys

Shortcut keys are keystrokes that execute equivalent mouse or menu commands. They are designed to make using UCM Navigator easier and more efficient. Shortcut keys are not case-sensitive.

Editing

- **C**: Selects the *Component* tool. Same effect as clicking on the *Component* button (located below the workspace).
- **P**: Selects the *Path* tool. Same effect as left-clicking on the *Path* button.
- **S**: Selects the *Selection* tool. Same effect as clicking on the *Select* button.
- **L**: Moves the last drawn component into the background one layer (allows components in behind it to become visible).
- **U**: Moves from a sublevel back up to a parent level.
- **Del**: Deletes the currently selected component.

Workspace Zooming and Scrolling

- **+**: Zoom in to the workspace
- **-**: Zoom out from the workspace
- **V**: Fit the whole virtual area into the workspace
- **=**: Fit the current map into the workspace
- **arrow keys** (UP, DOWN, LEFT, RIGHT): scrolls the workspace in the direction of the key pressed.

Element Fixing

- **E**: Switch to Full Editing mode (components and paths can be moved and resized)
- **F**: Fix Components (only paths can be moved and edited)
- **A**: Fix All Positions (paths and components may not be moved or resized)

3.5 Path Labels

Path labels are a useful way of distinguishing between paths, but their use can be somewhat confusing. The following is a description of the properties and effects of path labels in the UCM Navigator.

Path labels are only one character long, except in the case that the path is split or joined by either an AND join/fork, OR join/fork, or a stub. In that case, the path before the fork or after the join can be named as an addition of the two branches. (e.g. If the two branches out of a fork are named **A** and **B**, then the path before the fork can be named **A+B**.) Each point in a path segment inherits the path name from all the other points in the segment. Changing the path name at any point affects all other points in the path segment.

Path names are visible either through the menu option *Options / Show Path Labels* or by right-clicking on a path element and choosing the *Enable Path Label* option. *Show Path Labels* makes all points in the workspace display their path name, while *Enable Path Label* only affects the selected point. Path names can be set by selecting an empty point and choosing the *Edit Path Label* option from the Transformation menu.

All path names must be set manually if *Show Path Labels* is not turned on. However, when *Show Path Labels* is enabled, the UCM Navigator prompts for a pathname every time a new path is created. Whenever a fork is created, UCM Navigator asks for the names of all the new branches and also for the path leading into the fork. It also asks for the new merged path's name after a join is created.

3.6 Attributes and Single Element Transformations

Components

All components have identical options:

- *Attributes*: edits the attributes of the component:

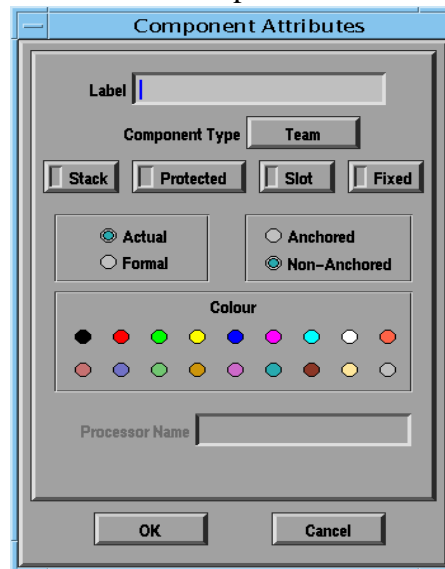


Figure 9. Component Attributes dialog box

- **Label**: The name appearing beside the Component.
- **Component Type**: The type of component - Team, Object, Process, ISR, Pool, or Agent.
- **Stack**: Controls whether the component appears as a single component or as a stack of identical components.
- **Protected**: If toggled, the component is protected by mutex and has a double-outline.
- **Slot**: Controls if the component is static (solid outline) or dynamic (dashed outline).
- **Fixed**: If toggled, the component cannot be resized or moved on the map.

- **Actual/Formal:** Decides whether the component is actual or usable in a formal call.
 - **Anchored/Non-Anchored:** Only used in plug-ins. Controls whether the component is wholly inside the stub (non-anchored) or present in the main map and merely called from inside the stub (anchored).
 - **Colour:** Sets the colour of the component
 - **Processor Name:** For performance analysis only - the name of the processor the component will be dealing with.
 - **Replication Factor:** Only available when the Stack option is set. Sets how many spots are available in the stack, and can be any alphanumeric pattern. The default (on non-numeric data) is 2, but the component will not be drawn with less than 2 or more than 6 (the correct number is saved in the component, however).
 - **Pool Type:** Only available when the component is of the type Pool. Possible pool types are None, Component (pool is a pool of components), and Plug-in (pool is of plug-in maps).
 - **Type:** Only available for components of type Pool, sub-type Component. Specifies the type of component contained within the pool.
 - **Choose Plugins:** Only available for components of type Pool, sub-type Plug-in. Controls what plug-ins (of all the plug-ins in the current map) are present in the current pool.
-
- *Cut Component:* Copies the selected component to the clipboard and then deletes it from the workspace.
 - *Copy Component:* Copies the selected component to the clipboard but does not delete it from the workspace.
 - *Paste Component:* Pastes the component on the clipboard into the workspace.
 - *Bind Enclosed Path Elements:* Binds all enclosed path elements into the component. Path elements are automatically bound if dragged or drawn inside a component's boundaries, but are not if the component is moved or drawn so that the path elements are inside its boundaries. Once bound, the elements move with the component as it is resized or dragged.
 - *Unbind Enclosed Path Elements:* When this option is selected, all path elements that fall inside the boundaries of the selected components are no longer bound to the component. Moving or resizing the component will not affect the paths.
 - *Bind Enclosed Components:* Similar to enclosing path elements, this command forces all sub-components that fall inside the boundaries of the selected component to be moved or resized along with the selected component. Components that are drawn completely within the boundaries of an existing component are automatically bound.
 - *Unbind Enclosed Components:* Unbinds the selected component from any components that are inside its boundaries. As a result, moving or resizing the selected component will have no affect on the size or placement of any other components that are inside it.

Start Points

- *Edit Waiting Name:* This allows the start point to be given a name.

- *Edit Arrival Characteristics*: Sets the type of distribution that arrivals will follow. Only needed for performance and time testing. Available types of distributions are: exponential, deterministic, uniform, erlang, and expert. Depending on the selected distribution type, the dialog box may present input fields for additional information.
- *Delete Path*: This will delete the currently selected path. Note that this function is only available for independent paths. To delete a path that merges with an AND fork or an OR fork, first cut the path prior to the fork, or use the Decompose From Join command.

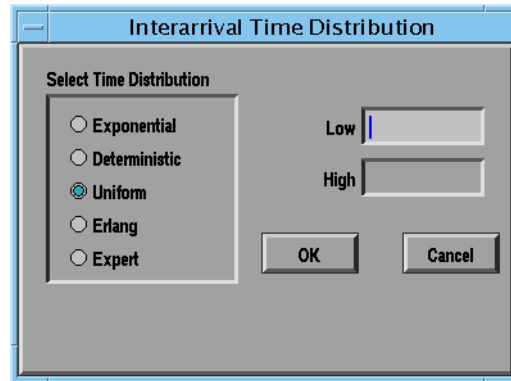


Figure 10. Arrival Characteristics dialog box.

Empty Points

- *Add Responsibility*: Adds a responsibility at the location of the current point. Each responsibility has a title, a description, and an execution sequence. A dynamic pointer can also be added to the point under this menu. Dynamic pointers may point in any of the 4 cardinal directions, and can be pointing into or out of a path. The pointer can also be of 5 different arrow styles, each symbolizing a different action: move, move-stay, create, destroy, or copy. The source pool name identifies the pool where the dynamic components influenced by the responsibility are kept when they are at rest.

A responsibility can contain references to data stores that it accesses. The *Specify Data Stores* button brings up a dialog box (see Figure 13) to select data stores. In a similar fashion, the *Specify Service Requests* button brings up a dialog box that allows the user to specify services (defined in *Device Characteristics*) which the responsibility requests access to.

Responsibility definitions may be added to more than one path point. The Edit Responsibility dialog box contains a list of all pre-existing responsibilities, with the option to select an existing responsibility to use in the current point, or to create a new one.

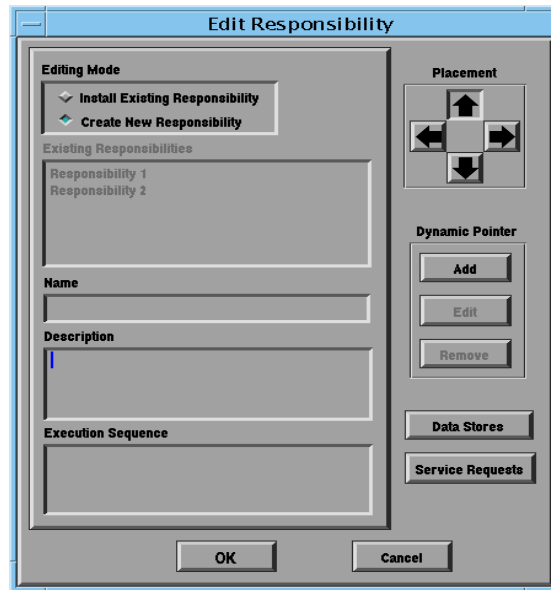


Figure 11. Add/Edit Responsibility Dialog

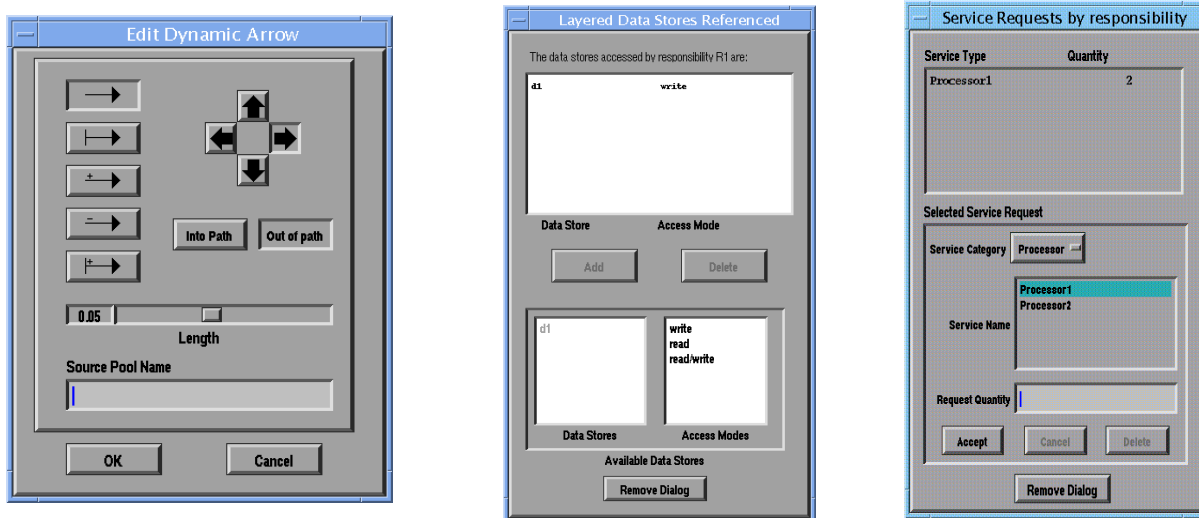


Figure 12. Dynamic Arrow, Data Stores, and Service Requests dialog boxes

- *Add OR Fork*: Splits the current path in two via an OR fork, located at the current point.
- *Add AND Fork*: Splits the current path in two via an AND fork, located at the current point.
- *Add Wait*: Adds a waiting place at the current point.

- *Add Stub*: Adds a static or dynamic stub at the current point.

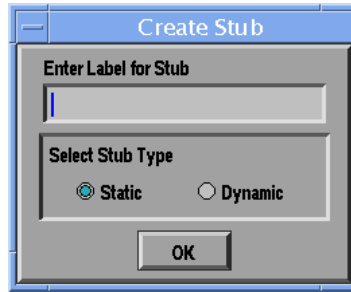


Figure 13. Create Stub dialog box

- *Add Timer*: Adds a timer at the current point.
- *Add Timestamp Point*: Adds a timestamp point at the current point. Response time requirements may be created from two timestamp points (see Section 3.7, **Double-Element Transformations**).
- *Add Goal Tag*: Adds a new goal tag at the current point. Goals may be created from two goal tags (see Section 3.7, **Double-Element Transformations**).
- *Add Point*: Adds another empty point further along the current path, half-way to the next path element. More empty points can be used to place paths with greater accuracy and aesthetics.
- *Delete Point*: Deletes the currently selected empty point.
- *Cut Path*: Cuts the current path into two separate paths at the selected empty point. A start point is created farther along the path, and a end point is created back along the path.
- *Enable Path Label*: Displays the current path label for the selected point only.
- *Edit Path Label*: Edits the current path label at the selected point.
- *Disable Path Label*: Removes the display of the current path label.
- *Add Direction Arrow*: Adds a visual arrow indicating the direction of the path at the selected empty point.
- *Add Failure Point*: Adds a failure point at the current empty point.
- *Add Shared Responsibility*: Adds a shared responsibility at the current point.
- *Decompose from Stub*: Is an option only on empty points immediately before or after a stub. Disconnects the path from the stub, adding a starting point or end point (depending on whether the path was an input to the stub or an output) automatically to the path.
- *Decompose from Join*: Appears only on the empty points immediately before an AND or OR join. Disconnects the current path from the join.
- *Specify Branch Selection*: Found only on empty points on branches immediately following OR forks, it specifies the probability for factors that will result in the current branch being taken. Is only used in performance prediction.
- *Delete Branch*: Deletes the current branch. Note that some branches cannot be deleted without first deleting other branches. This option appears only on empty points immediately following an AND or OR fork whose paths do not have another fork or stub further down. The current path is deleted and removed from the fork. Branches which end in a join can be deleted; this results in splitting the path into two separate paths.
- *Remove Direction Arrow*: Removes the currently selected direction arrow, turning it into an empty point.

Waiting Places

- *Delete Wait*: Deletes the waiting place.
- *Edit Waiting Name*: Changes the name appearing over the waiting place.

Stubs

- *Delete Stub*: Changes the stub back into an empty point.
- *Create New Sub Map*: Asks for the submap name, and then adds a submap of that name under the selected stub. Submaps can be edited in the same manner that maps are.
- *Edit Stub Attributes*: Brings up a dialog box allowing the name given to the stub to be changed, as well as the type. A stub can be changed from Static to Dynamic, and from Dynamic to Static if the dynamic stub has no more than 1 submap installed.
- *View Sub Map*¹: Opens the submap for editing in the main window. This can also be accomplished by double-clicking on the Stub. For dynamic stubs with more than 1 installed submaps, a dialog is popped up asking which submap to view.

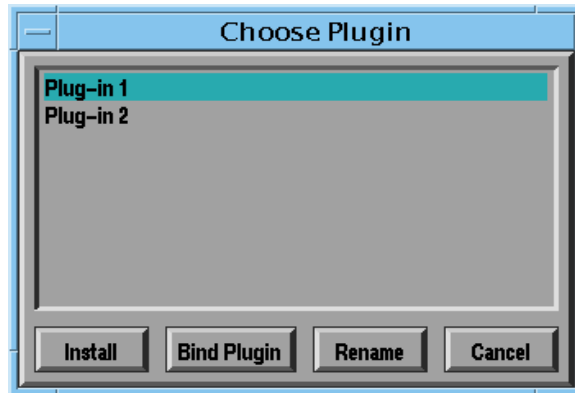


Figure 14. View Sub Map dialog box (multiple submaps)

Install: Opens the Submap in the main workspace.

Bind Plugin: Same effect as *Bind plugin to Stub*.

Rename Plugin: Renames the currently selected Submap.

Cancel: Cancels the *View Sub Map* operation.

- *Remove plugin*¹: Deletes the stub's submap. If there are more than one, this command asks which submap to delete before deleting it.
- *Rename plugin*: Only available for Static stubs. Allows for renaming of the stub's plug-in. To rename a plugin from a Dynamic stub, access the View Sub Map dialog box and select Rename from there.

1. Only available when the selected Stub has a submap.

- *Bind plugin to Stub*¹: Allows binding of Stub input and output to start points and end points in the submap. Requires that connections have unique names to distinguish between them.

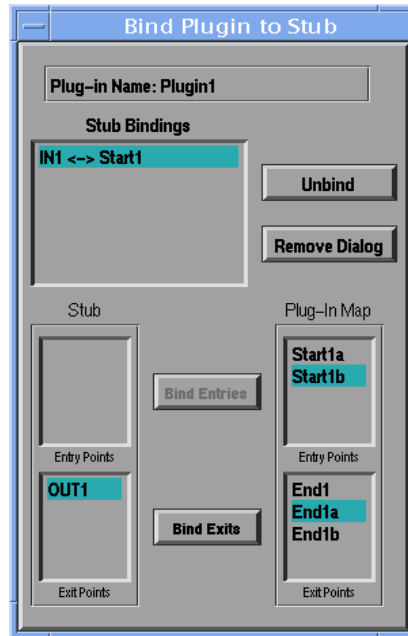


Figure 15. Bind Plugin to Stub dialog box

- *Install Existing Plugin*²: Adds a submap found under another Stub into the selected dynamic stub's list of available plugins. Modifications made to that submap are reflected globally in all Stubs which contain the submap.
- *Image of Existing Plugin*²: Identical to *Install Existing Plugin*, but makes a copy of the submap so any modifications to it are local and do not affect other stubs or submaps.
- *Import Plugin From File*: Installs a submap from a UCM file into the current Stub.

Timestamp Points

- *Delete Timestamp Point*: Deletes the timestamp point.
- *Edit Timestamp Point*: Allows for changing of the timestamp name, where the name appears in relation to the timestamp, and how the timestamp is used in performance evaluation.

1. Only available when the selected Stub has a submap.

2. Only available when 2 or more stubs are present and at least one other than the currently selected stub has an existing plug-in.

- *View Response Times*: Brings up the *Response Time Requirement* dialog box to display all the response time requirements defined within the UCM.

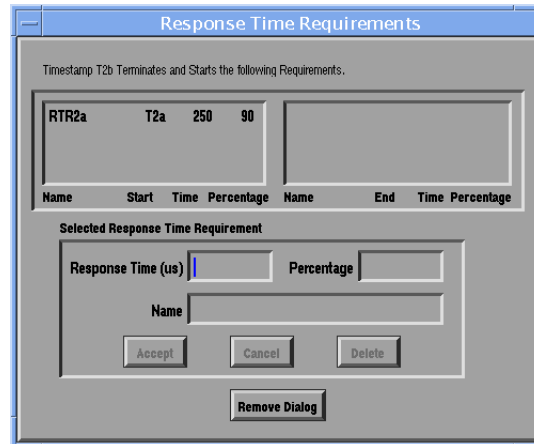


Figure 16. Response Times Requirements dialog box

Goal Tags

- *Rename Goal Tag*: Renames the currently selected goal tag.
- *Delete Goal Tag*: Deletes the currently selected goal tag.

End Points

- *Edit Path End Name*: Changes the name of the end of the path.
- *Flush Label*: Changes display alignment of the end label to flush left.
- *Center Label*: Displays the end label centered above the end point.

AND Joins/Forks

- *Add Branch*: Adds another branch to the AND join/fork.
- *Rotate*: Shows a popup menu allowing the user to specify one of four orientations to display the fork: forward, backward, upward, and downward. Each represents a 90 degree rotation from the previous setting. The orientation of the fork affects how it, and all connected paths, are drawn on screen.

OR Joins/Forks

- *Add Branch*: Adds another branch to the OR join/fork.
- *Rotate*: Allows for a selection of orientations for the OR fork to be displayed with, similar to that in the AND fork command.

3.7 Double-Element Transformations

Empty Point - Starting Point

- *Connect*: Attaches the Starting Point to the Empty Point in an asynchronous coupling.
- *Disconnect*: Decouples the Starting Point from the Empty Point, if previously connected.

Empty Point - End Point

- *AND Join*: Creates an AND join at the empty point and connects the ending path to it.
- *OR Join*: Creates an OR join at the empty point and connects the ending path to it.

Starting Point - End Point

- *Merge*: Attaches the end Point to the starting point so that the two paths become one.
- *Connect*: Connects the end point to the starting point in a synchronous coupling.
- *Disconnect*: Disconnects the two points from each other.

End Point - Stub

- *Join with Stub*: Connects the end Point to an input on the selected stub.

Starting Point - Stub

- *Merge*: Connects the starting point to an output on the selected stub.

Timer - End Point

- *Connect*: Attaches the end Point to the timer so that there is a synchronous coupling between them.
- *Disconnect*: Removes the synchronous coupling from the two points.

Waiting Place - End Point

- *Connect*: Attaches the end Point to the waiting place so that there is a synchronous coupling between them.
- *Disconnect*: Removes the synchronous coupling from the two points.

Waiting Place - Empty Point

- *Connect*: Attaches the empty point to the waiting place so that there is an asynchronous coupling between them.
- *Disconnect*: Removes the asynchronous coupling from the two points.

Timer - Empty Point

- *Connect*: Attaches the empty point to the timer so that there is an asynchronous coupling between them.
- *Disconnect*: Removes the asynchronous coupling from the two points.

AND join/fork - End Point

- *Synchronize Paths*: Adds the end point onto the AND join/fork as an additional input.

AND Join - Empty Point

Note: The empty point must be the first point before the join on one of the paths that are inputs to the join.

- *Add Timeout*: Adds a timeout to the selected branch of the join.
- *Remove Timeout*: Removes the timeout from the selected branch of the join.

Timestamp Point - Timestamp Point

- *Create Response Time Requirement*: Creates a response time requirement between the two timestamp points. The two timestamp points will change from being the outline of a triangle to a filled triangle.

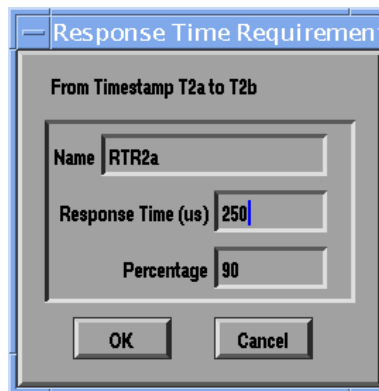


Figure 17. Create Response Time Requirement dialog box

- *Edit Response Time Requirement*: Edits the Response Time Requirement between the two selected Timestamp Points.
- *Delete Response Time Requirement*: Removes the Response Time Requirement from between the two selected Timestamp Points.

Goal Tag - Goal Tag

- *Create Goal*: Creates a goal between the two selected goal tags. Goals may have descriptions, preconditions and postconditions.
- *Edit Goal*: Brings up a dialog box to modify the properties of the currently selected goal.

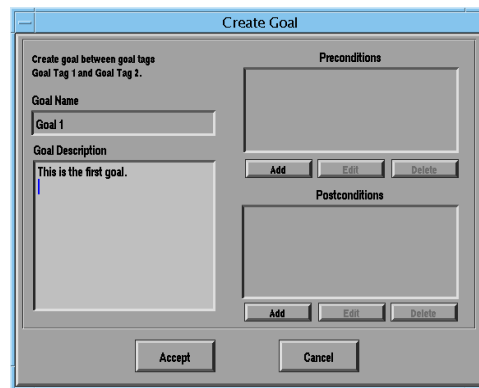


Figure 18. Create Goal dialog box

End Point - End Point

- *And Join*: Combines the two paths with an AND join.
- *Or Join*: Combines the two paths with an OR join.

List of References

Amyot, D. and Miga, A.: *Use Case Maps Linear Form in XML*, version 0.13, May 1999.
<http://www.UseCaseMaps.org/UseCaseMaps/xml/>

Buhr, R.J.A. and Casselman, R.S.: *Use Case Maps for Object-Oriented Systems*, Prentice-Hall, USA, 1995.
http://www.UseCaseMaps.org/UseCaseMaps/pub/UCM_book95.pdf

Buhr, R.J.A.: "*Use Case Maps as Architectural Entities for Complex Systems*". In: Transactions on Software Engineering, IEEE, December 1998, pp. 1131-1155.
<http://www.UseCaseMaps.org/UseCaseMaps/pub/tse98final.pdf>

Miga, A.: *Application of Use Case Maps to System Design with Tool Support*. M.Eng. thesis, Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada, 1998.
http://www.UseCaseMaps.org/UseCaseMaps/pub/am_thesis.pdf

Miga, A.: *Use Case Maps Navigator*, version 1.10, May 1999
<http://www.UseCaseMaps.org/UseCaseMaps/ucmnav/>

Use Case Maps Web Page and UCM Users Group, 1999.
<http://www.UseCaseMaps.org>

W3 Consortium: *Extensible Markup Language (XML) 1.0. W3C Recommendation*, 10 February 1998. <http://www.w3.org/TR/REC-xml>

List of Figures

Figure 1. The UCM Navigator Main Screen.	2
Figure 2. Single point transformation.	6
Figure 3. Double point transformation.....	7
Figure 4. Print dialog box	8
Figure 5. Response time requirements dialog box	10
Figure 6. Device Characteristics dialog box	10
Figure 7. Data Store Characteristics dialog box	11
Figure 8. Add/Edit Condition/Events dialog box	12
Figure 9. Component Attributes dialog box	14
Figure 10. Arrival Characteristics dialog box.	16
Figure 11. Add/Edit Responsibility Dialog.....	17
Figure 12. Dynamic Arrow, Data Stores, and Service Requests dialog boxes .	17
Figure 13. Create Stub dialog box.....	18
Figure 14. View Sub Map dialog box (multiple submaps).....	19
Figure 15. Bind Plugin to Stub dialog box	20
Figure 16. Response Times Requirements dialog box	21
Figure 17. Create Response Time Requirement dialog box	23
Figure 18. Create Goal dialog box	24

Glossary

A

- And Fork A split in the sequence of causal events where the system follows both paths concurrently. (p. 17)
- AND Join Describes a location where multiple paths will arrive and synchronize with each other before proceeding. (p. 21)

D

- Data Store A representation of a place where data is at rest (a database, filing system, etc.). (p. 10)
- Device A representation of a physical aspect of a computer system. (p. 10)

F

- Failure Point A location where a serious error might occur that would prevent the system from following the path any further. (p. 18)

O

- Or Fork A split in the sequence of causal events where the system will follow one of two paths, depending on the system's state and external stimuli. (p. 17)
- OR Join A location where multiple paths join and follow the same sequence of events. (p. 21)

P

- Path A causal sequence of events that can be followed through a system's components. (p. 3)
- Plugin Another name for a Sub Map. (p. 19)

R

- Response Time Requirement . . . A requirement for the system that the path between two points has to take a certain amount of time or less a set percentage of times. (p. 10)
- Responsibility A requirement that the system has to perform a certain function at a certain location or time. (p. 16)

S

- Shared Responsibility A notation that signifies that there is a responsibility at that location and that the responsibility may entail multiple components. (p. 18)
- Stub A black-box view of part of the system, used to unclutter the di-

agrams. (p. 18)
Sub Map A part of the UCM that is hidden by the black-box view of stubs.
(p. 19)

T

Timer..... A place where the system will pause for a signal, proceeding
when the signal is received or following a timeout path if no sig-
nal is received in a certain length of time. (p. 18)

U

UCM..... Use Case Map (p. 1)

W

Wait..... A place in the path where the system will pause for a signal be-
fore proceeding. (p. 17)