Nest User Interface Manual

Alexander Dupuy
Jed Schwartz

Computer Science Department
Columbia University
New York, NY 10027-6699

Wednesday June 15th, 1988

CUCS-376-88

Abstract

This manual describes the user interface client provided with Nest Version 2.5. Nest is available from Columbia University. For information, please contact the authors.

This research was supported in part by the Department of Defense Advanced Research Project Agency, under contract F29601-87-C-0074, and by the New York State Science and Technology Foundation, under contract NYSSTF CAT (87)-5.

Table of Contents

1. Using the Nest User Interface

To start the Nest user interface you should be running Suntools. If you aren't familiar with suntools, you may want to read the manual page for it before you continue. Once suntools has been started, you can run the user interface. You will probably want to run it in the background, so that your shell can still be used. After a moment or two, the user interface should display itself on your framebuffer display. Note that the Nest user interface takes all the standard suntools arguments for window and icon positioning, colors, etc.

When the user interface has displayed itself, you can see that it is divided into a number of subwindows, or panes. At the bottom is a large pane, where the network will be displayed, and where it can be manipulated. Just above it are three smaller panes, where the global simulation parameters are displayed, and can be manipulated. Above these is another panel, where the simulation to which the user interface is connected can be specified, and where graphs can be saved to and loaded from files. Finally, there is a small panel at the top where messages are displayed.

1.1. Control operations

All control operations are performed in the second panel. The host and port to which the simulation will be connected are text items which can be edited. Once the correct values have been set up, you can connect to a simulation by clicking on the cycle item (which should say "Detached"). Once a connection has been established, two addition cycle items will appear. The upper one allows the simulation to be paused and resumed. The lower one allows the simulation to be locked, so that other monitors cannot modify it. These use the control messages described in more detail in the Nest Overview.

Graphs can be saved to and loaded from files using the Save and Load buttons. These will read or write the file specified in the adjacent text item.

1.2. Graphlanguage header parameters

The values of graphlanguage header parameters are displayed in the three panels just above the network window. The panel on the left displays red-only parameters which cannot be changed by the user interface. The other panels contain editable parameters in text items, and in the case of the selected monitor function a cycle item. As with all cycle items, clicking the right menu on the item will pop up a menu with the available choices.

The values in these panels are updated whenever a new graph is received from the simulation. If the user edits the values in these panels, the changed parameters will be used in the graph the next time it is sent to the simulation (by selecting the Send action in the network window).

1.3. Network manipulations

The primary area of user interaction is the network window, where a diagram of the current network is displayed. Nodes and edges can be added quite easily using the left and middle mouse button. To create a node, simply position the cursor and click the left button. To create an edge, position the cursor over an existing node, then press and hold the middle button. Then move the cursor to another node (an edge line will follow the cursor) and release. It is even possible to combine the two actions by clicking the left button while an edge is being made with the middle button held down. This creates a node, which can be connected to the edge by releasing the middle button.

Deletions and other modifications can be made using the right, or menu button. When the menu button is pressed while the cursor is on (or very near) a node or edge, an appropriate menu is called up. The menu for nodes includes options to remove the node, stop or start it, or to display detailed information. A similar procedure allows edges to be removed and/or modified.

If the Show Data item is selected, the detailed information is displayed in a pop-up window, where it can be modified. The pop-up node and edge windows are also used as default values whenever new nodes or edges are created. Note that to actually modify a node or edge, not only must the values in the pop-up window be modified, but the Set Data menu item must be selected for the node. This action causes the values in the pop-up window to override the current settings of the node or edge.

If the menu button is pressed while the cursor isn't on any node or edge, a general menu is called up, which allows recent deletions to be undone, or a graphlanguage message containing the modifications made to be sent to the simulation.

2. Customizing the Nest User Interface

It is possible to customize the Nest user interface so that in the network window, different actions will be taken in response to mouse buttons. When the Nest user interface is started, it looks in the current directory for a file called "Nest.config". If this file is found, it is read and scanned for Event table and Menu definitions.

Comments can be placed in the Nest.config file by beginning a line with '#'.

2.1. Event tables

The first kind of customization which is possible is to modify some or all of the four event tables which are used in the Nest user interface. The first event table is called **Any**, and if an event is found in this table, the associated action is taken, regardless of where the mouse cursor may be. The second table is called **Node**, and if an event is found in this table, and the mouse cursor is over a node in the network window, the associated action is taken. Similarly, the third table, called **Edge**, is used when the mouse cursor is over an edge in the network window. The last table, called **Loc** is used only if an event has not been found in any of the other appropriate tables, depending on where the mouse cursor is. The only difference between the Any and Loc tables, is that the Any table takes precedence over the Node and Edge tables, even if the mouse is over a node or an edge.

The format of an event table definition is this: An event table definition starts with a line of the form

Events Any | Node | Edge | Loc *<number of events>*

The first word is "Events", followed by one of the four event table names, followed by the number of event lines which follow. Each event must be on one line, and no comment lines are allowed in these lines. The format of the following event lines is

<eventcode> <actioncode> <argument>

An eventcode can be any of "Left", "Middle", or "Right", prefixed by any combination of "Shift-", "Control-", or "Meta-". An actioncode is the name of the action to take, such as "Message" to display a message or "Menu" to pop up a menu. An argument is either a number, or a quoted string, for those actions which take additional arguments.

An example event table definition in a Nest.config file is the following:

```
Events  Edge     4
        Right            Menu     "Edge Menu"
        Shift-Middle     Delete_Edge
        Shift-Right      Show_EdgeData
        Meta-Right       Set_EdgeData
```

This defines the Edge event table so that pressing the right button over an edge will bring up the "Edge Menu" menu, pressing the middle button and shift key over an edge will delete the edge, and so forth.

Any event table definitions in the Nest.config file will override the defaults. If any of the four event tables are not defined in the Nest.config file, those undefined tables will retain the defaults.

2.2. Menus

The second kind of customization which is possible is define custom menus. All menus are referenced by a name, which is a quoted string. A menu can be tied to an event by placing a line in one of the event table definitions with an actioncode of Menu and an argument of the menu's name, in quotes.

The format of a menu definition is this: A menu definition starts with a line of the form

Menu " *<menu name>* " *<number of events>*

The first word is "Menu", followed by the name of the menu, in quotes, followed by the number of item lines which follow. Each item must be on one line, and no comment lines are allowed in these lines. The format of the following item lines is

" *<item label>*" *<actioncode>* *<argument>*

The item label is the string (in quotes) which will appear in the menu. The actioncode and argument are just the same as in an event line, and specify the action which will be taken if the menu item is selected. It is prefectly legal to put Menu actions in menu items; this creates a pullright submenu item.

An example menu definitions in a Nest.config file is the following:

```
Menu    "Nest Menu"     3
        "Redisplay"     Redisplay
        "Send"          Send_Graph
        "Graph"         Menu            "Graph Menu"
```

This defines a menu called "Nest Menu", with three items. The first, which is labeled "Redisplay", causes the network graph to be redisplayed, the second, labeled "Send", causes the current network graph to be sent to the simulation, and the third, labeled "Graph" is a pullright menu.

2.3. Defined Actions

Table 2-0 is a table of the actioncodes which are defined. Since some actions expect to be invoked from a particular event table, they can only be placed in that table. Be careful that all actions in menus and submenus are valid from the event table where the original menu was placed.

Table 2-1: Defined Actions

Actioncode	Argument type	Valid tables	Meaning of argument
Message	*string*	all tables	message to be displayed
Error	*string*	all tables	error message to be displayed
Menu	*string*	all tables	name of menu
Undo	*none*	all tables	
Redisplay	*none*	all tables	
Clear_Graph	*none*	all tables	
Send_Graph	*none*	all tables	
Make_Node	*string*	[Any, Loc]	name of function for node
Delete_Node	*none*	[Node]	
Move_Node	*none*	[Node]	
Set_NodeData	*none*	[Node]	
Show_NodeData	*none*	[Node]	
Set_NodeFunc	*string*	[Node]	name of function for node
Set_NodeOpt	*integer*	[Node]	[integer bitmask of node flags
Clear_NodeOpt	*integer*	[Node]	to be set, cleared, or x-or-ed
Toggle_NodeOpt	*integer*	[Node]	1=Start, 2=Halt, and 4=Repeat]
Start_Edge	*integer*	[Node]	edge weight for new edge
Delete_Edge	*none*	[Edge]	
Set_EdgeData	*none*	[Edge]	
Show_EdgeData	*none*	[Edge]	
Set_EdgeFunc	*string*	[Edge]	name of channel function
Push_EdgeFunc	*string*	[Edge]	name of channel function
Pop_EdgeFunc	*none*	[Edge]	

Appendix I
Default Configuration

What follows is a Nest.config file which duplicates the default configuration which is compiled into the Nest user interface. There is no reason to use this file, since if no Nest.config file is found, the compiled-in defaults will be used. It is presented as an example Nest.config file.

```
#
# Default Nest display configuration file
#
# Copy this file to Nest.config and edit it to taste.
# The configuration here is the same as the compiled tables which
# are used if no Nest.config file exists in the current directory.
#

#
# Event Tables.  These map keystrokes to actions.
#
# The format is:        Events  Any | Node | Edge | Loc  #
#
# where the second field specifies which of four event tables is
# being set, and the third field indicates the number of entries
# (lines) in the table.
#
# followed by # lines:  <eventcode> <actioncode> <argument>
#
# where eventcode is one of: Left, Middle, Right, prefixed by any
# combination of: Shift-, Control-, Meta- (multiple prefixes, e.g.
# Shift-Meta-, are okay).
#

Events  Any     1
        Control-Right   Send_Graph

Events  Node    6
        Left            Move_Node
        Middle          Start_Edge      -1
        Right           Menu            "Node Menu"
        Shift-Left      Delete_Node
        Shift-Right     Show_NodeData
        Meta-Right      Set_NodeData

Events  Edge    4
        Right           Menu    "Edge Menu"
        Shift-Middle    Delete_Edge
        Shift-Right     Show_EdgeData
        Meta-Right      Set_EdgeData

Events  Loc     3
        Left            Make_Node       ""
        Middle          Error           "You aren't on a node"
        Right           Menu            "Graph Menu"
```

```
#
# Menu Definitions.  These define named menus.
#
# The format is:        Menu     " <menu name> "  #
#
# where the second field specifies the name of the menu, and the
# third field indicates the number of entries (lines) in the menu.
#
# followed by # lines:  " <item label> " <actioncode> <argument>
#

Menu     "Node Menu"      9
         "Delete Node"            Delete_Node
         "Node Function"          Menu              "Function Menu"
         "Start Node"             Set_NodeOpt       1
         "Halt Node"              Set_NodeOpt       2
         "Reset Node"             Set_NodeOpt       3
         "Repeat Node"            Set_NodeOpt       4
         "Clear Flags"            Clear_NodeOpt     7
         "Show Node Data"         Show_NodeData
         "Set Node Data"          Set_NodeData

Menu     "Edge Menu"      6
         "Delete Edge"            Delete_Edge
         "Set Channel Function"   Menu              "Channel Menu 1"
         "Push Channel Function"  Menu              "Channel Menu 2"
         "Pop Channel Function"   Pop_EdgeFunc
         "Show Edge Data"         Show_EdgeData
         "Set Edge Data"          Set_EdgeData

Menu     "Graph Menu"     4
         "Redisplay"      Redisplay
         "Send"           Send_Graph
         "Clear"          Clear_Graph
         "Undo"           Undo
```