

**Artisynth:
Navigation Panel &
Selection Manager**

Software Design Overview

Version 1.0

April 27th, 2006

Revisions


<i>Version</i>	<i>Primary Author(s)</i>	<i>Description of Version</i>	<i>Date Completed</i>
1.0	Chad Decker	Initial version of the design overview document.	April. 27, 2006

Table of Contents

1.1 System Overview.....	4
1.2 Supporting Materials.....	4
2.1 Operational Context.....	6
2.2 MainFrame.....	6
3.1 Overview.....	10
3.2 The Navigation Panel in depth	10
3.2.1 JTree	11
3.2.2 NavPanelMutableTreeNode.....	11
3.2.2 ComponentInfo.....	11
3.2.3 TreeNodeSelectionListener.....	12
3.2.3.1 SelectedPathsManager.....	12
3.3 The Selection Manager in depth.....	13
3.3.1 SelectionManager.....	13
3.3.1.1 Selection Handler.....	14
3.3.1.2 Selection Window (Under Development).....	14
3.3.2 PokeablePopup.....	15
3.3.3 PopupRequestListener.....	16
4.1 Conclusion.....	17

1 Introduction

1.1 System Overview

ArtiSynth is a modular component-based system for performing dynamical 3D simulations of the human vocal tract and face. It is intended to provide a testbed for anatomical modeling and acoustic research in areas such as speech synthesis, linguistics, medicine, and dentistry. The framework provided by the system enables researchers to construct, refine, and exchange models of rigid bodies (such as bone) and soft tissues (such as a tongue), and integrate these into a comprehensive model of the vocal tract and surrounding structures. ArtiSynth provides base support for a variety of parametric and dynamic models, including rigid bodies, mass-spring and finite element models, and spline-based curves and shapes. 

Artisynth provides an API for model creation, along with a driver application for viewing and controlling the dynamical simulation of models. This specification document discusses revisions which were made to the User Interaction framework to allow for model traversal, selection, and editing through two main subsystems: the Navigation Panel, and Selection Manager.

1.2 Supporting Materials

Presentation on Artisynth

<http://www.artisynth.org/externdoc/AVSP-2005-2.pdf>.

Artisynth

<http://www.artisynth.org>.

Wikipedia, [Design Patterns](https://en.wikipedia.org/wiki/Design_Patterns)

http://en.wikipedia.org/wiki/Design_Patterns.

Sun Microsystems, [Class JTree](http://java.sun.com/j2se/1.4.2/docs/api/javawx/swing/JTree.html)

<http://java.sun.com/j2se/1.4.2/docs/api/javawx/swing/JTree.html>.

2 High Level Design Summary

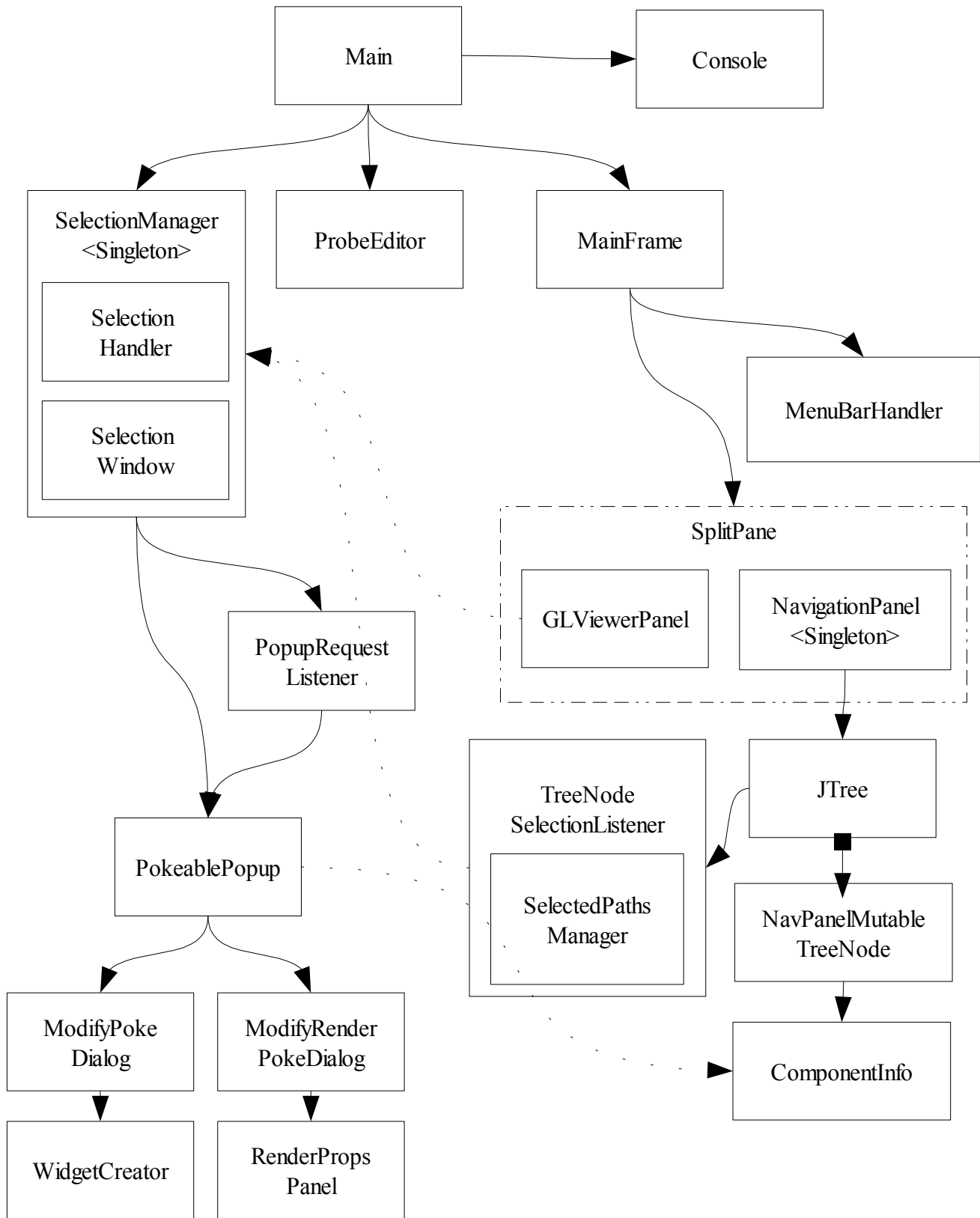


Figure 1.0 UML Diagram of the SelectionManager and NavigationPanel operational context.

2.1 Operational Context

Figure 1.0 above shows the current placement of the NavigationPanel and Selection Manager Packages within the Artisynt framework. Mainframe, which is the main viewer for Artisynt consists of three main components: the MenuBarHandler, GLViewerPanel, and NavigationPanel. The latter two panels are the principle sources of User Interaction with the 3D models, and have a great deal of interaction with the SelectionManager package. The ProbeEditor, and Console modules hook into the SelectionManager, and MainFrame objects respectively, to offer extended system functionality.

2.2 MainFrame

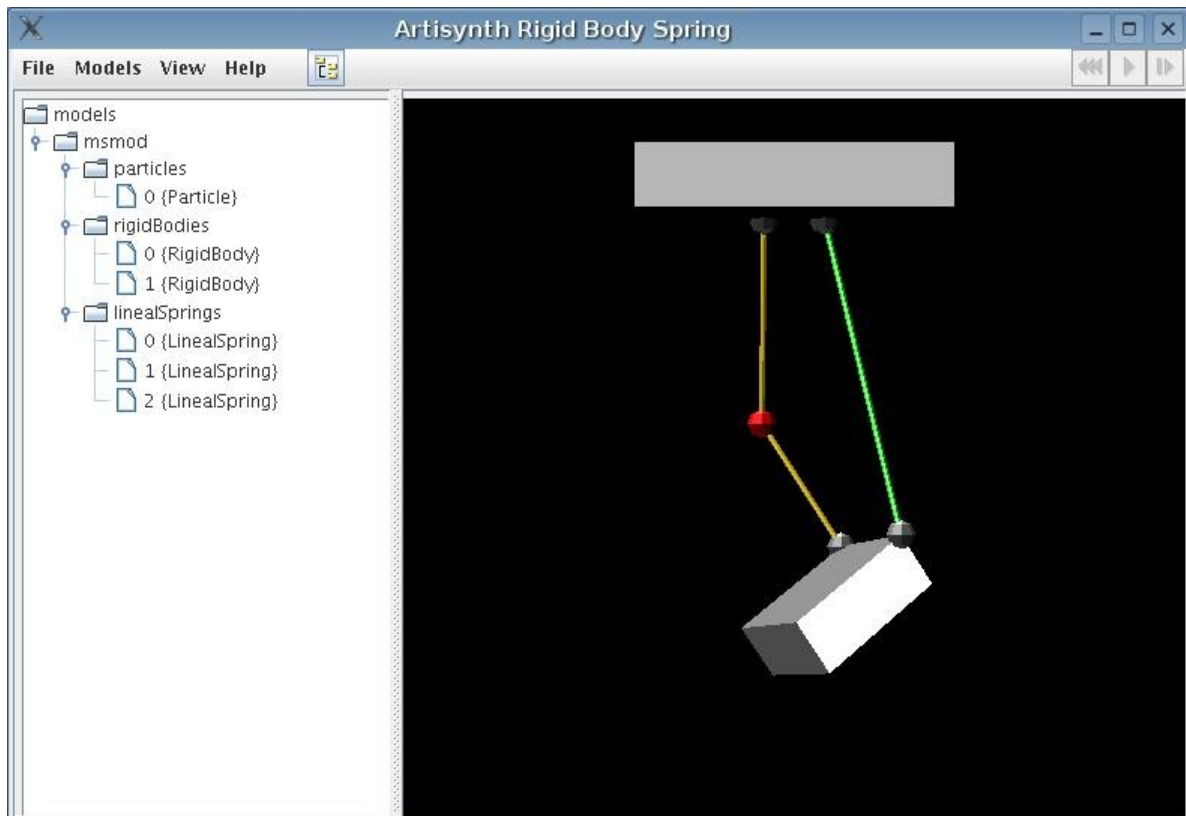


Figure 2.0 Image of MainFrame

Figure 2.0 shows the three principle components in MainFrame, on the left is the Navigation Panel which scans and presents the unit components of the 3D model which is shown graphially to the right in the GLViewerPanel. The Navigation panel is capable of scanning models, which are shown in the hierarchal format shown above. In addition, to being able to collapse and view the components, the Navigation Panel

like the GLViewerPanel, has the ability to select the individual model components, but also allows a user to select model components which may not be visible in the GLViewer panel. Depending on the viewing frustum, model components may be outside the viewing boundaries of the viewer, or components may overlap, hiding one another. In addition to Navigation Panel offering a user the ability to view and select items which cannot be selected in the GLViewerPanel, it also allows a user to select model components such as plane constraints or particle constraints, which aren't renderable, as shown below in figure 2.1.

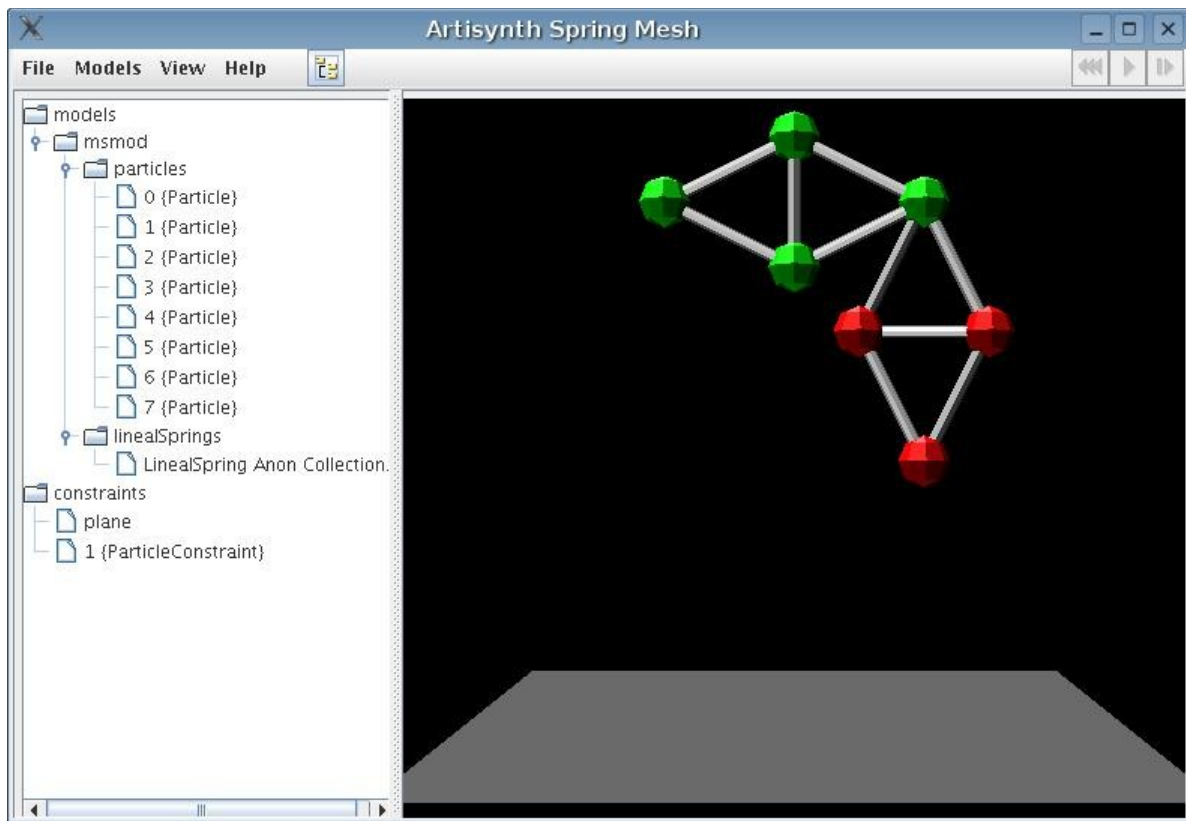


Figure 2.1 Image of MainFrame with Constraints

Both NavigationPanel, and GLViewerPanel have their own selection mechanism, which feeds into the Selection Manager Subsystem as mentioned in section 2.3. NavigationPanel allows three methods of selection and deselection: individual component selection, multiple component selection (using the Ctrl + mouse selection), and list selection (using the Shift + mouse selection); while the GLViewerPanel offers users both individual and multiple component selection. The GLViewerPanel and NavigationPanel (via the TreeNodeSelectionListener) provide notification of selected objects to the SelectionManager subsystem which is discussed in greater detail in selection 3.2. The SelectionManager provides a

centralized point for selection handling, where different selection mechanisms can send notifications of objects being selected, and the change is reflected in the other selection mechanisms. Figure 2.2 below shows the updating of the Navigation Panel taking place when objects in the GLViewerPanel are selected, just as how GLViewerPanel objects are updated when selections are made in the NavigationPanel.

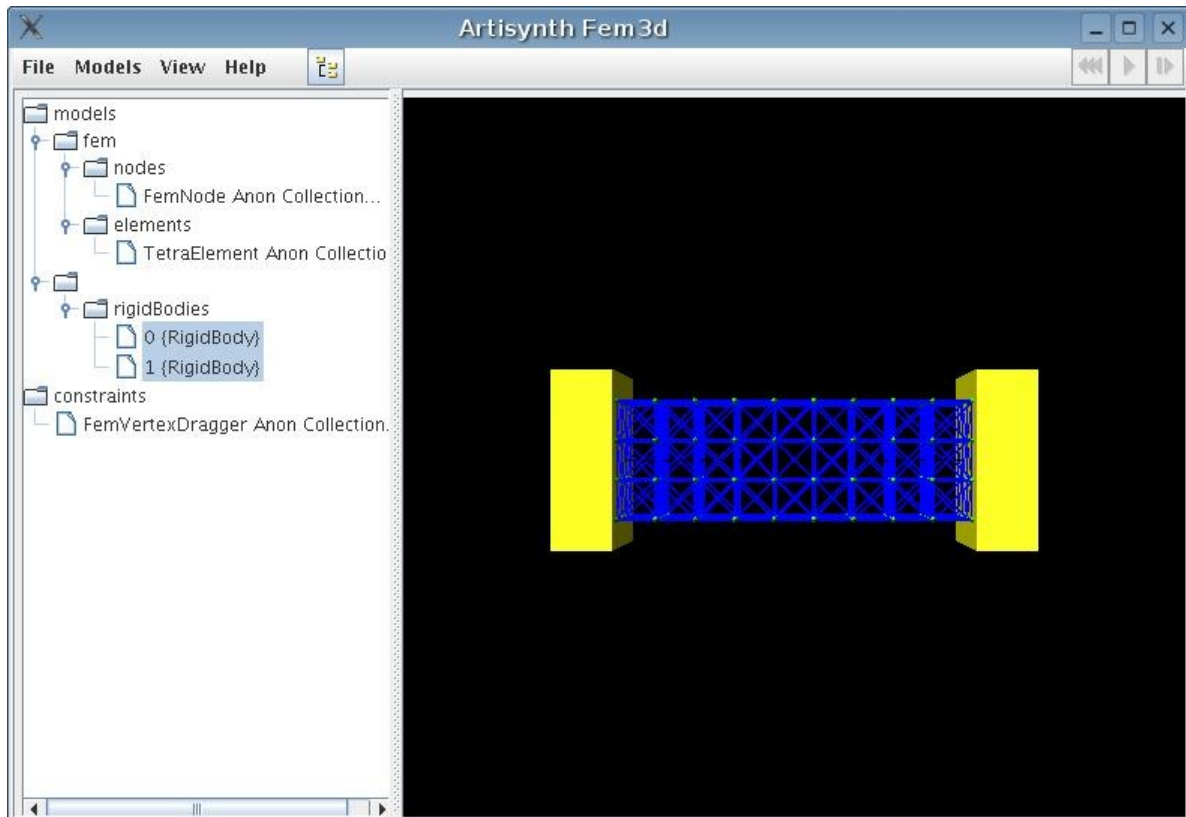


Figure 2.2 Updating of the MainFrame as a result of selection events.

Once a component is selected, and the other selection mechanisms updated appropriately, the Selection Manager builds a popup menu which displays the properties of the components which can be selected as shown below in figure 2.3. This popup menu can be accessed with a right click selection on the MainFrame. Various dialogs will become visible for the selections in the popup menu, and when affirmed, the changes applied to the objects in the model. If multiple components are selected then the intersection of their common properties is displayed such as in figure 2.4.

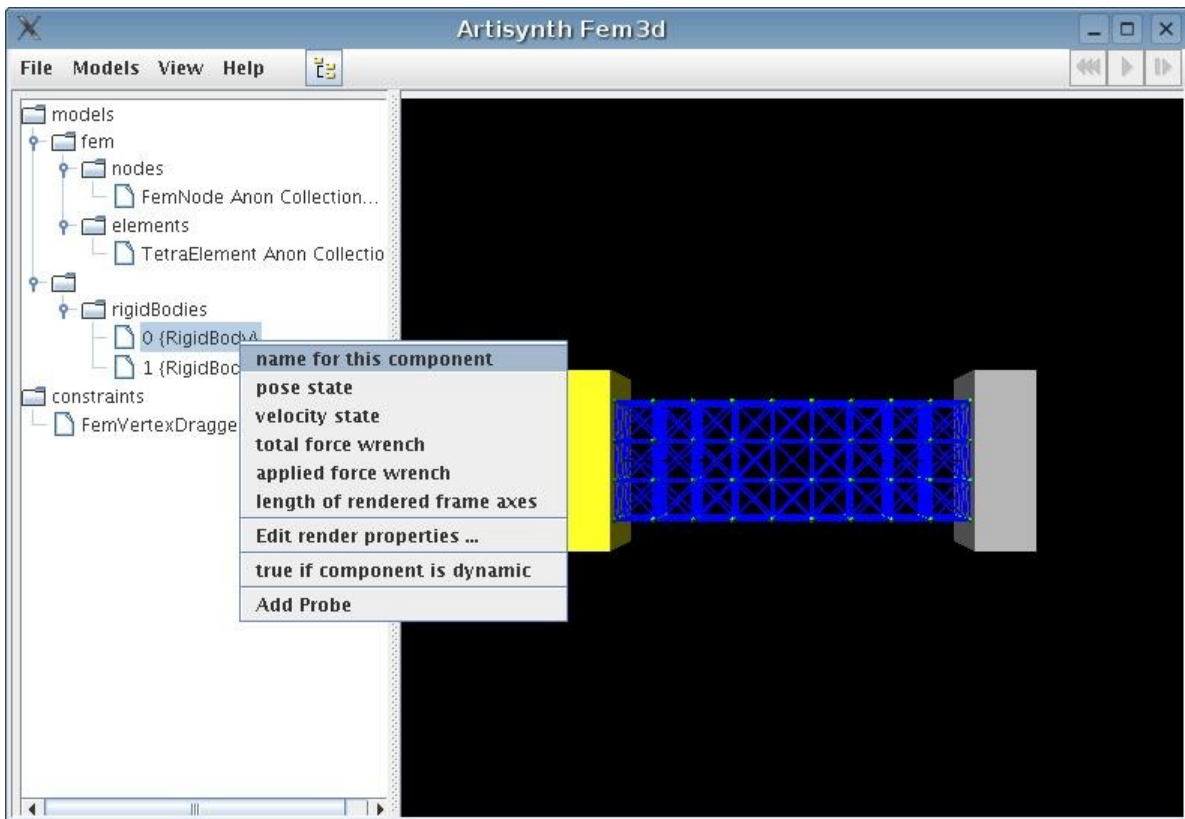


Figure 2.3 Popup menu composed by the Selection Manager.

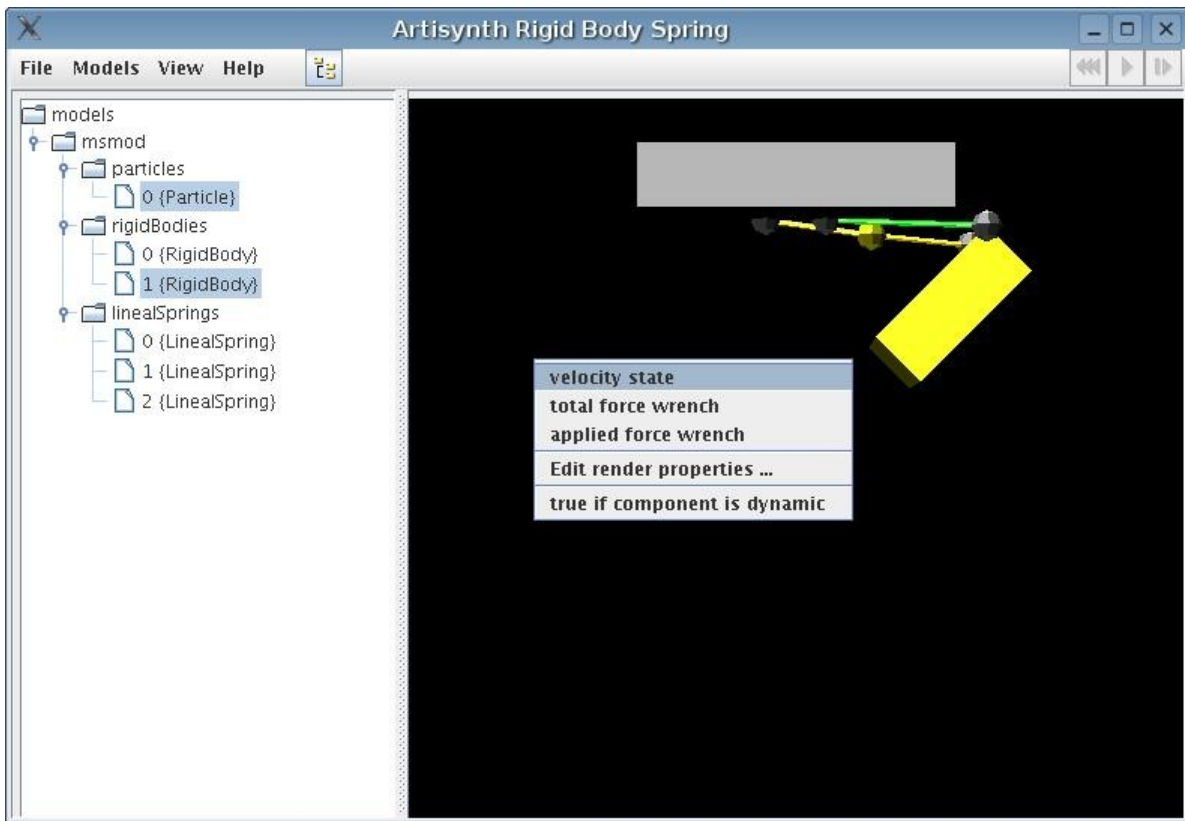


Figure 2.4 Popup menu composed by the Selection Manager intersecting common properties.

3 Low Level Design Summary

The section provides a low level overview of the Navigation Panel and Selection Manager packages.

3.1 Overview

The following sections discuss the inter workings of the NavigationPanel, and SelectionManager packages. The NavigationPanel consists of two main functionalities, first the generation and display of the model components, and second the action handling which takes place then nodes in the tree are selected. The Selection manager consists of two main functionalities: first providing a centralized object for selection mechanisms such as NavigationPanel and GLViewerPanel to provide, and receive information on the selection, and deselection of objects and second the Selection Manager manages the PokeablePopup property editor window.

3.2 The Navigation Panel in depth

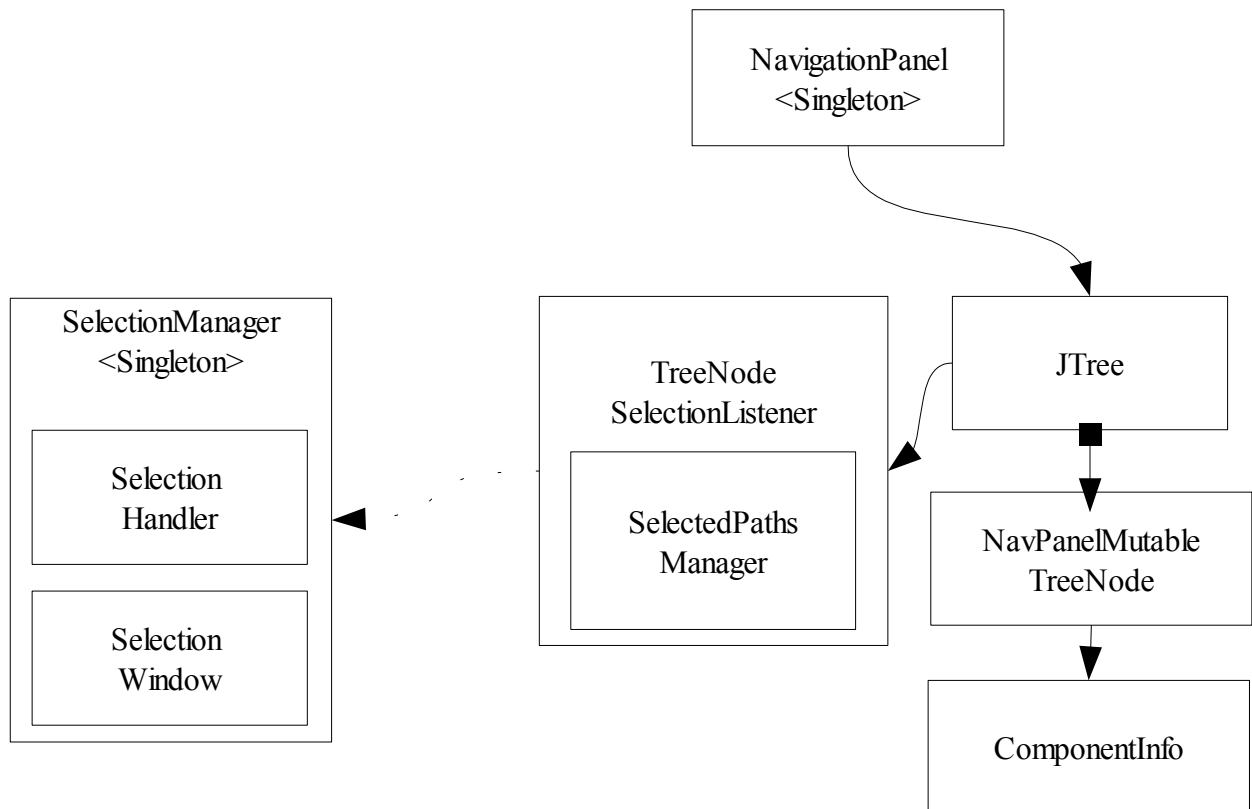


Figure 3.0 The Navigation Panels interaction with the Selection Manager.

Figure 3.0 above shows the decomposition of the Navigation Panel into 5 principle components, this selection provides a description of responsibilities and implementation for each of these classes.

3.2.1 JTree

Description: The JTree is a Java Swing Component which allows the display of hierarchical data. The Jtree object doesn't actually contain the data but simply provides a view of the data. Like any non-trivial Swing component, the tree gets data by querying its data model.

Implementation: The data in the JTree is organized in a nodal fashion with each element being a mutable tree node. In the NavigationPanel, a wrapper class for these nodes are used to embed additional operational information in the nodes.

3.2.2 NavPanelMutableTreeNode

Description: The NavPanelMutableTreeNode is a wrapper object for Java's DefaultMutableTreeNode with added information which is used to determine state information for the expansion/collapse of collections of anonymous model component collections.

Implementation: State information such as if the node, as a category, contains an anonymous object (either a model component or a collection of model components) is stored here. This allows the querying of categories when selected to see if the Popup option to expand/collapse categories is displayed to a user.

3.2.2 ComponentInfo

Description: The ComponentInfo is an object which is embedded in the NavPanelMutableTreeNode, that when selected, is passed to the SelectionManager. It serves as a wrapper class for two types of objects, Artisynth ModelComponent objects or NavPanelMutableTreeNode objects, as discussed above in 3.2.2. In the case of ModelComponent the ComponentInfo class offers a routine to extract a listing of ModelComponent's physical and renderable properties.

Implementation: Pointed to by the NavPanelMutableTreeNode, the ComponentInfo's toString function contains the name which is displayed in the JTree, as currently assigned by the NavigationPanel. The object inside, either a ModelComponent objects or NavPanelMutableTreeNode objects are used in two fashions: to manipulate the appearance of the tree, and for object passing when a selection is made which

are discussed in section 3.2.3.

3.2.3 TreeNodeSelectionListener

Description:

Implementation:

To manipulate the appearance of the tree, specifically with the collapsing, and expansion of Anonymous Collections of ModelComponents, the NavPanelMutableTreeNode of the node which contains the anonymous child nodes is used by the NavigationPanel to manipulate the structure of the JTree about this node. When components are selected using the GLViewerPanel, and a ModelComponent selected or deselected, the TreeNodeSelectionListener, discussed in 3.2.3 searches for the appropriate NavPanelMutableTreeNode which contains the ModelComponent and selects or deselects the path to the found node appropriately.

3.2.3.1 SelectedPathsManager

Description: The responsibility of the selected paths manager is to maintain the state of nodes which have been added or removed via the NavigationPanel. When a multiple selection takes place in the NavigationPanel, the selected paths manager is able to determine which tree nodes, have been added, or removed from the selection space.

Implementation: The Selection Manager maintains a copy of the previous selection list which was provided by the JTree. The Manager then performs a two way difference on the previous list, and the new selection list passed to it as a parameter, to determine which NavPanelMutableTreeNodes have been added or removed from the since the last selection.

3.3 The Selection Manager in depth

The Selection manager has two main responsibilities: to provide a centralized object for selection mechanisms to interact with one another, and to manage the PokeablePopup property editor window.

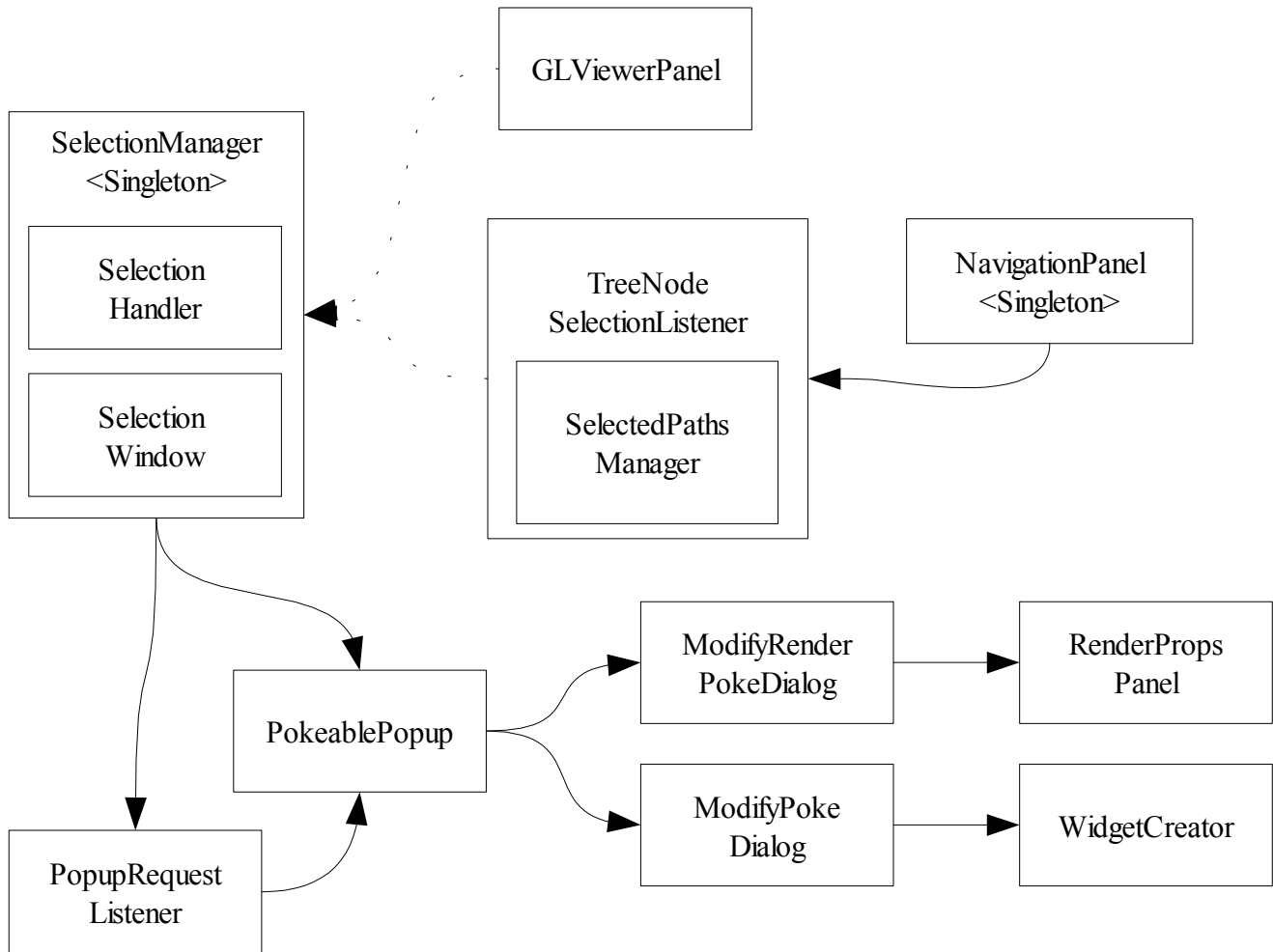


Figure 3.1 Selection Manager's interaction with Selection Mechanisms such as the GLViewerPanel, and TreeNodeSelection Listener

3.3.1 SelectionManager

Description: As shown above in Figure 3.1, the Selection manager employs a Mediator pattern, by which the various selection mechanisms, can notify the Manager, of the changes, and the selection manager delegates this information to other selection mechanisms, and the PokeablePopup window accordingly.

Implementation: The components of the model which are selected are maintained in a linked list structure of ModelComponents called mySelectedItems.

3.3.1.1 Selection Handler

Description: The SelectionHandler which implements the GLSelectionListener interface provides an entry point into the SelectionManager, for the GLViewerPanel. Its responsibilities include handling GLSelection events, which involves sending a notification to the SelectionManager, where the SelectionManager selects or deselects the component in the GLViewerPanel, and sends a request to the NavigationPanel to appropriately select or deselect the corresponding path in the NavigationPanel.

Implementation: When a selection event takes place in the GLViewerPanel, a call is made to the itemsSelected method in the SelectionHandler. The SelectionHandler, then sends a notification to the SelectionManager notifying it that a selection, or deselection has taken place in the viewer panel.

3.3.1.2 Selection Window (Under Development)

Description: Is a window, currently used for software development, which displays the objects which are currently selected according to the Selection Manager. Its application for future development, is particularly useful in largescale models with multiple components. It can serve as a method by which a user can interact with a small subset of the components present in the model as shown below in Figure 3.2.

Implementation: When a component selection or deselection takes place, through a selection mechanism the component is added or removed respectively from the linked list structure of ModelComponents called mySelectedItems. The Selection Window offers a graphical representation of the components which are in the linked list.

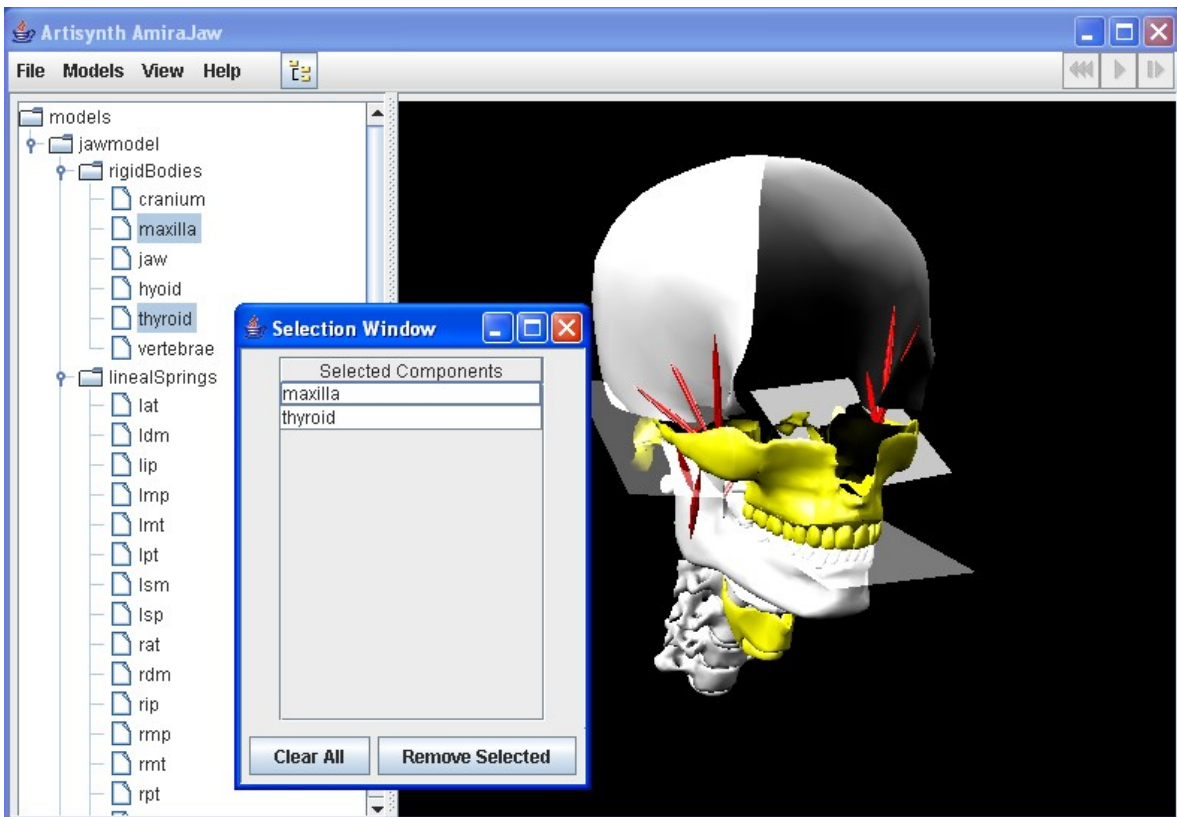


Figure 3.2 Selection Manager's Selection Window showing the Components which are currently selected through interactions with both the Navigation Panel, and GLViewerPanel.

3.3.2 PokeablePopup

Description: The PokeablePopup object is a popup window which displays the selectable options of the components which are currently selected. In the case of multiple components, the PokeablePopup object will compute the intersection of the common options for the objects selected, as shown in figure 2.4. The PokeablePopup object is also responsible for handling the selection made by the user and creating the corresponding modification dialog as show in figure 3.1.

Implementation: The PokeablePopup object is created and maintained by the SelectionManager. When the SelectionManager is informed of a selection, or deselection, the SelectionManager notifies the PokeablePopup of the revised list of selected lists, and informs it to modify the popup options accordingly. The PokeablePopup window decides which options are exposed when the loadComponents method is called. The PokeablePopup object is an extension of the JpopupMenu, and is activated by the PopupRequestListener when a right click occurs as describe below in section 3.3.3.

3.3.3 PopupRequestListener

Description: The PopupRequestListener is responsible for displaying the PokeablePopup window when the user has right clicked with the mouse button.

Implementation: There are two entry points to the PopupRequestListener. The first entry point is the mousePressed() method which is attached to the JTree in the NavigaionPanel and is fired when a right mouse button click occurs in the JTree panel. The second is the showPopup method which is called manually, as opposed to being a registered listener, by the GLViewerPanel when a right click occurs in the GLViewerPanel.

4 Design Summary

4.1 Conclusion

Previous to the addition of the two subsystems, the Navigation Panel and Selection Manager, Artisynt was a system where property editing was limited to a programmer manually editing code to change values within a model. With the extension of these new subsystems, a user, without programming experience, can interact with models dynamically by changing component properties such as mass, velocity and tension, and revisiting the simulation instantaneously. In addition, the NavigationPanel provides the user the ability to traverse through the model components and, combined with the Selection Manager, provides a uniform mechanism for model component selection and editing.

These two subsystems offer a framework in which the system can be expanded to offer features such as the creation, attachment and deletion of model components, which will ensure that Artisynt will continue to be an essential tool for anatomical modeling and acoustic research in areas such as speech synthesis, linguistics, medicine, and dentistry.