

## UML Tool Actions - Diagram Elements Actions 1.0 Requirements Specification

### 1. Scope

#### 1.1 Overview

The Diagram Elements Actions component provides the Actions related to the DiagramElements declared in Diagram Interchange component. The actions are strategy implementations of the action interfaces in the Action Manager component. The provided actions are for removing / copying / cutting and pasting diagram elements.

#### 1.2 Logic Requirements

##### 1.2.1 *Supported DiagramElements*

This actions will support at least the GraphNode, the GraphEdge, the TextElement and the Polyline.

Note that the TextElement must be contained in a GraphNode, in order to have the position and size attributes. It will be the only contained element of the GraphNode it belongs to.

The position and size of the Polyline is determined by its waypoints.

##### 1.2.2 *Remove / Copy / Cut / Paste DiagramElement action*

This component will provide a concrete action for each operation.

The Remove / Cut / Paste actions are UndoableActions.

The Copy action is a TransientAction.

Note that the Cut action can be implemented as a CompoundUndoableAction, made of a transient Copy action (wrapped in TransientUndoableAction) and an undoable Remove action.

The Add operations will be implemented in separated components, as the actual semantic model bridge nodes used and the actual compartments are different for each element.

##### 1.2.2.1 The Remove action will be configured with:

- the ModelElement

The action will remove the element from the diagram. However, it is not responsible for removing the owned elements, or the relations connected to it or its owned elements.

##### 1.2.2.2 The Copy action will be configured with:

- the ModelElement
- the Clipboard (defaults to the system clipboard)

The action will copy the element. However, it is not responsible for copying the owned elements, or the relations connected to it or its owned elements.

The copy information will be placed in the clipboard. Note that the Copy and the Paste action must function together.

The DataFlavor of the Transferable object used should be documented.

##### 1.2.2.3 The Cut action will be configured with:

- the ModelElement

This action will Copy and Remove the element, as specified above.

1.2.2.4 The Paste action will be configured with:

- the Transferable content representing the ModelElement
- the parent Namespace (optional)

The action will paste the element into the model in the same namespace it was in, or in the provided namespace. It will get the information from the received Transferable object.

### **1.3 Required Algorithms**

None.

### **1.4 Example of the Software Usage**

The component will be used in the TopCoder UML Tool to perform diagram element related actions.

### **1.5 Future Component Direction**

None.

## **2. Interface Requirements**

### *2.1.1 Graphical User Interface Requirements*

None.

### *2.1.2 External Interfaces*

The design must follow the interface found in the class diagram with the component interfaces. The designer is encouraged to add to the existing interface, but not to remove anything.

### *2.1.3 Environment Requirements*

- Development language: Java 1.5
- Compile target: Java 1.5

### *2.1.4 Package Structure*

com.topcoder.uml.actions.diagram.elements

## **3. Software Requirements**

### **3.1 Administration Requirements**

#### *3.1.1 What elements of the application need to be configurable?*

None.

### **3.2 Technical Constraints**

#### *3.2.1 Are there particular frameworks or standards that are required?*

None

#### *3.2.2 TopCoder Software Component Dependencies:*

- Action Manager 1.0
- Diagram Interchange 1.0
- UML Model Manager 1.0
- Configuration Manager 2.1.5 - recommended

# [ TOPCODER ]

## SOFTWARE

\*\*Please review the [TopCoder Software component catalog](#) for existing components that can be used in the design.

### 3.2.3 *Third Party Component, Library, or Product Dependencies:*

None

### 3.2.4 *QA Environment:*

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003

## 3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

## 3.4 Required Documentation

### 3.4.1 *Design Documentation*

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

### 3.4.2 *Help / User Documentation*

- Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.