[TOPCODER]
SOFTWARE

UML Tool Actions - Diagram Elements Add Actions 1.0 Requirements Specification

1. Scope

1.1 Overview

The Diagram Elements Add Actions component provides the Actions related to the concrete diagram elements - add and update. The actions are strategy implementations of the action interfaces in the Action Manager component.

1.2 Logic Requirements

1.2.1 Add / Update class diagram elements actions

1.2.1.1 Add / Update Package GraphNode action

This component will provide a concrete action for add and remove operations.

The actions are UndoableActions.

1.2.1.1.1 The Add action will be configured with:

- a GraphNode instance with the size and position set and with the semantic model set to the Package
- the parent GraphNode (optional)

The action will update the GraphNode with the proper compartments and will add the Package's GraphNode to the parent graphNode's contained elements, or directly to the diagram.

The action will also pass the element to the ProjectConfigurationManager, to apply any initial formatting.

1.2.1.1.2 The Update action will be configured with:

- the Package's GraphNode
- the change type
    - name changed
    - namespace changed
    - stereotypes changed
    - size changed

The action will react to these changes, updating the compartments. It should support different types of changes and action handlers.

1.2.1.2 Add / Update Interface GraphNode action

The actions are similar to the ones above.

The Update action has these change types

- name changed
- namespace changed
- stereotypes changed
- size changed

- compartments were hidden / shown (package, operation and attributes)
- attribute was added / removed / updated
- operation was added / removed / updated

1.2.1.3 Add / Update Class GraphNode action

The actions are similar to the ones for the Interface.

1.2.1.4 Add / Update Exception GraphNode action

The actions are similar to the ones for the Interface.

1.2.1.5 Add / Update Enumeration GraphNode action

The actions are similar to the ones for the Interface.

1.2.1.6 Add / Update Association GraphEdge action

The actions are similar to the ones above.

The Update action has these change types

- name changed
- namespace changed
- stereotypes changed
- association ends' properties changed

1.2.1.7 Add / Update Dependency GraphEdge action

The actions are similar to the ones for the Association.

The Update action doesn't have the association ends change type.

1.2.1.8 Add / Update Abstraction GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.1.9 Add / Update Realization GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.2 Add / Update use case diagram elements actions

1.2.2.1 Add / Update Actor GraphNode action

The actions are similar to the ones for the Interface.

The Update action doesn't have the change types related to the attributes and operations.

1.2.2.2 Add / Update Subsystem GraphNode action

The actions are similar to the ones for the Actor.

1.2.2.3 Add / Update UseCase GraphNode action

The actions are similar to the ones for the Actor.

1.2.2.4 Add / Update Include GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.2.5 Add / Update Extend GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.3 Add / Update activity diagram elements actions

1.2.3.1 Add / Update Initial Node GraphNode action

The actions are similar to the ones for the Actor.

1.2.3.2 Add / Update Object Flow State GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.3 Add / Update Action State GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.4 Add / Update Send Signal Action GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.5 Add / Update Accept Event Action GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.6 Add / Update Fork Node GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.7 Add / Update Join Node GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.8 Add / Update Decision Node GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.9 Add / Update Merge Node GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.10 Add / Update Flow Final Node GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.11 Add / Update Final Node GraphNode action

The actions are similar to the ones for the Initial Node.

1.2.3.12 Add / Update Transition GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.4 Add / Update sequence diagram elements actions

1.2.4.1 Add / Update Object GraphNode action

The actions are similar to the ones for the Actor.

1.2.4.2 Add / Update Create Message GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.4.3 Add / Update Create Message GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.4.4 Add / Update Synchronous Message GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.4.5 Add / Update Asynchronous Message GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.4.6 Add / Update Return Message GraphEdge action

The actions are similar to the ones for the Dependency.

1.2.4.7 Add / Update Send Signal Message GraphEdge action

The actions are similar to the ones for the Dependency.

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.class

com.topcoder.uml.actions.diagram.elements.usecase

com.topcoder.uml.actions.diagram.elements.activity

com.topcoder.uml.actions.diagram.elements.sequence

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
- UML Model Manager 1.0
- UML Project Configuration 1.0
- UML Model components
- Diagram Interchange 1.0
- Configuration Manager 2.1.5 - recommended

**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.