[TOPCODER]
SOFTWARE

XMI Writer 1.0 Requirements Specification

1. Scope

1.1 Overview

The XMI Writer component provides the ability to write XMI files. The writer will provide a framework for element transformers according to their type. The transformers will transform the element they receive, by writing the response to an output print stream.

1.2 Logic Requirements

1.2.1 XMIWriter

This class has registered XMITransformers to which it will pass elements to be saved in XMI. The class must at least support to save the data from the UML Model Manager component: the Model, the ActivityGraphs and the Diagrams (meaning that it will have a registered transformer for each type).

The class has a method to transform the model to a file.

If it is not requested to save the diagram data, only the Model will be saved.

1.2.2 XMI root element

The XMI output file will contain a root <XMI> node and the other elements will be added to the <XMI.content> node.

1.2.3 XMITransformer

The component will provide this interface. The transformers will receive the object and the output print stream.

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

The component will be used in the TopCoder UML Tool to write the XMI files.

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.xmi.writer

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:

- 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.