XMI Reader 1.0 Requirements Specification

1. Scope

1.1 Overview

The XMI Reader component provides the ability to parse XMI files, using SAX, as the usage of DOM requires too much memory. The parser will provide a pluggable framework for node handlers according to their type and a map index of the elements based on their xmi ids (for the elements that are not loaded a map of properties is kept).

1.2 Logic Requirements

1.2.1 XMIReader

This class provides a mapping of ContentHandlers according to the element types. The class will delegate the SAX events from the elements to these handlers. If there is not a mapped handler for a node, the events are passed to the active handler.

1.2.2 Referenced elements

The XMI file contains references to other elements. To be able to locate them, the XMIReader will keep the mappings between the xmi ids of the elements and the actual elements. If the reference is met before the element is actually declared, the XMIReader will keep a Map of properties for each referenced element - the id of the element that references this element and the name of the property. There will be two distinct methods: one that returns the referenced element, in case it exists and one that returns the properties, in case they exist.

The handlers will add the elements that can be referenced to the XMIReader, as they create them.

1.2.3 Handlers

The handlers are of org.xml.sax.ContentHandlers type.

Each handler will keep a reference to the XMIReader. The XMIReader will provide methods to maintain the list of registered handlers.

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

The component will be used in the TopCoder UML Tool to parse 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.reader

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.

