Standard Class Data Loader 1.0 Requirements Specification

1. Scope

1.1 Overview

The Standard Class Data Loader component provides the ability to load from configuration the standard namespaces and classes for a specific language. This component provides an easy to build and read storage file, as opposed to the XMI format for UML.

1.2 Logic Requirements

1.2.1 StandardClassDataLoader

This interface / class provides a method to load the namespaces for a specific language. The Namespace interface and the other required interfaces are from UML Model Core and UML Model Core Classifiers components. The returned namespaces will contain sub-namespaces and / or classifiers.

The classes used from UML Model are Namespace and concrete Classifiers: Class, Interface, DataType, Primitive and Enumeration.

1.2.2 Concrete configuration storage

The component will provide a concrete configuration storage using the Configuration Manager component or a custom XML file. The idea is that the format of the file should be easy to read and build.

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

The component will be used in the TopCoder UML Tool to load the standard classes for the design project, according to the project's language.

1.5 Future Component Direction

- Optionally, the classes can contain the methods and / or the attributes.
- The ability to load the information from concrete source files or from jar files.
- The ability to load the classes only from certain package paths.

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

©TopCoder Software, Inc. 2002

2.1.4 Package Structure

com.topcoder.uml.standardclassloader

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- The data source.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

None

3.2.2 TopCoder Software Component Dependencies:

- UML Model Core
- UML Model Core Classifiers
- 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.

©TopCoder Software, Inc. 2002