Software Documentation : Java Generic Swing Birds Eye View Control

This page last changed on Jul 29, 2008 by ghostar.

1. Scope

1.1 Overview

The UML Tool displays diagrams used for software modeling. These diagrams tend to get very large, and usually the diagrams are too large to be displayed in their entirety on a single display. To address this issue, this component will display a thumbnail view of a diagram, allowing the user to drag a view port to various areas on the diagram, allowing for easier manipulation of the displayed areas.

1.1.1 Version

1.0

1.2 Logic Requirements

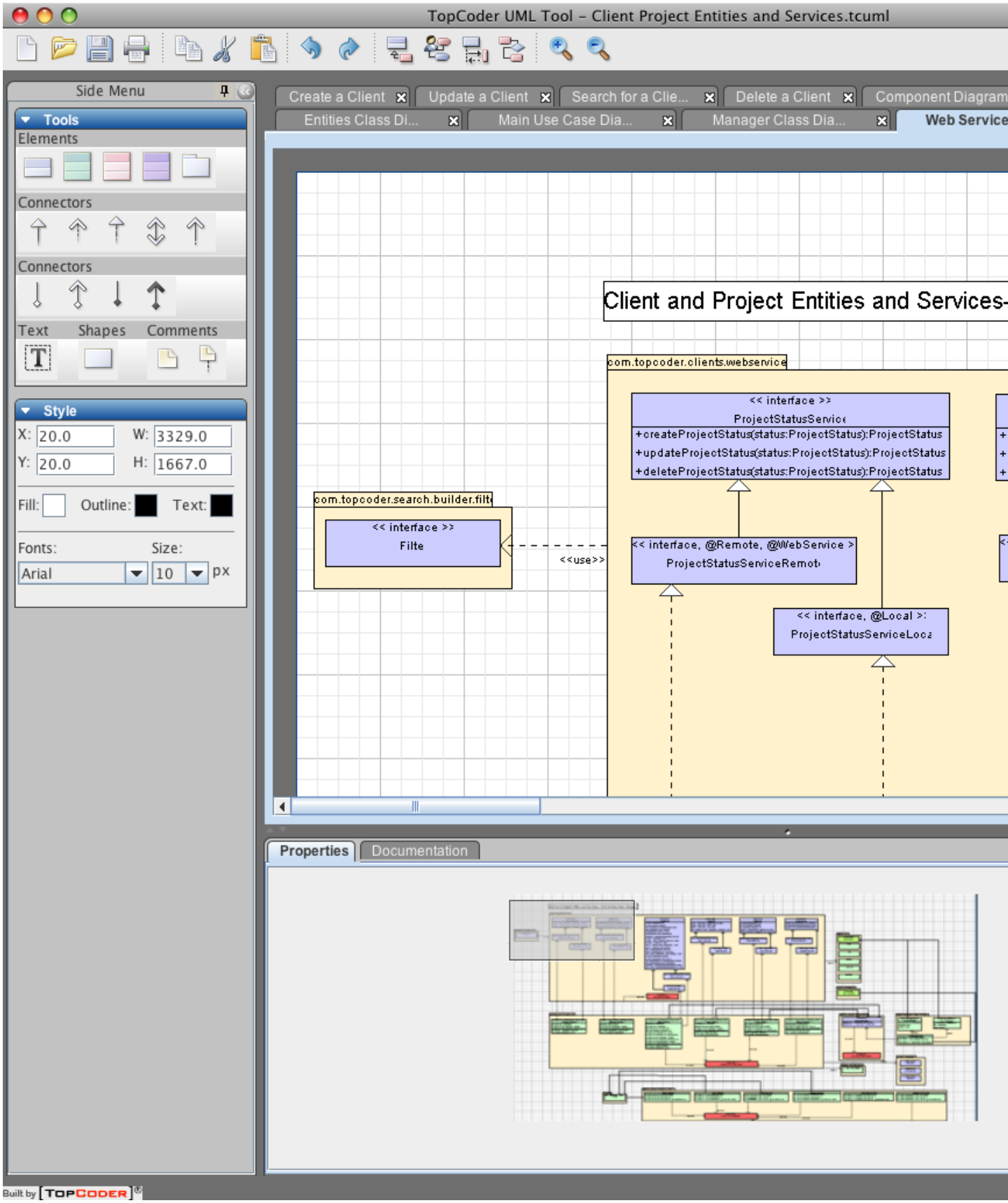
1.2.1 Input

This component will display a thumbnail view of a ZoomPanel instance, taken from the Zoom Panel component. This is the core requirement, but if the designer chooses to make the component generic enough to support any JComponent for input, it would be considered additional functionality. The component being displayed in the birds eye view control must be changeable through setters, as the user will switch diagrams on a regular basis, and the birds eye view must be updated to address this.

1.2.2 Thumbnail display

The output of this component will be a "bird's eye view" of the given Zoom Panel to watch, basically a thumbnail of the entire control, or in the case of the UML Tool, a thumbnail of an entire diagram. The thumbnail display must fit entirely into the given bounds of the Bird's Eye View Control. No vertical or horizontal scrollbars should ever be shown.

Here is a prototype of what the intended output should look like. Note that this is just a sample of what the component itself should look like. Placement of the control in the UML Tool will likely differ when this component is deployed



1.2.3 Viewport

A viewport overlay must be displayed over the top of the bird's eye view, with a size matching the actual part of the diagram displayed on the screen. This means that what is displayed in the viewport overlay in the bird's eye view must match the display of the actual Zoom Panel. The overlay should become bigger or smaller in proportion to the actual diagram. For instance, if the diagram is extremely large, the viewport overlay will be small, but if the diagram is small, the viewport overlay will be bigger.

1.2.3 Viewport Dragging

The viewport overlay must support dragging to change the displayed area in the main zoom panel. While the viewport overlay is dragged, the main zoom panel diagram display must be updated during the drag, showing the areas the viewport overlay covers.

1.2.5 Updates

If the main Zoom Panel is updated, to add a new element, like a new class or interface, the bird's eye view must reflect that change immediately. If the main Zoom Panel dimensions get bigger as a result, the viewport overlay should get smaller, and the overall size of the bird's eye view shown should also get smaller.

1.2.5 Colors

The viewport overlay color and opacity level properties must be able to be set through the UIManager, with names like "BirdsEyeView.overlayColor", and "BirdsEyeView.opacity".

1.2.6 Efficiency

Overall efficiency is important for this component, as it shouldn't slow down the UML Tool overall. It is important for this component to work as fast as possible, with as little overhead as possible.

1.3 Required Algorithms

How the bird's eye view is created must be described in detail, as well as how the viewport overlay will be created and used.

1.4 Example of the Software Usage

This component will provide an easy way for the user to navigate large diagrams in the UML Tool.

1.5 Future Component Direction

At some point in the future, the zoom level of the diagram may be tied to a resizable viewport overlay, so if the user resizes the viewport overlay to be bigger, the zoom panel being displayed will have its zoom level set smaller, etc...

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

This component will be a Swing component, and will manipulate other Swing components.

2.1.2 External Interfaces

None

2.1.3 Environment Requirements

- Development language: Java 1.5

- Compile target: Java 1.5

2.1.4 Package Structure

com.topcoder.swing.birdseyeview

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- The viewport overlay color (BirdsEyeView.overlayColor)
- The viewport overlay border color (BirdsEyeView.overlayBorderColor)
- The viewport overlay opacity (BirdsEyeView.overlayOpacity)

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

- Swing

3.2.2 TopCoder Software Component Dependencies:

- Zoom Panel http://software.topcoder.com/catalog/c_component.jsp?comp=7400223

3.2.3 Third Party Component, Library, or Product Dependencies:

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 the TopCoder UML Tool.