

UI Architecture

The main Element in the UI Architecture is the *GuiController* class. This Class defines all control flowing methods triggered whenever the UI is somehow influenced. However the *GuiController* never modifies the UI itself, furthermore it delegates these tasks to the *Cleaner*, *MenuHandler*, *PopupManager* and mainly to the *GuiViewElementHandler*.

While the names of *Cleaner*, *MenuHandler* and *PopupManager* are somehow self-descriptive and there influence is restricted to the named UI Part, the *GuiViewElementHandler* is responsible for maintain all specialized *ElementHandlers* throughout the UI, coordinating and arranging them in a proper way.

The specialized *ElementHandlers* themselves (*Code_ElementHandler*, *Requirement_ElementHandler*, *Commit_ElementHandler*, *Configuration_ElementHandler*, *Impact_ElementHandler*, *Linkage_ElementHandler*, *PathDE_ElementHandler*, *Requirement_Detail_ElementHandler*) arrange a list of corresponding *Presenters*, add *ListListeners* and manage the current user selection.

Whenever the user triggers a certain UI action the *GuiController* can get the current selection state in form of *DataElements* from the needed *ElementHandlers* and define a new request to the *I_Collection_Model* with defines the interface to the Business Logic. Afterwards the *GuiController* queries a suitable *ElementHandler* from the *GuiViewElementHandler* and transfers all these parts to the *Transformer*.

The *Transformer* itself only calls and manages all parts of the user-triggered UI action. Therefore a *Visitor* pattern is used to transform all requested data to *Presenters*. The results, now available as *Presenters* with internal presentation logic, are passed to the defined *ElementHandler* and displayed to user again.

The following picture shows a section of classes that participate in such a user action.

