**Use Case – ClubUML – Enable/Disable ‘Smart Policy’ Runtime User Preferences**

|  |  |
| --- | --- |
| Document Type | Use case |
| Activity | Analysis |
| Document Version | 1.0 |
| Document Status | Draft |

**Version history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 09.23.2013 | 1.0 | Initial draft | Seth |

**Brief Description**

This use case portrays the scenario where a user chooses which ‘smart policy’ rules they would like to enable and disable to be applied to an individual ‘Auto-Merge’ or ‘Suggest Promote’ request when merging or comparing two UML diagrams.

**Actors**

User: The user is the actor who enables or disables his or her ‘smart policy’ runtime user preferences.

**Preconditions**

1. The user follows Login use case flow to access ClubUML.
2. The user follows Auto Merge or Compare Diagram use case flow.

**Flow of events**

1. The user follows Login use case flow to access ClubUML.
2. The user follows either the Auto Merge (to automatically merge two diagrams) or the Compare Diagram (to compare two diagrams and will use ‘Suggest Promote’ feature to enable the ‘smart policy’ rules to be applied to the diagram comparison).
3. The user is presented with an ‘Override Smart Policy’ button on either of the Merge or Compare screens.
4. The user clicks on ‘Override Smart Policy.’
5. The user is presented with a list of ‘smart policy’ rules that can checked or un-checked. These rules can be clicked for more detailed information.
   * The user un-checks a rule to disable it.
   * The user checks a rule to disable it.
   * The user clicks on a rule for more detailed information and is presented with a pop-up or another screen showing the details of the rule. Rules that have configurable values associated with them (e.g. a number or Boolean setting) are editable.
     + The user changes an editable field associated with a rule. The user clicks ’Save’ to retain the edited value(s). The editable value is retained and stored for the user if the rule is or becomes enabled.
     + The user clicks ‘Ok’ to exit the detailed rule information screen.
6. With the list of ‘smart policy’ rules reviewed and appropriate rules enabled or disabled, the user clicks ‘Save’ to retain their preferences.
7. The user is returned to the previous screen.
8. The application will apply the overridden list of ‘smart policy’ rules and their associated values to the ‘Auto-Compare’ or ‘Suggest Promote’ request. This override will apply only to the current request, and will not be applied to future requests.

**Alternate Flow**

**User exits the “Maintain ‘Smart Policy’” screen without saving changes or opts to ‘Cancel’**

If the user exits the “Maintain ‘Smart Policy’” maintenance screen without saving his or her changes, or opts to ‘Cancel,’ then the user’s default list of ‘smart policy’ rules will be applied to the ‘Auto-Merge’ or ‘Suggest Promote’ request.

**Key Scenarios**

If the button “save” has not been clicked, no ‘smart policy’ rule changes will be applied, and the user’s default ‘smart policy’ rules will be applied to the request.

**Post-conditions**

1. **Successful Completion**

The user’s ‘smart policy’ runtime rule choices are applied for use by the ‘Auto-Merge’ and ‘Suggest Promote’ features for this request, and not to future requests.

1. **Failure Condition**

The user’s default ‘smart policy’ rules will be applied to the request.