**Use Case - ClubUMLv3 – Manual Merge Diagram**

|  |  |
| --- | --- |
| Document Type | Use case |
| Activity | Analysis |
| Document Version | 3.1 |
| Document Status | Draft |

**Version history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 09.17.2013 | 3.0 | Copied over from Spring2013 , renamed it from ‘MergeDiagram’ to ‘Manual-Merge Diagram’ Use case | ClumUml |
| 09.18.2013 | 3.1 | Updated to fit into the new auto ‘smart’ merge paradigm. | Gautam Chongtham |

**Brief Description**

This use case describes a manual merge functionality, in which the user selects which elements to keep from two related diagrams.

**Actors**

User: The user selects the diagrams to merge and makes decisions on merging.

**Preconditions**

1. The user follows Login use case flow to access ClubUML.
2. The user has already uploaded two valid diagrams of the same type (following UploadDiagram flow of events).
3. The uploaded diagrams to be merged are from Papyrus (no Ecore merge support).

**Basic Flow of Events**

1. The user selects two diagrams of the same type (class or sequence), both Papyrus files, to be merged.
2. The user clicks the “merge” button.
3. The user is prompted to choose either ‘Auto-Merge’ or ‘Manual Merge’. The user chooses ‘Manual-Merge’.
4. The webpage shows the two diagrams side by side. Under each diagram, there is a list of:
   1. Classes in diagram A only.
   2. Classes in diagram B only.
   3. Classes in common with diagram A and diagram B.
5. The user can select any pair of classes to merge if they are not already in common with both diagrams, e.g. class Bike in diagram A and class Bicycle in diagram B.
6. The user makes the selection of which elements to be kept in the merged class by checking a box (or some similar type of input) next to each desired element.
7. The user clicks another “merge” button to merge the class.
8. The user will be prompted to resolve any conflicts with associations in the newly merged class by picking which type of association is appropriate in each situation.
9. The user can repeat steps 4-7 to merge other classes which are not in common.
10. When the user is finished merging classes, the user clicks another Merge button to finalize the process and generate the overall merged diagram.
11. The UI displays a new merged diagram with the versions of each element combined.
12. The user can save the merged diagram.
13. The user can export Papyrus files of the merged diagram (see DownloadProject use case).
14. The user can comment on the merged diagram.
15. The user can return to the main ClubUML page after doing any combination of saving the drawing, exporting the merged diagram, or commenting.
16. The merged diagram will appear on the main ClubUML page along with the other diagrams that have been uploaded.

**Alternative Flows**

**Incorrect type of diagrams**

If user selects two different types of diagrams, then:

1. An error dialog will be prompted and ask the user to select two diagrams with the same type.
2. The use case returns to basic flow step 1.

**Number of diagrams error**

If user selects more or less than two diagrams, then:

1. An error dialog will be prompted and ask the user to select more or less diagrams to be merged.
2. The use case returns to basic flow step 1.

**Two exact diagrams are selected**

If user selects two diagrams that are exactly the same, then

1. A dialog will be prompted notifying user that the two diagrams are the same. Neither selections of merging nor saving of the diagram is allowed.
2. The use case returns to basic flow step 1.

**Subflows**

**No merging**

1. The user doesn’t make the request to merge diagrams
2. The user can return to the main ClubUML page, and the use case ends.

**No saving diagram**

1. The user chooses not to save an image of the merged diagram.
2. No saving will be done.

**No exporting XMI**

1. The user chooses not to export an XMI file of the merged diagram.
2. No exporting will be done.

**No commenting**

1. The user chooses not to leave any comment.
2. No comment will be added to the diagram.

**Key Scenarios**

No diagrams will be saved unless the user makes the request.

**Post-conditions**

1. The merged diagram is saved with comments, if any.
2. No diagrams are lost.

**Special Requirements**

The merged diagram has the same type as the two original diagrams.

The merged diagram can be merged again with other diagrams, if the user makes the request.