ClubUML Extension Proposal

Lauren DiCristofaro

1/15/14

ClubUML Extension: Rationale Management

Rationale Management can be included in the comparison, promotion, and selection of UML diagrams.

Accessing Rationale Management:

In the current tool, the user opens a project, selects “Go to compare”, selects “compare”, and promotes a diagram. Promoting a diagram should bring up a dialog box with a summary box to display in the current comments area.

Rationale Dialog Box: Summary, autofills *Alternative* as the compared diagram, select or create *Issue*, select or create (issue relationship), select or create *Criteria*, select or create (criteria relationship).

*Issues* and *Criteria* can be linked to a project management tool (Jira, Redmine, etc.).

Decisions: A member with the correct (administrator) permissions can choose or change a UML diagram to the current selected *Decision,* which will correspond to the current project goals and project documentation. Selecting a new *Decision* aggregates all *Rationales* that list the *Decision* as an alternative. At this point, rationale details can be added or culled to support a clean history and *Rationale* documentation generation. Current *Decisions* and Decision *History* documentation can be accessed through project details.

Rationale History Documentation Generation:

Rationale *History* and Documentation shows a user-friendly linked history diagram of the *Decisions* that led to the current *Decision*. This history will include the *Rationale* for each *Decision*, so it can link the related *Issues, Criteria,* and *Alternatives* to each *Decision*.

ClubUML Extension: Role and Permissions Extension

-User groups should be extended to allow separate organizations to work on different groups of projects.

-Authentication should be enforced so that a user cannot view any database details unless they are a member of the organization that owns those details.

-The application should store cookies to avoid the necessity of re-login at the login page, and those cookies should expire after a certain idle period.

-Role should be extended to include at least administrators vs. users. Extensions could possibly include read/write privileges or different team member roles.