Papyrus Model to Document Component Requirements

Goals:

This component must allow to the user to generate a documentation (text file, tables files, …) from an EMF or an UML model. So this component is mainly a generator.

This component must be extensible to provide new output formats in the futur.

This component must allow to the user to configure the generation.

This component must allow to the user to reverse some document changes in the initial model.

Here the list of the requirements:

The user must be able to configure the generation

The user must be able to define the output format.

The user must be able to define the generator to user.

The user must be able to define the styling template to use in the generated document.

The user must be able to define the structure of the document throw a kind of template. This template will allow him to define how to travel in the model. So the user must be able to define:

* EClass/UML Metaclass to find
* EStructureFeature/UML Property to use
* Stereotype to find
* Property of Stereotype to use
* Filters used to refine the found object

The user must be able to define for each of the previous step what the generator must do:

* Generate nothing
* Generate Title
* Generate a custom Title
* Generate Table
* Generate Paragraph
* Generate image for the view associated to the object (Diagram/table)
* Generate existing table View as Table
* Generate from a user selected element (so not necessary from the root of the model)

The user must be able to define the Main title of the generated document.

The framework must provide a way to manage versioning of the document.

The framework must provide a way to store an history of the generation.

This new framework must be fully integrated to Papyrus into to have the same behavior than Diagrams and Tables. Nevertheless, as the final document won’t be an element of the Model strictly speaking, it will be represented in Papyrus by an intermediate element which will know the final document.

* Creation of new document must use the architecture framework.
* generated documents must be visible from the Papyrus Model Explorer,
  + Standard operation must be supported: Open/Close/Destroy and probably rename
* generated documents must be visible in the Di View
* generated document must provide a Papyrus Property View, with the standard papyrus fields. These fields must be renamed for a better user understanding, but they must be here) :
  + Owner (owner used in the ModelExplorer) -> renamed into Graphical Owner
  + Root Element (the semantic parent of the element represented in the View (Diagram) -> renamed into Semantic Owner (read-only)
  + Description : a description of the generated document
  + Document kind (read-only) : indication of the nature of the document

The generator will use the label instead of the name of the UML element according to the value of the internationalization field defined in the Papyrus model.

TODO : a TextDocument must have :

Title

Authors

Version

List of figures

Sommaire

Body

Préface ?

PostFace ?

Attention aux nommages:

Actuellement, on développe surtout un parcours, pour une profondeur définie, mais :

Quid des profondeurs inconnues ?

Comment répétée un bout de l’arbre plusieurs fois/infini

On pourrait aussi décrire ce que l’on fait à chaqse fois que l’on rencontre un élément dans le parcours, quelques soit sa profondeur…

a) Table of contents;

b) Index;

c) Appendixes.