

TOPCASED / PAPYRUS / MDT UML CREATION OF AN OFFICIAL GRAPHICAL UML TOOL

Modeling - Talk
Modeling Symposium

Company : Anyware Technologies
Presenters : Jacques Lescot (jacques.lescot@anyware-tech.com)
David Sciamma (david.sciamma@anyware-tech.com)
Phone : +33(0)5 61 00 52 90

Modeling tools are more and more present in development of new projects. The UML language is already recognized as a standard to model these processes : the UML2 project provide an efficient API and a good basis for other tools but it remains quite difficult to create such UML model without using a graphical tool.

Today, many UML editors are available freely or not. Their current states are very variable : experimental or advanced status, more or less stable, compliant with other tools (naturally or using a transformation) or not, ...

Our experience with Eclipse environment and our knowledge on significant tools, let us imagine to create a uniform tool that will provide the best of the existing tools. Thus, we planned to merge the following projects :

- ✓ Topcased UML
- ✓ PapyrusUML
- ✓ Eclipse MDT UML2 Tools

We already made a study about the migration issues of UML Topcased (1.0.0) and PapyrusUML (0.7.0) projects to Eclipse UML2 Tools (1.1) project on their current state. We identified common functionalities, and moreover differences of UML Topcased and PapyrusUML projects that could be reused. Moreover, this will help us to clarify and specify our needs in term of UML modeling tools.

The two main points that should be considered are :

- ✓ The UML diagram degree of implementation : we intend to implements the 13 diagrams defined by the UML2 language.
- ✓ Cover the whole UML functionalities (from the UML OMG Specification : Superstructure version 2.1) and elements.

We noticed that the three tools are based on the UML2 project, this implies that the model representation is fully compliant between the tools. The differences are only due to the different implementation of the Diagram files. The migration implies converters based on the initial tools so that diagrams could be recreated.

This study shows us that the migration most important prerequisite is to gather requirement concerning the UML model and the tool functionalities. Another burning issue will be to determine the part of code that will be re-used.