Software modeling with UML and the KDE Umbrello tool

One step at a time

The Unified Modeling Language is the accepted standard for graphic modeling of object oriented programs. A KDE tool called Umbrello provides facilities for simple development of various diagram formats and conversion into code fragments. This helps to inform all those involved with a project of the overall design and interactions, In turn this saves time and money and can help to avoid problems.

BY RITA ORTENBURGER

visipix.com

Umbrello [1], a free GUI tool, helps software developers to use diagrams to describe program algorithms and architectures in Unified Modeling Language (UML, [2]). Graphic representations literally help to put everyone involved in the planning and realization of a project in the picture, particularly at the analysis and design stage.

THE AUTHOR

Rita Ortenburger has many years of experience as a freelance program developer with various languages (C, C++, Java) and multiple platforms (Unix derivatives, Linux, NT).

Honesty, aptly, seems to be one of Umbrello's major strengths. Umbrello provides useful documentation in various languages, and makes it quite clear that the tool is not a magical code generator that will convert sketches into ready-to-run programs.

On the other hand, it will generate source modules for Java, C++ , and PHP based on class diagrams. Modules will contain editable default comments and a class including the variables and method stems with parameters specified in Umbrello.

Multi-faceted Views

Umbrello also supports sequential diagrams for chronological processes, state diagrams for state machines (see Figure 1), activity diagrams, application case diagrams (Figure 2) and collaboration diagrams that show how program components will interface. If you are currently facing the task of representing a UML standard program graphically, you should also refer to the specification [4] defined by the OMG (Object Management Group, [3]).

Umbrello allows developers to position the diagram components on a desktop, in typical GUI fashion. The tool provides a toolbar full of icons and drag and drop functionality for this purpose. Unfortunately, desktop refresh was slightly untidy for a combination of Umbrello, KDE 3.0.0 and QT 3.0.3, leaving residual lines visible within the diagrams that needed another refresh to remove them.

The tree view in the left window panel (see Figure 3) shows the diagrams and elements completed so far. Diagrams and elements are shown in alphabetical order in a tree view below the top-level application case and logical views.

In other words, Umbrello does not assign application cases, actors and classes to the diagrams that use them. Instead they are handled as independent structural components that are assigned a globally unique name throughout the system.

Use of Folders to Organize the Class Tree

This is probably designed to prevent classes that occur in multiple diagrams from occupying too much space in the structural tree. However, the lack of information on the elements used in specific diagrams, and the lack of structure, may cause confusion in large-scale projects.

The wishlist for future Umbrello versions should contain a view menu that allows the users to switch between current alphabetic sorting and a future diagram-element tree. The new variant continue to display orphaned elements that are not used in any diagrams at diagram level, in order to draw the programmer's attention to the orphans.

Improving Model Readability

Users can add folders to the tree view to improve readability (using the *Logical view* drop-down). However, these branches of the tree view have no effect on the directory structure below the Umbrello target path where the tool will store the source code it generates. To store a module in a specific subfolder, you need to specify the directory name as the package name in the properties dialog box for the class.

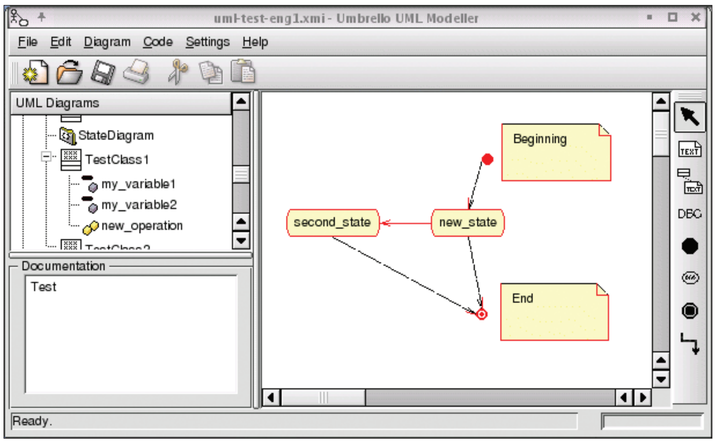

Figure 1: UML state diagrams describe objects in various states and the events that cause transitions between states

Elements can be created in at least three different ways: by clicking an icon in the toolbar on the right and then clicking on the desktop, using the *New* item in the drop-down menu (right mouse button on the desktop) or using the drop-down in the structure view. Existing elements that are appropriate to a diagram can be dragged from the tree view and dropped on the diagram. There is also a *Source code | Class Wizard*. Developers can use the source code menu to import existing C++ headers and avoid the need to create classes manually.

Changing modes

Names can be modified in the properties dialog box by double-clicking on an element or by clicking a component in the tree view. In the latter case the name of the structural element becomes a textbox.

Surprisingly, it is quite difficult to quit input mode and return to pointer mode. You might expect any mouse click outside the textbox to quit editing mode, but in fact you have to press [Return] to confirm your entry or [Escape] to cancel. Clicking on the documentation window or double clicking another structure element will also terminate input, although Umbrello will cancel your changes in this case.

On the subject of the documentation window, Umbrello provides help in a small static window, showing a comment on the element currently selected. Descriptions can be created and edited directly in the documentation window or the properties dialog. Assuming that the *Documentation comments* option is enabled in the Umbrello source code generation preferences, texts specified for classes will be incorporated in the source code.

Artistic Relationships

The major challenge when modeling a complex programming project is that of ascertaining and describing the relationships between the elements involved as coherently and clearly as possible.

It only makes sense for the toolbar to be mainly populated by icons for associations, depending on the kind of diagram you are creating.

One of example of this is the so-called generalization: in a class diagram a line with an simple arrow head points from a subclass to its base class. Umbrello will convert this inheritance association to a language specific notation in the source code for the target language. The association between the complete object and a

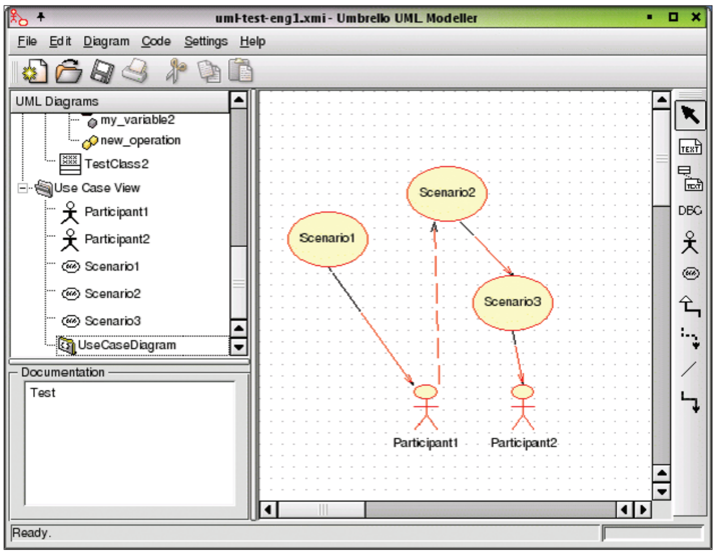

Figure 2: In the use case diagram, UML describes the associations and dependencies between various scenarios (use cases) and the actor. It describes what will take place but not how

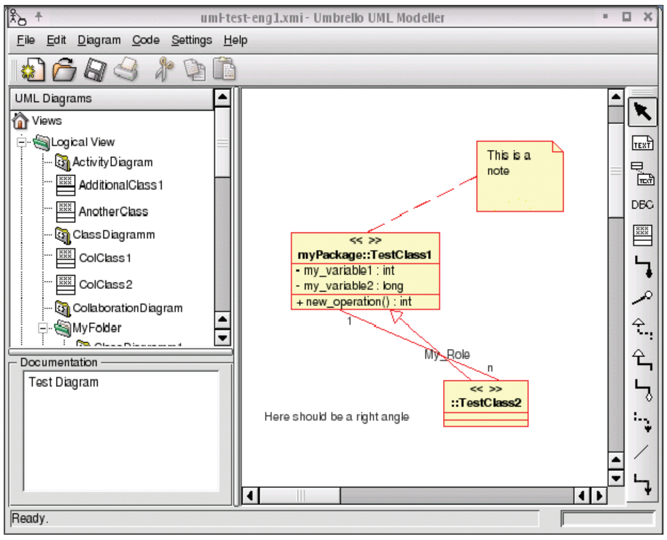

Figure 3: The Umbrello window is divided into multiple panels: the tree view on the left, the active UML diagram in the center, and the toolbar on the right

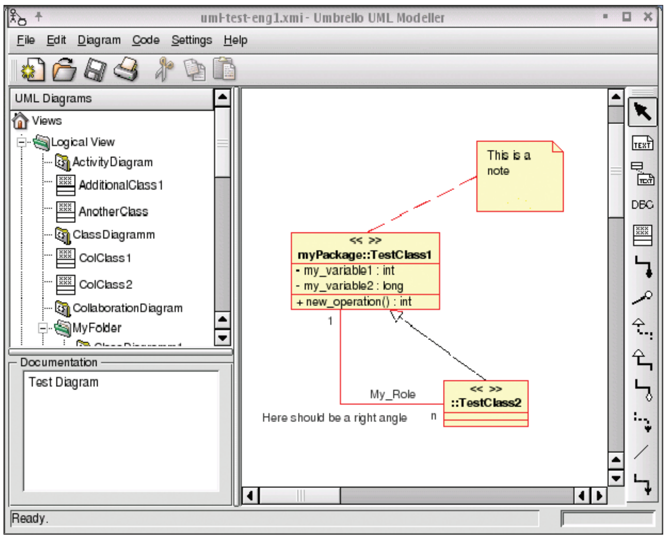

Figure 4: Umbrello can use angles to provide a better view of association lines. This effect is created by double clicking on the line to separate it, You should be aware of saving problems with angled lines

component (aggregation) will be rendered as a private variable of the component type in the sources for the class of the object.

Double clicking a line reveals one of the neater features of graphic associations in Umbrello: whereas this would typically open a properties dialog box, an additional handle appears on the line allowing you to re-route the line by clicking and dragging the handle (see Figure 4).

Strange behavior

Unfortunately, the current Umbrello version does not appear to save angled lines correctly, as an association with a 90 degree turn continually reverted to a straight line after restarting the tool.

Umbrello calculates the starting and endpoints of an association by reference to the number of links. This is why you cannot use the endpoints at the edge of a target object to move a line. Whether this behavior is useful in all cases, is debatable. After straightening out the right-angled association in our example, Umbrello went on to display two overlaid associations (see Figure 3).

One of the main candidates on the wishlist for future versions has to be an undo function. Element specific help on clicking the help button in a properties dialog box would also be useful – at present, this displays the table of contents from the manual.

Model Interchange

OMG has defined an interchange format called XMI (XML Metadata Interchange) for UML modeling data, and Umbrello stores diagrams in XMI format. However, when we attempted to import an XMI created by Umbrello into Oracle JDeveloper 9i development tool, the tool was unable to parse the XMI file.

Considering the fact that this brief overview did not look into the functionality provided in element specific pop-up menus, preferences for startup options, diagram type specific details and plausibility checks for associations between elements, assigning the version number 1.1 does smack of modest understatement.

Conclusion

Admittedly, the development team will need to put in a fair amount of work to help Umbrello mature into a professional tool. And instability in the form of frozen windows and a crash does tend to cloud the picture.

But having said that, version 1.1.1 is still highly recommended to developers looking to familiarize themselves with UML, by visualizing a small to medium-sized programming project. ■

Umbrello Update

Some of the issues mentioned in this article have been fixed in the current version, for example, the current version now has an undo function, drawing engine bugs have been fixed, and new source code generators have been added.

The new version is due to be released with KDE 3.2 in a few weeks time, and can be downloaded via CVS in the meantime. The project homepage provides further details [1].

Additionally, the Umbrello team has already embarked upon a complete re-write, which will eventually culminate in the release of Umbrello 2. Although work is in progress on the new version, it will definitely be a few months until a stable release is available. Some of the weaknesses pointed out in the article, such as the cluttered tree view of the model, have been target for revision in Umbrello 2.

The new development relies to a great extent on input from the user community; and the more ideas the community supplies, the more the developers can add to the new program design.

INFO

[1] Umbrello: *http://uml.sf.net/*
[2] Unified Modeling Language, UML: *http://www.uml.org/*
[3] Object Management Group, OMG: *http://www.omg.org/*
[4] UML specification: *http://www.omg.org/technology/documents/formal/uml.htm*