Enhancing the visual documentation artifacts in Envision

Project Description

Sascha Beat Dinkel

May 1, 2014

1 Background

Envision is a next generation development environment. It aims to improve the development experience with graphical representations of language constructs. Since the beginning of software development creating applications is basically writing source code in a text editor. Research has shown that code visualizations can benefit developers. The goal of Envision is to make program comprehension and navigation easier and therefore improve productivity by structuring the code graphically.
Apart from representing the executable code, Envision also has the ability to graphically support various kinds of software engineering artifacts. These artifacts are used to document the code and consist of enhanced comments (bold, italic etc.) as well as pictures and diagrams. This functionality has been introduced by Jonas Trappenberg in his Bachelor Thesis [1]. The goal of this thesis is to expand the documentation features of Envision.

2 Core Goals

2.1 Extend the documentation with tables

In addition to the already implemented features in the documentation artifacts, tables should be supported. Therefore the syntax for the documentation artifacts has to be extended to describe tables in a reasonable way.

2.2 Add support for adding source code blocks in comments

At the moment it is possible to have documentation artifacts in the source code, but it is not possible to have source code in the documentation artifacts. This feature should be added. The commented source code should then be displayed in an appropriate style using the standard code visualizations of Envision.

2.3 Improve the diagram editor

Envision already supports diagrams as a documentation artifact. This includes creating and editing diagrams consisting of simple shapes, connecting lines and text. The functionality of this diagram editor is sufficient but its usage is quite cumbersome. Therefore the diagram editor should be improved to make it easier to use. It should be conveniently possible to draw a diagram similarly to how it is done in standard diagram drawing programs like Microsoft Visio. To achieve that the usage of the mouse should be expanded and a toolbar with the different shapes and commands should be introduced.

2.4 Integrate documentation artifacts

Currently documentation can be inserted only at places where a statement can be inserted. Additionally it should be possible to integrate documentation artifacts with all programming constructs. For example a documentation artifact should describe a function.

3 Extensions

Once the core goals are completed we could explore these additional goals.

3.1 Generate and visualize Alloy models as documentation

Alloy is a language to describe specifications which can then be analysed by the Alloy tool. In the context of Envision, MIT's Alloy tool could be used to automatically generate Alloy models from code specifications. These models can then be visualized as a documentation artifact.

3.2 Recognition and visualization of standard patterns

Design patterns are often used in object-oriented software design. Envision could be extended in a way that it graphically displays the used patterns using the concrete information in the code. A UML class diagram of the pattern with the major components and relationships and a sequence diagram depicting the interactions between the different classes of the pattern should be supported.
This functionality can be achieved in the following two ways:

1. By automatically analyzing the class relationships and inferring what patterns are used there.

2. By manually annotating the code. These annotations could then be used by Envision to visualize the pattern.

The first option is clearly preferable since it does not require additional work from the user, on the other hand it requires a lot more effort to implement. Depending on the time available we will choose which option to explore.

3.3 Generate source code with Doxygen comments

Doxygen is a tool for generating software documentation using special comments in the code. Envision could be extended to generate source code with Doxygen comments using the information in its AST about the code and the documentation. In order to achieve this, the different documentation artifacts have to be translated into an appropriate Doxygen comment.

4 Schedule

This is a tentative time schedule of the project:

date	Description
2014.05.01	**Beginning of the thesis**
2014.05.02 - 2014.05.11	Get familiar with the Envision code
2014.05.12 - 2014.05.25	Implement the improvements of the diagram editor
2014.05.26 - 2014.06.08	Implement the source code blocks in comments
2014.06.09 - 2014.06.29	Implement the tables
2014.06.30 - 2014.07.20	Implement the integration of documentation artifacts
2014.07.20	**Milestone:** Finish implementation of the core goals
2014.07.21 - 2014.08.31	Implement the extensions
2014.08.31	**Milestone:** Finish implementation of the extensions
2014.09.01 - 2014.09.14	Write the Thesis
2014.09.15	**End of the thesis**

References

[1] Jonas Trappenberg, Bachelor Thesis, ETH Zürich, 2013
URL: http://www.pm.inf.ethz.ch/education/theses/student_docs/Jonas_Trappenberg/BA_report_Jonas_Trappenberg