

Supporting documentation artifacts in Envision

Project Description

Jonas Trappenberg

jonastr@ethz.ch

April 30, 2013

1 Background

Envision aims to become a next generation software development system. Instead of presenting the user with mere plain text, every language construct has a corresponding graphical representation. Plain text programming has proven to be invaluable over the past 60 years, but with evergrowing project sizes and therefore codebases, we begin to hit limits. A programmer that starts on a new project that has been running for years is usually confronted with a codebase too large to grasp in its entirety. So one has to ask: how can we enrich and enhance the understanding experience?

Envision aims to research how code can be visually represented in a meaningful way, in order to allow for a nicely structured code overview and therefore faster understanding than is possible with plain text.

Envision supports various language constructs, but thus far only those represented as executable code.

The goal of this project is to expand Envision by the ability to add and work with non-executable language constructs of various kinds to be able to document the program code with simple comments, diagrams, images, among others.

2 Core Goals

For simplicity, and because comments are historically of documenting nature, we will in the following be referring to all these *non-executable documenting artifacts* as *comments*.

For a successful completion of this project, at least the following kinds of *comments* shall be implemented:

- Textual *comments* that can be extended by rich formatting, such as bold, italic and underlined text, choosing different fonts and text sizes.
- Decorating *comments*, such as headers and lines, e.g. to add structure to very long *comments*.

- *Comments* which display an image that is loaded from disk.
- *Comments* that contain diagrams. It shall be possible to create these diagrams from within Envision, offering support for different shapes, colors, connectors and labels.
- Browser *comments*, i.e. an inline browser window that can dynamically fetch information from the internet, e.g. documentation.

Any two kinds of such *comments* shall be seamlessly mixable, such that any *comment* can be used to compose other *comments*. E.g. a diagram that uses richly formatted text labels, or *comments* that interleave text, images and diagrams. One exception thereof are browser *comments*, which can appear in other *comments*, but can themselves not consist of other *comments*.

3 Possible Extensions

Once the core goals have been achieved, some possible follow-up extensions include:

- Integrate MIT's Alloy Analyzer in order to display models of invariants that shall be specified using Code Contracts.
- Allow the programmer to mark classes to indicate what design patterns they participate in and which role they play. This information could then be enriched by automatically adding pattern diagrams as well as links to documentation.

4 Schedule

This is a tentative time schedule of the project:

days	deadline	
1	May 7	Finish project description and administrative tasks
4	May 14	Get familiar with Envision's code base
20	Jul 15	Implement core requirements
15	Aug 15	Tackle extensions
15	Sep 15	Write report and finalize extensions