

Task-specific views in Envision

Research in Computer Science Project description

Cyril Steimer

csteimer@student.ethz.ch

March 12, 2015

1 Introduction

Envision [1] is a visual programming tool, where a program's structure is presented using a combination of text and graphical objects. Most of today's mainstream programming interfaces work on purely textual and file based editors, often making a quick navigation or comprehension of programs cumbersome. By using graphical constructs and arranging the entire program on a single plane, Envision aims to achieve the facilitation of both quick navigation and comprehension of programs.

Envision currently has only a single view displaying the entire project on a large surface. In the following we will call this the *architecture view*. This architecture view makes the entire project and all its components visible to the user, allowing them to access all the information they might need. However in practice, often the programmer is interested only in a small subset of the entire project while working on some specific task, the so-called *working set*.

Within the context of this project, we will identify tasks for which a more specific view (which we will call *task view*) is useful. In a next step, we must find the working set necessary to solve such tasks and discover ways of displaying this information in a way that is beneficial to the user when working on the tasks. This can be done for example by visually indicating control flow through methods when the user is trying to debug a certain method. Lastly, these task views will be implemented within Envision. These task views are not meant to replace the current architecture view, but rather to complement it.

2 Goals

There are three main goals within this project:

1. Identify the tasks for which a task specific view should be created. Some such tasks are the implementation of a new feature, fixing a bug, debugging an error

message or efficient browsing through the code. The possible tasks are not limited to this list, but can be extended within the course of the project, if more tasks are identified.

Even more important than identifying the tasks themselves, however, is to identify the information necessary to solve these tasks. For one, it is important to identify the necessary components such as classes or methods. Furthermore, often also relations between these components can be very important to solve a task, such as a method referencing another method.

2. Design a concept for the task view necessary for the different tasks. Designing such a concept can be broken down into different subtasks:
 - a) Coming up with a visualization of the working set and the possible user interactions. The visualization should be closely related to the interactions, making them as effortless as possible.
 - b) Finding a way of realizing a uniform design over as many scenarios as possible, ideally even across all tasks that are identified within this project. For one, this allows for greater recognition by the user, as the task view will look familiar for all the different tasks. Furthermore, it makes the task of implementation within a single framework possible.
3. Finally the task view for the identified tasks must be implemented within Envision. The focus here should be on implementing a single framework which supports multiple tasks, where the view for each task is simply an instance of the framework. Putting the focus strongly on reusability allows visualizations for additional tasks to be introduced further down the road with little effort.

These goals should be targeted in an iterative process. Concepts for task views should be designed and implemented relatively quickly, before evaluating them and using the results of that evaluation for improving the concept.

3 Deliverables

The deliverables will consist of the following parts:

- The task view concepts and their implementation within Envision.
- A demo of the new task views within Envision.
- A written final project report.
- An initial project presentation within the first month after the start of the project, and a final presentation at the end of the project.

4 Schedule

Date	Milestone
2015-03-16	Project Start
2015-03-23	Getting familiar with Envision's architecture view
2015-04-06	Initial design mock-up
2015-04-27	Implementation of mock-up
2015-05-11	Evaluation & improvement of design
2015-05-25	Implementation of improvements
2015-06-01	Evaluation of task view
2015-06-15	Final presentation and project report

5 Bibliography

References

- [1] D. Asenov and P. Müller. Envision: A fast and flexible visual code editor with fluid interactions (overview). In *Visual Languages and Human-Centric Computing (VL/HCC)*, pages 9–12, 2014.