Master thesis in Computer Science
Fall 2010
Dimitar Asenov
Supervisor: Prof. Dr. Peter Müller

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Design and Implementation of *Envision* - a Visual Programming System

Description and schedule

October 12, 2010

1 Introduction

This Master thesis is a follow up research on "A feasibility study for a general-purpose visual programming system" [1]. It continues the development of the *Envision* visual programming system. This work constitutes further implementation of core features in order to support quantitative experiments with programmers.

2 Motivation and Goals

The initial feasibility study for *Envision* showed that the concept has a lot of potential to improve the programming experience. As the original research report suggests, it is essential to verify this by conducting experiments with programmers. This will allow stronger claims to be made about the improvements offered by our approach. It will also contribute some new experimental results in the field of visual programming - an area marked by a significant scarcity of empirical research [2].

In order to perform these studies however it is first necessary to further develop the implementation of *Envision*. The current prototype lacks essential features for the programming process and is built using an ad-hoc architecture. The core goal for this Master thesis is to define the underlying architecture for *Envision* and use it to implement essential features that will enable empirical studies with programmers. As time permits additional extensions to the core goal will also be included in the thesis.

The expected result is to have an implementation that demonstrates the benefits of our approach to visual programming in a measurable way. It should be suitable for conducting a scenario based empirical study where participants will have to perform typical programming tasks such as understanding new code, fixing a bug, refactoring, implementing a new feature, etc. As a possible extension such a study could be carried out and the results can be published.

The core goals and extensions are described in more details in the next two sections.

3 Core targets

3.1 Architecture

Designing a flexible architecture for the implementation of *Envision* is essential in order to facilitate coordinated and rapid implementation of key features. The main requirement for the architecture is that it should be based on different modules and extensible by user plug-ins. Three activities comprise the architecture design phase:

- **Requirements definition** - The major requirements for *Envision* need to be identified and reflected in the architecture design.
- **Modules and module interaction definition** - Based on the requirements a system with different module systems should be designed. The behavior of each type of module should be defined as well as interaction between the modules.
- **Documentation** - All relevant aspects of this process should be documented in a way that facilitates understanding of design choices and future work on the architecture.

The lessons learned during the development of the feasibility study prototype should be taken in consideration when making the new architecture.

3.2 Implementation

Once the architecture design phase is finished the implementation of key features of *Envision* can begin.

This will no longer be a prototype, but the first public version of the programming system. The main differences to the existing prototype will be:

- **Architecture** The new implementation will follow a well defined and carefully designed architecture. This will facilitate the development and further expansion of the software.
- **Features** Many new features will be implemented to bring the new implementation to a state usable for empirical studies. See below for details.
- **Higher quality implementation** The new implementation will be of production quality. In particular the code should be well documented and tested. This will allow to more easily distribute *Envision* as an open source project in the future.

The choice of features to implement is based on the assumption that empirical studies focusing on the visual interface of *Envision* will be conducted. Therefore functionality such as semantic versioning or content linking will not be implemented. Rather the focus will be on:

- **Application model** The application model (similar to an abstract syntax tree) will be fully developed and include a wide variety of programming constructs.
- **Visualization** A visualization module will be developed that can show all constructs from the application model.
- **Interaction** Interaction with all visualized program elements and built-in IDE tools (e.g. search) will be possible.
- **Persistence** The IDE will allow for the program under development to be saved or loaded from a persistent storage (disk file, database, etc.).
- **Compilation for Java** Once an application is developed inside *Envision* it will be possible to automatically create a Java equivalent. Only the more common Java features will be supported at this stage.

Overall the implementation should allow one to develop and work on a small-sized application - up to about 1000 lines of code. The idea is that such an application will be used during the empirical studies.

4 Possible extensions

As time permits some of the following extensions will be incorporated in the thesis after the work on the core is finished.

4.1 User studies with programmers

The goal of this extension is to provide concrete results about the benefits offered by the visual approach of *Envision*. The idea is to conduct studies with programmers where each participant will complete a number of typical programming tasks such as refactoring, fixing a bug or adding a new feature. Participants will perform these tasks either using *Envision* or a standard text-based

IDE such as Eclipse. The measurable quantities will be the time it takes to complete the task as well as whether the task was correctly completed or not.

The tasks for this study will be associated with a small toy application developed in *Envision*.

4.2 Publication

The concept of *Envision* and the results of the quantitative studies can be published as a workshop or a conference paper.

4.3 Custom visualization plug-in

To highlight the flexibility of the designed architecture a custom visualization plug-in can be developed. The plug-in can extend a new visual element or alter an existing one. This could be something like a stylized comments for example. It should also demonstrate how plug-ins are integrated into the system.

5 Time planning

This is a preliminary estimation of the time allocated for each activity:

Activity	**time**
Architecture design	1 Month
Core implementation	3 Months
Extensions and final writeup	2 Months
Total	**6 Months**

References

[1] Dimitar Asenov. *A feasibility study for a general-purpose visual programming system*, 2010. (accessed October 4th, 2010) `http://www.pm.inf.ethz.ch/education/theses/student_docs/Asenov_Dimitar/Report`.

[2] K.N. Whitley. Visual programming languages and the empirical evidence for and against. *Journal of Visual Languages and Computing*, 8(1):109–142, 1997.