Research in Computer Science II
Spring 2010
Dimitar Asenov

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

A feasibility study for a general-purpose visual programming system

16th February, 2010

Revision History

Date	Author	Version	Change Reference
2010-02-14	Dimitar Asenov	0.1	Initial draft
2010-02-16	Dimitar Asenov	1.0	Updated to reflect the plan established at the last meeting, adjusted the title, added a brief description of *Envision*

Contents

1 Introduction

This document describes an initial feasibility study to be carried out for the *Envision* project (see section 1.1). The study marks the project's beginning and is to be conducted during a "Research in Computer Science II" course at ETH in the Spring semester of 2010.

1.1 A brief description of the *Envision* system

Envision is a new platform for software development. It promotes a more interactive visually oriented programming style, integration of source code and documentation, better versioning control and more. It is a bold project that currently exists only on a conceptual level and its core ideas are still under discussion.

Some of the key features of *Envision* are:

- **Source code representation** - The programmer does not work directly with the source code but she is shown a representation of it. This can be any combination of textual and visual elements. Using a visually oriented representation can improve program comprehension, code navigation and retention.
- **Information integration** - Include graphs, animations, comments, contracts, software requirements, documentation, etc. directly in the "source code". Use one system for all software artifacts.
- **Content linking** - When all the information is integrated, it can be dynamically linked, so that changes in one component reflect in another linked component. This will reduce maintenance overhead and will improve the consistency of the various parts of the software product.
- **Higher expressiveness** - Allow for various programming constructs and compile them to the target language without the user's involvement.
- **Fine-grained versioning** - Keep track of all versions of methods, classes, functions, modules and other programming elements. This allows for a more controlled program evolution, additional verification conditions and can help comprehension of mature systems.

2 Motivation and goals

Currently only a very general description for the *Envision* project is available. It outlines its vision by providing some central new ideas and giving a few interesting use cases. However it also raises many questions about the features, design and implementation of the system. The feasibility study described in this paper is meant to provide answers to a few of those questions.

The purpose of the study is twofold:

1. To investigate some key concepts of *Envision* and provide information that can be used to specify the system in more detail.
2. To evaluate the proposed system and compare it with existing programming methods and development solutions.

Below these two aspects are described in more detail.

2.1 Investigated concepts

This study aims to develop a better understanding and further specification of the following key components of *Envision*:

- Source code representation
  What are good representations? What visualizations are useful for writing new code or understanding existing one? How can multiple representation modes enhance the programming activity?
- Programing and user input
  Given a visual representation, how can the user provide input to the system? Can this be made as efficient and as easy to use as text-based systems? How is the input transformed to an image?
- Error visualization
  How are errors reported in a visual system? How can different types of errors be visualized to aid the developer? What benefits does a visual approach provide for debugging?

2.2 Performance evaluation

A key question throughout the *Envision* project is: How does the proposed system compare against existing solutions? After the core concepts have been specified in more detail their performance in a number of use cases should be evaluated and compared to existing solutions. Here are a few possible factors for investigation:

- Program comprehension
- Code manipulation
- Code navigation
- Consistency between code and documentation

3 Investigation methods

The investigation approach differs in the two components of the study.

3.1 Concepts

The following three activities form the investigation methods for the conceptual part of the study.

3.1.1 Familiarization with relevant research

Envision is meant to provide a new platform for visual programming. Thus a good understanding of related research fields together with knowledge of the latest developments in those fields are required. Part of this study is to get acquainted with the latest relevant research in the following disciplines:

- Visual Programming
- Software Visualization
- Human-Computer Interaction

3.1.2 Further development of key ideas

The existing ideas for software development with *Envision* should be further developed. This includes specifying additional design details and identifying problems and challenges.

During this activity the corresponding research should be taken into consideration to arrive at better solution.

3.1.3 Development of use cases and solutions

Through this activity a number of different use cases with corresponding solutions using *Envision* should be developed. At this initial stage the focus will be on more basic scenarios. This can include:

- Input and visualization of basic language structures.
- Navigating code.
- Manipulation of code blocks.
- Displaying and correcting errors.

The different use cases will be briefly specified with a text description. The corresponding solution might include sketches, diagrams, source code and should be well explained.

3.2 Use case performance evaluation

The performance of the *Envision* approach should be evaluated by comparing it to existing solutions. For this purpose several use cases should be developed that will serve as a basis for the analysis. These must be typical situations that arise during software development.

To support the evaluation an early *Envision* IDE mock-up application should be created. Due to time constraints this application will only include the features that are absolutely necessary for this study. Its focus will be the visual representation of programs and the way the user provides input to the system.

4 Expected results

This study will result in a written report that will contain the following:

- A summary of the current state of the art in Visual Programming and Software Visualization and related research
- A detailed description of the following components of *Envision*
  - Visual code representation
  - User input
  - Error visualization
- An evaluation of the viability of this new visual approach and a comparison to existing solutions based on concrete use cases

The mock-up application's source code is also part of the final deliverables.

5 Schedule and milestones

Date	**Milestone**	**Description**
2010-03-11	Initial project presentation	
2010-04-01	Summary of current research	
2010-04-01	Preliminary description of key components	
2010-05-15	IDE mock-up	
2010-05-30	Experiments and evaluation	
2010-05-30	Final description of key components	
2010-05-31	Final presentation and project report	