

# Visualizing graphs with Vision2

## PROJECT PLAN

Semester project

Project period: SS-2006

Student name: Lukas Angerer

Status: 6-th semester

Email address: [angererl@student.ethz.ch](mailto:angererl@student.ethz.ch)

Supervisor name: Michela Pedroni

## 1. PROJECT DESCRIPTION

### *Overview*

Graphs are an essential part of software engineering as well as many other domains. The proper layout of a given graph structure can be quite challenging and time consuming and for this purpose many applications and libraries have been developed to simplify this process. One of them is the Graphviz [3] library, which uses the DOT language as abstract representation of graphs and has been selected as the basis for this project. The DOT language is a file format that allows to specify directed and undirected graphs in a plain (human readable/writable) fashion. Such a DOT file can be tagged with attributes describing the layout of the contained elements and such a tagged file will be referred to as an “attributed DOT file”.

The idea of the project is to provide an Eiffel Vision2 widget to display and modify graphs. A graph given by an abstract description in the DOT language will serve as input. The open source Graphviz library with its graph layout algorithms will be employed to create an initial layout of the graph yielding an attributed DOT file as intermediate output. This attributed DOT file will then be rendered within a Vision2 widget, where the user of the widget can modify the graph layout via an intuitive mouse interface. The modified graph can then be saved as a DOT file or any other output format supported by the Graphviz library. All interactions of the program with the Graphviz library will of course be hidden from the user.

### *Scope of the work*

The scope of this project will be the creation of a prototype of the widget as explained in the section “Overview”. Prototype here means that the widget should be able to display and modify very simple graphs that only contain a subset of the primitives that are supported by the Graphviz library, but the system should be constructed in a way that allows simple extension with other primitives.

### *Intended results*

The final result of the project is an Eiffel Vision2 widget to display and modify graphs in an intuitive way. The widget does not have to support the full range of primitives and functionalities provided by the Graphviz library since that would exceed the scope of a semester project, but it has to be designed with that goal in mind. Furthermore the project will contain an example application showing the capabilities of the widget.

## 2. BACKGROUND MATERIAL

### *Reading list*

Information about the DOT language

- The DOT Language: <http://www.graphviz.org/doc/info/lang.html>
- Graph Attributes: <http://www.graphviz.org/doc/info/attrs.html>
- Node Shapes: <http://www.graphviz.org/doc/info/shapes.html>
- Arrow Shapes: <http://www.graphviz.org/doc/info/arrows.html>

General Graphviz documentation:

- <http://www.graphviz.org/Documentation/Agaph.pdf>
- <http://www.graphviz.org/doc/libguide/libguide.pdf>

Vision2 programming:

- <http://docs.eiffel.com>
- Object-Oriented Software Construction, 2nd edition [2]

### 3. PROJECT MANAGEMENT

#### *Objectives and priorities*

The main objective is to build a widget that can render and modify simple graphs given by DOT files, and to build it in a way that allows a simple extension to encompass the full set of primitives supported by Graphviz. The next priority is to design the underlying structure of the widget in a way that allows an easy adaption to other graphical libraries like WEL or EiffelMedia. If all this is achieved, the system can be extended with a growing subset of primitives until full compatibility with the DOT language (and the attributed dot format used by Graphviz) is reached.

#### *Criteria for success*

The project is to be considered a success if the widget is able to display several sample graphs (using a subset of primitives supported by Graphviz) and save a user-modified version of these graphs in attributed DOT format.

#### *Method of work*

The first task will be to research the capabilities and limits of the Graphviz library and to gain insight in the process of Vision2 widgets (see reading list).

#### *Quality management*

##### **Documentation**

There will be an overview over the selected architecture of the system, as well as a separate documentation of the API the widget will provide, and the source code will of course also be documented.

##### **Validation steps**

To guarantee the quality of the created system, there will be a test-phase near the end of the project where the project supervisor (Michela Pedroni) and probably some other test-subjects will test the example application containing the widget.

### 4. PLAN WITH MILESTONES

#### *Project steps*

31.05.2006	Project plan
30.06.2006	Research
07.07.2006	Design
25.08.2006	Implementation

15.09.2006 Test / project end

### *Deadline*

15.09.2006

### *Tentative schedule*

<i>Name</i>	<i>Start</i>	<i>Finish</i>	<i>Work</i>
Write project plan	May 19	May 20	1d
Research Graphviz capabilities	Jun 2	Jun 10	2d
Research Vision2 widgets	Jun 10	Jun 30	3d
Design system	Jul10	Jul 13	4d
Implement system	Jul 17	Aug 9	5d
Test system (bug fixing)	Aug 10	Aug 17	5d

## **REFERENCES**

- [1] Chair of Software Engineering: *Semester-/Diplomarbeiten*; Online at: <http://se.inf.ethz.ch/projects/index.html>, consulted in October 2002.
- [2] Bertrand Meyer: *Object-Oriented Software Construction, 2nd edition*, Prentice Hall, 1997.
- [3] Graphviz – Graph Visualization Software: <http://www.graphviz.org/>