1 Introduction

Graph visualisations enable humans to perceive information represented by graph structures, in an orderly manner. GraphViz [1] presents both graph and digraph visualisations in 2D. These visualisations can be generated from a single specification and be generated according to different layouts. Examples of different graph representation layouts in GraphViz are given in Figure 1.

This project proposal requires 3D digraph visualisations to be generated from a single specification taking different criteria into account. These visualisations will be presented in 3D space either on a hyper-plane or a hyper-sphere.

2 Project vision and objectives

Work with regards to spatial, three-dimensional, visualisations of graphs has increased with graphics engines becoming more powerful. Interaction with graph models is also becoming more popular.

The problem to be addressed in this project is the development of a layout manager that displays the visualisation of the digraph on either a plane or a sphere. The viewer of the modelled digraph should also be able to interact with the model and manipulate the model.

The visualisations must make provision for at least digraphs which have been defined using the set of triple notation as proposed by Barla-Szabo et al. [2], implemented by Koopman [5] in a toolkit and applied by Marshall [6] to specify curricula as digraphs. The viewer (user) should be able to change their viewpoint to any point on or beyond the plane or sphere and view the visualisation from this vantage point.

The labels of edges of the digraphs are represented as tuples containing meta-data about the edges, this meta-data should be linked according to user-defined criteria and presented in relation to the hyper-shape (plane or sphere) by applying algorithms that ‘force direct’ the vertices linked on either side of the edge. Research into algorithms for
drawing ‘force direct’-ed graphs have been defined [4] [7] [3] and can be investigated and applied in this project.

3 Project owner

The project will remain the property of the University of Pretoria. The members of the team who accept this project will be members of the CSEDAR research group which is housed in the Computer Science department.

4 Project scope

The following basic functionality is required:

- The description of the digraph, specified as a set of triples is read into the visualiser. Digraph vertices are depicted as labelled spheres and the edges between vertices are labelled using an n-tuple representing data associated with the edge.

- Digraphs are either placed on a hyper-plane or a hyper-sphere. The hyper-plane visualisation is similar to the 2D GraphViz models.

- The viewer must be allowed to navigate the hyper-shape.
  - change the viewpoint and zoom in and out
navigate the digraph by selected vertices(s) and following edge(s). Navigation
may be constrained by user defined criteria.

- select topics of interest and drill down into the details of the selection

- The viewer must be able to set criteria, such as parameters for force-directed-
ion, for the generation of meta-graphs. The criteria will be defined using the data presented
in the labels of the edges.

Additional functionality to ‘edit’ the visualisation of the digraph and save the updated
specification as a file represented by a set of triples may be implemented if time permits.

5 Architectural requirements

Initially, the visualisations need to be presented on a single desktop computer. Once it
has been shown that this goal is attainable and if time permits, the visualisations can be
ported to work on a client-server architecture.

The visualisations must be manipulated in real-time. Digraphs may be large, 1000 or
more vertices, which needs to be rendered on a standard computer screen.

It would be preferable if the initial desktop system be written in C++. This require-
ment however is open for negotiation. The system however must execute on Linux and
Mac OS X.

6 Skills requirements

It is assumed that the team members can and will learn the technical skills related to
text parsing, and the necessary algorithms to render the digraphs in a pleasing manner.
Team members need to understand that some members will excel in specific areas of the
project, but that a coherent team will result in a successful project.

7 Development and Roll-out

7.1 Incremental development

The system design has to provide for incremental development. Basic functionality has
to be implemented first. The system design has to provide for modular development
of additional functionality. All documentation, test-harnesses, source code and other
artifacts required by the COS301 management team are required by the client as well.

7.2 On-site customer

The client, being a member of staff in the Department of Computer Science, is therefore
available in the capacity of an on-site customer. It would be preferable for the team
to meet on a bi-weekly basis. All team members are required to attend this meeting.
Decisions made at these meetings will be minuted and actions need to be taken before the
next meeting. These meetings may result in changes to the requirements of the project.
7.3 Intellectual ownership

If successful, the project artifacts should be released into the public/open-source domain by creating a SourceForge project.

References


