



Vectorial Preview



Difficulty: Medium

Required skills: Java, SVG basics

Gephi is able to export its network as SVG or PDF format. Using vectorial drawings for graph have many benefits like infinite zooming or clear shapes. The aim of this proposal is to develop a preview module in Gephi for seeing how the output will exactly looks like with the given parameters.

Whereas the embedded 3D engine is designed for efficient network exploration the vectorial export concentrate on clarity, readability and outstanding design. This module has many different settings like edge thickness, arrow size, white borders around labels, node border and so on. In the vein of WYSIWYG editors we would like to see direct setting change on a Vectorial Preview window. The module must be scalable to support large networks.

To resume there are two requirements, previewing the network exactly how it will looks-like in output and support huge graphs efficiently.

Technically the aim is to display the network exactly as it will be when opening the SVG, PDF or EPS in a postscript editor (Inkscape, Adobe Illustrator). This can be done with different approach, the student will have to compare and choose the most suitable. We propose following approach but we are open for your ideas as well.

Approach #1:

- Experiment existing SVG and Postscript Java interpreter. Use the output generated from Gephi Rich SVG Export module to reconstruct and display the network.

Approach #2:

- If the first approach fails, more things will have to be done from the root. A basic Java 2D previewer can be built for displaying the network. Given the fact the only objects we use are basic shapes, edges and labels, developing a mini interpreter isn't unattainable. Actually this approach is similar to creating a 2D graph visualization module, which is well documented on Internet.

Resources:

- Gephi Rich SVG Export (<http://gephi.org/2009/rich-svg-export>)
- Batik (<http://xmlgraphics.apache.org/batik>)
- Java2D (<http://java.sun.com/products/java-media/2D/index.jsp>)