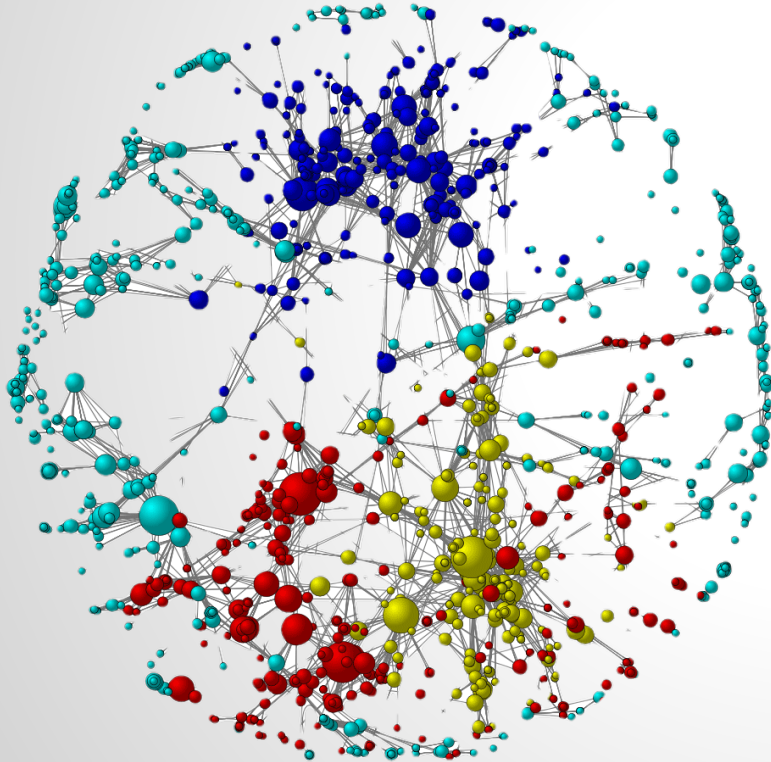


Gephi



Team: Hypergraphs

Aggelos Gickas <aggelosGks@gmail.com>

Charalampos Tsiligiannis <htsiligiannis@gmail.com>

Welcome!

- Introduction to Gephi
- Why we picked Gephi?
- Gephi's Architecture
- Type of contribution
- Progress plan

Introduction to Gephi

Gephi is...

- A project started in 2007 by Matthew Bastian
- An Exploratory Network Analysis and Visualization Tool
- An open-source platform licensed under GPL v3
- Written in Java, Swing, OpenGL

Introduction to Gephi

The architecture and functionality of Gephi offers :

- Performance, Usability and Modularity
- The platform, easy to use and extend
- Networks. No diagrams or other visualizations.
- Standards and open innovation

Why we picked Gephi?

- Networks exists everywhere both mentally and actually
 - Computer Networks,Business Networks,Geographic Networks Power grid
 - Nowadays increasing importance is evidenced : Social Networks, Biological Networks
- Many application related to networks
 - Better support for decision making
 - Accurate metrics for development and expansion
 - Research and development in medicine and biology
 - Artificial Intelligence (Neural Networks)
- As sector expands visualization is demanded.
 - Information Display
 - Information Retrieval

Why we picked Gephi?

- After our familiarity with the software for personal use (analyze graphs to lead to results)
 - Room for optimization and improvement
 - Utilities are missing both in functionality and user experience side.
- Great experience with Java programming in many projects so far, allow us to focus directly to contribution.
 - Significant contribution
 - Focus on quality

Gephi's Architecture

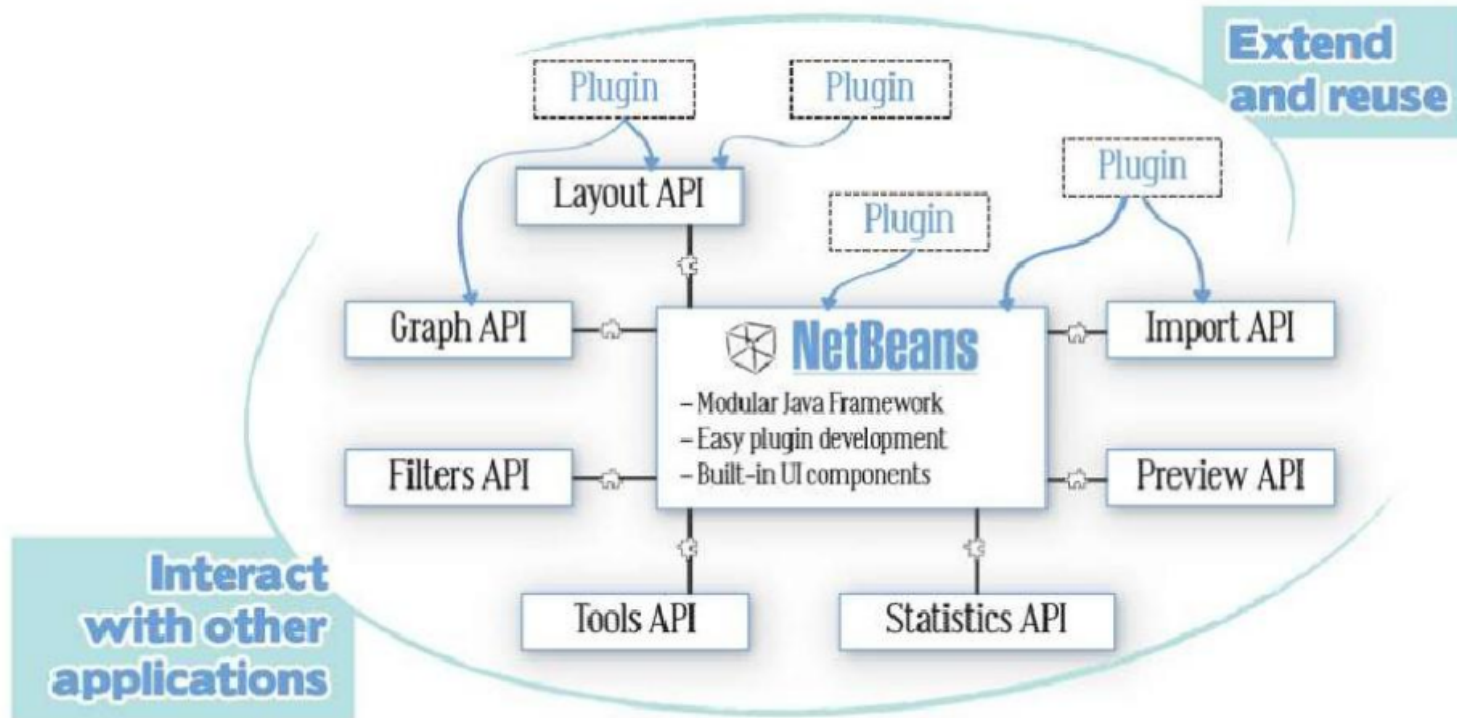
- Gephi has a modular and extensible architecture
- Gephi is built on top of Netbeans Platform
 - Netbeans platform provides to Gephi the module system, the window system, the localization system, the preferences system and more...
- Modules expose public APIs and depend on each other (without circular dependencies)
- Gephi follows the MVC design pattern

Gephi's Architecture

SPI vs API

- SPI
 - SPIs define how Gephi can be extended. For example a plug-in extends the Layout interface to add a new layout.
- API
 - Each API has a role and a set of packages other modules can use. For instance a module call the LayoutAPI to run a layout algorithm for 100 iterations.

Gephi's Architecture



Type of contribution

Having read the ideal list in the wiki section of the project(Github), we focused on two specific categories of things that can be improved. Our main goal is

- The first category emphasizes on changes that can be implemented to improve the user experience ,including visualisation, utilities and edit tools. (e.g delete options applied on graph, quick shortcut fixes)
- The second category emphasizes on changes that can be implemented to improve the functionality, including new features (metrics on graph, new quantitative methods) about networks and performance within time interval.

Type of contribution

User Experience

- Tools
 - Zoom in / Zoom out
 - Node Eraser: removing nodes under the cursor's click surface; the diameter of the surface is variable (like Photoshop)
 - Undo last action
- Usability
 - Fullscreen mode with only the graph, and maybe the timeline.
 - Copy/paste from an opened Excel file
 - Copy/Paste selection in Edit menu

Type of contribution

Functionality

- Statistics
 - Add the support of edge weight in the centrality measures: betweenness, closeness, eccentricity
 - When computing Shortest Paths, add an option in the metrics panel to save the paths in a new attribute column (of type String List)
- Graph
 - Create adjacency matrix from Graph
- Reliance
 - Autosave the current project every xx minutes
- Dynamics
 - Video recording, with output either video format or a series of PNG
- Layouts
 - Bipartite (and multipartite) graph layout

Progress plan

April

- Read the javadoc and the documentation of the project
- Understanding the key elements that compose the software
- Detect ways the software can be extended
- Fix a few small bugs in order to decide where our main contribution will focus

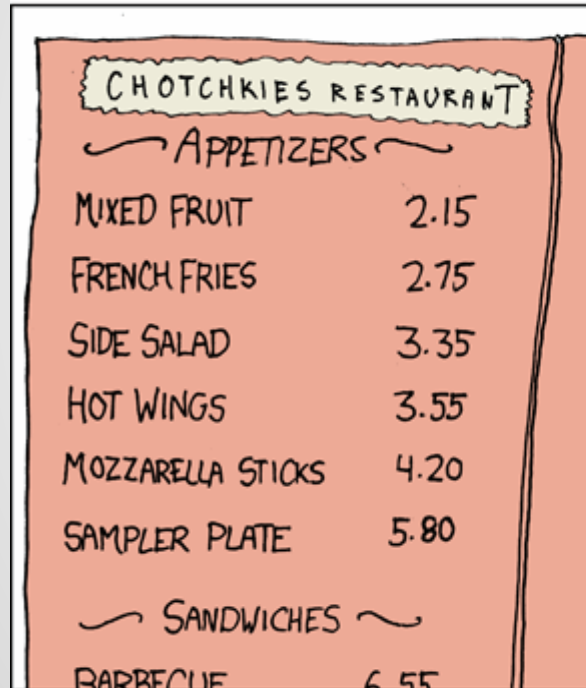
Progress plan

May

- Work on the main contribution
 - Meet the coding and designing standards
 - Developing process
 - Unit test our code
 - Create documentation of our code
- Pull request
 - Improve user experience and functionality
 - Give back to the community

Hypergraphs

Thank you! Questions?



CHOTCHKIES RESTAURANT

— APPETIZERS —

MIXED FRUIT	2.15
FRENCH FRIES	2.75
SIDE SALAD	3.35
HOT WINGS	3.55
MOZZARELLA STICKS	4.20
SAMPLER PLATE	5.80

— SANDWICHES —

BARBECUE	6.55
----------	------

