Graph visualization using OpenSceneGraph and Eclipse

Currently a bio informatics research project is underway in which among others aims to visualize large biological networks. The MOVE (Multiple Ontology Visualization and Exploration system) has been developed to mine information contained in a set of biological graphs. The system uses a number of modules which implement specific functions. The visualization module represents a graph as a network. This network is constructed dynamically from the underlying graph model.

The user can manipulate the underlying data model as well as the visualization. Some interesting features that the visualization will support:

- Graph element (node/edge) selection.
- Automatic layout of selected graph areas.
- Mapping of data model attributes to visual attributes (e.g., Gene type to node fill color mapping)
- Managing multiple network visualizations of a single graph or different graphs.
- Fast global graph panning and zooming actions
- Graph complexity management: a.o folding groups of nodes together.

One of the most difficult problems of interactive graph visualization is getting enough performance out of the application. An exciting new framework to create fast visualizations using Open-GL is Open Scene Graph (OSG). While OSG performs the actual network rendering a solid GUI is needed to control and manipulate the visualization. A consistent GUI is provided by Eclipse, a general purpose editor, which can be extended through plugins. The overall architecture of the complete system is presented in the figure at bottom of this project proposal. You will be responsible for the design of the "view module" (depicted in orange).

Your project:
Develop an Eclipse (view) plugin in which visualizes the graph provided by the system. Subtasks:

- Integrate OSG in a plugin. (Some tests have already been conducted)
- Build a OSG scene structure based on the input graph
- Design (Eclipse) navigation and manipulation actions.
- Multiple camera support of a single graph. This should be relatively easy to design as OSG already supports multiple cameras.
- Extension: use OSG open-gl effects to create distortion lenses.

If desired a workplace with computer is available at the Biological Center in Haren.

Requirements:
Good understanding of Java. No experience is needed in Eclipse or OSG.

Contact:
J.B.T.M. Roerdink: roe@cs.rug.nl, D.W.J. Bosman: dinne@cs.rug.nl