Reducing the Time Complexity of Node Selection Operation in SBGNViz.js by Making Use of Quadtree Data Structure

SBGNViz.js is a web application based on cytoscape.js to visualize the pathway models represented by SBGN process description notation. SBGNViz.js accepts the pathway models represented in SBGN-ML format[2].

Cytoscape.js is an open-source graph theory (a.k.a. network) library written in JavaScript. Cytoscape.js can be used for graph analysis and visualisation[1].

SBGNViz.js perform the graph operations by making use of cytoscape.js and cytoscape.js enables the selection of the graph elements that are nodes and edges in linear time by calling checkPoint function, a function performing in constant time, it is defined for each node type in cytoscape.js and returns whether the given point is inside that node shape.

In this project we are proposing to decrease the time complexity of node selection operation from linear time to logarithmic time by constructing a region quad tree to store the nodes information.

In SBGNViz.js the nodes are geometrically represented by rectangular regions called bounding boxes. In the region quadtree I will keep AABB(Axis Alligned Bounding Box) objects to represent these bounding boxes.

In retrieving an object from the region quadtree when a node is attempted to be selected by the user. I will retrieve the AABB's colliding with the desired point by using the corresponding function of the region quadtree and call the checkPoint function of the corresponding nodes, finally will select the one returning true at first. The time complexity of this operation is O(logn+n') as the time complexity of the checkPoint function is constant, where n is the number of nodes in the graph and n' is the number of the retrieved AABB's.

Also, SBGNViz.js enables adding, deleting, moving, resizing the nodes and panning operation for the entire graph which changes the maximum and minimum coordinates in the graph. The region quadtree should be designed in a way that it is updated after these operations effectively. Therefore, it should have delete and update functions besides the basic quadtree functions.

References