# Graph Structure

## Data Conversion

At first the relational database was simply converted into a graph. This was done by creating nodes of the type:

* Ward
* Premise
* Application Status
* Legislation
* Application Type
* Application (based on reference)

These nodes are simply a copy of the columns in the table. Each node has therefore a similar meaning to the table columns. The relationships between the nodes ,thus edges, which were added in this initial stage are the following:

* Made application, from premise to Application.
  + Attribute: Time stamp when received, thus temporal edge
* Application has ward, from application to handling ward
* Has status, from application to application status
  + Attribute: approved date
  + Attribute: refused date
* Is of legislation type, from application to legislation
* Is of type, from application to application type

This simple conversion resulted in the following graph:

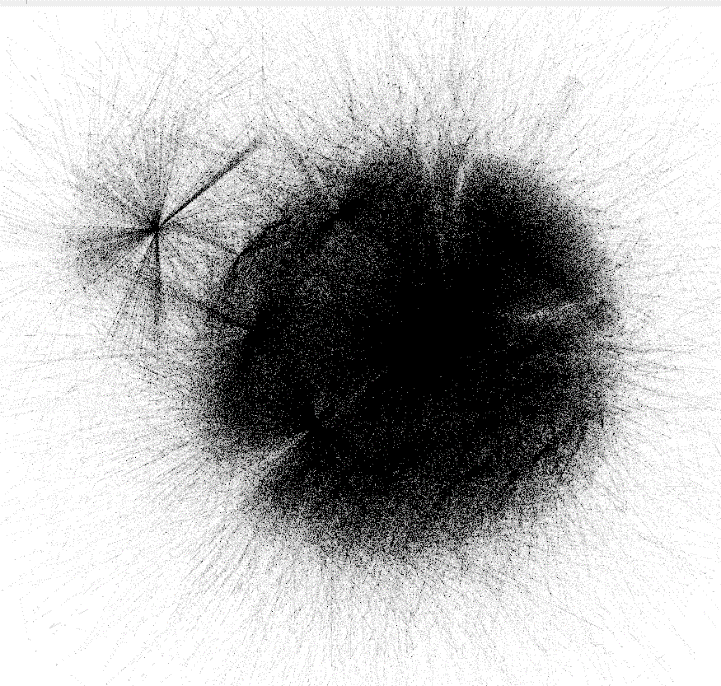


Figure 1 graph after conversion in Gephi Yifan Hu layout

The graph is unreadable in this state but after considering some of the many layout options this Yifan Hu layout shows somewhat decently how the nodes are concentrated.

### Statistics

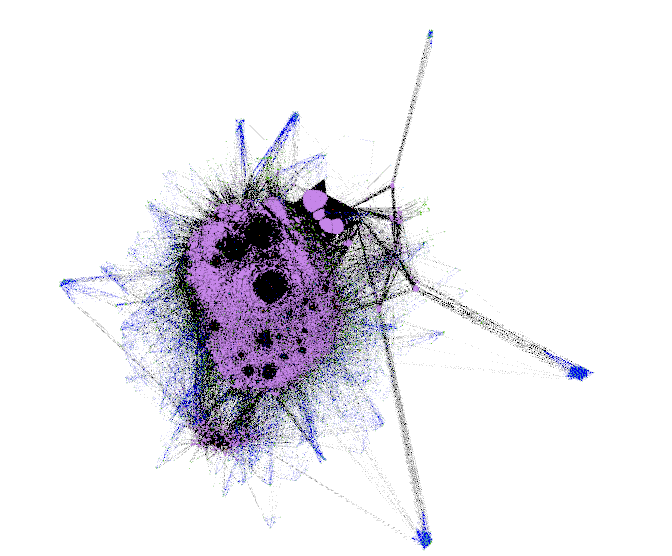
|  |  |
| --- | --- |
| Nodes | 39872 |
| Edges | 137882 |

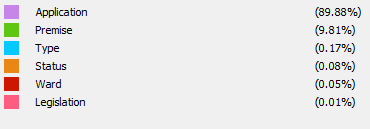
## Additional relations

The resulting graph from the conversion is rather big but does not display any details which were not easily discoverable with the relational database. Therefore additional relations were added all of them based on physical distance. Each application has a location in terms of Longitude and Latitude these Longitude and Latitude pairs can be compared to discover the distance between 2 applications. Note some applications share the same location pair. Then a heuristic was made to decide when 2 applications were “close” to each other. This heuristic is set in the script as a variable and thus can be adjusted for analysis purposes. Creating the full graph results in the following statistics:

Figure 2 Graph with neighbors in forest layout

|  |  |
| --- | --- |
| Nodes | 39872 |
| Edges | 555683 |

Thus, a vast amount of edges has been added due to the neighbors, this is ofcourse also because of duplicate neighbors. Every premise is checked versus every other premise thus cleanup can be done to reduce the duplicates in the undirected graph.



# Appendix A: Distance and Neighbor relationship code

