1. Motivation

The project Scatex is a product developed by the company to the energetic sector. It includes a group of applications to allow the supervisory and data acquisition in electrical distribution networks. Among the modules in the product, there is a schematic diagrams editor.

The scope of this dissertation is to specify tools that reduce repetitive tasks to the user in the process of creating schematic diagrams in the editor, making his task easier, faster and iterative.

The motivation is the improvement of the schematic diagrams editor, to be more competitive according with the current technology and market requirements.

2. Main Goals

The study should specify a new group of features to support development and maintenance of schematic diagrams. It should focus mostly on the possibility of automatic generate parts of diagrams, arranging the components in the space. During an automatic generation, the information about the components must be read from a database which specifies the equipment, the connections between them, and the terminals through which connection occurs. Notice that the possibility of automatically generate an entire diagram without any interference from the human user has never been considered a realistic goal.

3. Description of Work

In order to meet these objectives, the study has been divided in four subsections.

It started with the review of the state of the art for the area in study, next focused on achieve a group of features that might be useful in the diagrams editor, then a possible method to provide the automatic generation of schematic diagrams was presented and finally it has been developed a framework to test the proposed methods of automatic generation.

This chapter is divided in four parts to cover the topics.

3.1. Graph Theory

Due to the fact that the theme is related with the drawing of diagrams and automatic layout of components, the state of the art focused on the study of graph theory, particularly in the area of graph drawing.

There were presented the most common methods used in graph drawing and some different algorithms of automatic graph drawing, each one using different methods covering different aesthetic criteria.

Three different types of graph drawing algorithms were studied. A force based algorithm, an orthogonal algorithm and an hierarchical algorithm.

3.2. Suggested Features to add to the schematics editor

This section of the study defines a group of tools that might be useful in the schematic diagrams editor. It is not objective of this part of the study define possible technical methods to allow the implementation of these tools, only their system requirements and a possible interface with the user.

The specified features were divided in two types: tools to allow automatic generation of components in a diagram and tools to detect errors and inconsistencies between the diagrams and the database.

It has been identified an unique tool for automatic generation, which should offer the possibility to generate and arrange a set of components connected to a device where the user could decide initiate an expansion. The components should be generated until a certain level of depth assigned by the user, till the appearance of a substation (main equipment in the diagrams) or until find an equipment already drawn.

Features related with the validity of the diagrams were also presented. There were distinguished tools to detect errors of fact, that occur immediately after an erroneous action by the user (for instance, connect two incompatible equipments) and tools to detect inconsistencies between the diagrams and database.

The group of tools defined should make easier the drawing and maintenance of diagrams.

3.3. Automatic Generation of Branches in schematic diagrams

Among the proposed tools, the automatic generation of branches in a schematic diagram is the most complex feature but certainly the most useful.

This part of the study has focused in research methods to allow the automatic generation of branches of schematic diagrams.

The problem consists in find methods to arrange the components in the draw area avoiding crosses between edges and overlap of equipments.
The Sugiyama algorithm for hierarchical layout of graphs has been chosen due to the aesthetic criteria that it covers: planarity (avoid crosses between edges), linearity in the edges, uniformity in nodes and edges and keep the same flow to the majority of the edges (and consequently to the generated branches).

Fig. 1 - Hierarchical Graph with 5 layers

The Sugiyama algorithm has 4 basic steps:

1) Assure that the graph is aciclic (if necessary changing the direction of some edges)
2) Assign a layer to each node
3) Assign the sequence of nodes in each layer
4) Assign the absolute horizontal and vertical coordinates of each node

As barriers to the use of the Sugiyama algorithm, there were identified some specific characteristics in schematic diagrams that are not part of standard graphs and for that reason are not described in the Sugiyama algorithm.

The main problem are the connectors that each equipment has which means that the edges should connect exactly to a specific terminal and not only to the equipment (node). Indeed, this characteristic brings considerable differences in some steps of the algorithm that must be considered.

Other problem is find a direction to the edges before run the Sugiyama algorithm because it considers directed graphs. The solution was define the direction of the edges according with the sequence of the search performed from the expanded equipment to its connected equipments. In order to find uniformity in the orientation of the edges, specially in case of cicles in the diagrams, the breadth-first search (BFS) was the method used.

The results obtained after applying the layout algorithm of Sugiyama still had a few points that differed significantly from those produced by humans and for that reason there were introduced two additional steps to the algorithm:

- Normalization of short branches (which allows the diminution of the number of layers)
- Orthogonalization of the majority of edges

The step of Normalization should be performed immediately after the third step of the Sugiyama algorithm. Hence, it causes some differences in the forth step of the algorithm.

The additional step of Orthogonalization should be performed in the end.

3.4. Testing framework

In order to test the proposed layout methods and at the same time support the study about modifications to apply to the Sugiyama algorithm to consider the specificities of the schematic diagrams, it has been developed a framework that performs the layout to branches of schematic diagrams following the proposed methods.

In the end of the study, using the framework, it has been proved that the followed method returns very reasonable results (despite the fact that, as predicted, they need some adjustments from the user) in the generation of branches of schematic diagrams, with satisfying computing time, even to branches with several components.

4. Conclusions

As a final result, it is thought that it has been given a group of features that would save a lot of time to an human user in the design and maintenance of schematic diagrams.

It has been proved, performing tests, that an automatic generation following the referred methods would be possible and it is almost certain that such functionality would drastically reduce the time needed to draw a new diagram.

Therefore, there is a high level of satisfaction with the obtained results from the study, due to the fact that it is believed that once integrated in the editor, the proposed tools would be useful to the user.

Future studies in the area might focus on finding new support tools for drawing diagrams, research for alternatives to the layout algorithms presented or combine layout algorithms in order to automatically generate larger parts of schematic diagrams.