ELEC-A7151 - Object oriented programming with C++

Circuit simulator project plan

Group: circuit-sim-2

Vincent Eurasto, 711690
Tuukka Jaakkola, 608923
Perttu Niskanen, 460132
Ilari Ojakorpi, 609113

1. Introduction

Our goal is to implement circuit simulator that can calculate RMS (root mean square) values for voltage and current. User can interact with the program via GUI (Graphical User Interface). The user can add, connect, move and remove components. Thus the user can build an electronic circuit from various components and sources. After the user has constructed a circuit he/she can solve the circuit. The program will use MNA (Modified Nodal Analysis) when solving the voltages and currents in the circuit. After solving the circuit, the user can view calculation results visually by selecting/clicking nodes or components. After the user has selected a component or a node, the program will show the results graphically in different plots. The user can also save and load circuits from files.

We will start the project by implementing basic components (resistors, inductors and capacitors) and sources (constant voltage and current sources). We would also like to implement additional features after we have implemented basic functionality and everything is working well. Additional features could be for example AC sources and operational amplifiers.

2. Software structure

We used UML-diagram to visualize the program structure. A preliminary software structure is presented below. Non-mandatory features are marked with an asterisk (*).

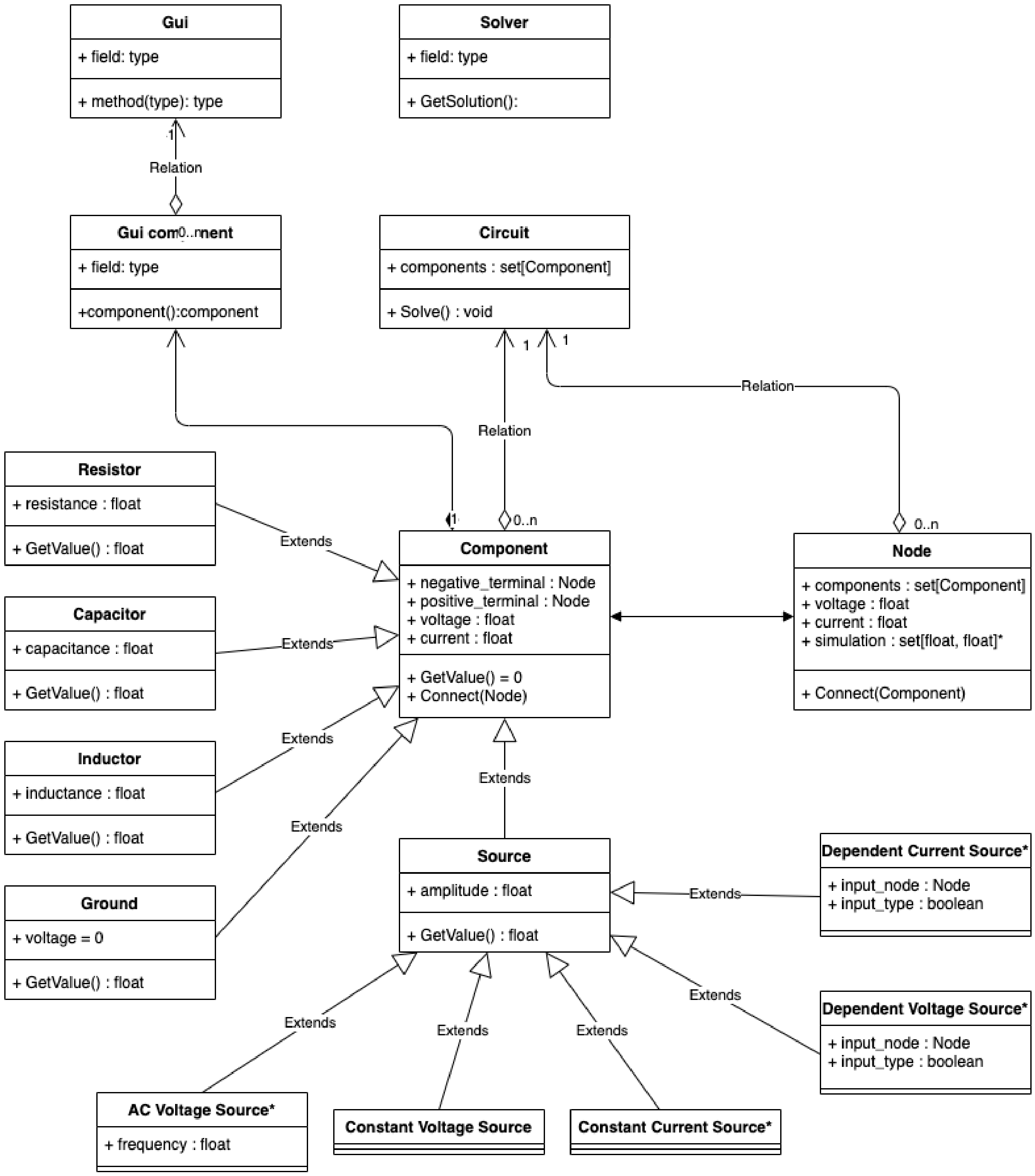

Figure 1. Diagram of the program structure

3. External libraries

Although we are planning to write a lot of code from scratch we don't have enough time to implement all necessary building blocks to get this program working properly. That's why we will use some external libraries to make our job easier and faster. We will use a library called *odeint* for solving differential equations and we will build our graphical user interface with a library called *IMGUI*.

We will start the project by using these two libraries but also the following four external libraries could be used if needed:

- *SFML* (a simple multimedia library for graphics and media playback)
- *IMGUI-SFML* (a library which allows you to use ImGui with SFML)
- *Qt* (an application framework with extended capabilities, like UI features)
- *Eigen* (template library for linear algebra)

4. Division of work

Our coarse division of work is written in the table below. Each person is responsible of an area but can divide the workload to other people. We are all motivated to work in all areas of the program.

Vincent	Perttu	Tuukka	Ilari
MNA	GUI	Math (odeint)	Writing models for components and circuits

Table 1. Coarse division of work

5. Schedule and milestones

We are all motivated to learn more about C++ programming language and work for a good project grade. Thus we should also implement some additional features into the project. We would also like to learn more about team working skills and version control tools (Git). We are also eager to learn about GUI development and complex math involved in circuit simulators. Circuits are our passion.

We are planning to do work weekly and meet regularly with group members. We will also meet weekly with our group advisor at agreed time. Testing and debugging is done throughout the project.

Preliminary schedule:

3.11: Git and coding environments ready

10.11: Program structure and solving algorithms preliminary ready

17.11: Basic components and circuit models ready

24.11: GUI should be started

1.12: Non-mandatory additional features added

8.12: Project should be nearly finished, final improvements and debugging

11.12: Project deadline