HAW
HAMBURG

CJ1 Report

Andre Alves, Divyesh Joshi, Myungjun Kim

Web App that Generates Electric Circuit LaTeX Code

Faculty of Computer Science and Engineering
Department of Information and Electrical Engineering

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Andre Alves, Divyesh Joshi, Myungjun Kim

Web App that Generates Electric Circuit LaTeX Code

CJ1 Report based on the examination and study regulations
for the Bachelor of Engineering degree programme
Bachelor of Science Information Engineering
at the Department of Information and Electrical Engineering
of the Faculty of Engineering and Computer Science
of the University of Applied Sciences Hamburg
Supervising examiner: Prof. Dr.-Ing. Martin Lapke

Day of delivery: 14 September 2022

Contents

1 Introduction

As a CJ1 project, this team chose to build a webapp that would allow a user to draw a circuit graphically and use that circuit to obtain LaTeXcode to draw a circuit using the CiruiTikZ package. The team developed a mostly functional MVP, and its source code is publicly available on GitHub [1]. Instead of building the webapp from scratch, the team forked the draw.io software by JGraph Ltd [4], which was renamed *diagrams.net* while working on this project. Starting from that source code was essentially necessary due to the scope of the project, although that came with many of its own challenges that are captured in this document.

1.1 Technology

This project employed a combination of Javascript and the LaTeXpackage CircuiTikz [2]. The Javascript project itself was built using Apache Ant, as was already implemented by diagrams.net. Ant takes a source structure and transpiles the code into minimized javascript files that are a prepared webapp. Due to its capabilities, IntelliJ IDEA Ultimate is the recommended IDE for this project.

1.1.1 Diagrams.net

Diagrams.net is an open-source, but closed-contribution, Javascript-based webapp that is used to draw diagrams, flowcharts, and similar visual documents. It also includes support for drawing electrical circuits. While diagrams.net offers support for drawing electrical circuits, the resulting circuits can only be exported as images. Consequently, if included in a LaTeXdocument, such as a thesis or a journal article, the resulting circuit does not match the overall style of the paper.

1.1.2 CircuiTikZ

CircuiTikZ is an open-source LaTeXpackage that is used for drawing circuits within LaTeXdocuments. It is a superset of the TikZ package. As a text-based diagramming tool, creating CircuiTikZ diagrams is very labor-intensive, and the learning curve turns many people away from using it. However, its text-based design results in a tremendous amount of flexibility, and the resulting circuits match the style of the paper they are made for. Due to its incredible flexibility, the goal of this project was never to build a complete GUI for CircuiTikZ. Instead, the goal of this project was to get a basic CircuiTikZ diagram that could then be further enhanced by hand, since the most difficult part of making a CircuiTikZ diagram is laying out the components.

1.1.3 IntelliJ IDEA Ultimate

Jetbrains' flagship IDE IntelliJ IDEA Ultimate is a full-featured IDE that includes support for a number of languages and frameworks [3]. In fact, it supports every framework and tool that is included in the original diagrams.net project, including Apache Ant. When combined with the size and scope of the pre-existing project, this became the obvious tool for the job.

Although there is a free community edition of IntelliJ, the community edition lacks several of the features required by this project. Luckily, students may receive a free education license of the tool as long as it is not used for commercial purposes.

1.2 Completed Tasks Against Diagrams.net

The following is a list of tasks that were carried out directly against the diagrams.net project. In retrospect, some of these tasks were unnecessary and were built into the application, already, but without public documentation, this was not known until they had already been performed. Additionally, the number of necessary changes was such that this project will remain incompatible with the original project, anyway.

- Remove non-electronics components, as well as components that are not a part of CircuiTikz.

- Reduce the number of options from the `File -> New` menu.
- Limit available themes to those that are based around components, only.
- Designed a LaTeXexport dialog.
- Added LaTeXas an export option

1.3 Additional Complete Tasks

In addition to tasks that were performed directly against the diagrams.net project, the following is a general list of tasks that were performed independently. Details of the parser structure are included separately in Chapter 3.

- Classes were created to represent electrical components in an object-oriented way.
- An XML parser was built to parse the diagram's XML into those classes.
- All electrical components from both diagrams.net and CircuiTikz were cataloged and compared.
- A look up table was built to look up the above electrical components.

1.4 Tasks Remaining

While not all-inclusive, this is a short list of tasks that we recognize as remaining unsolved. Further details for the parser itself can be found in Chapter 3.

- Support needs to be added for CircuiTikz node-style components.
- Large sections of spaghetti code need to be refactored.
- Some components may still need to be removed from the webapp.
- The images of some removed components still need to be removed from the webapp.

2 Editing Components

2.1 Finding draw.io components within CircuiTikz manual

There are two types of components in CircuiTikz, node type and path type. The information about all the components along with there type and code to draw them is available within Circuitikz manual.
To find available draw.io components in CircuiTikz manual following steps were performed:

1. Open *CircuiTikz* manual and Draw.io.
2. In draw.io look at the component's name and shape.
3. Try to find the relevant section in circuiTikz manual and try to match the name and shape of the component.

2.2 Removing components from the sidebar which are not a part of CircuiTikz

To add/remove components from sidebar following steps were taken:

1. Open *src/main/webapp/js/diagramly/sidebar/Sidebar-Electrical.js* file.
2. Look for the pallet in which the component belongs, all the components of a pallete are locted within addPaletteFunctions of that particular pallet.
3. To remove the component from the sidebar, remove the createVertexTemplateEntry line of the component within addPaletteFunctions.

2.3 User Stories

There were two viable user stories for this project:

1. As a *researcher/student/educator/engineer*, I want to draw circuits for LaTeXdocuments using a GUI so that I can draw the circuits more quickly than I can code them.

2. As a *researcher/student/educator/engineer*, I want to draw circuits for LaTeXdocuments using a GUI so that I do not need to be a LaTeX*expert* to draw circuits.

Since any person who is writing a LaTeXdocument must have some level of coding knowledge, expecting users to slightly modify circuit code is a reasonable expectation. Therefore, the following user story was considered, but ultimately not included in the planning process:

- As a researcher/student/educator/engineer, I want to draw circuits for LaTeXdocuments using a GUI so that I do not require any LaTeXknowledge.

2.4 Other test

dssg

3 Parser Structure and Next Steps

sadf

3.1 User Stories

There were two viable user stories for this project:

1. As a *researcher/student/educator/engineer*, I want to draw circuits for LaTeXdocuments using a GUI so that I can draw the circuits more quickly than I can code them.

2. As a *researcher/student/educator/engineer*, I want to draw circuits for LaTeXdocuments using a GUI so that I do not need to be a LaTeX*expert* to draw circuits.

Since any person who is writing a LaTeXdocument must have some level of coding knowledge, expecting users to slightly modify circuit code is a reasonable expectation. Therefore, the following user story was considered, but ultimately not included in the planning process:

- As a researcher/student/educator/engineer, I want to draw circuits for LaTeXdocuments using a GUI so that I do not require any LaTeXknowledge.

3.2 Other test

dssg

Bibliography

[1] Myungjun Kim Andre Alves, Divyesh Joshi. drawio-circuitikz, 2022.

[2] CircuiTikZ contributors. Circuitikz, 2022.

[3] JetBrains. Intellij idea, 2022.

[4] JGraph Ltd. drawio, 2022.