Rebar Addon for FreeCAD

Submitted for partial fulfilment of the Degree
of
Bachelor of Technology
(Information Technology)

Submitted by:
Amritpal Singh
1411234

Submitted to:
Sukhjit Singh Sehra
Training Coordinator
CSE Department

Information Technology
Guru Nanak Dev Engineering College
Ludhiana 141006

Abstract

Rebar Addon for FreeCAD is the project that I worked upon for my 6-month training and also as Google Summer of Code project. It is under the umbrella organization of BRL-CAD. FreeCAD is an open-source parametric 3D modeling application, made primarily to design real-life objects. Parametric modeling describes a certain type of modeling, where the shape of the 3D objects you design are controlled by parameters. For example, the shape of a brick might be controlled by three parameters: height, width and length. In FreeCAD, as in other parametric modelers, these parameters are part of the object, and stay modifiable at any time, after the object has been created. Some objects can have other objects as parameters, for example you could have an object that takes our brick as input, and creates a column from it. You could think of a parametric object as a small program that creates geometry from parameters.

FreeCAD is also multiplatform (it runs exactly the same way on Windows, Mac OS and Linux platforms), and open-source. Being open-source, FreeCAD benefits from the contributions and efforts of a large community of programmers, enthusiasts and users worldwide. FreeCAD is essentially an application built by the people who use it, instead of being made by a company trying to sell you a product. And of course, it also means that FreeCAD is free, not only to use, but also to distribute, copy, modify, or even sell.

My project is to create a rebar addon for Arch Workbench of FreeCAD to ease up the process of creating reinforcement in structural element. The main purpose of this project is to enable the user to create reinforcement through an easy and intuitive way.

This project is purely related structural engineering, aimed at easing up the reinforcement process in the FreeCAD. The current rebar functionality in FreeCAD is very limited by its UI and creating a reinforcement system using it is quite tedious. Currently, the user has to create a sketch for the rebar profile and define the required set of constraints. This becomes very time-consuming task even for an expert level user when he/she has a building model with several structural objects. This project is aimed at easing up the process of rebaring in FreeCAD. In this project, list of rebars will be provided to user in the form of dropdown. On selecting a rebar from dropdown, a dialog box will popout with input fields where can provide data related to selected rebar. The entire project will be delivered as a FreeCAD addon. The input fields in the dialog box are further categorised and presented in the form of tabs. User can easily switch to any tab to see contained input fields, thus enriching the experience by keeping the natural flow of user. With successful completion of this project, FreeCAD user will have an easy and professional way to create rebars for their projects with less efforts in less time.

This addon is completely open source (under the LGPLv2+ license) and the entire code is available to the user as and when required. There is also Complete developers Documentation, User manual and video tutorials alongwith it that helps using it a lot easier.

Acknowledgement

I, student of Guru Nanak Dev Engineering College, Ludhiana, have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

The author is highly grateful to Dr. M.S. Saini Director, Guru Nanak Dev Engineering College, Ludhiana for providing him with the opportunity to carry out his Six Weeks Training at Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana.

The author would like to wholeheartedly thank Dr. H.S. Rai Dean, Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana who is a vast sea of knowledge and without whose constant and never ending support and motivation, it would never have been possible to complete the project and other assignments so efficiently and effectively.

Finally, I would thank's Nirbhay Chauhan and My Mentors at FreeCAD organization Yorik van Havre, Bernd Hahnebach and whole FreeCAD community. Without their encouragement and Guidence, it would not have been possible to complete this project in such an efficient manner.

Amritpal Singh

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION OF ORGANIZATION

Figure 1.1: Guru Nanak Dev Engineering College

I had my Six Month Industrial Training at TCC-Testing And Consultancy Cell, GNDEC Ludhiana. Guru Nanak Dev Engineering College was established by the Nankana Sahib Education Trust Ludhiana. The Nankana Sahib Education Trust i.e NSET was founded in memory of the most sacred temple of Sri Nankana Sahib, birth place of Sri Guru Nanak Dev Ji. With the mission of Removal of Economic Backwardness through Technology Shiromani Gurudwara Parbandhak Committee i.e SGPC started a Poly technical was started in 1953 and Guru Nanak Dev Engineering College was established in 1956.

The main goal of this institute is:

- To build and promote teams of experts in the upcoming specialisations.
- To promote quality research and undertake research projects keeping in view their relevance to needs and requirements of technology in local industry.
- To achieve total financial independence.
- To start online transfer of knowledge in appropriate technology by means of establishing multipurpose resource centres.

1.1 Testing and Consutancy Cell

I had my Six Month Institutional Training at TCC i.e Testing And Consultancy Cell, GNDEC Ludhiana under the guidance of Dr. H.S.Rai Dean Testing and Consultancy Cell. Testing and Consultancy Cell was established in the year 1979 with a basic aim to produce quality service

for technical problems at reasonable and affordable rates as a service to society in general and Engineering fraternity in particular.

Consultancy Services are being rendered by various Departments of the College to the industry, Sate Government Departments and Entrepreneurs and are extended in the form of expert advice in design, testing of materials & equipment, technical surveys, technical audit, calibration of instruments, preparation of technical feasibility reports etc. This consultancy cell of the college has given a new dimension to the development programmers of the College. Consultancy projects of over Rs. one crore are completed by the Consultancy cell during financial year 2009-10.

Ours is a pioneer institute providing Consultancy Services in the States of Punjab, Haryana, Himachal, J&K and Rajasthan. Various Major Clients of the Consultancy Cell are as under:

- Northern Railway, Govt. of India
- Indian Oil Corporation Ltd.
- Larson & Turbo.
- Multi National Companies like AFCON & PAULINGS.
- Punjab Water Supply & Sewage Board

CHAPTER 2

INTRODUCTION TO PROJECT

2.1 Overview

Figure 2.1: FreeCAD's logo

Rebar Addon for FreeCAD is the project that I worked upon for my 6-month training and also as Google Summer of Code project. It is under the umbrella organization of BRL-CAD. FreeCAD is an open-source parametric 3D modeling application, made primarily to design real-life objects. Parametric modeling describes a certain type of modeling, where the shape of the 3D objects you design are controlled by parameters. For example, the shape of a brick might be controlled by three parameters: height, width and length. In FreeCAD, as in other parametric modelers, these parameters are part of the object, and stay modifiable at any time, after the object has been created. Some objects can have other objects as parameters, for example you could have an object that takes our brick as input, and creates a column from it. You could think of a parametric object as a small program that creates geometry from parameters.

FreeCAD is also multiplatform (it runs exactly the same way on Windows, Mac OS and Linux platforms), and open-source. Being open-source, FreeCAD benefits from the contributions and efforts of a large community of programmers, enthusiasts and users worldwide. FreeCAD is essentially an application built by the people who use it, instead of being made by a company trying to sell you a product. And of course, it also means that FreeCAD is free, not only to use, but also to distribute, copy, modify, or even sell.

My project is to create a rebar addon for Arch Workbench of FreeCAD to ease up the process of creating reinforcement in structural element. The main purpose of this project is to enable the user to create reinforcement through an easy and intuitive way.

This project is purely related structural engineering, aimed at easing up the reinforcement process in the FreeCAD. The current rebar functionality in FreeCAD is very limited by its UI and creating a reinforcement system using it is quite tedious. Currently, the user has to create a sketch for the rebar profile and define the required set of constraints. This becomes very time-consuming task even for an expert level user when he/she has a building model with several structural objects. This project is aimed at easing up the process of rebaring in FreeCAD. In this project, list of rebars will be provided to user in the form of dropdown. On selecting a rebar from dropdown, a dialog box will popout with input fields where can provide data related to selected rebar. The entire project will be delivered as a FreeCAD addon. The input fields in the dialog box are further categorised and presented in the form of tabs. User can easily switch to any tab to see contained input fields, thus enriching the experience by keeping the natural flow of user. With successful completion of this project, FreeCAD user will have an easy and

professional way to create rebars for their projects with less efforts in less time.

This addon is completely open source (under the LGPLv2+ license) and the entire code is available to the user as and when required. There is also Complete developers Documentation, User manual and video tutorials alongwith it that helps using it a lot easier.

The core part of this project is implemented using Python and PySide. Github is used to manage code and IRC/Gitter to communicate with the mentors.

My training being not based on particular language or technology, different type of open-source software's and technologies are used in this project and many during my training which are not used in this project like Django, Facebook's Graph API for *Love Ludhiana* WebApp.

2.2 The Existing System

At present, the rebar functionality in FreeCAD is very limited and creating a reinforcement system is quite tedious. The current approach is followed by creating a sketch for the rebar profile and defining the required set of constraints. This becomes very time-consuming task when user has building model with several structural objects. We can only creates planar rebars by using current rebar tool.

Limitations of previous system

- The AST of Rebar object was not strong.
- It creates only planar rebars.
- Circular dependencies issue with Arch structural object.
- Lack of User interface.
- No default rebar shapes present.
- Complex workflow.
- Rebar object not calculates its rebar length.
- Reinforcing rebars in real building model is very time-consuming task.

2.3 User Requirement Analysis

User Requirements Analysis for a software system is a complete description of the requirements of the User. It includes functional Requirements and Non-functional Requirements. Non-functional requirements are requirements which impose constraints on the design or implementation.

Users of the System:

1. Provides full GUI (Graphical User Interface) for reinforcing rebars in the structural object.
2. They also requires pre-built standard rebar shapes.
3. Different rebars will have their own view and data properties.
4. User can edit the parameters of the group of rebars from the view & data properties itself.

5. These rebars will be fully parametric. Hence the parameters of the rebars will automatically adjust themselves if the changes are made to their parent structure.

6. Easy to use.

2.3.1 Functional Requirements

- **Specific Requirements**: This phase covers the whole requirements for the system. After understanding the system we need the input data to the system then we watch the output and determine whether the output from the system is according to our requirements or not. So what we have to input and then what well get as output is given in this phase. This phase also describe the software and non-function requirements of the system.

- **Input Requirements of the System**:

  - **Straight Rebar**: Orientation, Front Cover, Right Cover, Left Cover, Cover along, Bottom Cover, Top Cover, Amount and Spacing.
  - **UShape Rebar**: Orientation, Front Cover, Right Cover, Left Cover, Bottom Cover, Top Cover, Rounding, Amount and Spacing.
  - **LShape Rebar**: Orientation, Front Cover, Right Cover, Left Cover, Bottom Cover, Top Cover, Rounding, Amount and Spacing.
  - **BentShape Rebar**: Orientation, Front Cover, Right Cover, Left Cover, Bottom Cover, Top Cover, Anchor Length, Bent Angle, Rounding, Amount and Spacing.
  - **Stirrup**: Front Cover, Right Cover, Left Cover, Bottom Cover, Top Cover, Bent Angle, Bent Factor, Rounding, Amount and Spacing.
  - **Helical Rebar**: Side Cover, Top Cover, Bottom Cover, Pitch and Diameter.
  - **Rebar Distribution**: Number of rebars and Spacing in segment 1, segment 2 and segment 3.

- **Output Requirements of the System**: Reinforced structure is produced which contains user selected rebar shapes.

- **Interactive mode**: The user can operate rebar addon from an interactive mode of FreeCAD.

- **User interaction mirroring on the console**: Everything the user does in the FreeCAD interface executes Python code, which can be printed on the console and recorded in macros.

2.3.2 Non-functional requirements

1. Extensible: It should be able to support future functional requirements

2. Usability: Simple user interfaces that a layman can understand.

3. Modular Structure: The software should have structure. So, that different parts of software would be changed without affecting other parts.

2.4 Feasibility Study

Feasibility study aims to uncover the strengths and weaknesses of a project. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility analysis should provide a historical background of the project, description of the project or service, details of the operations and management and legal requirements. Generally, feasibility analysis precedes technical development and project implementation. These are some feasibility factors by which we can determine that the project is feasible or not:

- **Technical feasibility**: Technological feasibility is carried out to determine whether the project has the capability, in terms of software, hardware, personnel to handle and fulfill the user requirements. This whole project is based on Open Source Environment and is part of an open source software which would be deployed on any OS.
- **Economic feasibility**: In Economic feasibility, we determine whether the benefit is gain according to the cost invested to develop the project or not. If benefits outweigh costs, only then the decision is made to design and implement the system. It is important to identify cost and benefit factors, which can be categorized as follows:

  1. Development costs.
  2. Operating costs.

  Rebar Addon for FreeCAD is also Economically feasible with as It could be developed and maintain with zero cost as It is supported by Open source community. Plus This project is started with no intention of having any economic gain but still there is an option for donations.

2.5 Objective of Project

To ease up the rebaring process in FreeCAD, an interactive addon is developed where user will input the required data as per the design requirements and they will need not to draw rebars from the Sketcher workbench for creating reinforcement in the structural object.

One of the primary benefits of this addon is the ability to create parametric rebars. These are designs which are parametrized using parameters or top-level variables.

1. Provides full GUI (Graphical User Interface) for reinforcing rebars in the structural object.
2. They also requires pre-built standard rebar shapes.
3. Different rebars will have their own view and data properties.
4. User can edit the parameters of the group of rebars from the view & data properties itself.
5. These rebars will be fully parametric. Hence the parameters of the rebars will automatically adjust themselves if the changes are made to their parent structure.
6. Easy to use.

CHAPTER 3

PROJECT DESIGN

3.1 Product Perspective

This product is supposed to be part of an open source, under the LGPLv2+ license. It is a CAD software for 3D modeling application, made primarily to design real- life objects. Rebar Addon is the idea given by the user's only to fullfill their needs and It had been in demand even before it consentment. It will extent this already powerfull system by allowing the user to create reinforcement through an easy and intuitive way. It will also provide a way to modify the set of parameters using FreeCAD Python console.

The following are the main features that are included in Rebar Addon for FreeCAD

1. **Cross platform support:** Offers operating support for most of the known and commercial operating systems in form of binaries and also it can be compiled on other platforms.
2. **Allows to create rebars using GUI:** The system allows the user to reinforce rebars by simply filling in input fields.
3. **Backward Compatibility:** FreeCAD should be backward compatibility even after new features are implemented.
4. **Rebar parameters:** The created rebars will have their own view and data properties. User can edit the parameters of the group of rebars from the view & data properties itself.
5. **Parametrization:** These rebars will be fully parametric. Hence the parameters of the rebars will automatically adjust themselves if the changes are made to their parent structure.
6. **Different operations:** The rebars will also have the functionality to cut, copy, clone, move and rotate.

3.2 User Characteristics

We have identified three potential classifications of users of our system:

1. Designers: Designers are the people who create model for their own use or for commercial use.
2. The Client: These are the people which will customzie model to their need made by modelers before ordering or printing model drawing themselves.
3. Developers: These are people who might want to integrate this new feature of FreeCAD into their systems.

3.2.1 The General User

All users can be assumed to have the following characteristics:

1. Ability to read and understand English.
2. Familiarity with the operation of the basic Graphical User Interface (GUI) components of FreeCAD.
3. Beyond the above, no further facility with computer technology can be assumed.

3.2.2 Designers

The Designer can be assumed to have the following characteristics:

1. Basic Knowledge of FreeCAD.
2. Basic Knowledge of Creating models.
3. Basic coding skills.
4. Optional experience of cmd-line.

3.2.3 Developers

The Developers can be assumed to have the following characteristics:

1. Basic Knowledge of programming.
2. Basic Knowledge of FreeCAD.
3. Ability to program in cmd-line.

3.3 Flowchart

A flowchart is a type of diagram that represents an algorithm, work flow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows and the flowchart 3.1 of Rebar Addon showing the flow of control and Data in the software.

3.3.1 Detailed Description

The basic implementation of this project is almost done in form of prototype. There is need to modify the structure of the project. We have to divide the task into there parts:

1. **Front end** It will deal with how the Rebar Addon will look to the user like in form of toolbars, menus etc. This part will include two parts:
   (a) **Rebar Addon toolbar/menus** It contains a list of different rebars.
   (b) **Dialog box** A number of different inputs widgets present in the dialog box according user's selected rebar.

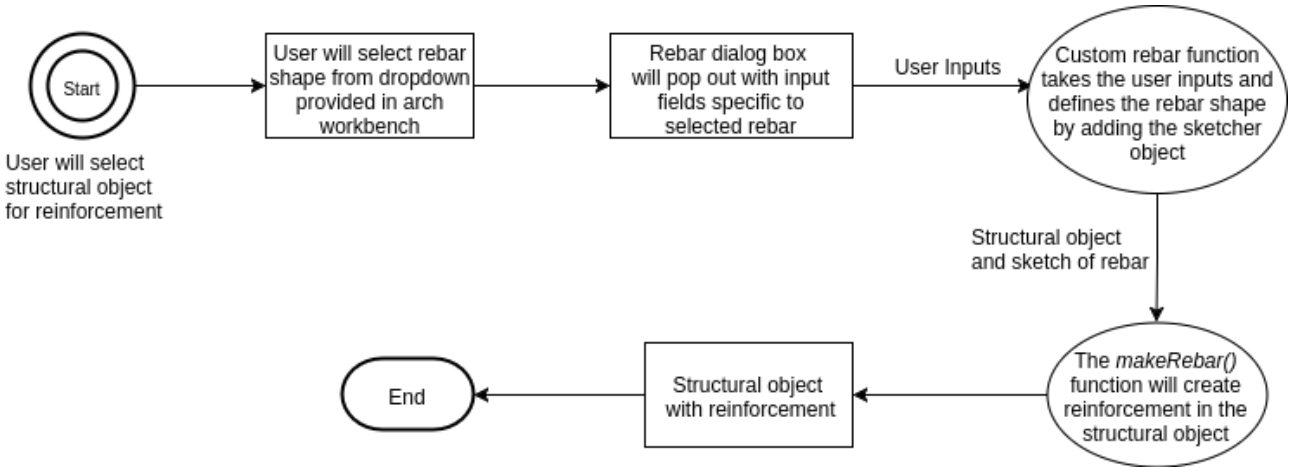

Figure 3.1: Flowchart of Rebar Addon

2. **Back End** On backend, a top layer is created over FreeCAD rebar object which holds rebars properties (likes side cover, left cover, orientation and bent angle etc.) and then a rebar profile is created from these properties. This profile is now pass to Rebar object which will create rebar.

3.4 Dependency Graph

A Dependency Graph is a graphical representation of the which module is dependent on which other modules. A Dependency Graph is often used as a preliminary step to creating an overview of the system. Dependency Graph also gives overview of how good is the design of the system. FreeCAD being were huge software it would be difficult to make the dependency graph of whole software. So, here is Dependency Graph of Rebar addon is as following-:

1. **Caller graph of FreeCAD Rebar object:** Figure 3.2 shows the modules that use rebar object.
2. **Caller graph of StraightRebar._StraightRebarTaskPanel.clicked:** Figure 3.3 shows the modules that use the module StraightRebar._StraightRebarTaskPanel.clicked.
3. **Caller graph of StraightRebar._StraightRebarTaskPanel.accept:** Figure 3.4 show the modules that uses the module StraightRebar._StraightRebarTaskPanel.accept.
4. **Caller graph of Rebar Distribution:** Figure 3.5 show the modules that uses the module Rebar Distribution.

3.5 Class Diagrams

Class Diagrams describe the static structure of the system. Following classes diagram represent the relationship between different classes in FreeCAD and Rebar Addon:

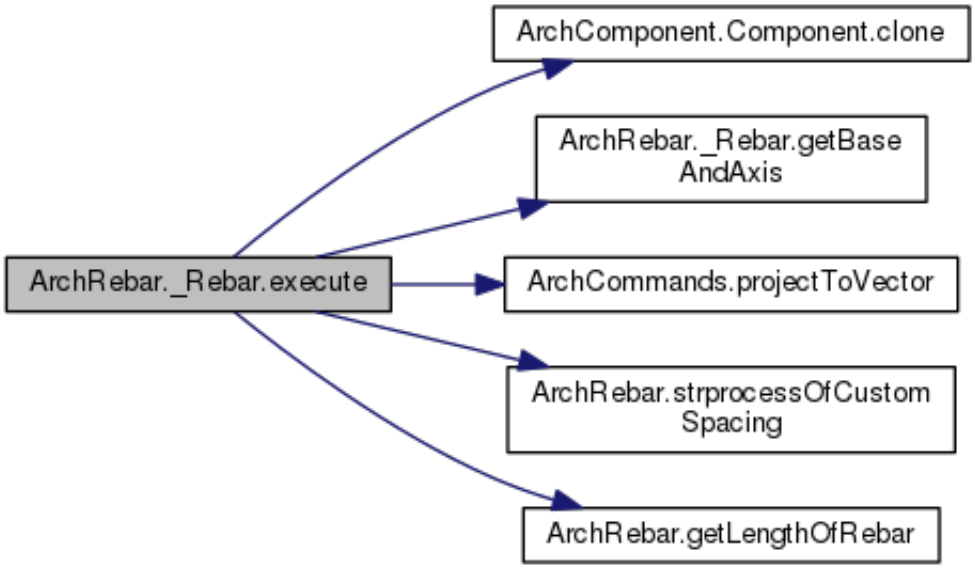

Figure 3.2: Dependency graph of Rebar object

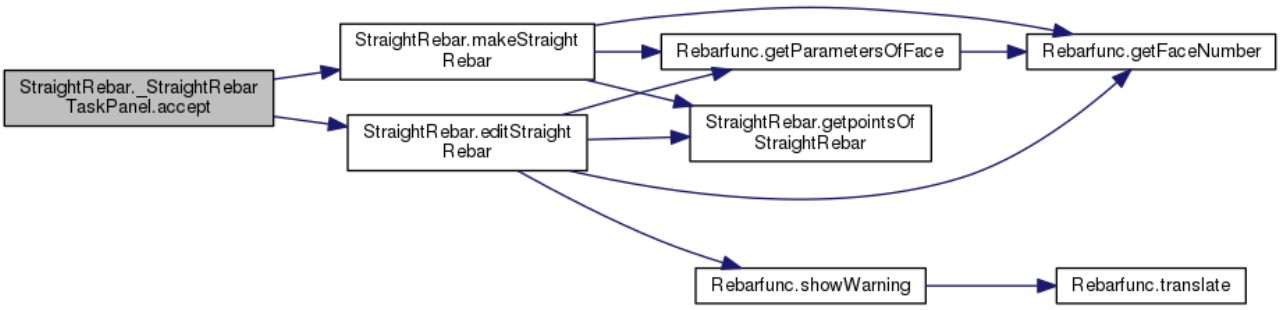

Figure 3.3: Caller graph of StraightRebar._StraightRebarTaskPanel.clicked

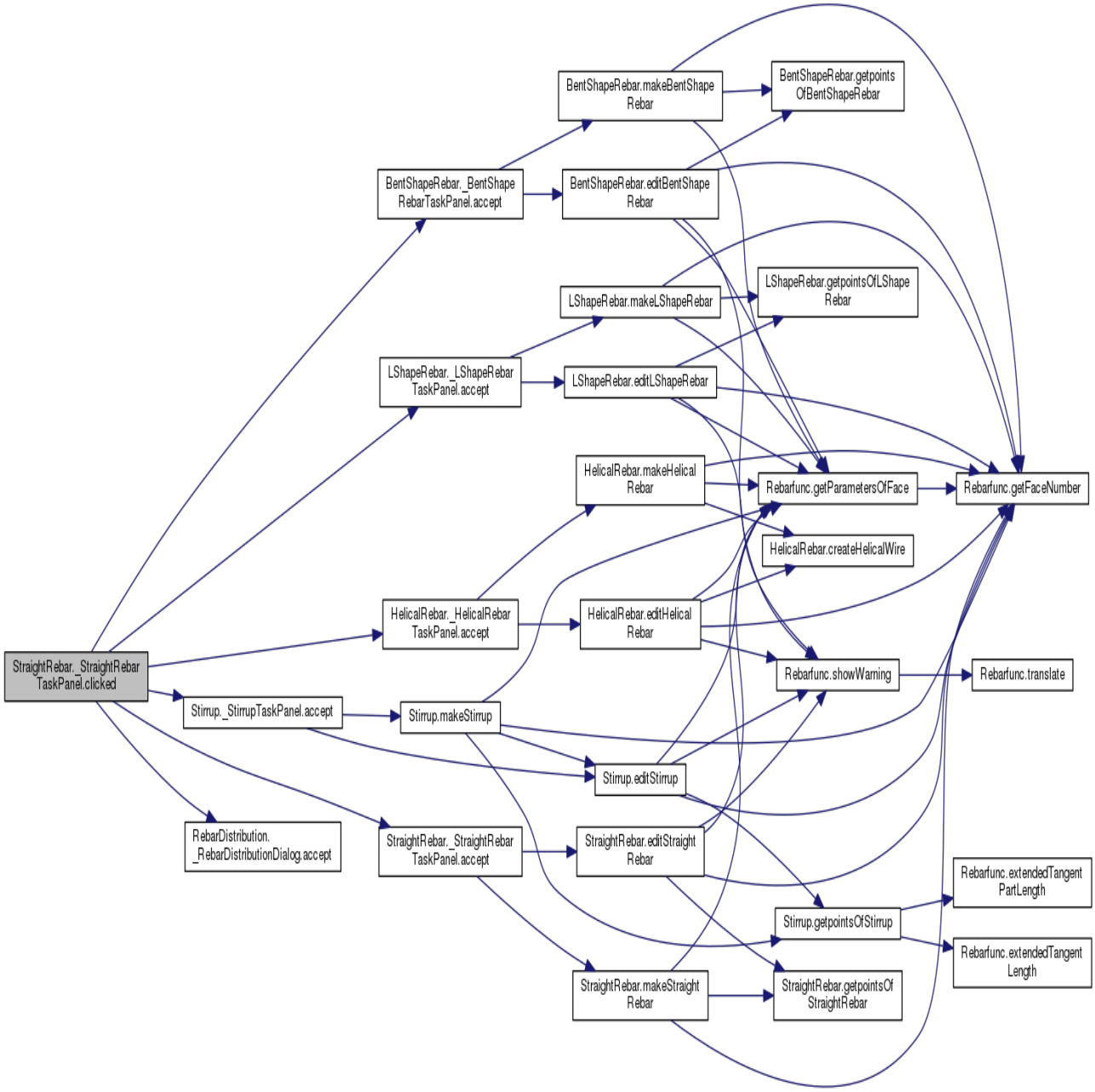

Figure 3.4: Caller graph of StraightRebar._StraightRebarTaskPanel.accept

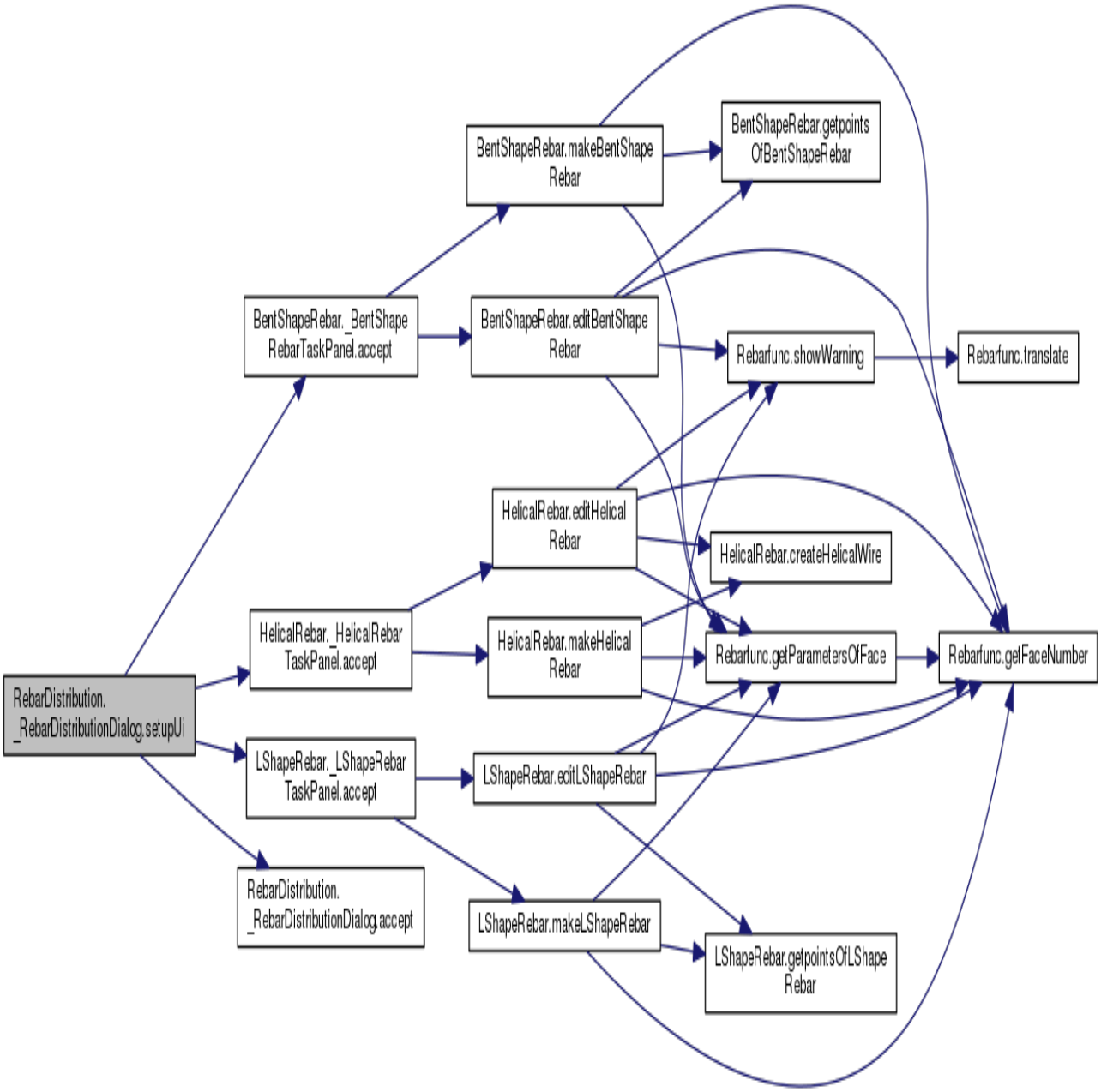

Figure 3.5: Caller graph of RebarDistribution

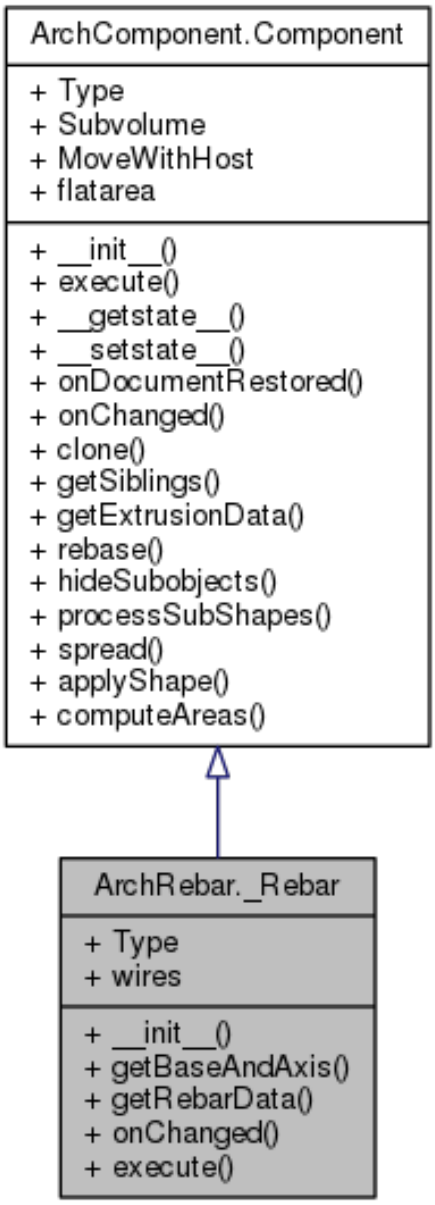

Figure 3.6: Class Diagram for ArchRebar._Rebar

1. Figure 3.6 shows the class diagram of the ArchComponent.Component class which is the basse class of ArchRebar._Rebar i.e ParameterCheckbox, ParameterVector,ParmeterSpinbox, ParameterComboBox,ParameterSlider, ParameterText.

2. Figure 3.7 shows the class diagram of the StraightRebar._StraightRebarTaskPanel class which is the main class for whole the Straight rebar dialog box.

3. Figure 3.8 shows the class diagram of the RebarDistribution._RebarDistributionDialog class which is the main class on rebar distribution dialog box.

4. Figure 3.9 shows the class diagram of the QtGui::QDialog which is the base class of PipUpImage.PopUpImage.

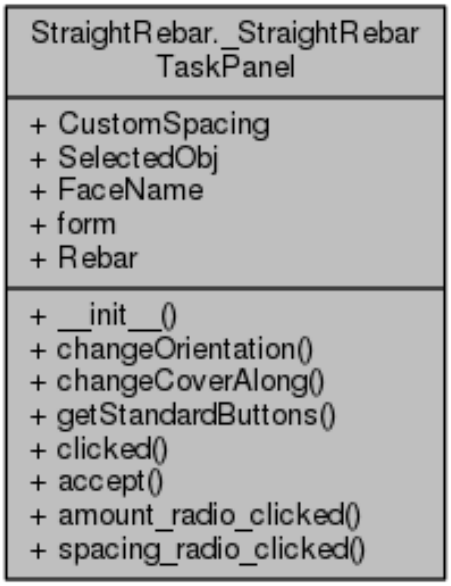

Figure 3.7: Class Diagram for StraightRebar._StraightRebarTaskPanel

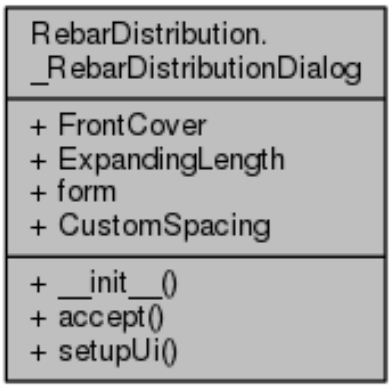

Figure 3.8: Class Diagram for RebarDistribution._RebarDistributionDialog

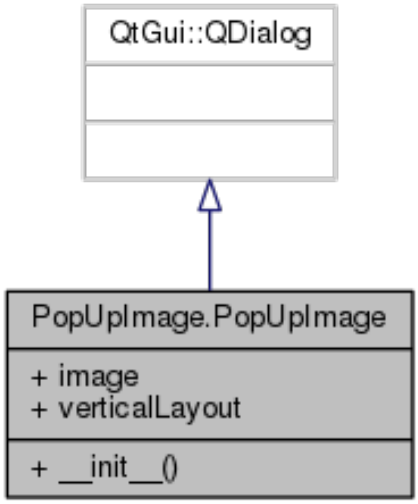

Figure 3.9: Class Diagram for PipUpImage.PopUpImage

3.6 Dependencies

Dependencies include softwares or framework that need to be installed for proper working of this software.

There is no software dependencies to Install this software.

This addon could be installed on any given list of operating system.

1. Mac OS X
2. Windows
   (a) XP or newer on x86 32/64 bit
3. Linux
   (a) Debian
   (b) Ubuntu
   (c) Kubuntu
   (d) Arch Linux
   (e) openSUSE
   (f) Fedora
4. BSD
   (a) NetBSD $\geq$ 6.1
   (b) FreeBSD $\geq$ 10
   (c) OpenBSD

The main dependencies of rebar addon is FreeCAD (version ¿=0.17). But If you want to build FreeCAD this software from soucre code on *any OS*, you need some libraries and tools. The version numbers in brackets specify the versions which have been used for development. Other versions may or may not work as well.

If you're using a newer version of Ubuntu, you can install these libraries from aptitude. If you're using Mac, or an older Linux/BSD, there are build scripts that download and compile the libraries from source. Follow the instructions for the platform you're compiling on below. Following are dependencies of FreeCAD software.

1. A C++ compiler supporting C++11
2. build-essential
3. python
4. python-matplotlib
5. libtool
6. libcoin60-dev (Debian Wheezy, Wheezy-backports, Ubuntu 13.04 and before)
7. libsoqt4-dev

8. libxerces-c-dev
9. libboost-dev
10. libboost-filesystem-dev
11. libboost-regex-dev
12. libboost-program-options-dev
13. python-pyside
14. libqt4-opengl-dev
15. qt4-dev-tools
16. libboost-thread-dev
17. libboost-python-dev

CHAPTER 4

DEVELOPMENT AND IMPLEMENTATION

4.1 Python

Figure 4.1: Python logo

Python is a dynamic language, as in python coding is very easy and also it require less coding and about its interpreted nature it is just exellent. Python is a high level programming language and Django which is a web development framework is written in python language.

Python is an easy to learn, powerful programming language.Python runs on Windows, Linux/Unix, Mac OS X. Python is free to use, even for commercial products. Python can also be used as an extension language for existing modules and applications that need a programmable interface. Python is free to use, even for commercial products, because of its OSI-approved open source license.

4.1.1 Features of Python

- Very clear, readable syntax.
- Strong introspection capabilities.
- Intuitive object orientation.
- Natural expression of procedural code.
- Full modularity, supporting hierarchical packages.
- Exception-based error handling.
- Very high level dynamic data types.
- Extensive standard libraries and third party modules for virtually every task.
- Extensions and modules easily written in C, C++ (or Java for Jython, or .NET languages for IronPython).
- Embeddable within applications as a scripting interface.

4.1.2 Installation of Python

Installation of python is a very easy proccess. The current python versions are: Python 2.7.1 and Python 3.2. Type the commands in the terminal:

$ wget http://www.python.org/ftp/python/2.7/Python-2.7.tgz

$ tar xzf Python-2.7.tgz

This will install the python on your pc/laptop.

4.2 PySide

Figure 4.2: Python logo

4.2.1 Introduction

PySide is an open source software project providing Python bindings for the Qt framework. Qt is a cross-platform application and UI framework, allowing the developers to write applications once and deploy them across many operating systems without rewriting the source code, while Python is a modern, dynamic programming language with a vivid developer community.

Combining the power of Qt and Python, PySide provides the wealth of Qt framework for developers writing software in Python and presents a first-class rapid application development platform available on all major operating systems.

4.2.2 Licensing

PySide has been published as a response to the lack of suitably licensed Qt bindings for Python. PySide is licensed under the LGPL version 2.1 license, allowing both Free/Open source software and proprietary software development.

4.2.3 Project scope and goals

PySide consists of a full set of Qt and Qt Quick bindings for multiple platforms as well as the automated binding generation tools required to produce the bindings. Due to the availability of the whole toolchain, PySide will be of interest not only to developers requiring the Qt bindings, but to developers willing to generate other Qt and C++ based bindings as well.

Although based on a different technology than the competing PyQt bindings, in the Python-level compatibility between the two projects is still rather high, allowing for easy porting of software between PyQt and PySide.

The PySide project has been initiated and much of the initial development sponsored by Nokia. PySide is now run as a true open source project, and currently most of new work is based on volunteer contributor efforts. PySide is a Qt add-on, utilizing the same licensing and the same infrastructure as Qt itself.

PySide aims to support all the platforms Qt itself does. Feel welcome to assist in porting the code to your platform of choice.

4.2.4 Installation of PySide

To install PySide, you can choose from the following options:

- Use pip to install the wheel binary packages:

  ```
  pip install -U PySide
  ```

- Use setuptools to install the egg binary packages (deprecated):

  ```
  easy_install -U PySide
  ```

4.2.5 Hello World example

```
# Import PySide classes
import sys
from PySide.QtCore import *
from PySide.QtGui import *

# Create a Qt application
app = QApplication(sys.argv)

# Create a Window
mywindow = QWidget()
mywindow.resize(320, 240)
mywindow.setWindowTitle('Hello World!')

# Create a label and display it all together
mylabel = QLabel(mywindow)
mylabel.setText('Hello World!')
mylabel.setGeometry(QRect(130, 110, 60, 10))
mywindow.show()

# Enter Qt application main loop
sys.exit(app.exec_())
```

4.3 Introduction to LaTeX

LaTeX, I had never heard about this term before doing this project, but when I came to know about its features, found it excellent. LaTeX (pronounced /letk/, /letx/, /ltx/, or /ltk/) is a document markup language and document preparation system for the TeX typesetting program. Within the typesetting system, its name is styled as LaTeX.

Within the typesetting system, its name is styled as LaTeX. The term LaTeX refers only to the language in which documents are written, not to the editor used to write those documents. In order to create a document in LaTeX, a .tex file must be created using some form of text editor. While most text editors can be used to create a LaTeX document, a number of editors have been created specifically for working with LaTeX.

Figure 4.3: Donald Knuth, Inventor Of TEX typesetting system

LATEX is most widely used by mathematicians, scientists, engineers, philosophers, linguists, economists and other scholars in academia. As a primary or intermediate format, e.g., translating DocBook and other XML-based formats to PDF, LATEX is used because of the high quality of typesetting achievable by TEX. The typesetting system offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout and bibliographies.

LATEX is intended to provide a high-level language that accesses the power of TEX. LATEX essentially comprises a collection of TEX macros and a program to process LATEXdocuments. Because the TEX formatting commands are very low-level, it is usually much simpler for end-users to use LATEX.

4.3.1 Typesetting

LATEX is based on the idea that authors should be able to focus on the content of what they are writing without being distracted by its visual presentation. In preparing a LATEX document, the author specifies the logical structure using familiar concepts such as chapter, section, table, figure, etc., and lets the LATEX system worry about the presentation of these structures. It therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system
  for the \TeX{} typesetting program.
   \par
```

```
   $E=mc^2$
\end{document}
```

LaTeX

August 10, 2013

LaTeX is a document preparation system for the TeX typesetting program.
$E = mc^2$

Figure 4.4: LaTeX output of above program.

4.3.2 Installing LaTeX on System

Installation of LaTeX on personal system is quite easy. As i have used LaTeX on Ubuntu 13.04 so i am discussing the installation steps for Ubuntu 13.04 here:

- Go to terminal and type

  $ sudo apt-get install texlive-full

- Your Latex will be installed on your system and you can check for manual page by typing.

  $ man latex
  in terminal which gives manual for latex command.

- To do very next step now one should stick this to mind that the document which one is going to produce is written in any type of editor whether it may be your most common usable editor Gedit or you can use vim by installing first vim into your system using command.

  $ sudo apt-get install vim

- After you have written your document it is to be embedded with some set of commands that Latex uses so as to give a structure to your document. Note that whenever you wish your document to be looked into some other style just change these set of commands.

- When you have done all these things save your piece of code with .tex format say test.tex. Go to terminal and type

  latex path of the file test.tex Or pdflatex path of the file test.tex

eg: pdflatex test.tex
for producing pdf file simultaneously.
After compiling it type command

$ evince filename.pdf
eg: evince test.pdf
To see output pdf file.

4.3.3 Graphical Editors for LaTeX

LaTeX is not restricted to command line only there are so many graphical based editors available to be used. These GUi based editors provide an easy interface to user so as to do typesetting in an efficient manner. Some of them are listed below:

- Texmaker

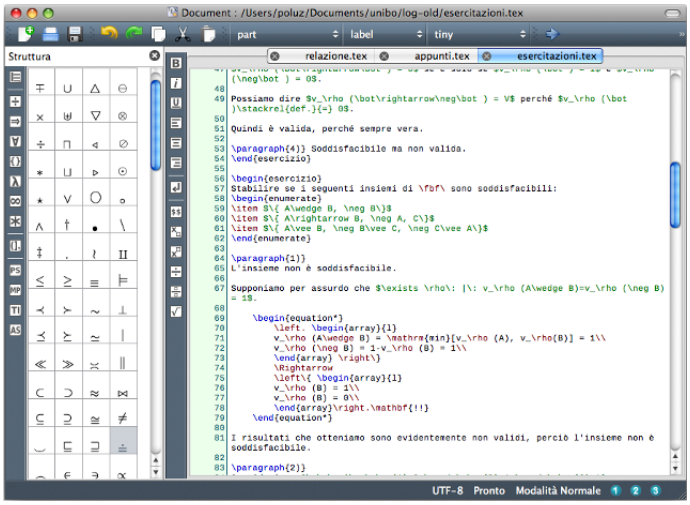

Figure 4.5: Texmaker, A Graphical LaTeX Editor

- LEd

And many more but the preferred method to produce LaTeX document is through console mode only.

4.3.4 Pdfscreen LaTeX

There are some packages that can help to have unified document using LaTeX. Example of such a package is pdfscreen that let the user view its document in two forms-print and screen. Print for hard copy and screen for viewing your document on screen. Download this package from www.ctan.org/tex-archive/macros/latex/contrib/pdfscreen/.
Then install it using above mention method.

Just changing print to screen gives an entirely different view. But for working of pdfscreen another package required are comment and fancybox.

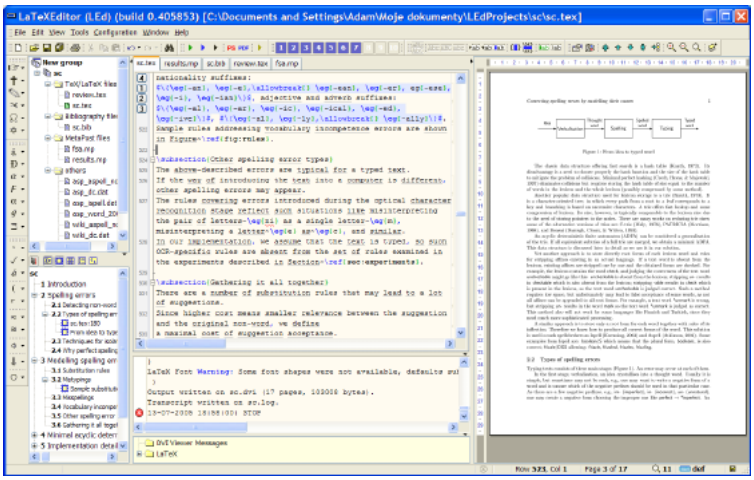

Figure 4.6: LEd, A Graphical LaTeX Editor

The fancybox package provides several different styles of boxes for framing and rotating content in your document. Fancybox provides commands that produce square-cornered boxes with single or double lines, boxes with shadows, and round-cornered boxes with normal or bold lines. You can box mathematics, floats, center, flushleft, and flushright, lists, and pages.

Whereas comments package selectively include/excludes portions of text. The comment package allows you to declare areas of a document to be included or excluded. One need to make these declarations in the preamble of your file. The package uses a method for exclusion that is pretty robust, and can cope with ill-formed bunches of text.

So these extra packages needed to be installed on system for the proper working of pdfscreen package.

4.3.5 Web based graphic generation using LaTeX

LaTeX is also useful when there is need of generating the graphics from browser. For example to draw a circle by just entering its radius in html input box. So this kind A of project can be conveniently handled using LaTeX. Basic idea behind this generation process is that when user clicks on submit button after entering radius a script will run that enter the radius in already made .tex file and recompiles it on server and makes its pdf and postscript file. After that user can view those files by clicking on link provided to view the files. See some screen shots of such a graphic generation project made by Dr. H.S. Rai:

So here in the above input page which is also the index page user can enter input for length of rectangle, breadth of rectangle and for radius of circle after that user can submit the values. After the values get submitted a script get runs by php code at server side. This script first enters the dimensions of rectangle and circle that were selected by user in to an already existing .tex file and replace with the older dimensions there. After that script recompiles the the tex file and make it available for user.

In above figure it gets clear that .tex file has been compiled and pdf and postscript files are available to user and user can download the graphics so produced. Hence graphics can be

generated in LaTeX through web interface.

Figure 4.7: Web based graphic generation using LaTeX(input page)

4.4 Introduction to Doxygen

Figure 4.8: Doxygen Logo

Doxygen is a documentation generator, a tool for writing software reference documentation. The documentation is written within code, and is thus relatively easy to keep up to date. Doxygen can cross reference documentation and code, so that the reader of a document can easily refer to the actual code.

Doxygen supports multiple programming languages, especially C++, C, C#, Objective-C, Java, Python, IDL, VHDL, Fortran and PHP.[2] Doxygen is free software, released under the terms of the GNU General Public License.

4.4.1 Features of Doxygen

- Requires very little overhead from the writer of the documentation. Plain text will do, Markdown is support, and for more fancy or structured output HTML tags and/or some of doxygen's special commands can be used.
- Cross platform: Works on Windows and many Unix flavors (including Linux and Mac OS X).
- Comes with a GUI frontend (Doxywizard) to ease editing the options and run doxygen. The GUI is available on Windows, Linux, and Mac OS X.
- Automatically generates class and collaboration diagrams in HTML (as clickable image maps) and LaTeX (as Encapsulated PostScript images).
- Allows grouping of entities in modules and creating a hierarchy of modules.
- Doxygen can generate a layout which you can use and edit to change the layout of each page.

- Can cope with large projects easily.

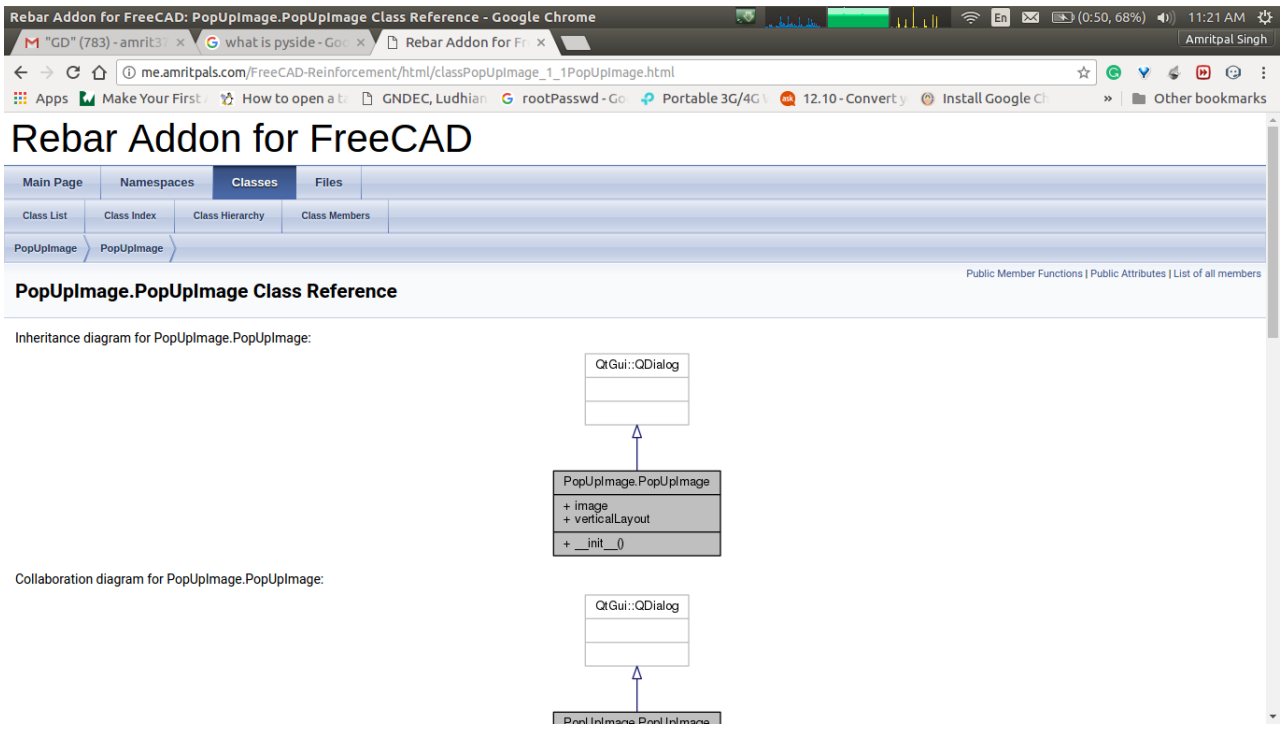

Figure 4.9: Documentation using Doxygen (main page)

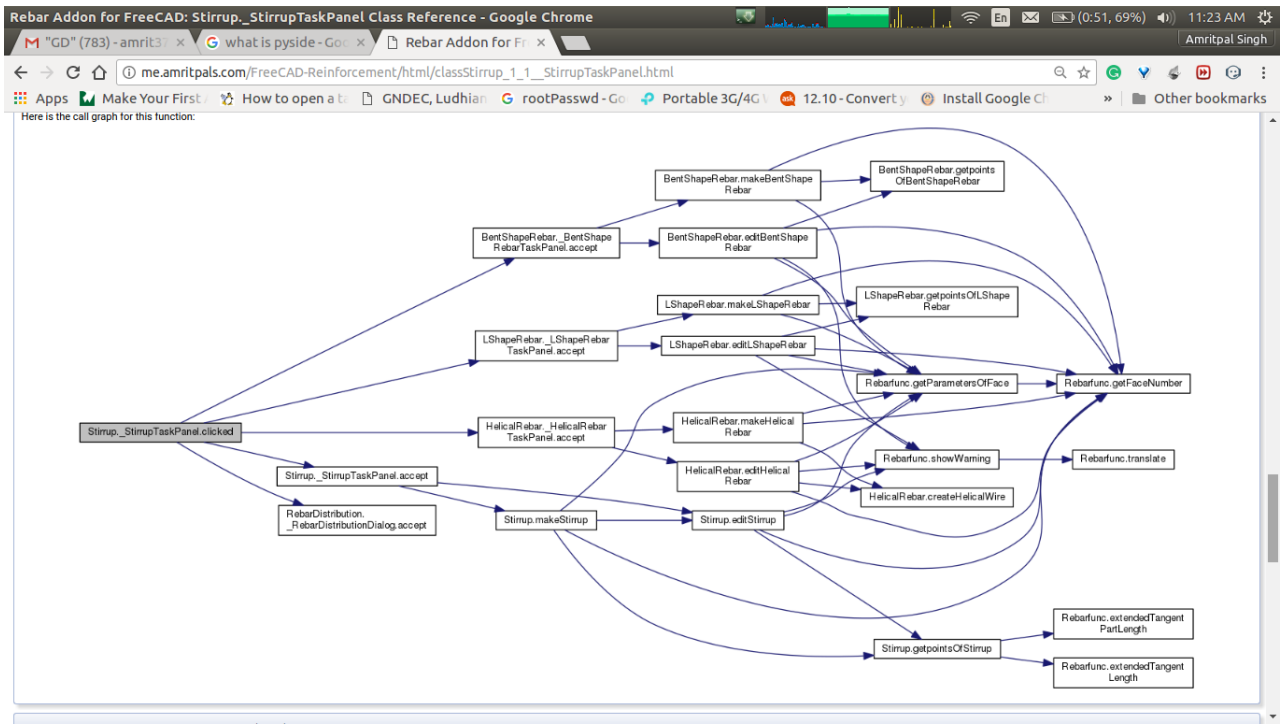

Figure 4.10: Doxygen documentation of a function

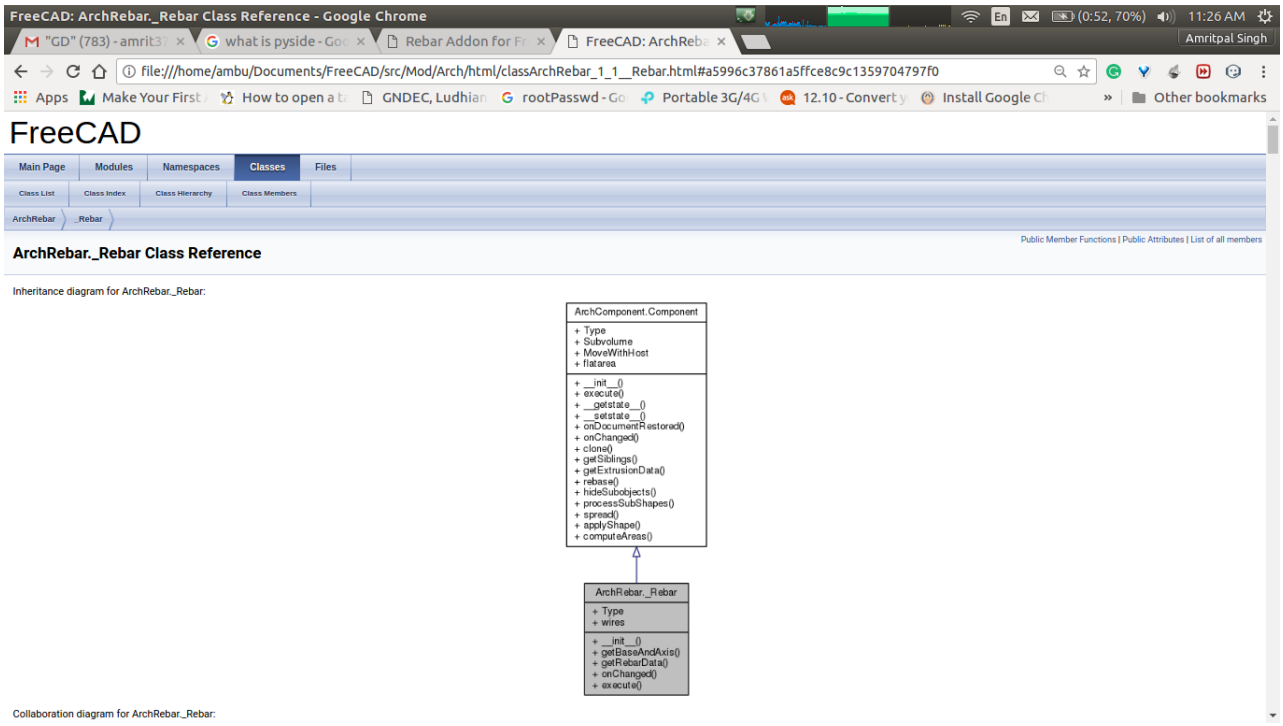

Figure 4.11: Documentation using Doxygen (classes inheritance)

4.5 Introduction to Github

Figure 4.12: Github Logo

GitHub is a Git repository web-based hosting service which offers all of the functionality of Git as well as adding many of its own features. Unlike Git which is strictly a command-line tool, Github provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

GitHub offers both paid plans for private repto handle everything from small to very large projects with speed and efficiency. ositories, and free accounts, which are usually used to host open source software projects. As of 2014, Github reports having over 3.4 million users, making it the largest code host in the world.

GitHub has become such a staple amongst the open-source development community that many developers have begun considering it a replacement for a conventional resume and some employers require applications to provide a link to and have an active contributing GitHub

account in order to qualify for a job.

The Git feature that really makes it stand apart from nearly every other Source Code Management (SCM) out there is its branching model.

Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

This means that you can do things like:

- Frictionless Context Switching.
  Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.
- Role-Based Codelines.
  Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.
- Feature Based Workflow.
  Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.
- Disposable Experimentation.
  Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the workwith nobody else ever seeing it (even if you've pushed other branches in the meantime).

Notably, when you push to a remote repository, you do not have to push all of your branches. You can choose to share just one of your branches, a few of them, or all of them. This tends to free people to try new ideas without worrying about having to plan how and when they are going to merge it in or share it with others.

There are ways to accomplish some of this with other systems, but the work involved is much more difficult and error-prone. Git makes this process incredibly easy and it changes the way most developers work when they learn it.

4.5.1 What is Git?

Figure 4.13: Git Logo

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was

initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development.

As with most other distributed revision control systems, and unlike most clientserver systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is free and open source software distributed under the terms of the GNU General Public License version 2 to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

4.5.2 Installation of Git

Installation of git is a very easy process. The current git version is: 2.0.4. Type the commands in the terminal:

$ sudo apt-get update

$ sudo apt-get install git

This will install the git on your pc or laptop.

4.5.3 Various Git Commands

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. The commonly used Git command line instructions are:-

4.5.3.1 Create Repositories

Start a new repository or obtain from an exiting URL

$ git init [**project-name**]
Creates a new local repository with the specified name

$ git clone [**url**]
Downloads a project and its entire version history

4.5.3.2 Make Changes

Review edits and craft a commit transaction

$ git status
Lists all new or modified files to be committed

$ git diff
Shows file differences not yet staged

$ git add [file]
Snapshots the file in preparation for versioning

$ git reset [file]
Unstages the file, but preserve its contents

$ git commit -m "[descriptive message "]
Records file snapshots permanently in version history

4.5.3.3 Group Changes

Name a series of commits and combine completed efforts

$ git branch
Lists all local branches in the current repository

$ git branch [branch-name]
Creates a new branch

$ git checkout [branch-name]
Switches to the specified branch and updates the working directory

$ git merge [branch]
Combines the specified branchs history into the current branch

$ git branch -d [branch-name]
Deletes the specified branch

4.5.3.4 Save Fragments

Shelve and restore incomplete changes

$ git stash
Temporarily stores all modified tracked files

$ git stash pop
Restores the most recently stashed files

$ git stash list
Lists all stashed changesets

$ git stash drop
Discards the most recently stashed changeset

4.5.3.5 Synchronize Changes

Register a repository bookmark and exchange version history

$ git fetch [**bookmark**]
Downloads all history from the repository bookmark

$ git merge [**bookmark** /[branch]]
Combines bookmarks branch into current local branch

$ git push [**alias** [branch]]
Uploads all local branch commits to GitHub

$ git pull
Downloads bookmark history and incorporates changes

4.6 Implementation

1. Every workbench is nothing more than a folder containing an Init.py and/or InitGui.py. I just need to define FreeCAD commands, that can be made into menu items or toolbar buttons.

2. The inputs provided by user from rebar dialog box will be passed to our custom rebar function. This function will use the inputs to define the shape of rebar and in this function the sketcher object will be added to the FreeCAD active document which will hold the profile of the rebar by calculating coordinates of vertices and drawing the shape of rebar from user inputs.

3. Then this sketcher object and the selected structural object will pass to the prebuilt function of FreeCAD which will create the rebar. Below is the detailed description of that function.

   makeRebar([baseobj,sketch,diameter,amount,offset,name]): Adds a reinforcement bar object to the given structural object, using the given sketch as profile.

   By following the above approach I can reuse the existing implementation of reinforcement system in the FreeCAD and at the same time proposing something new and interesting which will enrich user experience. Figure 4.14

4.7 Activation of Rebar Addon

1. Open the FreeCAD Addon Manager (Tool →Addon manager).

2. When an addon manager will open, select Reinforcement from a list of workbenches shown by an addon manager.

3. After selecting, click on Install/Update button.

4. Restart FreeCAD.

5. Now you will see different rebars in a drop-down list of rebar tools (Arch →Rebar tools →Different rebars).

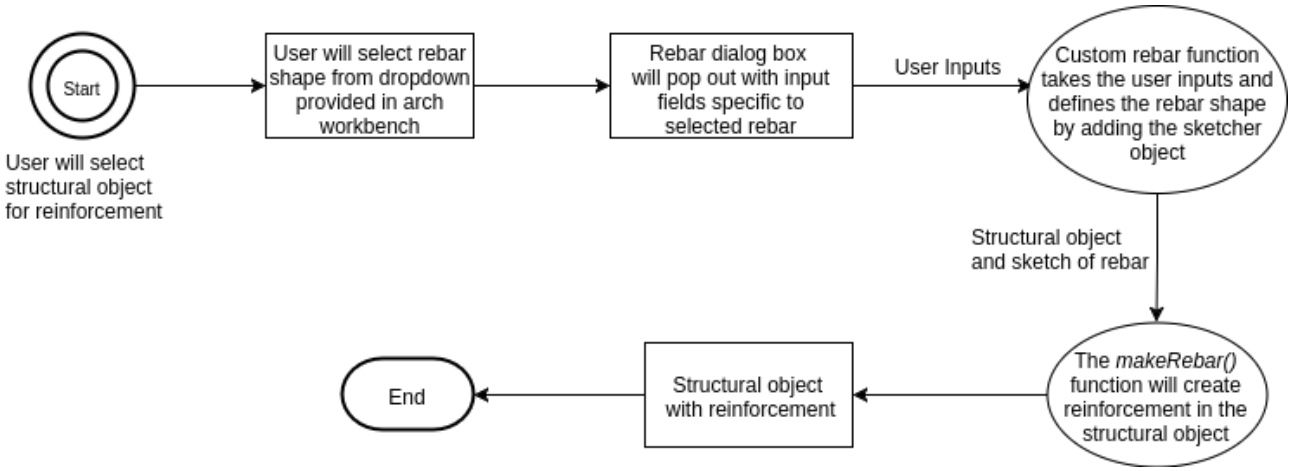

Figure 4.14: Represenation diagram of data flow from Rebar addon to FreeCAD object

After activation, Rebar Tool is added in the FreeCAD Arch Workbench as shown in Figure 4.15

4.8 Tools of Rebar Addon

4.8.1 Creates a Straight reinforcement bar in a selected structural element

4.8.1.1 Description

The Straight Rebar tool allows user to create a straight reinforcing bar in the structural element as show in Figure 4.16.

4.8.1.2 How to use

- Create a structure element
- Select any face of the structure
- Then select Arch Rebar Straight Rebar from the rebar tools
- A task panel will pop-out on the left side of the screen as shown in Figure 4.17.
- Select the desired orientation
- Give the inputs like front cover, right side cover, left side cover, bottom cover and diameter of the rebar
- Select the mode of distribution either amount or spacing
- If spacing is selected, a user can also opt for custom spacing
- Pick selected face is used to verify or change the face for rebar distribution
- Click OK or Apply to generate the rebars

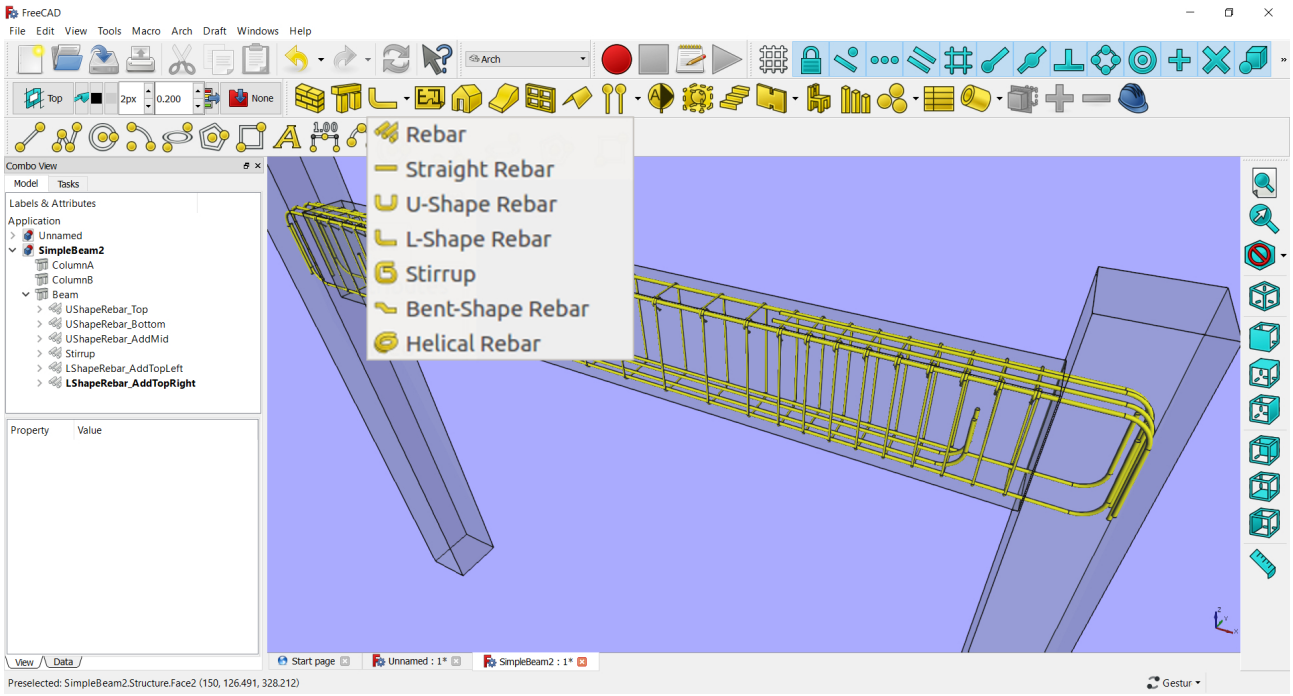

Figure 4.15: FreeCAD Window

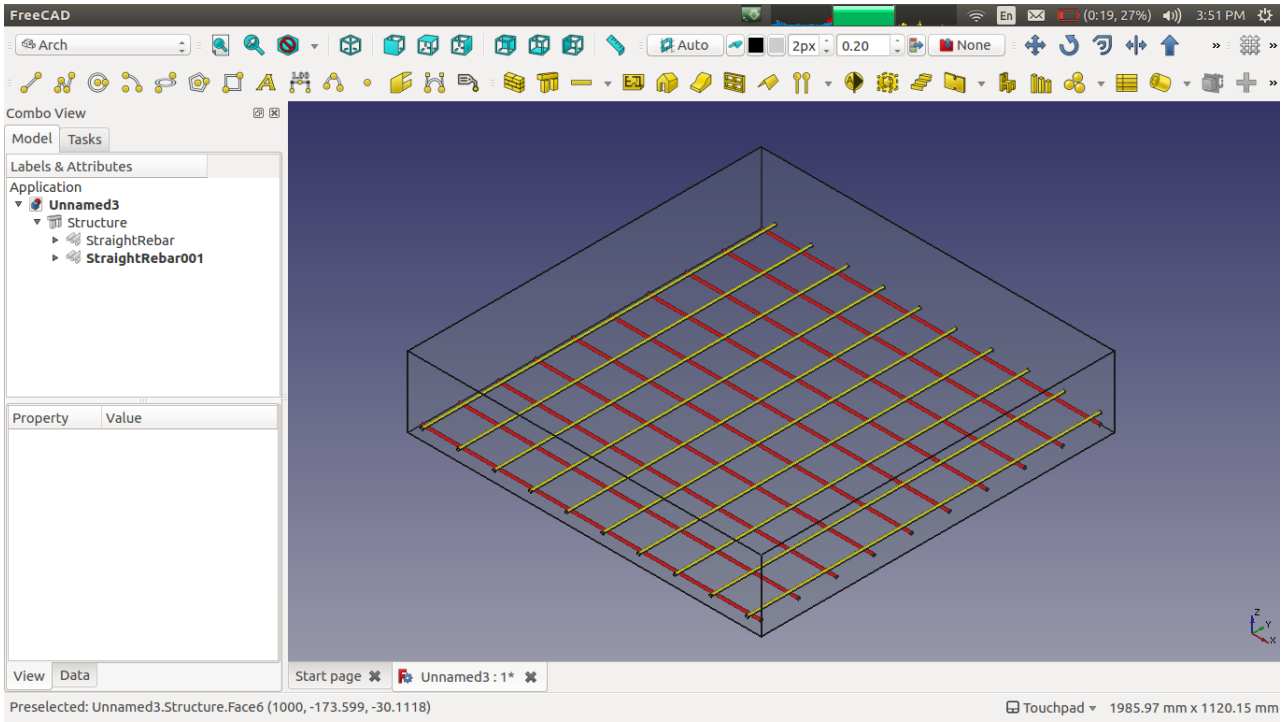

Figure 4.16: Straight rebar reinforcement

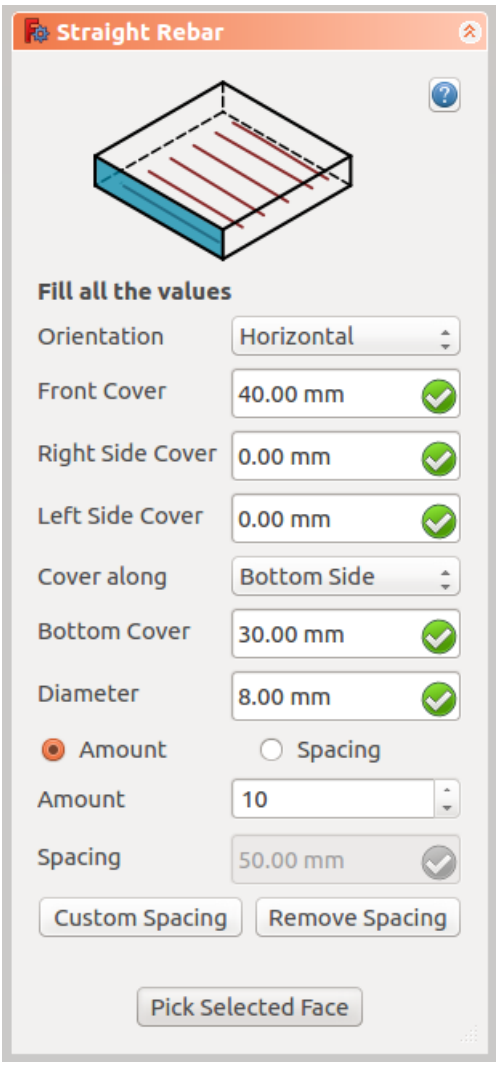

Figure 4.17: Dialog box of Straight Rebar

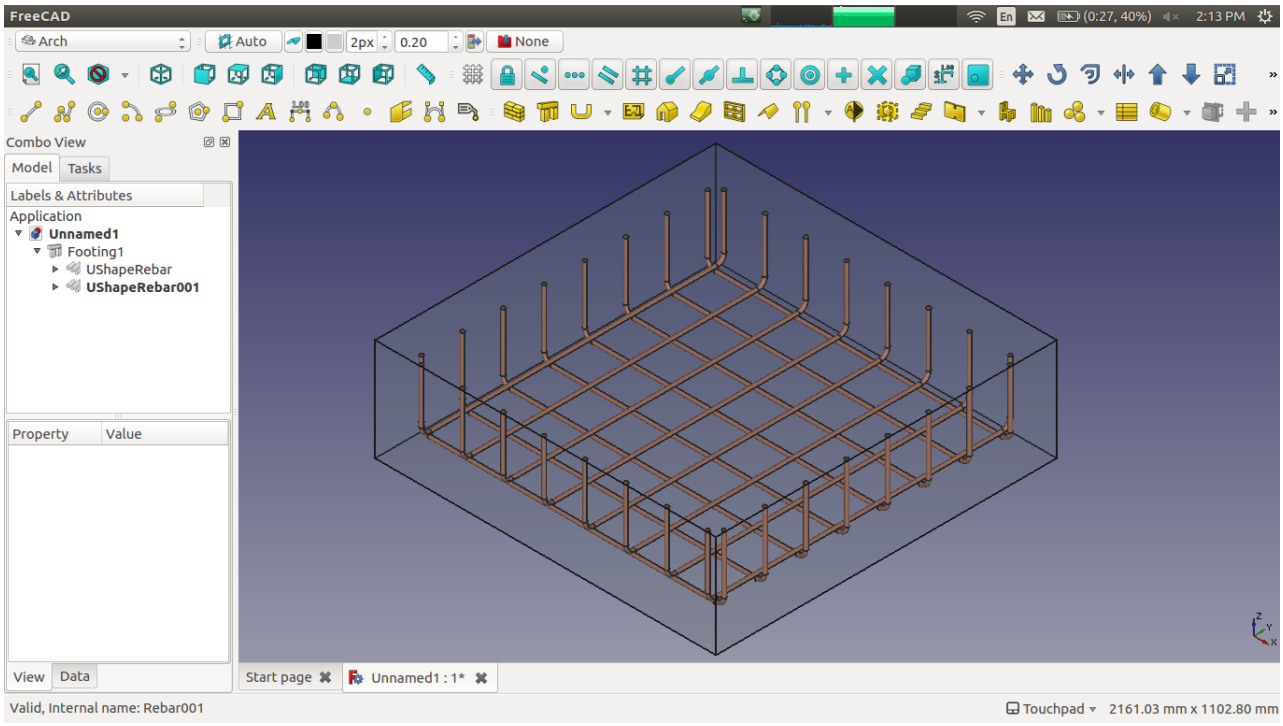

Figure 4.18: UShape rebar reinforcement

- Click Cancel to exit the task panel

4.8.2 Creates a UShape reinforcement bar in a selected structural element

4.8.2.1 Description

The UShape Rebar tool allows user to create a UShape reinforcing bar in the structural element as show in Figure 4.18.

4.8.2.2 How to use

- Select UShape Rebar tool from the rebar tools.
- A task panel will pop-out on the left side of the screen as shown in Figure 4.19
- Click OK or Apply to generate the rebars.

4.8.3 Creates a LShape reinforcement bar in a selected structural element

4.8.3.1 Description

The LShape Rebar tool allows user to create LShape reinforcing bar in the structural element as shown in Figure 4.20.

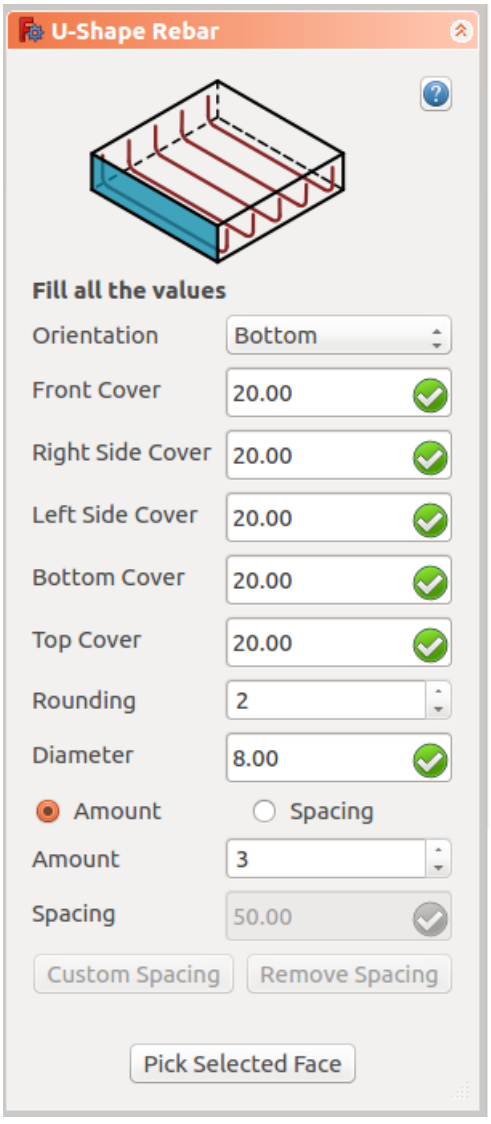

Figure 4.19: Dialog box of UShape Rebar

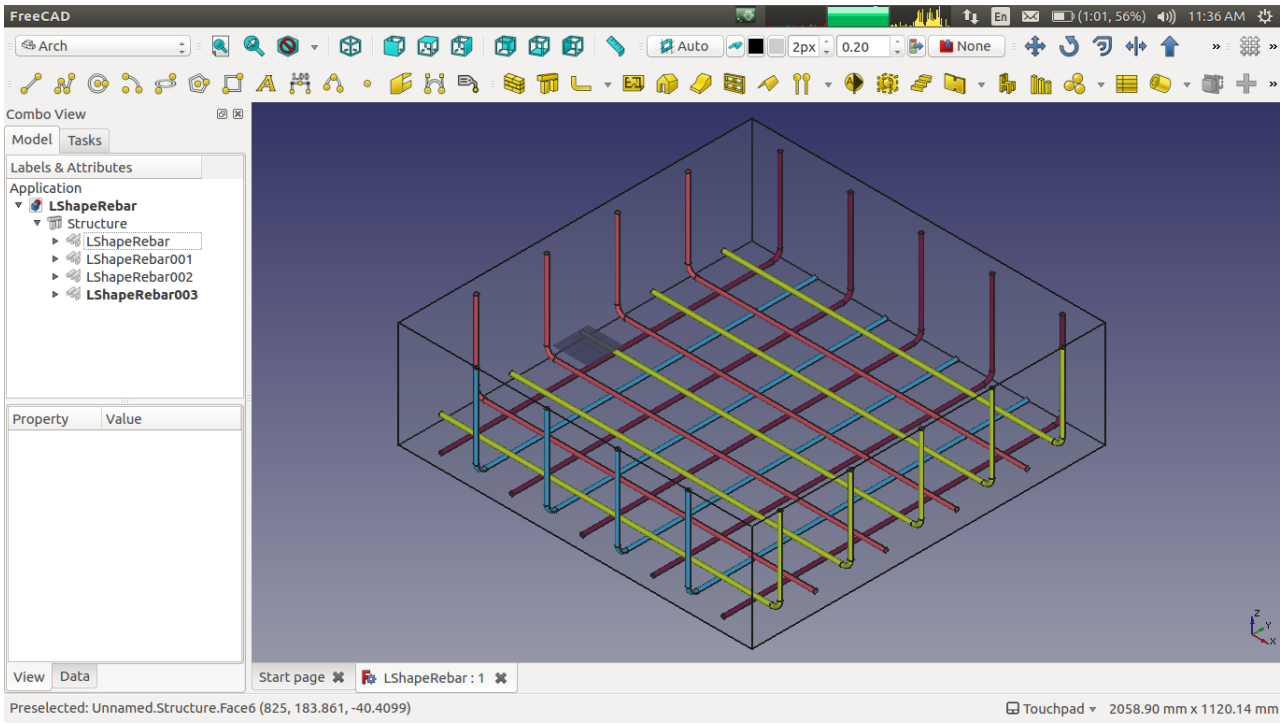

Figure 4.20: LShape rebar reinforcement

4.8.3.2 How to use

- Select LShape Rebar tool from the rebar tools.
- A task panel will pop-out on the left side of the screen as shown in Figure 4.21
- Click OK or Apply to generate the rebars.

4.8.4 Creates a Bent Shape reinforcement bar in a selected structural element

4.8.4.1 Description

The Bent Shape Rebar tool allows user to create a bent shape reinforcing bar in the structural element as shown in Figure 4.22.

4.8.4.2 How to use

- Select Bent Shape Rebar tool from the rebar tools.
- A task panel will pop-out on the left side of the screen as shown in Figure 4.23
- Click OK or Apply to generate the rebars.

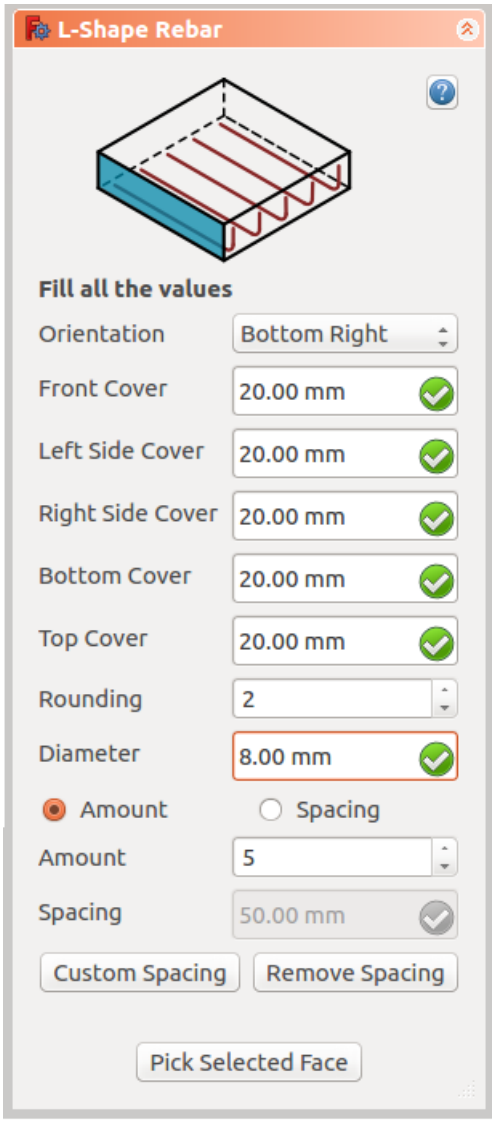

Figure 4.21: Dialog box of LShape Rebar

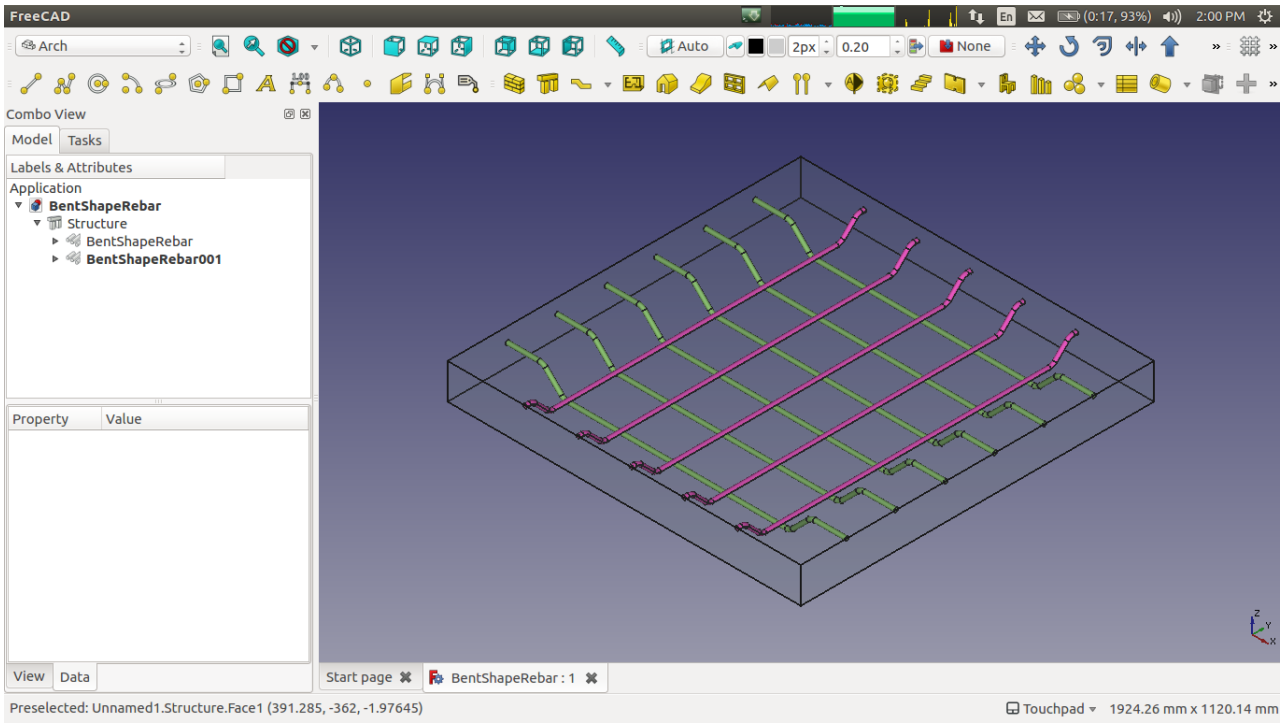

Figure 4.22: Bent Shape rebar reinforcement

4.8.5 Creates a Stirrup reinforcement bar in a selected structural element

4.8.5.1 Description

The Stirrup Rebar tool allows user to create a stirrup reinforcing bar in the structural element as shown in Figure 4.24.

4.8.5.2 How to use

- Select Stirrup Rebar tool from the rebar tools.
- A task panel will pop-out on the left side of the screen as shown in Figure **??**
- Click OK or Apply to generate the rebars.

4.8.6 Creates a Helical reinforcement bar in a selected structural element

4.8.6.1 Description

The Helical Rebar tool allows user to create a helical reinforcing bar in the structural element as shown in Figure 4.26.

4.8.6.2 How to use

- Select Helical Rebar tool from the rebar tools.

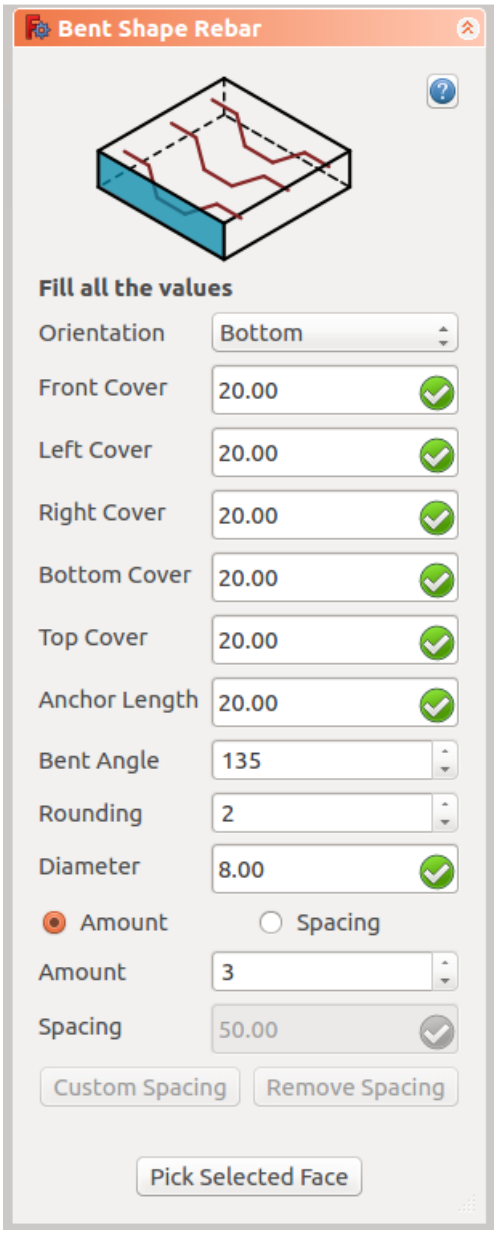

Figure 4.23: Dialog box of Bent Shape Rebar

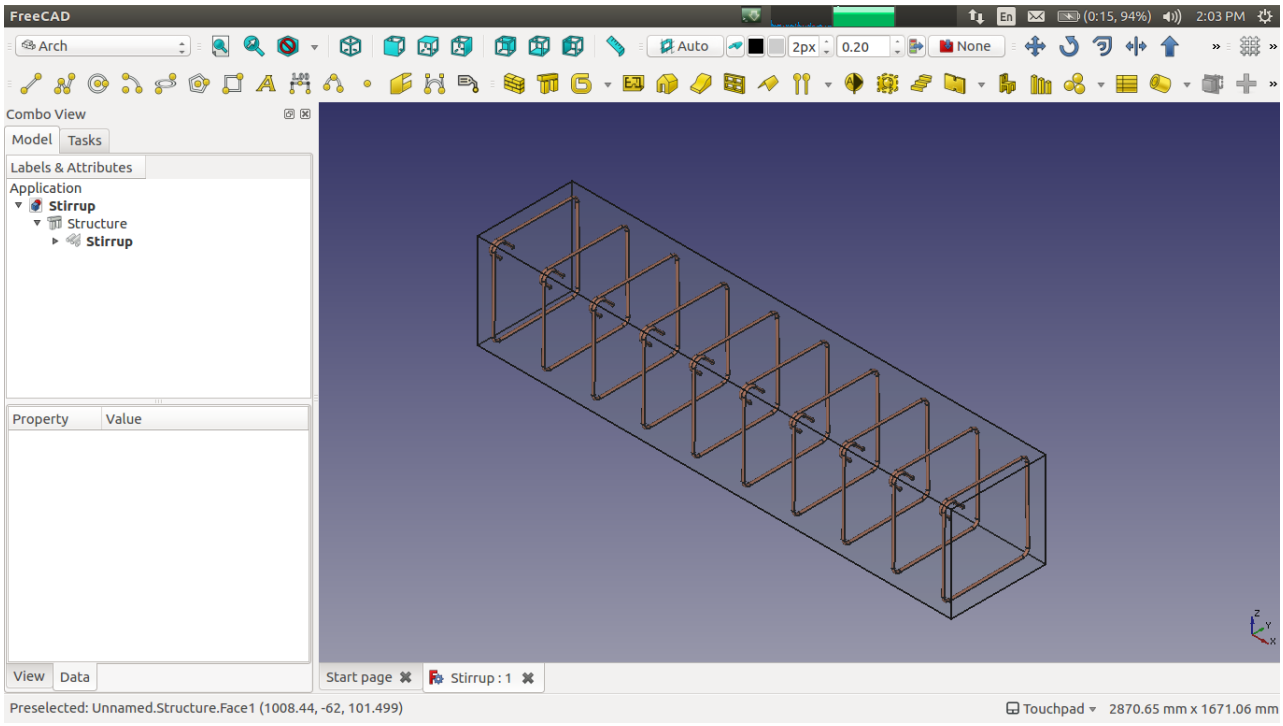

Figure 4.24: Stirrup rebar reinforcement

- A task panel will pop-out on the left side of the screen as shown in Figure 4.27
- Click OK or Apply to generate the rebars.

4.8.7 Custom Spacing

4.8.7.1 Description

The Custom Spacing tool allows a user to create rebar distribution in the structural element. You can define three segments for the distribution. For the first and third segments, you can give both a number of rebars and spacing between rebars. But for the second segment, you can only give either a number of rebars or spacing between rebars because one value automatically determines other.

For eg.: Given input values to Rebar Distribuiton dialog as shown in Figure 4.28:

Output produces by Rebar Distribution dialog when user click on OK button as shown in Figure 4.29:

4.9 Testing

FreeCAD is the software used by many of people all over the world and many web services as backend. so, it has to be perfect, Robust, reliable and bug-free. So, A very strong test suite is developed for testing this software and any component or feature that is to be integrated into FreeCAD. And Rebar Addon was able to pass all the task before getting Integrated into FreeCAD.

Test Suit for FreeCAD is developed using following two software:

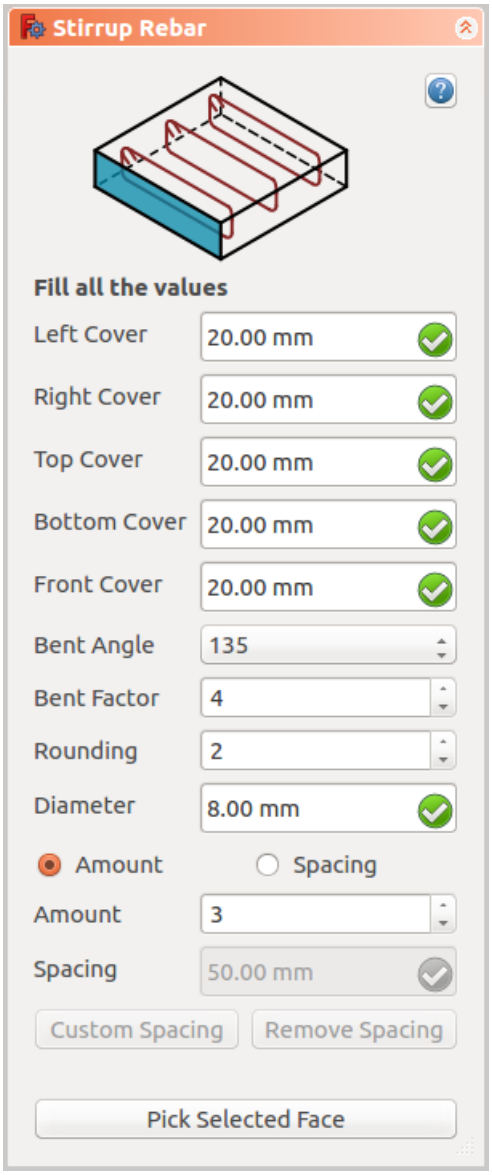

Figure 4.25: Dialog box of Stirrup Rebar

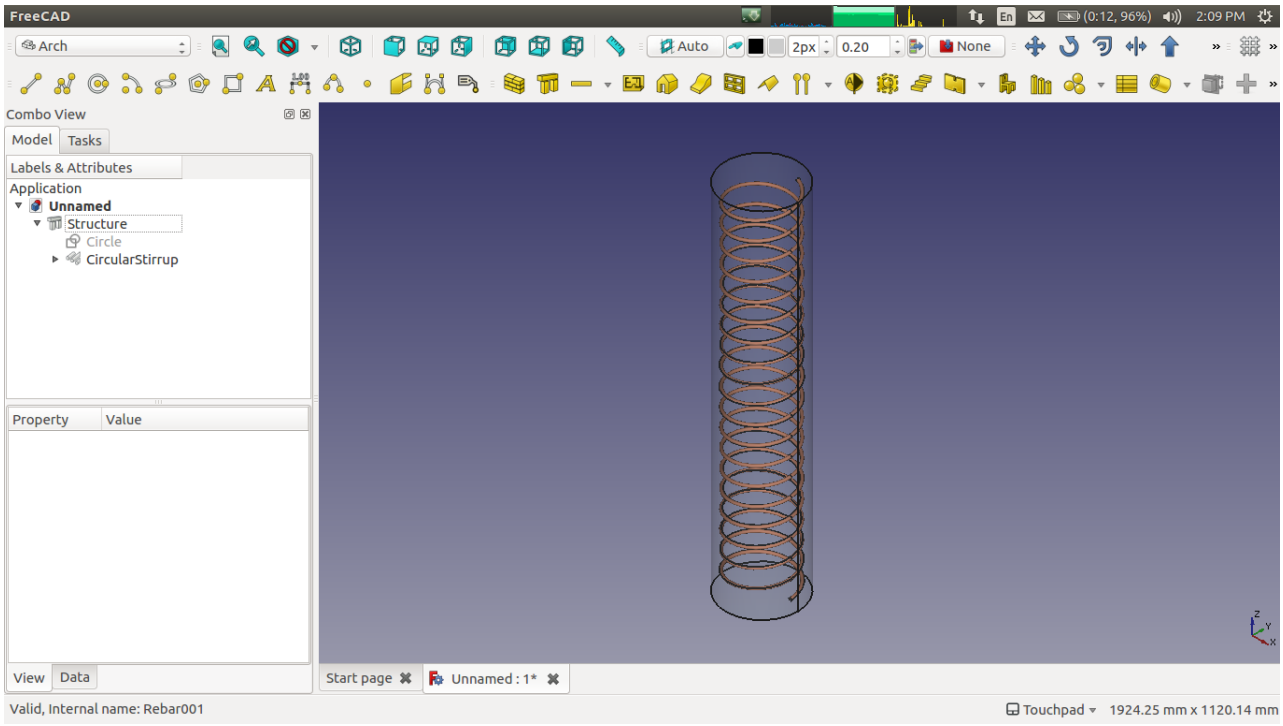

Figure 4.26: Helical rebar reinforcement

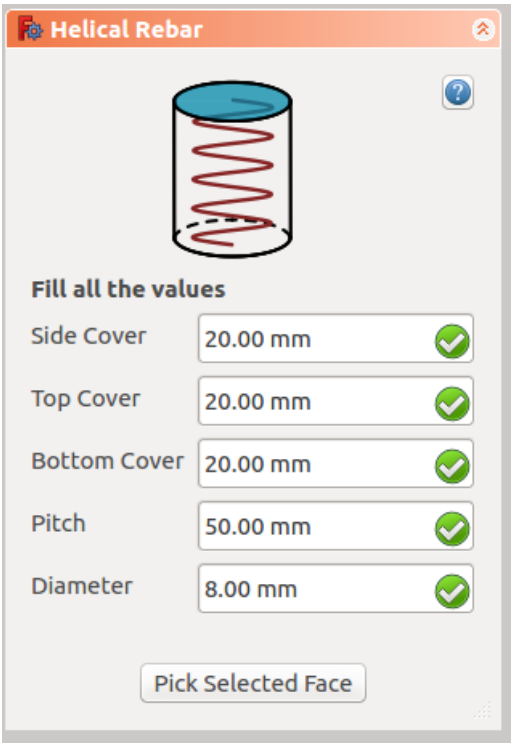

Figure 4.27: Dialog box of Helical Rebar

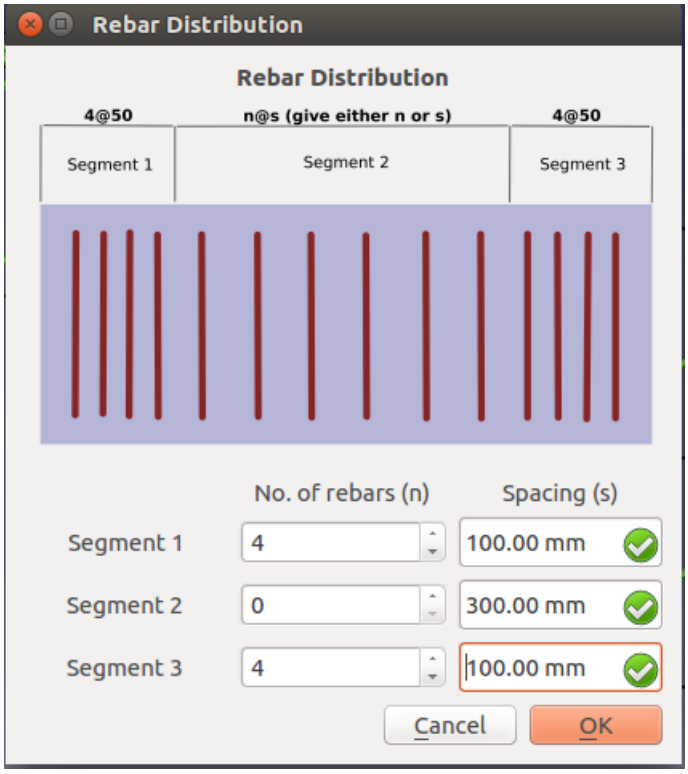

Figure 4.28: Dialog box of Rebar Distribution

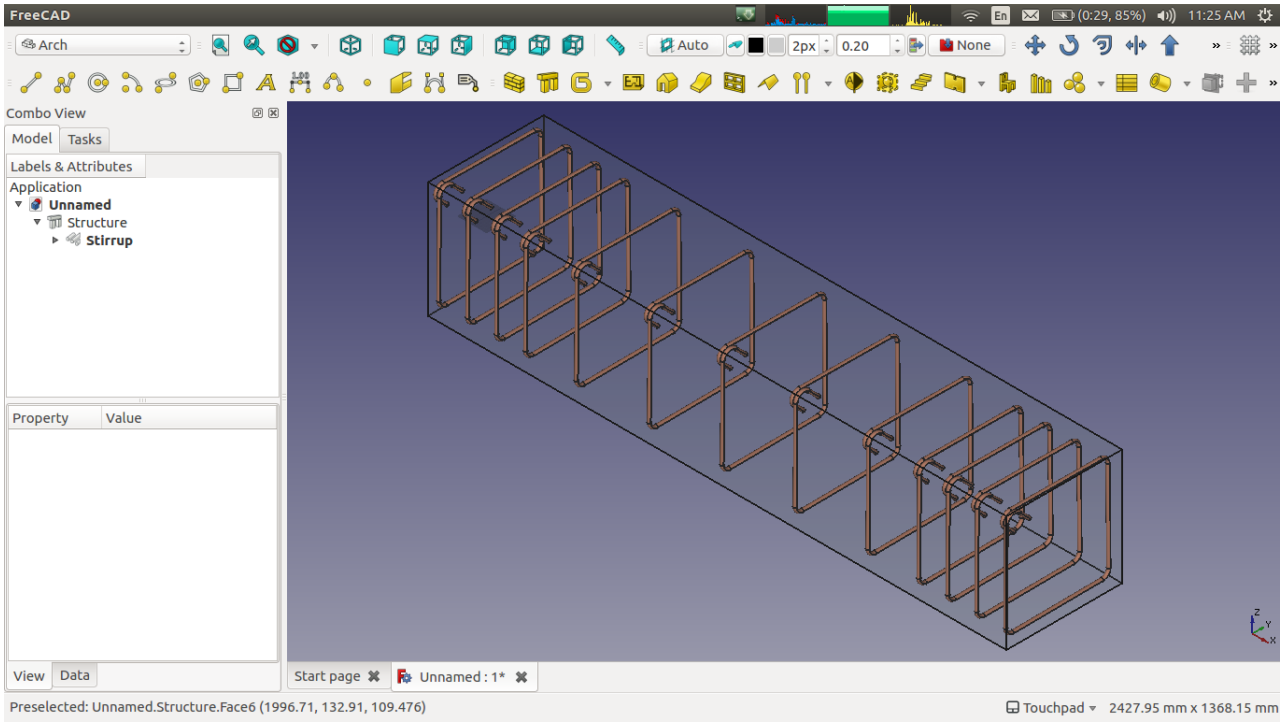

Figure 4.29: Rebar distribution of Stirrup reinforcement

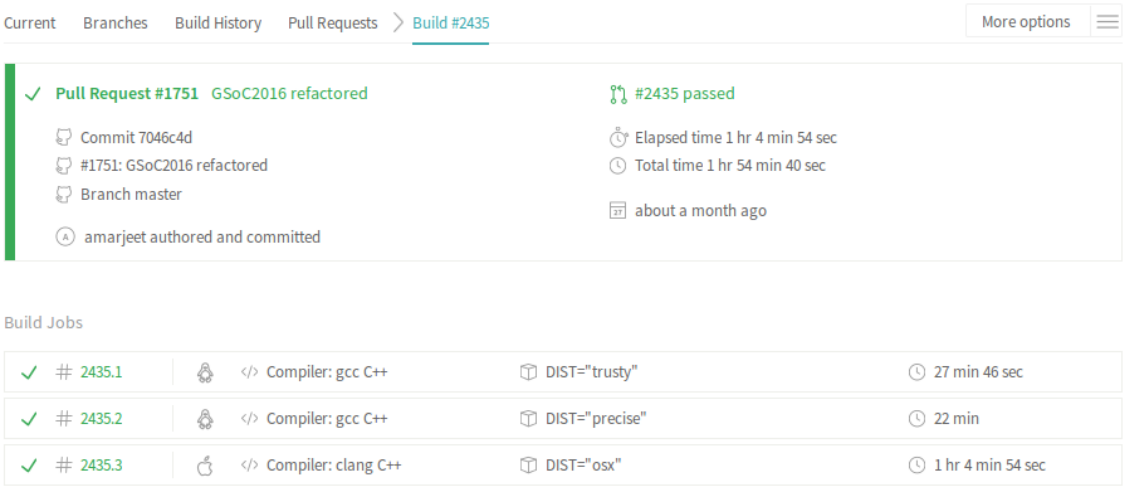

Figure 4.30: Travis test summary for 3 different OS

1. **AppVeyor:**AppVeyor is a hosted, distributed continuous integration service used to build and test projects hosted on GitHub on a Microsoft Windows virtual machine. AppVeyor is configured using a Web UI, or by adding a file named appveyor.yml, which is a YAML format text file, to the root directory of the code repository. Visual Studio Team Services (formerly Visual Studio Online) includes AppVeyor integration.

2. **Travis CI:** Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted on GitHub.Open source projects may be tested at no charge via travis-ci.org. Private projects may be tested at the same location on a fee basis. Travis CI is used mainly for Installation testing on different OS and It also perform black box and regression testing on different operating systems.

Test Suit developed for FreeCAD perform following type of testing:

1. **Installation Testing:** It is done by building the FreeCAD from scratch on totally fresh installation of different operating systems. Figure 4.30 and 4.31 shows installation testing of FreeCAD

2. **Regression Testing:** It is done by running the test cases developed for the OpenSCAD before the New changes are made to check whether new Changes doesn't produce abnormal behavior. Figure. 4.32

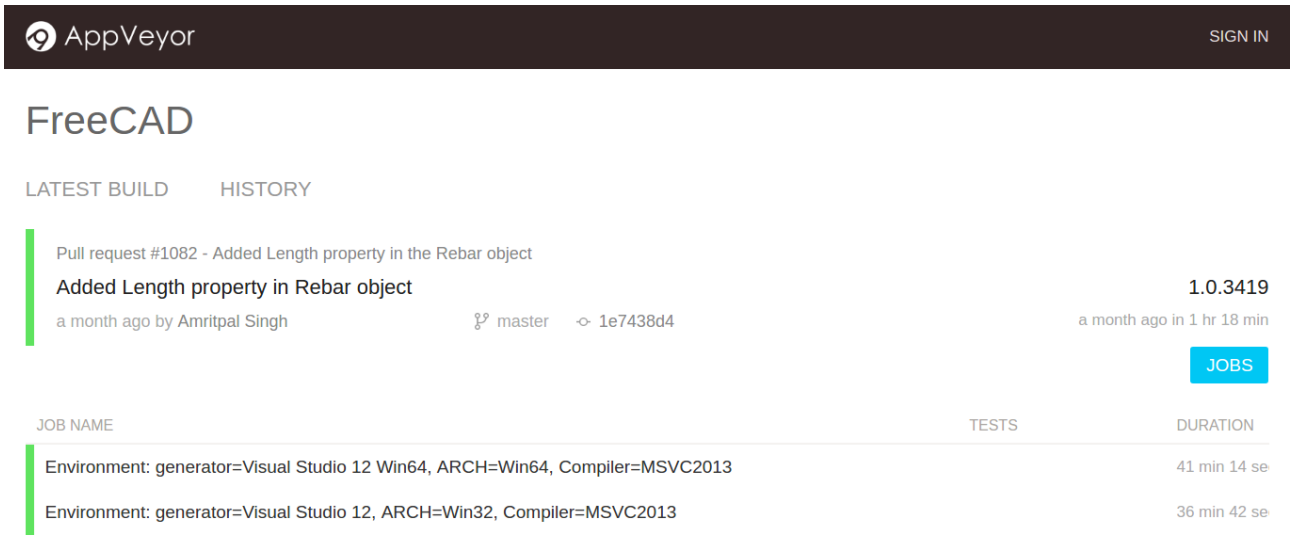

Figure 4.31: AppVeyor test summary for Microsoft Windows virtual machine

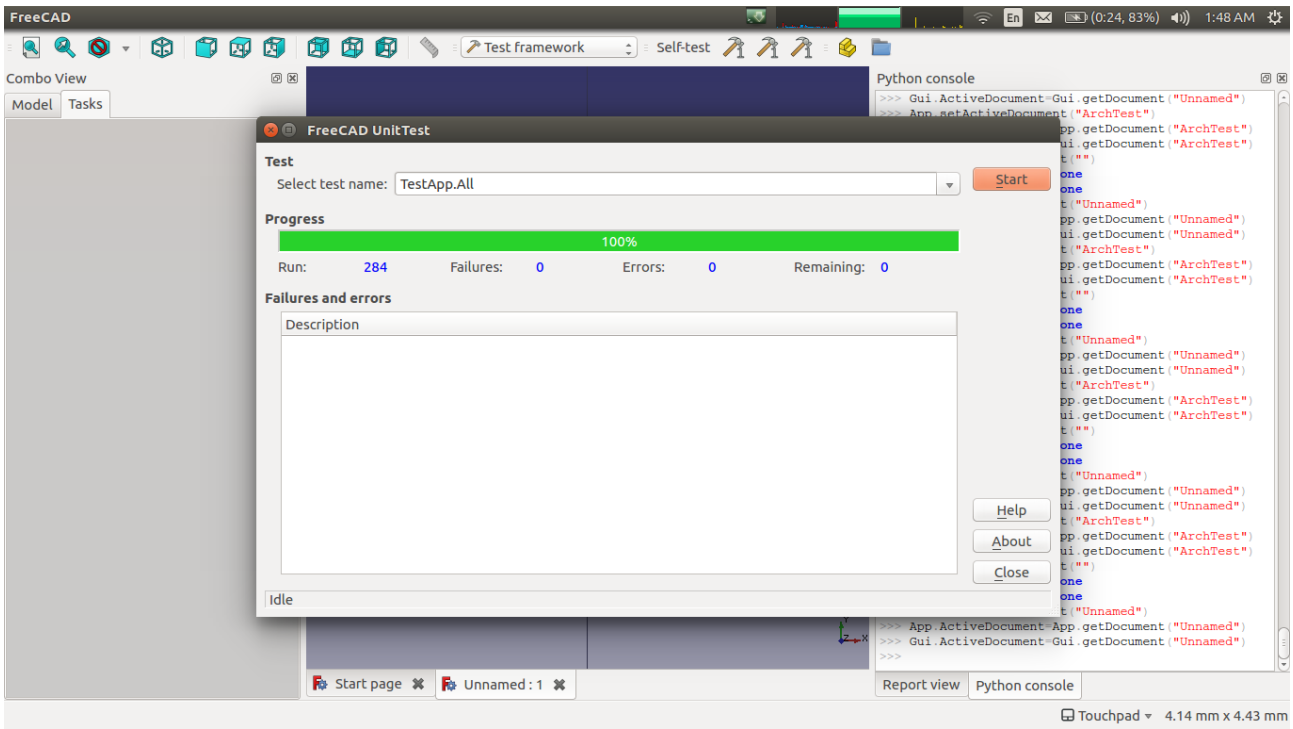

Figure 4.32: Unit testing

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

This project was something which required in-depth view in structural engineers field and analysis of your problems. Worked on a Google Summer of Code (GSoC) project with FreeCAD. This project is purely related structural engineering, aimed at easing up the reinforcement process in the FreeCAD. For better user experience, list of basic shapes of rebars are implemented under Rebar tools in the form of dropdown. While normal users can use them in real world complex structures, the same codebase can be used as a platform for implementing advanced rebar shapes.

A great deal of things are learned while working on this project. The learning was not limited to project only but the whole experience of working as a trainee under Dr. H.S. Rai at TCC, developer at FreeCAD and also as a mentor for Google Code In was immensely educational. Working with the Open Source Community and a variety of people of different age group, one is always challenged by the fundamental difference between classroom coaching and real World experience. But such a challenge is exactly the purpose of six months training.

The whole experience of working on this project and contributing to a few others has been very rewarding as it has given great opportunities to learn new things and get a firmer grasp on already known technologies. Here is a reiteration of some of the technologies I have encountered, browsed and learned:

1. Operating System: Linux
2. Language: Python, PySide, IfcOpenShell, C++, LaTeX
3. FrameWork: Qt, Django, Django Rest framework
4. API: Faebook Graph API, Social Auth
5. Softwares: Git, QtCreator, Inkscape, Doxygen
6. Building Tools: CMake, make
7. Communication tools: IRC, Gitter

So during this project, I learned all the above things. Above all, I got to know how software is developed and how much work and attention to details is required in building even the most basic of components of any project. Apart from above I also learned things like:

1. Planning
2. Designing
3. Developing code
4. Working in a team
5. Testing
6. Licensing Constraints

7. How Open source community work?
8. Writing Readable code
9. Coding Standards

And these are all very precious lessons in themselves.

5.2 Future Scope

FreeCAD being an open source project and supported by a large open Source community have a lot of scope for future improvements and additions as other individuals can also contribute in it and add additional functionality.

Being an Open Source project there is a constant flow of suggestion and demands by people for additional functionality and improvement in existing features. Rebar Addon being no exception constant flow of feature of suggestions and feature requests even before the start of development. Here is a small list of the Features that would be added in near future.

1. Adding more rebar shapes under Rebar tool.
2. Implementation of bill of material.
3. Some rebar object properties not export to IFC (Industry Foundation Classes) format.
4. Rebar distribution is missing in helical rebar.

BIBLIOGRAPHY

[1] "User:Amritpal_singh/gsoc_proposal - BRL-CAD", Brlcad.org, 2017. [Online]. Available: https://brlcad.org/wiki/User:Amritpal_singh/gsoc_proposal. [Accessed: 27- Nov- 2017].

[2] "Dev logs of Rebar Addon for FreeCAD - GSoC project", FreeCAD Forum, Amritpal Singh, 2017. [Online]. Available: https://forum.freecadweb.org/viewtopic.php?f=8&t=22760. [Accessed: 27- Nov- 2017].

[3] "FreeCAD User Manual/Documentation", 2017. [Online]. Available: https://www.freecadweb.org/wiki/Main_Page. [Accessed: 27- Nov- 2017].

[4] "FreeCAD Tutorial: New Rebar Tools" Youtube, 2017 [Online]. Available: https://youtu.be/BYQQjEKmx5E. [Accessed: 17- Nov- 2017]

[5] "FreeCAD/FreeCAD", GitHub, 2011. [Online]. Available: https://github.com/FreeCAD/FreeCAD. [Accessed: 27- Nov- 2017].

[6] "User:Amritpal_singh/GSoC17/logs - BRL-CAD", Brlcad.org, 2017. [Online]. Available: https://brlcad.org/wiki/User:Amritpal_singh/GSoC17/logs/. [Accessed: 27- Nov- 2017].

[7] "amrit3701/FreeCAD-Reinforcement", GitHub, 2017. [Online]. Available: https://github.com/amrit3701/FreeCAD-Reinforcement. [Accessed: 27- Nov- 2017].

[8] "Doxygen: Main Page", Doxygen.org, 2017. [Online]. Available: http://www.doxygen.org. [Accessed: 27- Nov- 2016].

[9] "FreeCAD", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/FreeCAD. [Accessed: 27- Nov- 2017].

[10] http://www.freecadweb.org/

[11] "MakerBot Customizer", Customizer.makerbot.com, 2016. [Online]. Available: http://customizer.makerbot.com/docs. [Accessed: 27- Nov- 2016].

[12] "User Interface for Customizing Models", amarjeetkapoor1, 2016. [Online]. Available: https://amarjeetkapoor1.wordpress.com/2016/07/04/user-interface-for-customizing-models/. [Accessed: 27- Nov- 2016].

[13] "PySide", Wiki - qt, 2017. [Online]. Available: https://wiki.qt.io/PySide_Tutorials. [Accessed: 27- Nov- 2017].

[14] "Qt (software)", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Qt_(software)#/media/File:Qt_logo_2015.svg. [Accessed: 27- Nov- 2017].

[15] "Git", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Git. [Accessed: 27- Nov- 2017].

[16] "Qt (software)", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Qt_(software). [Accessed: 27- Nov- 2017].

[17] "Python", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Python. [Accessed: 27- Nov- 2017].

[18] "Travis CI", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Travis_CI. [Accessed: 27- Nov- 2017].

APPENDIX A

IMPLEMENTATION EXAMPLE

Example 1: Create LShape reinforcement in structural element.

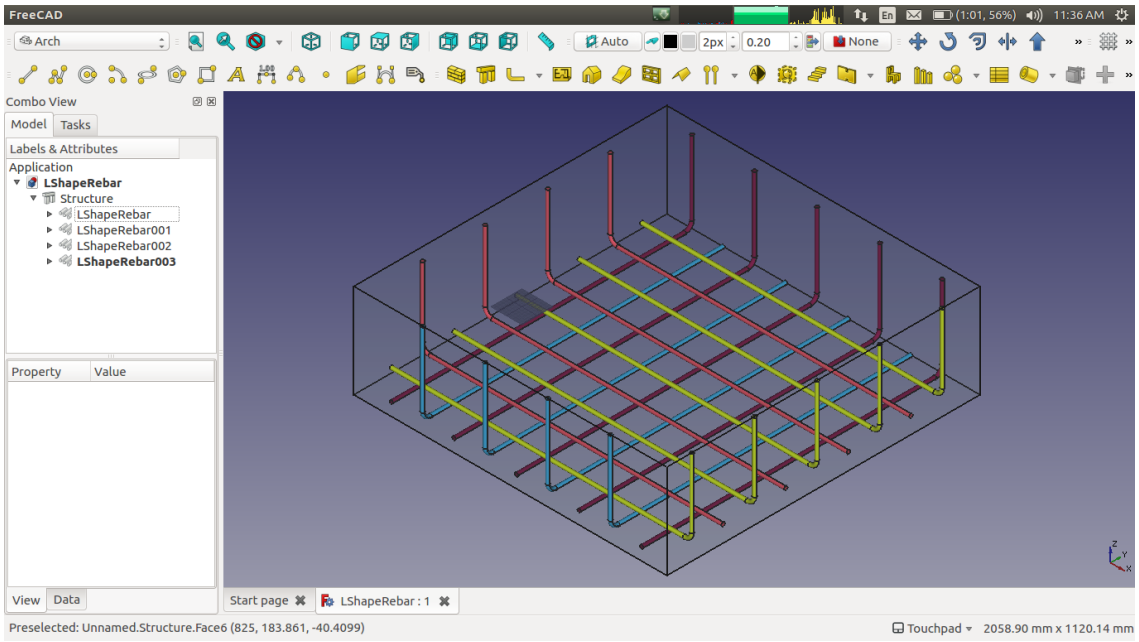

Figure A.1: LShape reinforcing bar in the structural element.

```
# Creating LShape rebar.
import Arch, LShapeRebar
structure = Arch.makeStructure(length=1000.0, width=1000.0, height=400.0)
structure.ViewObject.Transparency = 80
FreeCAD.ActiveDocument.recompute()
rebar = LShapeRebar.makeLShapeRebar(20, 20, 20, 20, 8, 20, 2, True,\
    10, "Bottom Left", structure, "Face1")

# Changing properties of LShape rebar.
import LShapeRebar
LShapeRebar.editLShapeRebar(50, 50, 20, 20, 8, 20, 2, True, 5, "Top Left")

# Written by Amritpal Singh <amrit3701@gmail.com>
#
# To the extent possible under law, the author(s) have dedicated all
# copyright and related and neighboring rights to this software to the
# public domain worldwide. This software is distributed without any
# warranty.
#
# You should have received a copy of the CC0 Public Domain
# Dedication along with this software.
# If not, see <http://creativecommons.org/publicdomain/zero/1.0/>.
```