

Plugin based system for data visualization.

Rahul Mule¹, Prasad Tupe², Nilesh Birajadar³, Harshal Patil⁴

^{1,2,3,4} BE Student, Dept. of Computer Engineering, Terna Engineering College, Maharashtra,

ABSTRACT

Data Visualization is the best method to understand the information easily. Understanding flow of data is very impossible. But with the help of the data visualization, it is nearer to possible. Gaining the insights from data means achieving goal of that data. The large sets of data isn't always straightforward. Sometimes, data sets are so large that it's downright impossible to discern anything useful from them. That's where data visualizations come in. Creating data visualizations is rarely straightforward. It's not as if designers can simply take a data set with thousands of entries and create a visualization from scratch. Sure, it's possible, but who wants to spend dozens or hundreds of hours plotting dots on a scatter chart? That's where data visualization tools come in. The research aims to describe the plug-in-based framework for data visualization. The main aim of developing this plug-in-based system is to improve the data visualization easier for the users. We are developing a software for data visualization using plug-in based system. In this more plug-in can be added in order to perform different types of visualization. A framework will be developed to extract data from .csv file. The user can extract the data from file and visualize data in different type eg. Graph, Pie Chart etc. The result of the research been implemented is supposed to provide accurate visual of the given data the output, which specifies that plug-in-based system works in an intended way.

Keyword: - Plugin, Data visualization, DLL

1. INTRODUCTION

In daily life, large no. of data is produced every second. One of main important feature of data is understanding flow of data. To achieve this goal, we have to visualize it into an understandable charts or graphs. For example, these days stock market changes every seconds which is very unpredictable.[2] We can't understand the market without graph that whether it is bullish or bearish. But one point is certain that after analyzing market graph, we can predict what will next LTP which will not be accurate. Data visualization is technique of converting data into visual context i.e. graph, chart to make it easier for human to understand and gain insights from it.[1]The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large data sets. It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new pattern.

The main objective of research is to design framework which allows the user to visualize the data. This framework will take input of file of extension with .csv file and extract data from it and save it in array and represent it various types of graph and chart. For this kind of framework, the data and view both could vary hence a plugin based system will be implemented in order to make it extendable[3]. The framework will implement line graph as a plugin, hence demonstrating the plugin system and functional UI. Extendible software which the developers can add support for various graphs and data types as per the need.[3]

2. LITERATURE REVIEW

There are lots of example of using of visualizing method in history and modern life The concept of using pictures to understand data has been around for centuries, from maps and graphs in the 17th century to the invention of the pie chart in the early 1800s. Several decades later, one of the most cited examples of

statistical graphics occurred when Charles Minard mapped Napoleon's invasion of Russia. The latter half of the 20th century is what Friendly calls the 'rebirth of data visualization', brought on by the emergence of computer processing. Computers gave statisticians the ability to collect and store data in increasingly larger volumes, as well as the ability to visualize the information quickly and easily. The 1960s and 1970s saw the emergence of researchers like John W. Tukey in the United States and Jacques Bertin in France, who developed the science of information visualization in the areas of statistics and cartography, respectively. The early 80s saw the emergence of Edward Tufte, whose seminal work,[2] *The Visual Display of Quantitative Information* is still used today in university courses for data visualization and statistical analysis. Tufte also introduced us to the sparkline, which gives the general shape of a trend in a small amount of space.

Nowdays, there are lots of application developed for this purpose. Here are some of them: tableau, data wrapper etc.[1] For web application, library like D3.js has been developed.[2] One of the famous library is Matplotlib which is compatible with python.[2] Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Those applications and libraries have a drawback which is non-extensibility. This drawback is resolved by plugin-based system. Some of example of plugin based system are Abode Flash, Quicktime, Realplayer. We think that everybody should have experienced use of Abode Flash. After running it will ask you to install flash plug-in. Adobe Flash Player is a plugin that allows browsers such as Firefox to display Flash media on web pages.

3. PROPOSED SYSTEM

The proposed system is a plug-in based framework which allows the user to visualize the data. For this kind of framework, the data and view both could vary hence a plugin based system will be implemented in order to make it extendable. This is an Extendible software which the developers can add support for various graphs and data types as per the need.[3] This application supports dynamic loading of Dynamic Link Library (DLL) which is intended to made so customization and/or enhancement of the existing application without having to have it recompiled or, as that matter of fact, have access to its source code. This framework is divided into three parts:[4]

1. Application UI,
2. Plug-in Manager,
3. Plug-ins.

Plug-ins are nothing but Dynamic Link Library(DLL), [6] Those DLL are created for perform certain function. In our system, we have developed DLL for loading data file, extracting data from it, and for every type of graph, etc. All of these plug-ins are saved in plugins folder. Plug-in Manager manages the life-cycle of the plugins and exposes them to the main system. It can find and load plugins, initialize them, register factory functions and be able to unload plugins. It should also let the main system iterate over loaded plugins or registered plugin objects.

The application UI loads the plugins and creates plugin objects via the plugin manager.[5] Once a plugin object is created and the main system has some pointer/reference to it, it can be used just like any other object. The application loads the DLL files using the standard OS API calls for dynamically loading libraries like LoadLibrary on Windows and dlopen on UNIX/Linux[5]. The validity of the plug-in is verified by making calls to certain functions expected to be implemented by the plug-in developer

The most important part of developing plug-in is creating plugin header interface. Our plugin interface represented by structure **pluginstruct** which hold following functions:[4]

- Creating
- Destroying
- Getting plugin name
- Running

```
struct _pluginstruct{
```

```

CREATEFP create;

NAMEFP name;

RUNFP run;

DESTROYFP dest;

}

```

The function pointers are to those functions expected to be implemented by the plugin developer. The Windows API function LoadLibrary loads the plugin dll into the memory and returns a handle to it from the function shown above. [6]

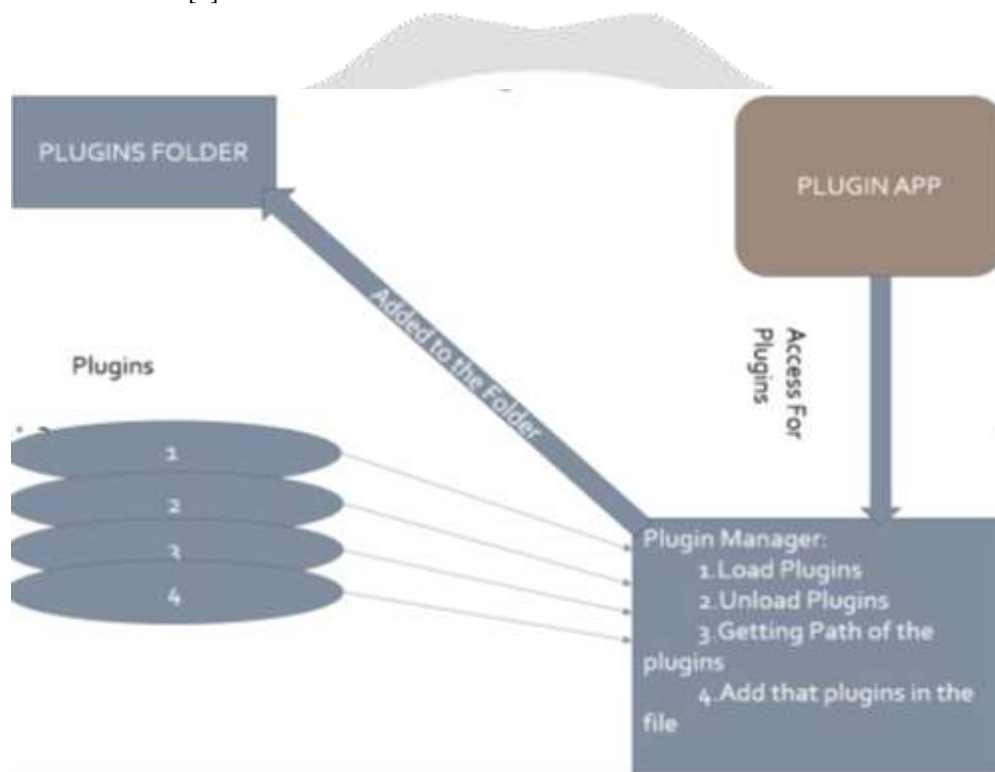


Fig 1. Architecture diagram

After running the framework, application will search for it in plugin folder via plugin manager. If it is there in folder, then plugin manager will load that plugins. And it will do its function. Suppose we want visualize data into pie chart, then it will search for it in plugins folder and load into in application represent data into pie-chart.

4. SCOPE

In modern life, quantity of data is increasing exponentially with every second. Representing such data is most difficult task in itself. This application can be developed into new updated develop which will be able to read and extract data from any type file i.e. .csv, .xls, .txt. Suppose we are trying to represent data from .xls, then application should convert that data file into .csv if possible otherwise extracting data into arrays and represent it.

Another feature can introduced that is there should be an option to add new dll directly into application without interfering source code of application. With help of multidimensional visualization, we can understand data very easily.

5. CONCLUSION

The plug-in based framework for data visualization is system which extracts data from .csv file and represent into charts, graphs with various patterns like slash, solid slash...etc.

There are lots of application for data visualization. But in these application if there is a type of graph not available in application, user can't represent it. So using plug-in based architecture user can represent it by creating and loading new DLL for that graph separately. This is the ability of extending use of application. We can add new feature as plugin in system which will help extendibility of system.

6. ACKNOWLEDGEMENT

This research work would not have been possible without the research efforts of many individuals. We would like to pay our sincere gratitude to our professor, Prof. Vishwajit Gaikwad for the opportunity that inspired us on this accomplishment.

7. REFERENCES

- [1]. Omkar, Bhushan, Shweta, Amrita and Prof Mantale(2019) "Plug-in Based System for data visualization. IRJET Volume: 06 Issue: 04 e-ISSN: 2395-0056,p-ISSN: 2395-0072
- [2]. Jiaying Liu, Tao Tang, Wei Wang, Bo xu & Feng Xia(2018) "A Survey of Scholarly data visualization", 10.1109/ACCESS.2018.2815030
- [3]. Boto Bako, Andreas Borchert, Norbert Heidenbluth, Johannes Mayer "Plugin-based Systems with hierarchical Presentation"
- [4]. H. Cervantes, H.Almeida, J. Mayer, A. Perkuish(2006), "Plugin-Based Software Architecture"
- [5]"Grphical User Interfaces composed of plug-ins" in Proceedings of fourth European GCSE Young Researchers. Ser. Fraunhofer IESE Technical Report vol 053.02/E, Germany pp25-29
- [6] Robert Chatley, Susan Eisenbach, Jeff Magee "Modelling Framework for Plugins" ESEC/FSE 2003 9th European Software Engineering Conference and 11th ACM SIGSOFT Symposium on the Foundations of Software Engineering Helsinki, Finland