

# Prototype of Platform Independent Editor Using Unified Modeling Language

Challapalli Venkata Vijay Chaitanya<sup>1</sup> and Koduganti Venkata Rao<sup>2</sup>

<sup>1</sup>III/IV Computer Science and Information Technology

<sup>1,2</sup>Gayatri Vidya Parishad College Of Engineering,  
Accredited by National Board of Accreditation,  
Affiliated to Jawaharlal Nehru Technological University,  
Madhurawada, Visakhapatnam,  
Andhra Pradesh, India. Pin – 530041

**Abstract:** This paper describes a prototype of Editor. This prototype deals with web-based technology and occupies diminutive amount of space, which enables a team of users to work simultaneously on different systems with the same Editor. This has the capability of running in the server and client system independently. More over this Editor have achieved the goal of platform independency and it has it's own capabilities to run on different platforms. Along with these features this also have the feature to send messages to the server from the client.

To enable concurrent access and platform independency the editor uses some of the concepts of JVM, html, UML, and scripting languages. These features are used to facilitate editing and operating. With the help of JVM the messages are being sent to the server from the client.

## 1 Introduction

Operating systems play a major role in computing. Every operating system has its own advantages and disadvantages. But it is not fair to restrict a computer intellectual to a particular operating system, since the trend in the technology always varies from time to time. With the improving technology one cannot survive by having knowledge of only one operating system. But as the operating system changes it is becoming more and more difficult for a particular individual to have an idea of all the applications present in the operating system. Hence if the applications are made in such a way that they are platform independent then one can easily handle those applications and execute them to have their corresponding results. Here it decreases the necessity to remembering the name of the same application in different operating systems. Hence it is now required to have an application or software that can be run on any platform. More over it would be more useful if the same application has the features of forwarding the messages to the server from the client so that the server-client relationship can also be achieved through the same application.

The prototype described in this paper packs all the above features and helps the user to operate on any platform with the same application. This application mainly

deals with the text and the graphics modes (formats) of operation and has the capability of handling the text files and the graphic files.

Text files are handled in this editor using the concepts of html and scripting languages and these can be stored in the client system.

Graphics files are typically handles in the system. These files are handled using the concepts of JVM and html.

UML plays a major role in the design issue and every part of the system is designed in the UML and the diagrams in the UML play a key role in the design issue of the entire prototype.

## **2 Example Scenarios**

### **2.1 Example 1**

The editor in the DOS operating system is restricted only to that of the DOS and this can be opened by the command "EDIT". As this opens, one can type the corresponding file and can store it. This editor has it's own advantages and disadvantages. The main disadvantage of this editor is that it cannot handle the image files. This command individually occupies a total space of 80692 bytes.

### **2.2 Example 1**

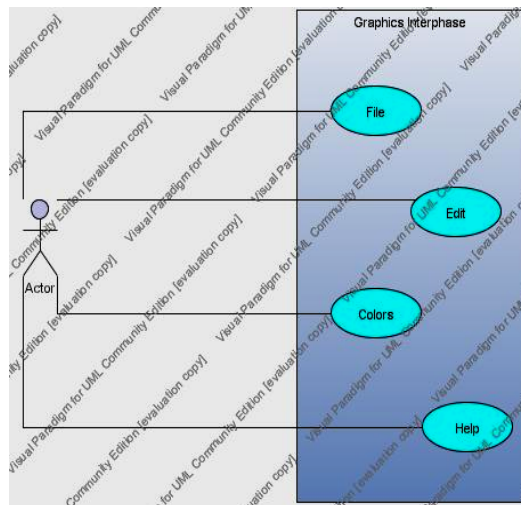
The editor in the WINDOWS operating system is "Notepad". This also acts in the similar fashion to that of the general dos editor have almost the same options. For image files one has to open the "Paint" in windows and continue with his work. One cannot have the concurrent access of Notepad and Paint from the same application and they as a whole occupies a disk space of 73,728 bytes.

## **3 System Overview**

As mentioned above the Editor described in this paper is completely web-based and browser running Editor. This Editor is a composite of both text mode of input and graphic mode of input. The Editor is likely to start at a page where the user is given a choice to choose whether he wants to go for the text mode of input or the graphic mode of input.

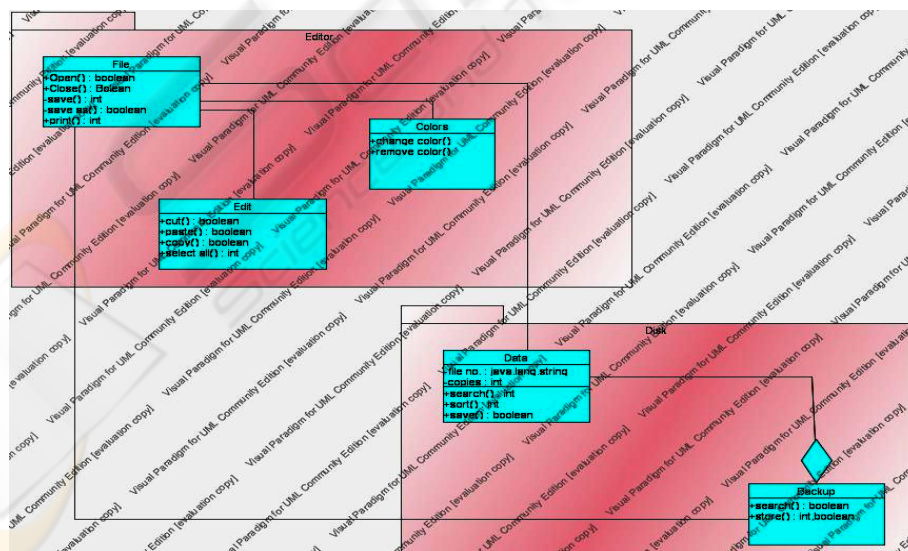
With in the project each user is given permissions to

- Create a new file
- Edit the files
- Change the colors of the background
- Add figures to the files.

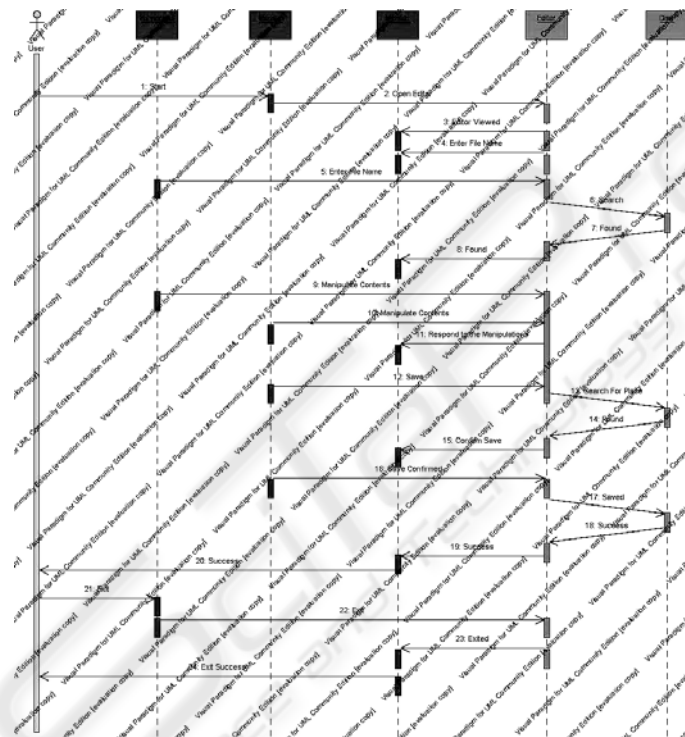


The original user deals with the file use case. Here he opens a new file and has permissions to input the text or to draw the figures. The edit use case helps him to edit the contents of the file and the colors use case helps him to add colors to the total project and the help use case to provide necessary help to the user. The user can directly interact with the graphics editor package, which provides all the above-mentioned use cases and can draw the required figures or to input the required text for him.

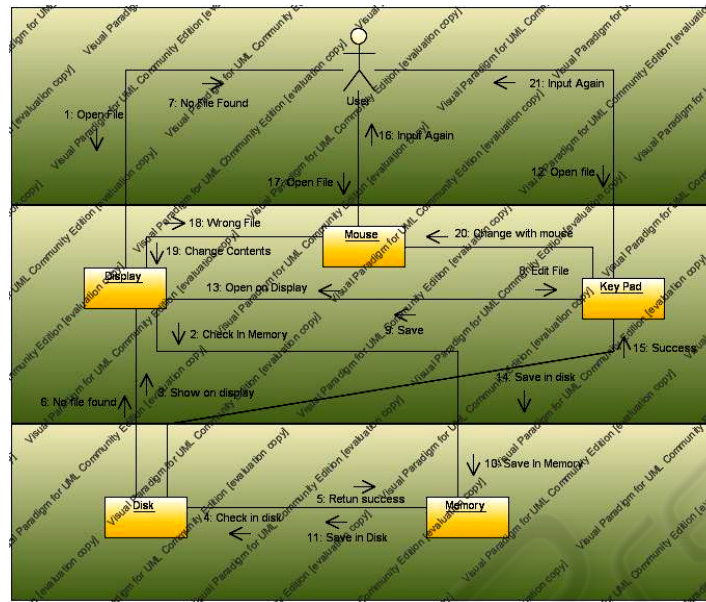
The general architecture or the class diagram of the above mentioned graphics editor have the corresponding operations to handle the files and their objects. Here the class diagram consists of the methods such as open, close, save, save as, exit for the classes such as file, edit, etc., This diagram mainly deals with the major operations of the file handling and achieves the total concept of the project.



The major design issue lies in the sequence, which the total editor follows. Here the total sequence is to be taken care of and have the capability of representing the total diagram in the single sequence diagram. Our editor directly opens through any platform and has the capability to ask the user about the operation he wants to use and the mode in which he wants to enter the editor. Here he is given a chance to select the options. Since the complete user interface is designed in html the user even without having a minimum knowledge of the computer can also easily access the complete system. Then he can select the respective option and can follow the mentioned respective operations on the screen and can easily handle the complete software. The sequence diagram is as shown in the figure and user can directly handle the respective operations and can achieve the result.



The design issue also lies in the collaboration diagram where one can view the total concept in the top-down approach and he can realize the total concept laid inside it and the path through which the messages take the choice, which the user opts. Here the objects of the same category lie in the same swim lane and this helps the user to identify easily where the object belong to the which category and he can come to a final understanding where he can directly interact with the system more easily and conveniently and effectively. The more the user gets into the details of the system the more he can adjust with it.



## 4 Discussion

Although the development of the prototype presented here is still going on it would be useful to review the benefits and the limitations of the Editor.

### 4.1 Benefits

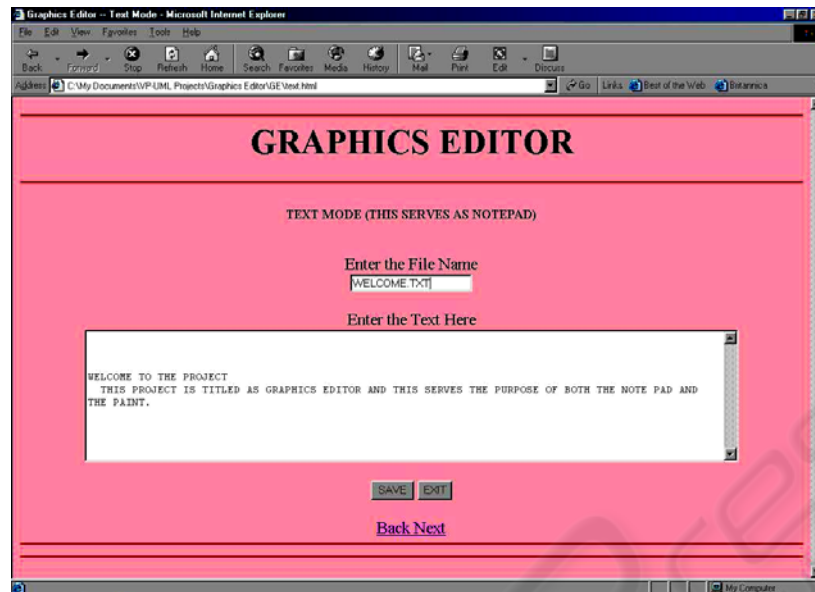
This Editor has its unique benefits, which are not present in the other editors.

This is completely a web-based editor and hence all the benefits that are provided in the browser can be directly used in the editor with out any limitations.

This provides a chance to use the editor and the Internet at one instance of time since this is completely a browser-based editor.

This has the capability of forwarding the messages (graphical) to the server directly (only when the net is connected) through this editor and hence message passing becomes so easy.





Since the Editor is completely designed using the html and JVM, bagged up all the features of both of these, this have the capability of running in any operating system with out the intervention of the operating system.

The major benefit in this is that the operating system (i.e., on which it is running) only provides maximum options to it and hence the designer of the system does not have to spend more time in providing all the options to it.

This provides the most user friendly nature and this can be operated by any user who do not have that much knowledge about the computer as well as the operating system especially.

## 4.2 Limitations

Because of the editor is designed completely in the browser there are certain limitations to it.

We have first designed a prototype so that the original project time decreases and the prototype can increase the understandability of the user.

Since our Editor is a web based editor we cannot have the page numbers and other additional features with cannot be available in the browser.

Present system saves the text files only in the .txt extension.

It is limited only to create graphics files and is not capable of modifying the already created files, which are present in the system.

Since we have used the concept of applets in JVM, it is not capable of saving the file in the local machine.

This system does not support any paragraph counting, word counting and spell checking functions.

This is the complete overview of the system and this system has its own benefits and the limitations and more over there is so much to enhance in this system. The future enhancements are discussed below.

## 5 Conclusions and Future Works

Having implemented this prototype of a Editor which deals with the text and the graphics formats we are confident that this achieves some of the major goals of the present trend such as platform independency, less space utilization on the disk, more reliability, more user friendly and combined features of several applications. We can also have a look to the future works.

It is planned to implement the complete features of the WORD such as formatting, macros, tables, etc., in the Editor, which helps the user to edit his text completely.

An import function helps in importing the graphics file and helps in manipulating the contents of the graphics file. This can help the user more and the usage of the system improves.

Last but not the least: It is important to integrate these systems into the daily teaching and research practices to gain more experience and more impulses for the further developments. It should become a general practice to use these systems in the daily life to improve more and more.

## References

1. The Unified Language User Guide by Grady Booch, James Rumbaugh and Ivar Jacobson
2. Real Time UML by Bruce Powel Duglass
3. Analysis Patterns by Martin Fowler
4. Surviving Object-Oriented Projects
5. Object Solutions by Grady Booch
6. UML Distilled by Martin Fowler
7. Software Reuse by Ivar Jacobson
8. The Unified Software Development Process by Ivar Jacobson
9. Visual Modeling with Rational Rose and UML by Addison-Wesley Object Technology Series
10. Software Project Management by Walker Royce
11. The Unified Model Reference Manual by James Rumbaugh
12. Applying Use Cases by Geri Schneider
13. The Object Constraint Language by Jos Warmer
14. Enterprise Computing with Objects by Yen-Ping Shan
15. <http://user-mode-linux.sourceforge.net>
16. <http://www.onesmartclick.com>
17. <http://www.tigris.org>
18. <http://hotscripts.com>