

A Novel Method for Making Cut-Copy-Paste Operations Using Clipboard

Mayur M Patil^{#1}, Akkamahadevi R Hanni^{%2}, Priyadarshini M Patil^{*3}

[#]Student (2015 Batch), Dept. of CSE, B.V.B. College of Engineering and Technology, Hubballi 580031, Karnataka, India

[#]Student, Dept. of CSE, B.V.B. College of Engineering and Technology, Hubballi 580031, Karnataka, India

^{*}Assistant Professor, Dept. of ISE, B.V.B. College of Engineering and Technology, Hubballi 580031, Karnataka, India

Abstract— With the rise in browser usage and editors, it is becoming essential to find an easy way to manipulate text information that is used. The aim of this project was to create a tool for text data, the main criteria being to provide the options like multiple cut/copy/paste. We find ourselves in situations where we may have to enter the same data repeatedly. For this we have chosen java which has vectors and with the use of the system clipboard we achieve multiple cut/copy/paste. The application has a GUI which allows doing the same.

Keywords— cut, copy, paste, vectors, java, clipboard.

I. INTRODUCTION

A **clipboard manager** is a computer program that is used to add more functionality to the system clipboard. Clipboards provide only one buffer, overwritten by each new "copy/cut" operation. The main task of a clipboard manager is to store data copied in the clipboard in a way which allows user to use the data multiple times.

Clipboard manager allows the user to keep multiple objects which have been previously used, available for later use. This feature is beneficial in Form-filling, URL auto fills and text editing and it also provides multiple cut-paste operations that helps in making data transfer and editing operations.

The main objective is to provide user a tool that can facilitate in data handling by having options like multiple cut , multiple copy, multiple paste.

II. SYSTEM STUDY

A. PROPOSED SYSTEM

Clipboard Manager is a tool that provides options like Recently added, History, Skin chooser and Help that helps the user to manipulate text.

The advantages of the proposed system being it can be used to transfer text between applications and the availability to perform multiple cut copy paste operations. [1]

The limitation of the application at present being the application handles only string datatype.

The application has been built on SuseLinux and shall be available for Operating systems like Windows 7, Suselinux, MacOS.

III. EXISTING APPLICATION STUDY

We look at few of the existing applications that perform similar actions.

Among the available applications, we look at

1. CLCL
2. ClipMate
3. ArsClip
4. Ditto

A. CLCL

CLCL allows you to access your clipboard history with a simple press of ALT+C or from the icon in the system tray. Frequently used clippings can be saved to the Templates category, and you can customize the paste hotkey based on the program you're using. It stores the clippings you want, retrieves them with a simple keystroke-summoned menu, and is light on your system resources. [2]

Fig. 1 shows the GUI of the application.

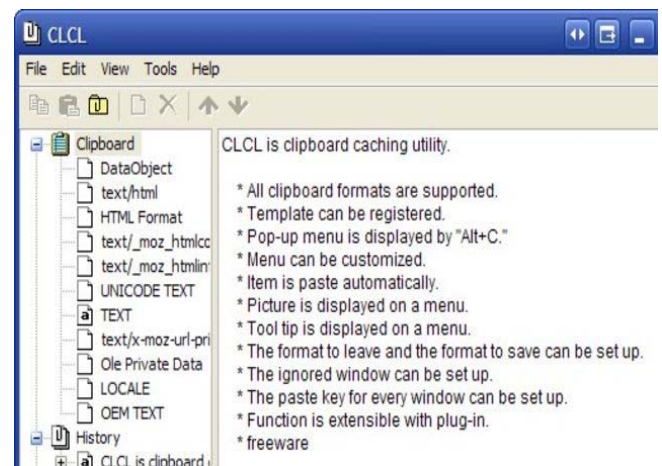


Fig. 1 CLCL application

B. ClipMate

ClipMate view gives you a toolbar with drop down options, the Explorer view—seen here—is the master management panel where you can group and edit your clippings. ClipMate is a non-freeware application. [2]

Fig. 2 shows the GUI of the application.

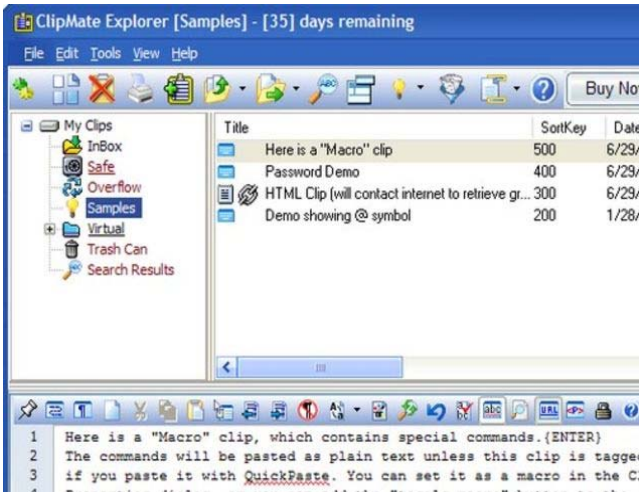


Fig. 2 Clipmate application

C. ArsClip

ArsClip supports and stores pictures, formatted text, Unicode, and HTML. ArsClip also supports files, allowing you to store frequently used/pasted files right in ArsClip. [2]

Fig. 3 shows the GUI of the application.



Fig. 3 ArsClip application

D. Ditto

Ditto is a portable clipboard manager that supports plain text, but it can also copy formatting and even images. You can group together your clippings to keep things organized by task or project, as well as assign hotkeys to frequently used clippings. Ditto also has built-in hotkeys for entering, searching, and retrieving your clippings [2]

Fig. 4 shows the GUI of the application



Fig. 4 Ditto application

IV. IMPLEMENTATION

The application has four main modules. They are the following:

- Recently added module
- History module
- Skin chooser module
- Help module

The application has various screens. The application can be made to act like a widget thus allowing users to use other applications with ease. The moment the focus is moved away from the application, the screen becomes semi-transparent and continues to run in the background. When the application is launched the vectors are initialized. Vectors are chosen for handling the data as there can dynamically grow in size as and when its necessary. Fig. 5 shows the main screen of the application with Add, History, Help and Skin chooser options.

Fig. 5 shows the main screen of the application

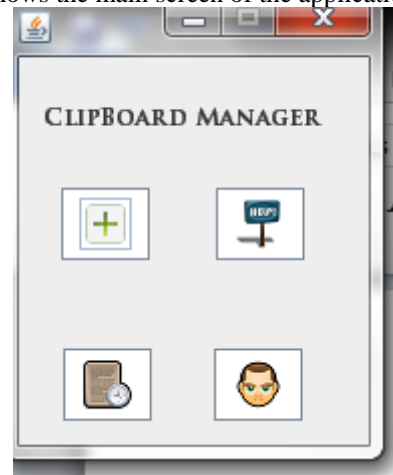


Fig. 5 main added screen

A. Recently added Module

The user chooses 'select' or 'select all' the items which acts as the input.

The selected item(s) shall be written to the system Clipboard ready to be pasted which is the output.

Pseudocode –

```

Select(int index)
{
    Item= Vector_data_item(index);
    Write_To_SysClip(Item);
}
Select_all()
{
    Item[]= Vector_data_item();
    While(Vector!=Null){
        Write_To_SysClip(Item);
    }
}
    
```

Fig. 6 shows the Recently added screen of the application

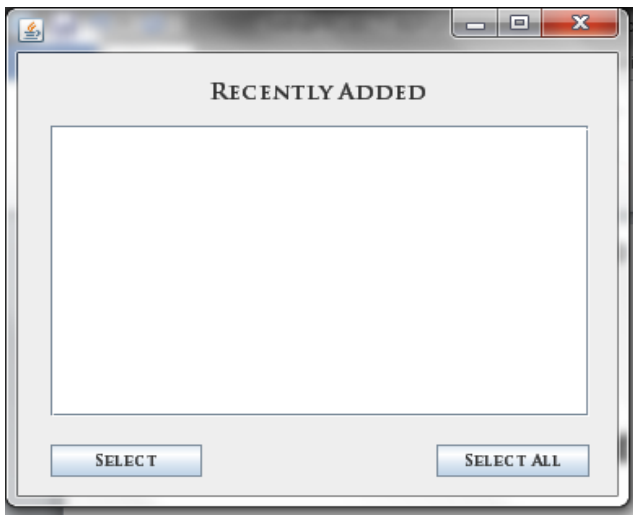


Fig. 6 Recently added screen

B. History Module

The user chooses select and clear or clear all the items which acts as the input.

The selected item(s) shall be cleared from the vector

```
Pseudocode –
Clear(Select (int index))
{
    Delete_Vector_data_item(index);
}

Clear_all()
{
    While(Vector!=Null){
        Delete_ Vector_data_item();
    }
}
```

Fig. 7 shows the History screen of the application

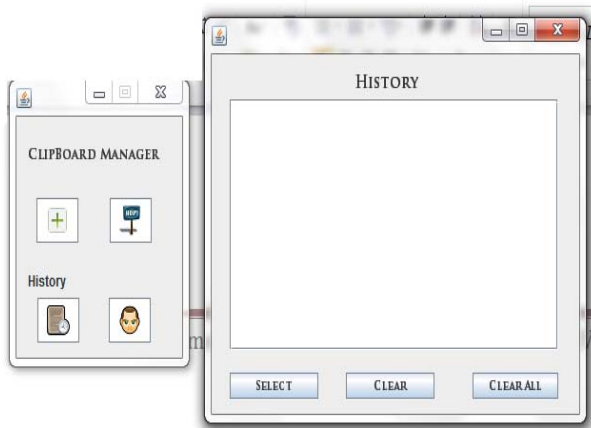


Fig. 7 History screen

C. Skin Chooser Module

The user chooses select the color options or revert back to the default option of the main panel and the buttons which acts as the input.

The selected settings shall be applied to the tool after showing a preview.

Pseudocode –

```
Apply_Color(Background_color,Button_color)
{
    Change_Panel_Settings(BgC,BtnC);
}

Preview_Apply_Colour(Background_color,Button_color)
{
    Show_Panel_Settings(BgC,BtnC);
}

Apply_Color_Default(Background_color,Button_color)
{
    Change_Panel_Settings(BgC_Def,BtnC_Def);
}
```

Fig. 8 shows the Skin chooser screen of the application

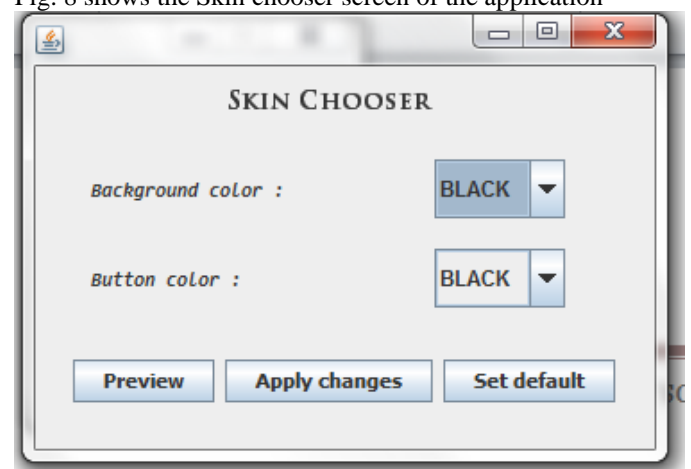


Fig. 8 Skin chooser screen

D. Help Module

The user chooses the watch tutorial option which acts as the input.

The video showing the tutorial shall be played.

```
Pseudocode -
Watch_Tutorial()
{
    Open_Tutorial_video(Path);
}
```

V. WORKING OF THE APPLICATION WITH MEMORY CONSUMPTION DETAILS

The application can handle data in the form of string datatype. When the user opens the application, the main screen opens up. The four options available for the user are Add, Help, Skin Chooser and History. When the user selects add, the data which has been copied gets moved into the vector and appears in the recently added section of the application. The more the number of copy action performed, the more are the number of items that appear on the list. This happens as the application checks the content of clipboard upon the event "CTRL+C" or right

click “Copy”. The content if present in the system clipboard is of string datatype, it gets written on the vector. [4] The Recently added section holds only the previous fifteen items that have been copied. Refer Fig. 6 which also has Select and Select all options allowing the user to choose the ‘item(s)’ that has to be copied.

The next option is the History section which stores the previous data items that have been copied. The previous two hundred and fifty records are stored. This data is also persisted by the use of files. All the items are listed in the history section that also has the option to clear selected or clear all the items that have been stored. Every time the application is closed, the file contents are rewritten. Please refer Table I for memory consumption changes without this option.

The skin chooser section allows the user to choose colours for the frames and the buttons of his choice. The user also has the option to set the colours to default. The user can preview changes before they make it permanent. The file used to persist data is also used to store configuration settings that allows user to keep the changes permanently.

Fig. 9 shows the Configuration file contents of the application that stores colour information and previously copied items

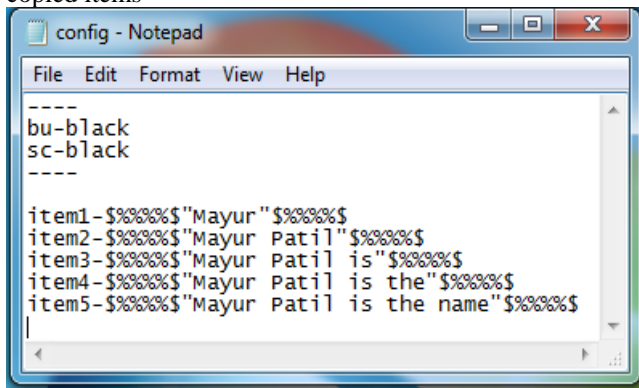


Fig. 9 config file

The file content (items) Refer Fig. 9 are amended with the format “\$%\$\$\$” at the beginning and at the end to avoid failure of capture of data or data being incorrectly represented.

The other option available for the user is the help option that has a hyperlink pointing to a video which shows the user how to perform the actions in the application.

Whenever the user selects paste after they have selected one or more items from either the recently added or the history section, the item gets automatically written on the system clipboard. The process of retrieval and writing from and to the system clipboard is done using the

“getSystemClipboard()“ function and by importing “java.awt.datatransfer.*“ and “java.awt.Toolkit”. [4]

The application produced is a portable application which can run on Windows 7, SseLinux and MaC OS operating systems. The application has a size of 346KB and configuration file is created when the application is invoked for the first time and persists itself. If the user deliberately removes the configuration file, a new one is created once the application is invoked again.

The application’s consumption of the memory has been observed and it is found that the consumption of the memory lies between 2.8 to 3.9 MB under standard conditions with the data and configuration settings as explained via Table I.

TABLE I
DATA AND MEMORY CONSUMPTION

Data (characters / item)	Memory Consumed		
	KB	MB	Without Persisting data (history option disabled) MB
100	2892	2.8	2.3
400	2980	2.9	2.5
1000	3021	3.0	2.5
1600	3102	3.1	2.7
2900	4216	4.2	3.9

VI. CONCLUSIONS

The Application ClipBoardManager has the capability of handing multiple cut copy paste operations using the system clipboard and built using java. It has four main options Add, history, Choose skin and Help with which the user can manipulate string datatype. The application is portable and works on Windows 7, SuseLinux and Mac OS operating systems. The limitation of the application is the application cannot handle media objects which can be the future scope of the application.

REFERENCES

- [1] Gerald Reif, Martin Morger and Harald Gall, “Semantic clipboard- semantically enriched data exchange between desktop applications”, Semantic Desktop and Social Semantic Collaboration Workshop at the 5th International Semantic Web Conference ISWC06, Athens, Geogria, USA, November 2006.
- [2] Jason Fitzpatrick. (2009) five best clipboard managers page on lifehacker. [Online]. Available: <http://lifehacker.com/5298615/five-best-clipboard-managers>
- [3] Robert C. Miller and Brad A. Myers, “Synchronizing Clipboards of Multiple Computers”, *Proceedings of the 12th annual ACM symposium on User interface software and technology*, p.65-66, November 07-10, 1999
- [4] Sun Microsystems, Inc. , *Java Code Conventions*, 1997.