ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

A Review On the Development of GEDIT 3 Plug-In: Go-To Definition for C Using Exuberant-Ctags

P.S. Bangare[1], Pramod Sonar[2], Pallab Pain[3] and Sadhna Advani[4]

Assistant Professor, Dept. of I.T, Sinhgad Academy of Engineering, Pune, India[1]

Team Leader, Persistent Systems, Pune, India[2]

Student Final Year B.E, Dept. of I.T, Sinhgad Academy of Engineering, Pune, India[3, 4]

ABSTRACT: A plug-in is a software component that adds a specific feature to an existing software. A software component that supports plug-ins makes it more customizable. Plug-ins are used in web browsers to add features such as web browsers, virus scanners etc. Nowadays plug-ins are becoming very popular with editors used for programming. Since programmers have started using editors, plug-ins have also become very popular to provide added functionality. A similar plug-in is already present in Visual C++. This plug-in will imitate the same behaviour and provide similar functionality in GEDIT 3.This paper discusses the development of the proposed plug-in in Python 3 using Exuberant Ctags for Linux based platforms.

KEYWORDS: GEDIT 3, Exuberant Ctags, Python 3, GTK+ 3, Linux.

I. INTRODUCTION

A software component that adds specific features to a software application is called a plug-in. A software application enables customization when it supports plug-ins. The most commonly used plug-ins are the ones used in web browsers such as search engines, virus scanners or being able to utilize a new file type like a new video format. The Adobe Flash Player, the QuickTime Player, and the Java plug-in, are the well-known browser plug-ins. The Java plug-in launches a user activated Java applet on a web page to its execution on a local Java virtual machine. There are many reasons why applications support plug-ins, the major reasons are:

- Enabling third party developers to create abilities which extend an application.
- To support the ease to add new features.
- Reducing the size of the application.
- To enable separating of source code from the application due to incompatible licenses.

Plug-ins are also supported by text editors to add features to them. There are certain text editors like Sublime text, ATOM and Notepad++ which allow developers to create custom plug-ins according to their need. These are the plug-ins that are not included in the basic package [1].

GEDIT 3 is a general purpose and default text editor of the GNOME desktop environment and supports third-party plug-ins to customize the text editor according to the user's need. This paper discusses the Go-To Definition plug-in which adds the feature of viewing definitions of identifiers used in a code. GEDIT 3 being the default editor with a GUI is getting more popular nowadays and adding this plug-in to it will make it more programmer friendly and provide a lot of help while coding.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

II. LITERATURE SURVEY

A. *Available plug-ins for GEDIT 3:*

Plug-ins have been present in GEDIT 3 since its release and along with time more third party plug-ins have been created by developers. A few plug-ins are shipped by default along with GEDIT 3 and other plug-ins can be downloaded from their website and added to GEDIT 3.

Shipped plug-ins with Gedit 3:

- *Check update*: Check for latest version of gedit
- *Spell*: Checks the spelling of the current document.
- *Sort*: Sorts a document or selected text.
- *Quick open*: Quickly open files
- *Python console*: Interactive Python console standing in the bottom panel [3].

You can find more plug-ins at *http://wiki.gnome.org./Apps/Gedit/ShippedPlugins*

Third-party gedit3 plug-ins:

- *Advanced Find/Replace:* Advanced find/replace functions. Find/replace documents.
- *Class Browser 3g:* List functions, classes, etc. In the side pane and supports the languages from ctags, special parsers for Python, HTML, XML, Diff, Ruby, Markdown and Changelogs.
- *Click Regex:* Gedit plug-in to provide configurable regular expression to apply on mouse click.
- *Control Your Tabs:* Switch between document tabs using Ctrl+Tab / Ctrl+Shift+Tab (most recently used order and Ctrl+PageUp / Ctrl+PageDown (tabbar order).
- *Open URI Context Menu:* Adds context menu item to open/browse an URI at the pointer position.
- *Project Manager:* Lists projects in the side panel, restorelast opened files for project, find directories that seems to be projects and more [4].

You can find more plug-ins at *http://wiki.gnome.org./Apps/Gedit/ThirdPartyPlugins-v3.8*

B. *Current existence*

Go-to-definition feature is already present in the Visual Studio IDE. Since Linux is a widely used platform now and many programmers use GEDIT 3 to write their code it has become a necessity that even GEDIT 3 should be equipped with the Go-To-definition plug-in. The implementation of the go to definition plug-in in Visual C++ is as follows:

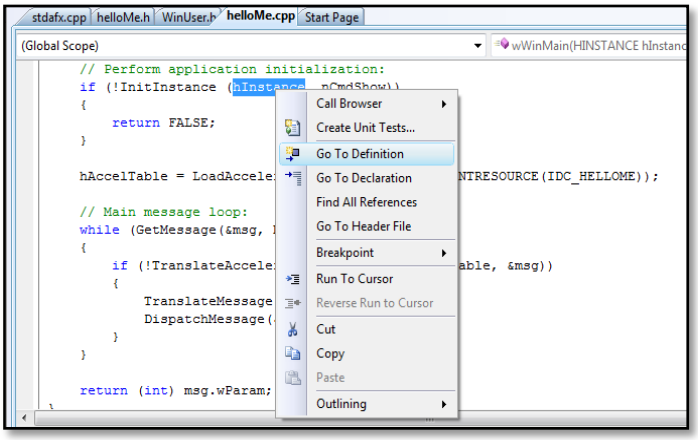

Figure 1. Selecting text to find its definition

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

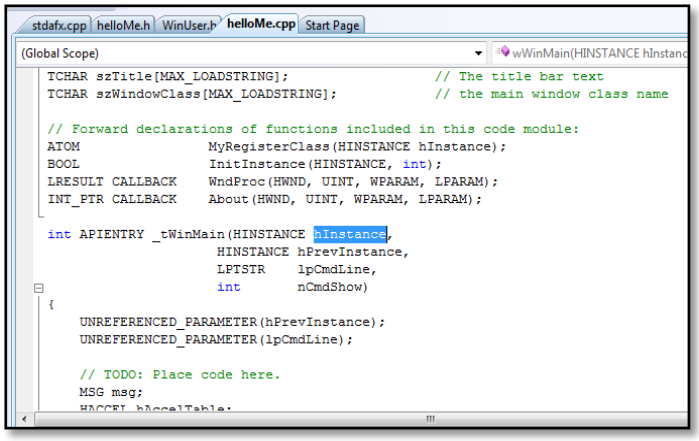

Figure 2. Displaying the file containing the definition.

In Visual C++ if we want to see the definition of a variable the user can select the text, right click, and click on go to definition. This will open the file containing the definition of that variable.There is no functionality similar to this in GEDIT 3 which is why the development is important.

III. PROPOSED METHODOLOGY AND DISCUSSION

To implement a plug-in in GEDIT 3 two files are required: a .py file and a .plugin file.
The format of a .plugin file is as follows:

```
[Plugin]
Loader=python
Module=<name of the python file>
IAge=3
Name=<Name of the plug-in>
Description=<Description of the plug-
in>
Authors=<Name of the author(s)>
Website=<Name of the website>
```

The .py file contains the code of the plug-in implementation. These two files, the .plugin and the .py files need to be placed in the *~/.local/share/gedit/plugins* folder. The .local folder is hidden by default. The two folders, 'gedit' and 'plugins' need to be created as they are not there by default. The two files should be placed in the plugin folder.
The implementation of this plug-in is made very efficient using a tool known called *exuberant-ctags*.

A. *What is Ctags?*

Ctags is a tool that sifts through a piece of code, indexing methods, classes, variables, classes and stores the index in a 'tags' file. The 'tags' file contains a single tag per line. According to the command line arguments and the language that ctags is run against we can obtain a lot of information from these indexes. Ctags currently supports 41 programming languages, and it is relatively easy to add definitions for more. Ctags makes it much easier to navigate a larger project, particularly if the code you are working with is unfamiliar. If you are unsure of what a method does or how it is supposed to be called, you can jump straight to its definition. If you are in the downward spiral of a 500+ line Perl script and want to know where a variable was defined a while ago, maybe 3 hours, you can jump right back to it. And later, you can come back to where you were working.

Therefore this technique can be useful for developing plug-ins for Gedit 3[2].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

Following is tag file format generated by Ctags:

```
!_TAG_FILE_FORMAT   {version-number}    /optional comment/
!_TAG_FILE_SORTED   {0|1}               /0=unsorted, 1=sorted/
!_TAG_PROGRAM_AUTHOR      {author-name}/{email-address}/
!_TAG_PROGRAM_NAME  {program-name}      /optional comment/
!_TAG_PROGRAM_URL   {URL}  /optional comment/
!_TAG_PROGRAM_VERSION     {version-id} /optional comment/
{tagname}<Tab>{tagfile}<Tab>{tagaddress}[;"<Tab>{tagfield}..]
...
...
...
...
...
{tagname}<Tab>{tagfile}<Tab>{tagaddress}[;"<Tab>{tagfield}..]
```

B. *Installing Exuberant-Ctags:*

On Ubuntu, Ctags can be installed by writing the following command on the terminal:

```
sudo apt-get install ctags
```

On Fedora :

```
sudo yum install ctags
```

C. *GTK+ 3*

GTK+ is is a cross-platform widget toolkit used for creating GUI.

D. *Flowchart:*

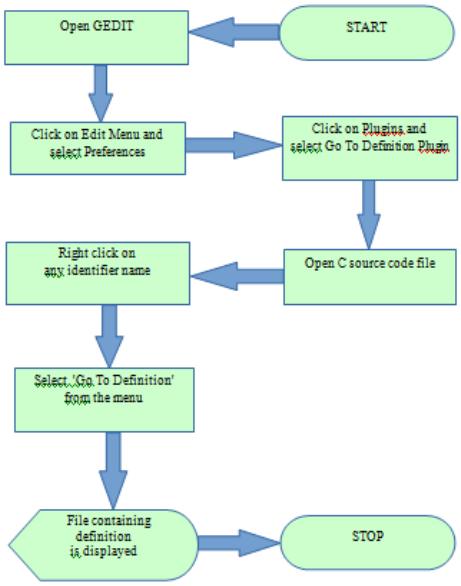

Figure 3. Flowchart depicting user interaction with the plug-in.

The above diagram shows the flowchart of how the user will interact with the system to enable the plug-in and use it. Let's say the user is going to try the plug-in for the very first time. He will have to enable the plug-in first of all. After the plug-in is enabled, he will have to select a project root folder and proceed with his coding. Whenever he has to view

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

the definition for anything in the code, he can simply right click on that part of the code and select Go-To Definition from the pop-up menu and the file in which the reference is present will open up in a new tab.

E. *Working of the plug-in*

The Go-To Definition plug-in will provide a project like environment to the user.

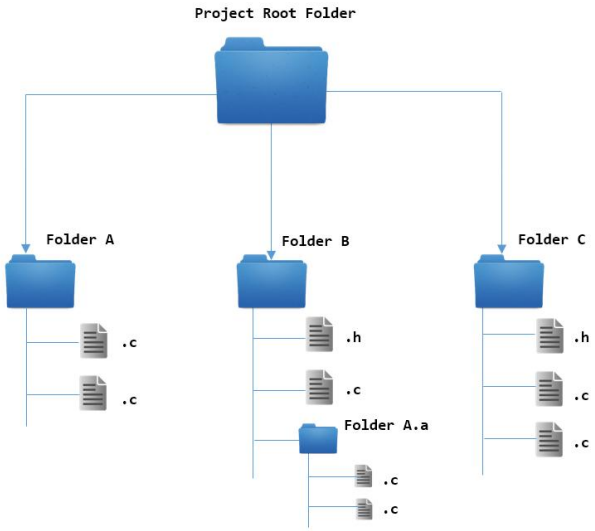

Figure. 4 A sample project hierarchy

When the user activates the plug-in, he will have to select the project root folder and once he selects the project root folder, a tag file will be created using Ctags in the project root folder for all the source files present in the hierarchy by executing the following command from the project root folder:

```
ctags -R
```

For every other tab the user opens, the project root folder will remain the same unless changed by the user. If a file does not belong to that hierarchy, the plug-in will not work for that file. Hence, the user will be able to view definitions for the project he selects. The user can then refresh the tags after making changes or adding new files to the hierarchy.

Figure 5. Selecting the project root folder

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

Once the initial setup is done, the user can just select the identifier name he wants to view the definition of and right click on it and select "Go-to definition" from the pop-up menu. The python code will then extract the text under the pointer and check for references in the tag file present in the project root folder. When a valid match is found, the file name in which the definition is present is extracted from it and new tab will be created which will open the file in which the definition is present.

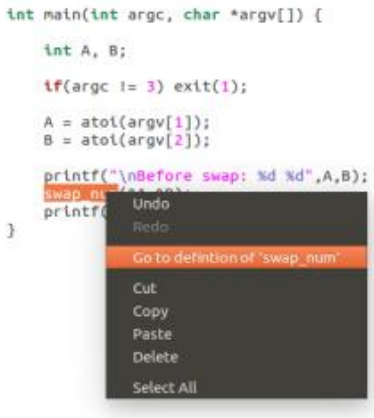

Figure 6. Using the Go-To Definition option

The plug-in will not show "Go-to definition" in the pop-up menu for reserved keywords and other irrelevant text in the code thus creating less confusion for the user.

GTK+ 3 will be used for all GUI related implementations. The folder selection will be made user-friendly by showing a folder selecting dialog box. For the convenience of the user, we will add keyboard shortcuts for selecting project root folder, refreshing tags and viewing definitions.

IV. CONCLUSION

GEDIT, the default text editor of the GNOME environment is considered as a simple text editor that can perform basic editing. But there is more to GEDIT than this.

Features like syntax highlighting, auto indentation and bracket matching make it a very good choice for programmers. But besides these features the most important feature of GEDIT is the powerful plug-in system. GEDIT allows custom made plug-ins to be added to the existing package to enable added functionality for the programmers.

There are many plug-ins that are already available to be used in GEDIT but the go to definition plug-in is one that has been requested by many programmers as well as the GEDIT community therefore we are focusing on the development of this plug-in. It will ease the use of programmers as they will not need to memorize the file where they have defined variables.

The discussed plug-in is still in making and will soon be available for download on GitHub.

V. FUTURE WORK

In this paper we have discussed the development of the go to definition plug-in for GEDIT for C. After the development of this plug-in, it can be modified to support more languages.

Another concept that can be looked into is the availability of the plug-in on other platforms such as Windows, Macintosh OS, etc.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

There are no boundaries to the future scope of plug-in development for GEDIT as it is becoming a very popular editor for programmers. It should contain as many features as possible to make the user experience very efficient.

ACKNOWLEDGEMENT

Sincere gratitude toMr.Parmod Sonar for suggesting us to develop a plug-in for Gedit 3 and helping us in the overall development process and to Mrs. P.S. Banagre for guiding us on writing this paper and monitoring our progress.

We would like to extend our gratitude to the GNOME community for providing us with the necessary tutorials and documentation for developing the plug-in.

REFERENCES

1. Gedit's Wikipedia Page *http://en.wikipedia.org/wiki/Gedit*
2. Vim and Ctags*http://andrew.strwt.ca/posts/vim-ctags*
3 Shipped plug-ins *http://wiki.gnome.org./Apps/Gedit/ShippedPlugins*
4. Third party plug-ins *http://wiki.gnome.org./Apps/Gedit/ThirdPartyPlugins-v3.8*

BIOGRAPHY

P.S.Bangareis an assistant professor in the I.T Department of Sinhgad Academy of Engineering, Pune.

Pramod Sonaris working as a team leader in Persistent Systems, Pune, India.

Pallab Pain is a final year student pursuing his B.E in Information Technology from Sinhgad Academy of Engineering.

SadhnaAdvani is a final year student pursuing her B.E in Information Technology from Sinhgad Academy of Engineering.