

## Proposal for Google Summer of Code - 2013 to KDE community

**Name:** Punit Mehta

**Email Address:** [punit9462@gmail.com](mailto:punit9462@gmail.com)

**Freenode IRC nick:** punit9462

**IM Service and Username:** Google talk with username punit9462

**Location (City, State, Country):** Ahmedabad, Gujarat, India.

**Time zone:** Indian Standard Time (IST) [UTC + 05:30]

**Proposal Title:** Polishing file structure, Application Actions and UI features in Khipu.

### **Short Description:**

Khipu is an advanced mathematical function plotter application which uses Analitza library of the KDE Education Project. Khipu is in playground now. The aim of this proposal is to polish necessary features of Khipu. This proposal includes improvement in the basic and necessary features which are some Application Actions, Persistence file and Plot Dictionary support as well as improving UI of Khipu so that it can be releasable at the end of GSoC.

### **Motivation for the project:**

In the analysis of Scientific and Mathematical theory, we often have to deal with the functions and their graphs. So, an interactive and user-friendly environment to plot the functions is needed. Sometimes, while dealing with the function behaviour, visualizing the plots in different interval (e.g. to get the knowledge of the maxima and minima) become more complicated and difficult especially when it comes to the 3D space. So, an application is required to make the user's task simpler. Khipu is such kind of application. It is an advanced mathematical function plotter. This application comes to the existence because of the efforts of Percy Camilo Triveño Aucahuasi and Aleix Pol in Google Summer of Code -2012 as a Replacement for KmPlot. Currently, the application has many features but it needs some polishing before its stable release. Working with most of the applications (e.g. MatLab, LabView) which deal with function plotting, is kind of complicated and not much interactive. As well as, in most cases, knowledge of some kind of programming is needed. But while working with Khipu, user does not need to know anything about programming. . So, the purpose of the application is very clear for the education and getting fun while learning mathematics. Therefore, it is important that the application gets alive and ready to be used by the millions of people around the world.

### **Project Goals:**

The main goals of the proposal are following:

- 1) Support of Khipu-specific file format (.khipu) to save the user's work.
- 2) Polishing Application's basic Actions (Type KActions) in Khipu.
- 3) Support of Plot Dictionary in the UI of Khipu
- 4) Making some existing GUI related features work
- 5) Creating appropriate test files and documentation for some features which include:
  - Tests for Application Actions.
  - Tests to check some crashes in Khipu. The idea here is to make Khipu less prone to be crashed.
  - Saving a plot space and reloading it.

### **Implementation Details:**

- APIs and Libraries to be used: -> Qt (C++) especially Model/View framework, some classes of KDE's API including KAction, Analitza library for Khipu backend.

### 1) Improving Khipu-Specific files :

Most of the times, the teachers and students need to save their work in the files permanently. This task will be accomplished with the idea of Khipu-specific file (Persistence file) .The important part of this feature is to save the contents of the variable and objects in runtime into a text file. The text file will have strict rules and format for the contents to be written (for example: - plotname := plotexpression := plotcolor := plotspacetitle := plotspacediscription) .These text files will be stored as (.khipu) files in the local disk.

When the user wants, he/she can reload contents of the variables and objects by opening up that file. This task will require a kind of parsing of the data stored in the .khipu file. As the format of the file is very strict, we can easily parse the contents using the delimiters (special characters which do not appear in Analitza Equation. In above example “:=”). Some portion of the code is currently available to save the plot space in Khipu’s current repository [1]. This can be reused according to the file format of the Khipu-specific file. Typically, this file will have the contents as listed here: Plots (its math expressions and name), Plot color, Plot space name and description, A thumbnail of a set of plots, X Interval, Y Interval , Z Interval.

### 2) Improving the state of Application Actions :

Currently, most of the actions in Khipu are not working in Menubar (New/Open/Save/Save As...). I will add a slot to each needed actions. After implementing the Khipu-specific file, it is easier to add slots for New file, Open file, Save file and Save As file. These features will require the knowledge of the KStandardActions of the KDE API and also some knowledge of Qt (C++) GUI. Another action that will be added is ‘Save Plot as PNG Image’. It will save the plot file as a PNG image as mentioned above. So now , the user can work with the other applications such as Gwenview , digiKam etc., using the same Plot Image. This will increase the compatibility between different KDE applications.

### 3) Support of Plot Dictionary in the UI of Khipu:

The plotdictionary file will contain plot name and expression. The key idea here is to user the function through its name rather than its equation. So, this files’ data (i.e. Plot name and equation) comes into the picture while typing the plot name in the line edit widget of Khipu .When the user is entering the equation, it should give a list of suggesting plots. This list of suggestion will be determined by the string entered in the line edit and the first matching string to it in the Plot dictionary file. This can be implemented using the Qt model/view framework in which we set the model to the data of the plotdictionary file and the view will be prompted as ListView with two columns (name and equation). The link to sample plotdictionary files is here [2].

### 4) Making some existing GUI related features work:

Currently, the GUI of Khipu has many buttons and actions. But some of them are not working. I will add slot to each following button /checkbox.

- A) Adding slot to Equation edit button (the one which is showed by a pen symbol) and to the checkbox showAxisEnable ( the one which is showed by a symbol of eye in the dashboard of Khipu)[3]
- B) Implementation of “Take Snapshot button” for 3D plots. It is working quite well for 2D plots.
- C) Diagram options for Plot in the Khipu dashboard is working well for 2D space, but not for 3D. So , I will implement that in 3D also[4].

### 5) Adding Test files and Documentation: I will be including the unit test files and documentation for the following features:

- A) Test file for Khipu-specific format which includes :
  - Saving a .khipu file
  - Opening a .khipu file
- B) Plot dictionary unit test which includes:
  - Testing File to check about UI adaptation of Plot Dictionary file
- C) I will be attaching documentation to the code sections of various features, methods, classes whenever necessary.
- D) Other unit tests to ensure that Khipu does not crash so often.

## Tentative Timeline:

**Up to June 17:** (Preproject Research) This will include the understating of important code sections in Khipu and Analitza (how the application handles the plotdictionary) in both model and view's perspective. I will also learn how to work with other KDE classes such as KAction. Apart from this, it is important to do some necessary research on the things, I will be implemented and learning the skills required. I will also fix some other small bugs and crashes found in Khipu. If everything works well in time and I get some free time, I will be going to start coding part as well.

**June17-July29:** (Persistence File phase) Defining a file format and implementation of Khipu-specific files.

**July29-August2:** (Mid-term evaluations) Evaluation of my work done till now. Getting some bugs and fixing them.

**August3-August9:** (Persistence File phase) Fixing bugs and some other improvement needed for Khipu-specific files and also implementation of test files.

**Aug10-Aug16:** (KAction Phase) Working for KActions in Khipu

**Aug17-Aug23:** (KAction Phase) Fixing some bugs and refactoring the code for necessary improvement needed.

**Aug24-Aug30:** (Plotdictionary Phase) Polishing the Plot dictionary feature support for Khipu

**Aug31-Sept6:** (Fixing the Plot Options) Implementation of Plot options (like Equation Edit and showAxisEnabled and so on) as mentioned above.

**Sept6-Sept13:** (Review Process) Reviewing the whole code, fixing some other code sections in Khipu in order to have a perfect release

**Sept14-Sept22:** (Pencil down) Including some unit test files for necessary features such as features included in Diagram tab for 3D , Testing for the correctness of KActions. Adding some required documentation for understanding of some classes, methods and features.

**Sept23- Sept30:** (firm pencil down) Fixing the code to improve its quality. Discussion with mentor about the Final Evaluation and submission of code.

**Sept30 – Later on:** (After GSoC ends) Now, we can have a perfect Khipu release. Most of the bugs would be fixed and features would have been polished. We will think over the state of the Khipu at that time and decide what to do for Khipu in the future in order to make it much interactive as well as user-friendly. We will also try to make it much better by implementing new features [5][6] as I discussed with the Khipu project manager - Percy Camilo Triveño Aucahuasi.

**Q: Do you have other obligations in the GSoC period (school, work, vacation, etc.)? Please note that we expect the Summer of Code to be a full-time, 40-hr a week occupation. It is important to be clear and upfront about other commitments that you may have during that time.**

**A:** I have vacation in my college from the late April to Early August. Then, I need to go to college to attend classes and labs. But I am completely free on weekends and partially free on two days in the week. I have no problem in working on weekends also. The available time for me to work on is as following:

- In vacation (July to Approximately Early August):

I can work at least 7 to 8 hrs. on an average in this time including the weekends. So, time per week = 7 hrs.\*7=49 hrs.

- After vacation (After Early August):

I can work at least 4 to 5 hours on an average in this time in weekdays and 10 hours in weekends. So, time per week = (4hours\*5) + (10 hours\*2) = 40 hours.

## About me (let us know who you are!):

I am Punit Mehta, a second year undergraduate student studying Information and communication Technology (ICT) in the college Dhirubhai Ambani Institute of Information and communication Technology (DA-IICT) , Gandhinagar, India . I live in Ahmedabad, India. From my school education, I am very fond of mathematics, computers and technology. I am a true lover of Programming. I have learnt Basic C programming from 9th standard (age: 14 years). I know about Open source technology from 12th standard (age: 17 years).I was introduced to KDE in the first year of graduation in my college (age: 18 years) .From that onwards, I have a thrilling potential and a desire to work on KDE projects. I have a Linux experience (Kubuntu 12.10) of 2.5 years. I have also some of my own repository and solved some Codechef problems available on Github [7]. Since this is a math project, I would like to introduce some of my past Math Achievements:

- I was selected for the Indian National Mathematics Olympiad -2011 based on my performance in the Regional Mathematics Olympiad (RMO) -2010.
- Since my school education, I am proficient in understanding math concepts and its application in scientific theories. For my detailed resume, please check out this page [8].

Apart from the basic programming knowledge, I am quite familiar with the low level details and system level implementation of Programs. I have a clear understanding about how the functions implemented on to the stack, memory allocated for different variables and their access is gained and how the way we can optimize the system performance of the programs.

I would assure to continue on improving this project even after GSoC. I am also going to maintain my code and do bug-fixing in the features whenever necessary or the bug is reported. Even after GSoC is over, I have also planned to implement some extra features from the page [6].

### **Why am I selected for this Project?**

- I have reported a major bug in Khipu [9] and also solved it. I have also solved some other bugs in Khipu and KAlgebra [10].
- I have worked for a Demo of plotting using Analitza library, which is going to be added in the master branch of analitzaplot, for a month with the potential developer and Khipu project manager Percy Camilo Triveño Aucahuasi [11].
- I have also submitted a mock-up for possible improvement in the Plots options available in Khipu in the mailing list [12]. So that we can get all the possible feedbacks of the community members and improve the capabilities of Khipu.
- From past 3 months, I am in a constant communication with the potential developers of the KDE Education Project, Percy Camilo Triveño Aucahuasi and Aleix Pol through the mailing lists, emails and Google talk. So, I have a clear understanding of the state of the project, in which way it can be improved, how it can be made user friendly to the target audience which are teachers and the students and which are the necessary features required for its perfect release.
- I have already learnt the necessary technology required to code the features mentioned in the Project which are:
  - Languages/APIs/Library: Qt (C++) Model/View framework, Analitza, C++.
  - Building Tools: CMake, GNU make
  - Version control System: Git and Github
- I have an extensive experience of Linux Environment and Open source technology. Apart from that, I have a clear understanding of compiling and building the application (KDE and Non-KDE projects) from its source code. I am also familiar with the type of communication within the community (IRC, Mailing list etc.)
- I have attended KDE-Meet up, India -2013 in my college DA-IICT. So, I have a clear idea about the goals of the community, its motto and working style.
- I am fully dedicated towards my project and my work, will take it very seriously.

### **References:**

- [1] : <https://projects.kde.org/projects/playground/edu/khipu/repository>
- [2]: <https://projects.kde.org/projects/kde/kdeedu/analitza/repository/show/analitzaplot/data/plots?rev=aucahuasi%2Fanalitzaplot>
- [3] : <https://www.dropbox.com/s/jpfc6r623alaed7/ui1.png>
- [4] : <https://www.dropbox.com/s/ti1uskm9j2hy9lz/ui2.png>
- [5] : [http://old.nabble.com/attachment/35228180/0/new\\_mockup.png](http://old.nabble.com/attachment/35228180/0/new_mockup.png)
- [6] : <http://community.kde.org/KDEEdu/Khipu/ToDo>
- [7] : <https://github.com/punit9462?tab=repositories>
- [8] : [https://www.dropbox.com/s/epyh5x4fxhw98aw/Resume\\_GSoC.pdf](https://www.dropbox.com/s/epyh5x4fxhw98aw/Resume_GSoC.pdf)
- [9] : [https://bugs.kde.org/show\\_bug.cgi?id=312215](https://bugs.kde.org/show_bug.cgi?id=312215)
- [10] : <https://github.com/punit9462/Diffs-KDE-EDU->
- [11] : <https://github.com/punit9462/MathPlot>
- [12] : <http://old.nabble.com/Regarding-GSoC-participation-and-Mock-up-for-Khipu-td35228180.html>