

Python Based VISA communication Tool

Ian Absher
Seth Ward

Lucien Armand Tamno;
Wenbo Hou; Chenliang Wang

October 14, 2016

Abstract

↓
With the desire to equip himself with an effective and efficient tool to work with, Tektronix has decided to revamp his software by enhancing some of the functionalities to better address problems related to his software workplace environment. To make this happens, Tektronix has decided to implement a new VISA talker/ listener tool, in which the acronym VISA stands for Virtual Instrument Software Architecture is a standard communication API (Application Programming Interface) for applications with test and measurement devices in Test and Measurement industry. However, the current Tektronix VISA communication applications merely provide inefficient interfaces with limited functionalities. Therefore, employees from Tektronix and our developing team have decided to created a Python-based VISA communication tool that will foster the Tektronix single integration software environment in 9 months. We also implement the principles of communications layers and ISO9241 to successfully complete our design. The new tool will have the capability of providing a graphical User Interface and helping engineers can easily carry out their tasks.

? not a person, so why use it? ?

1 Problem Statement

1.1 Problem Definition

The Python Based VISA Communication Tool project is being carried out for the current software applications at Tektronix are assessed to be clunky or clumsy, actually providing limited functionalities thus need some enhancements. the same applications also don't yield good performances when it comes to the time responsiveness.

In fact, the current software gives the user little of intuitive functionalities and features are so time consuming that the client can't really enjoy the work his or her is doing and these are some of the reasons why Tektronix wants to revamp its software environment by integrating a new efficient tool called VisaCommunicationTool which role is to get rid off the clumsiness and sluggishness by:

- providing Effective and efficient way of conveying with others instruments of its network via a user based interface called programmatic interface.
- Facilitating new users to access the same devices by providing intuitive programmatic interface.

That said, the project to be carried out is to, after clear analysis of the current awkward application and isolation of problematic parts, to device a subsequent delineated system termed VisacommunicationTool that will ultimately be the piece of the software requested and expected by the Tektronix company, to perform in its modules the following tasks:

- Automatic and intuitive syntax completion commands issued by a user.
- Provide a functionality of syntax errors correction by first, highlighting flaws when occurred and instantly drawing users attention by suggesting the right syntax to pick up.

In other hands, while running, the new tool should render less time of delay by providing quick response and in parallel effectively interfacing with its environment. The piece of embedded software should be able to efficiently run on any platform of user choice and has the capability to be: load and run without any prior installation. Meanwhile, the VisacommunicationTool has to be a resizable tool with sizes screen to fit devices like computer, tablet, Smartphone in the today mobile workplace environment. To top it all, when compared with the current tool at the Tektronix disposal, the incoming software has to enhance the buffering capability of characters displayed which gives to the user the chance of typing then viewing the entire command line, but also to store a significant list of previous commands.

1.2 Proposed Solution

As we elaborated on the aforementioned problems and converse with users, it plays out that the entire process will be divided into two main parts: interfaces and functionalities. First off, we will use others

*What does the
communication
Tool
do?*

*Please have
someone
proofread
this...
too many
errors.*

software tools to rebuild the interfaces. The interfaces are the places where the user interact with the system, so It should be looks professional and use friendly. From the experience of use the software before, the interfaces of the VISA Communication Tool should be user-friendly and simple and the design made up with:

- All interfaces will be consistent.
- A function bar somewhere can be find all function easily.
- As the tool has command input, the color of command and background will easy to read (maybe a few choices).

For the function part, we will develop more functions based on python. We will solve the problem one by one.

- The tools should communicate with device based on commands, we will create a command window where user can write or query commands.
- User can check recently 25 commands. We will design a button for check the history and reload the history from the database.
- We will create another windows where users can find available device in the Internet. Also, users can find a device by tracking its IP address.
- The tool has to offer a pathway to display all the related listed commands on either a popup windows or a box and that list shrinking down as long as the command typed is accurate. In order to develop the VISA Communication Tool successfully, we will keep in touch with Tektronix, and adjust our design.

At Expo, the VISA Communication Tool will be awesome. It will not only include the professional interfaces but also have multiple functions. The tool will never hard to use. It will be simple and easy even for new user and more automatic. We will complete all the function what Tektronix need base on the priority. All in all, after 9 months the problems of VISA Communication Tool will be solved and a great tool will appear.

1.3 Performance Metrics

Primarily, we will keep in touch with our clients, and adjust our design based on their feedback.

To confirm our solution functions well, we generate a comprehensive test plan. Basically, we will implement TDD (Test Driven Development) in the whole design pattern. Our team will divide the large project into units, where a unit represents either a class or a function in python programming language. Each unit has a unique test case that leads programmers to write the correct code. Then, we will prepare random tests for each functionality in our solution. For example, we will generate a random tester that creates thousands of incomplete commands and inputs them into our autocomplete function. Consequently, we can evaluate the software performance in completing typed commands, and then fix caught bugs. Finally, we will release a test version of our software and do the hallway usability testing

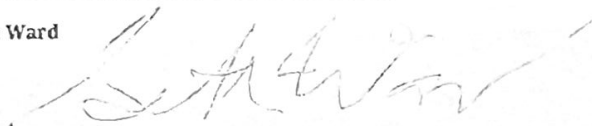
with Tektronix engineers. We will randomly pick individuals who never touch VISA communication tool before to use our software. Then, we collect test data and users advice to improve the test version. We will hold hallway usability tests for at least five times to make use our software have all necessary functionalities and be user-friendly enough. In the end, we will ask our mentor to do the ultimate test to check whether the software meets all requirements.

If possible, we would like to establish a database to collect useful messages to keep optimizing the software.

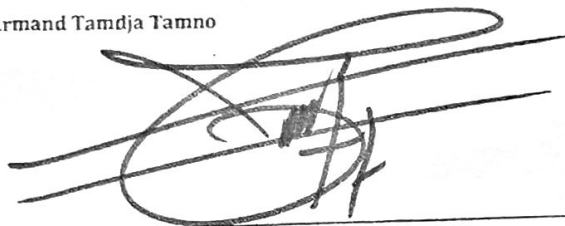
Ian Absher

A handwritten signature in cursive script, appearing to read 'Ian Absher', written in dark ink.

Seth Ward

A handwritten signature in cursive script, appearing to read 'Seth Ward', written in dark ink.

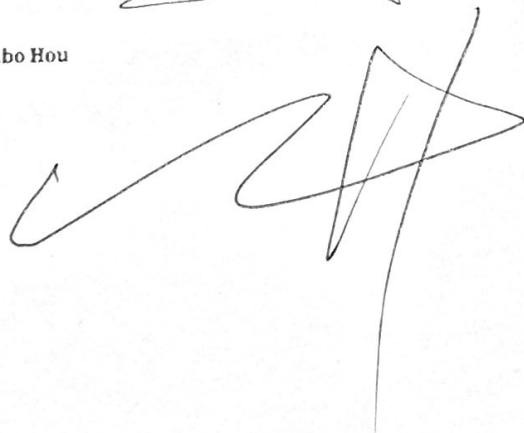
Lucien Armand Tamdja Tamno

A highly stylized and abstract handwritten signature, possibly representing 'Lucien Armand Tamdja Tamno', written in dark ink.

Chenliang Wang

A handwritten signature in cursive script, appearing to read 'Chenliang Wang', written in dark ink.

Wenbo Hou

A highly stylized and abstract handwritten signature, possibly representing 'Wenbo Hou', written in dark ink.