Sandhi
Open Source Visual Programming Software

Ambikeshwar Srivastava
FOSSEE, IIT Bombay
Manoj Gudi
CTO, Focus Analytics

December 16, 2015

Introduction

- Sandhi is a visual programming editor based on GNU Radio
- Basic data structure in sandhi is the flowgraph
- It has been named Sandhi as it means connecting and conveys our idea of connecting various blocks to come up with a robust visual program
- Sandhi is aimed to become a visual programming tool for replacing LabVIEW

Flowgraph

- Flowgraph represents the connections of the blocks through which a continuous stream of samples flows
- The concept of a flowgraph is an acyclic directional graph:
  - with one or more source blocks (to insert samples into the flowgraph)
  - one or more sink blocks (to terminate or export samples from the flowgraph) and
  - any functional blocks in between.

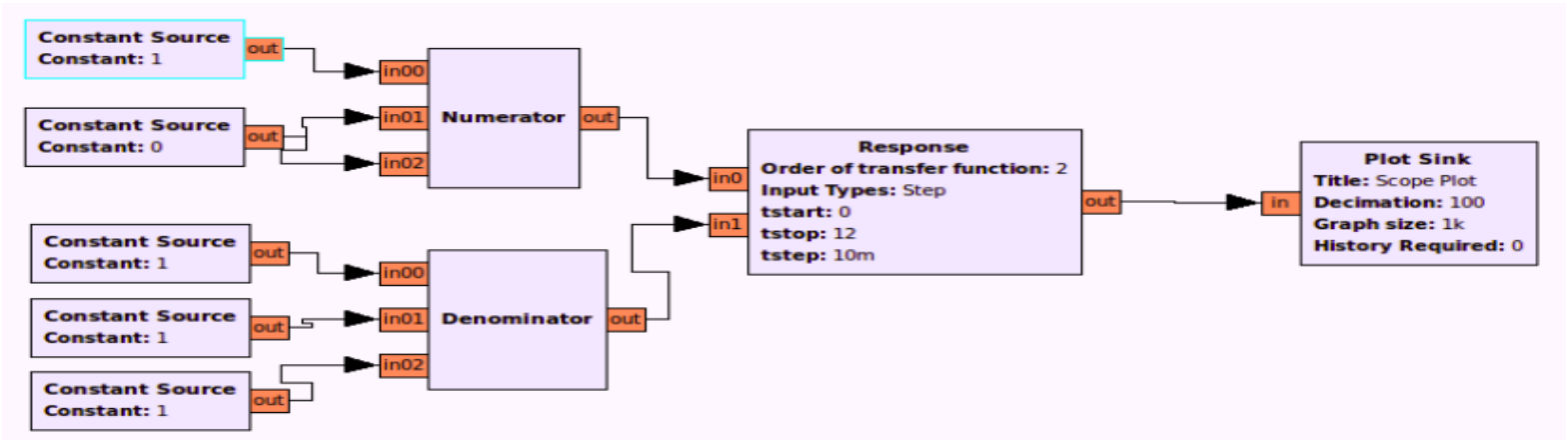


Figure 1 : Flowgraph

Motivation to develop Sandhi

- Lack of proper open source alternative to LabVIEW.
- Expensive proprietary software.
- Being FOSS, it gives you freedom to modify, share and sell your application without any permission.

Development of Sandhi

- GNU Radio
- sciscipy
- GRAS

GNU Radio

- GNU Radio is a free and open-source software development toolkit that provides signal processing blocks to implement software radios.
- Supposed to be used by the Electrical Engineering community for the purpose of digital signal processing
- It has a rich module of implemented device drivers and thereby supports a range of devices

Why GNU Radio?

- GNURadio is a very promising visual programming tool as:
  - it make very easy for the developer to abstract his code
  - provides a very easy to use framework to the developer
  - it is open source

sciscipy

- Sciscipy is an Application Programming Interface
- Aimed for Inter Process Communication with scilab when in workspace of Python programming language

Sample Code:

```
from scilab import Scilab
sci = Scilab()
x = sci.rand(20, 20)
y = x*x.transpose()
y_inv = sci.inv(y)
```

GRAS

- GRAS stands for GNU Radio Advanced Scheduler
- It was impossible to implement the feedback with GNU Radio, which uses stock application schedular

Note: *Application Scheduler is responsible for threading, controlling the data flow and managing the use of the computer resources like processor time to various processes.*

Blocks in sandhi

- Blocks are the basic building component of flowgraph
- Blocks have the property written in C++ or Python

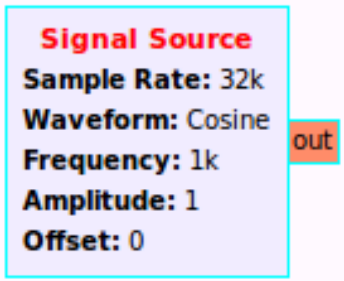


Figure 2 : Source

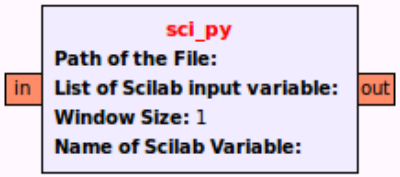


Figure 3 : Process

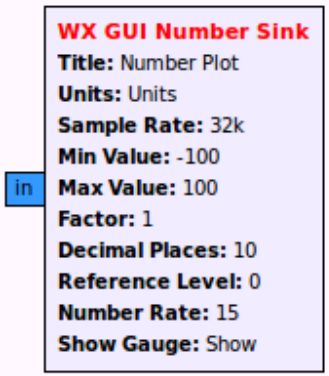


Figure 4 : Sink

Sandhi GUI

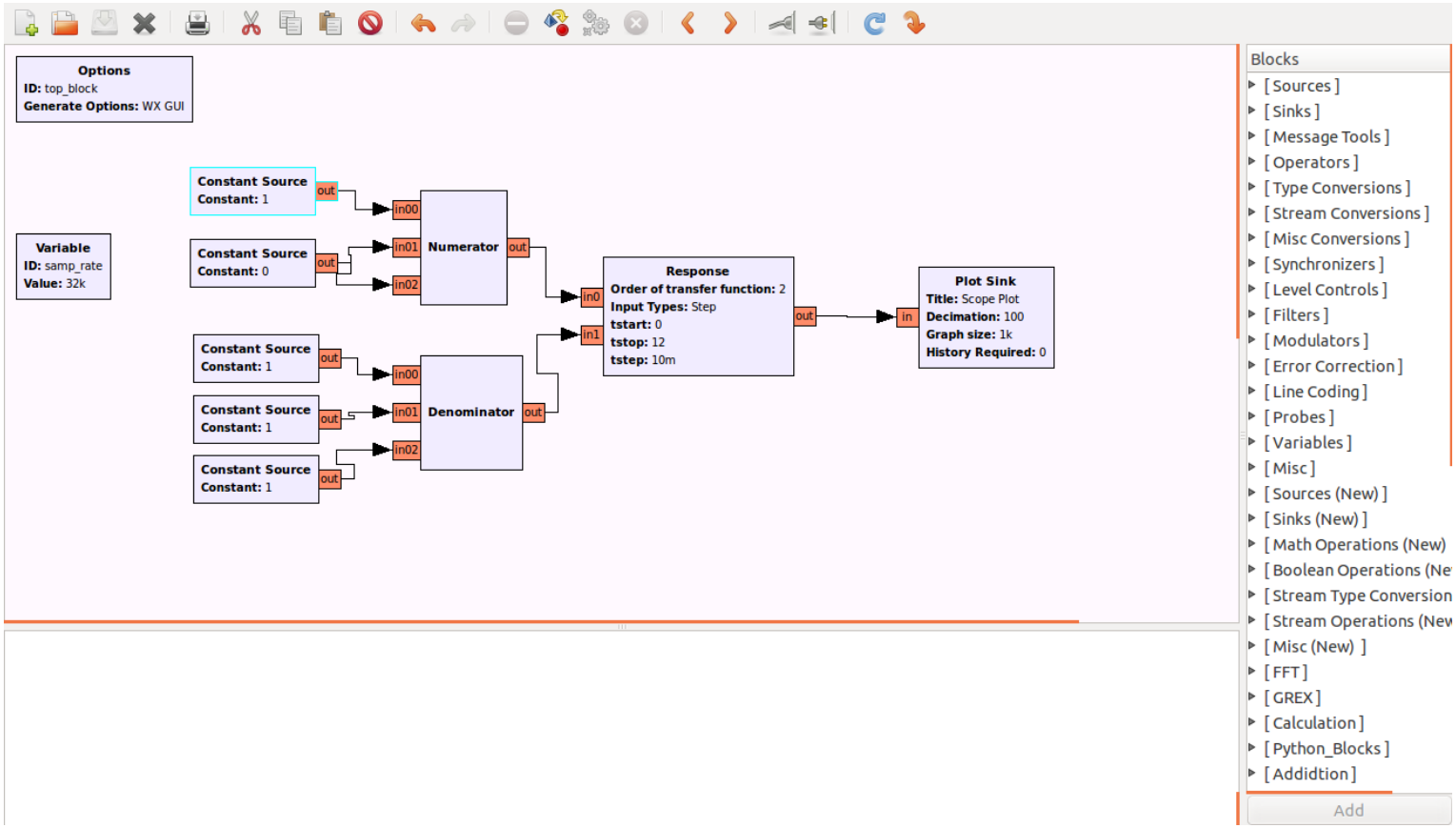
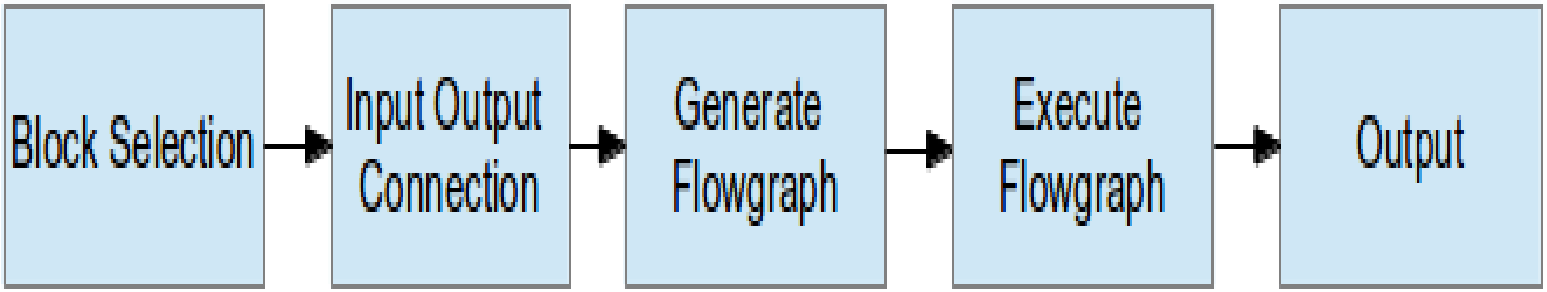


Figure 5 : Sandhi GUI

How to create a block

- One can create a customized block with knowledge of C++ or Python
- Block developer have access to any library available in Python
- There are two files needed to create a block in sandhi:
  - Functionality written in C++ or Python
  - Properties written in xml file

Work Flow



- **Block:** A functional processing unit with inputs and outputs.
- **port:** A single input or output of a block.
- **Source:** A producer of data.
- **Sink:** A consumer of data.

Features

- Applications based on flowgraph can be created in sandhi by connecting blocks as per requirement
- In sandhi user can create their own customized blocks using GNU Radio API
- It is capable of passing any practical types of data between blocks

Features

- User can use scilab script in flowgraph for computation using sciscipy wrapper
- Flowgraph with feedback can be create using GRAS
- Sandhi provides nice GUI to plot or show data.
- User can also change value in real time using slider.

Experiments on sandhi: Data Aquisition

- Single Board Heater System(SBHS) can controlled using sandhi
- Using Python serial library, one can set the fan,heat value to SBHS and receive temperature value from SBHS

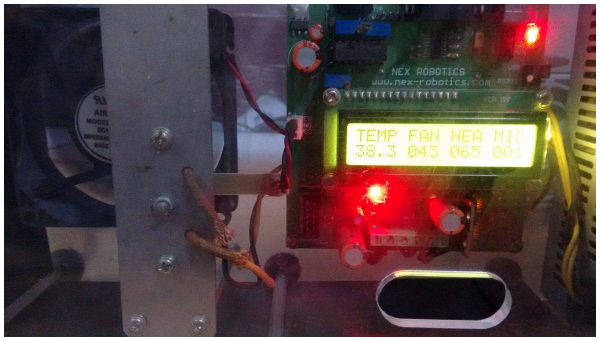


Figure 6 : SBHS setup

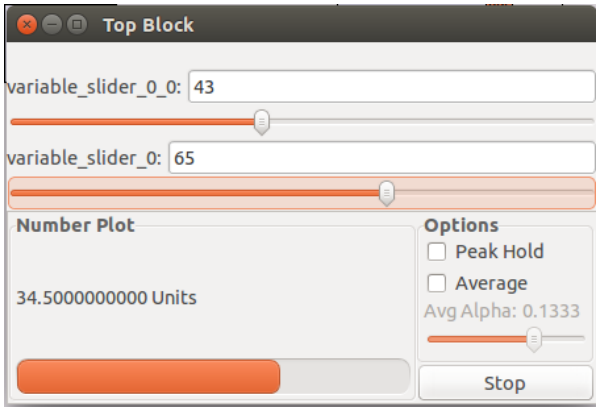


Figure 7 : Output Window with slider

Experiments on sandhi: step response of transfer function

- To perform step response the flowgraph is created as follows
- Flowgraph uses *Numerator*, *Denomenator*, *Response* and *plot-sink* block
- These blocks has been written in Python and response of system is calculated in scilab using sciscipy in Response block

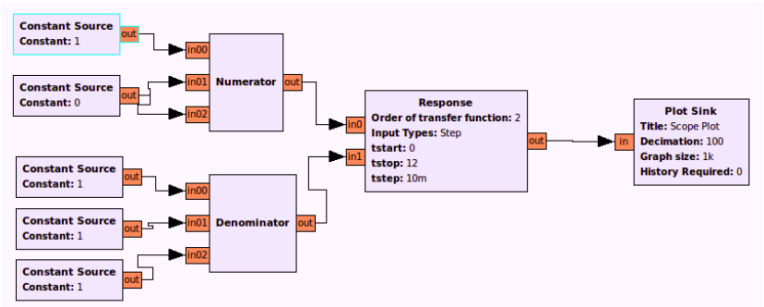


Figure 8 : Flowgraph

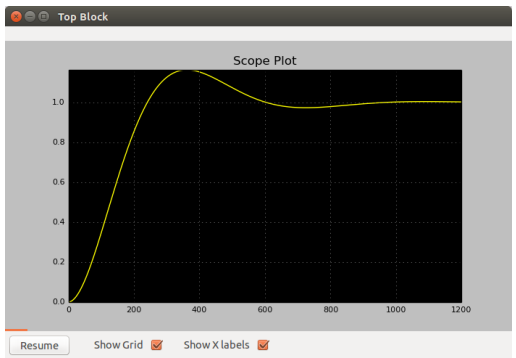


Figure 9 : Output Plot

Architecture

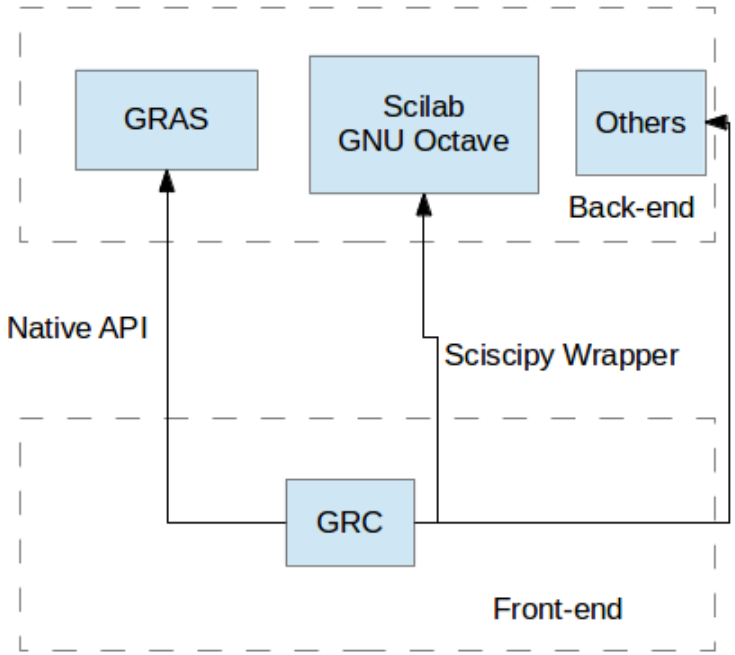
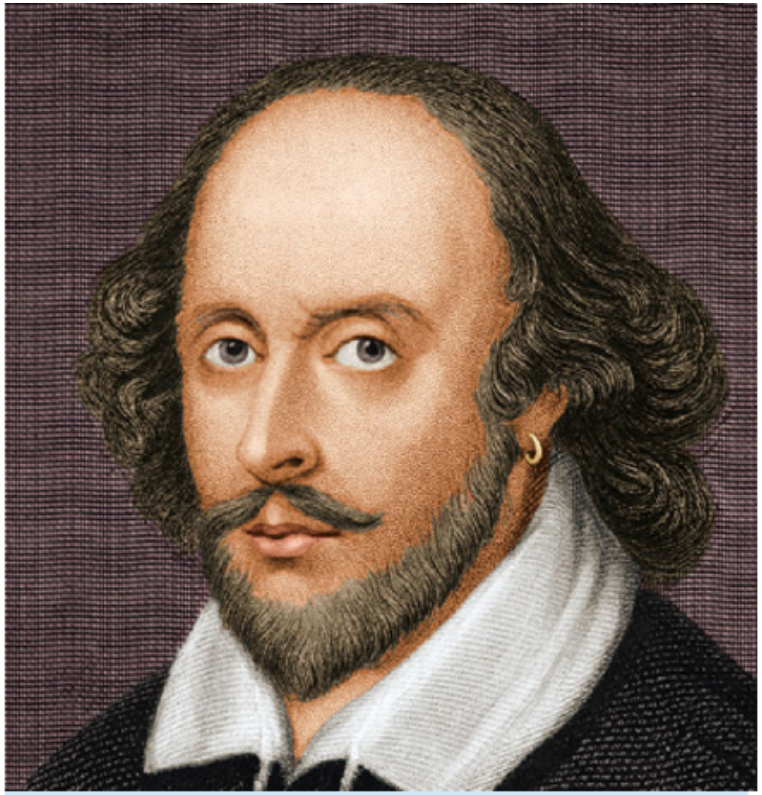


Figure 10 : Sandhi Architecture

Closed Loop System



"Cometh peace and reliability from within, and from closed loop system"
- **W. Shakespeare** 1573AD

Figure 11 : Ancient Wisdom

GRAS: GNU Radio Advanced Scheduler

- Written by Josh Blum (josh@joshknows.com)
- It is application scheduler
- Handles how the blocks(computational entity) should be formed, scheduled
- Provides easy API to write our own blocks in Python
- Uses Theron, PMC, Apology etc. libraries

Being Pythonic

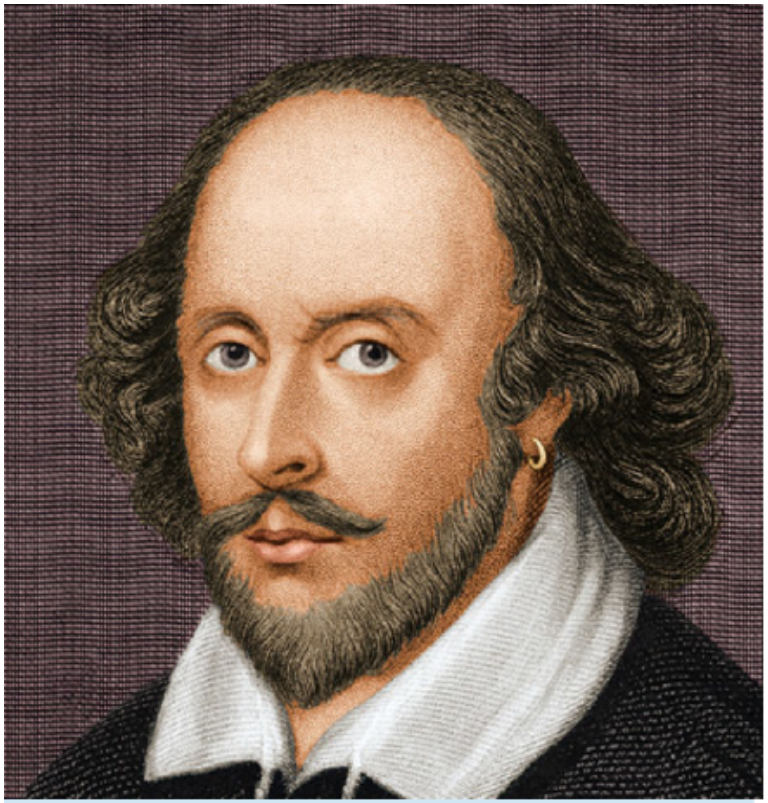


Figure 12 : Really ancient wisdom

Pythonic Sandhi

```
def work(self, input_items, output_items):

    # Limit output_items to just the size of window
    output_items[0][:] = output_items[0][:self.window_size]

    # Check number of input_instances
    n_input_items = len(input_items)

    # Men Tears were shed here..
    eval_function = getattr(self.scilab_instance, self.func_name)

    for i in range(n_input_items):
        output_items[0][i] = eval_function( input_items[i] )
```

Figure 13 : Work Function Code snippet

Ongoing work

- Shifting to cloud based architecture(using RPC)
- A client side containing smaller GRC, runtime environment
- Server side containing heaviy libraries
- Prototyped using RabbitMQ's RPC service(AMQP)
- Should work well for heavy simulations (if not realtime app)
- Functional block generator

References/Links

- Source code
  `https://github.com/gnu-sandhi/sandhi.git`
- Documentation
  `https://github.com/gnu-sandhi/docs.git`
- Problems using it? Found a bug?
  - Are you sure its a bug and **not** *a feature*?
  - Raise an issue on our github
  - Mailing List: *gnu_lc@googlegroups.com*

Contact Us

- If you are interested to contribute please write to us at contact-sandhi@fossee.in
- Go through our website sandhi.fossee.in
- You can post your queries on Forums at forums.fossee.in

THANK YOU