Software Design Concept

Audible-Plot

Daniel Birket

March 28, 2023

Abstract

This document outlines the concept for the software design.

The software design concept is to leverage existing well-documented and currently maintained open-source tools to produce a hybrid audible and visual rendering of a data series and associated descriptive text and meta-data, as well as a text table of the data.

The ‘python‘ language will be used to "glue" the tools together into an application to produce the outputs. The output text, graphics and audio will similarly be glued together with markdown text. The final output will be a HTML page or PDF page with one or more audible and visual plots and their meta-data and descriptive text.

The visual plot will be created by writing a ‘gnuplot‘ script and running it through ‘gnuplot‘ to produce a graphic file, which will be embedded in the ‘markdown‘ file, along with a link to the audible plot. (The ‘gnuplot‘ script may be a bash script to combine the gnuplot commands, data, and gnuplot command line options into one portable and readable file.)

The audible plot will be created by synthesizing tones of frequencies representing the data points.

The description and meta-data of the plot will also be converted to audio via a text-to-speech engine.

The final output will be created by writing a markdown text file that embeds the graphic and links to the audio and then running the markdown running it through ‘pandoc‘ to produce a HTML or PDF page (via LaTeX.)

Note that the python program, the gnuplot script, the markdown text file, and the LaTeX file are all simple text files, which are inherently blind-accessible. The final outputs of HTML or PDF files are also blind-accessible.

The Audible Plot module has three main phases of operation: describe, play, and plot.

Several audible-plot methods allow the programmer to describe the plot and provide the data to be plotted. These descriptive calls are outlined below.

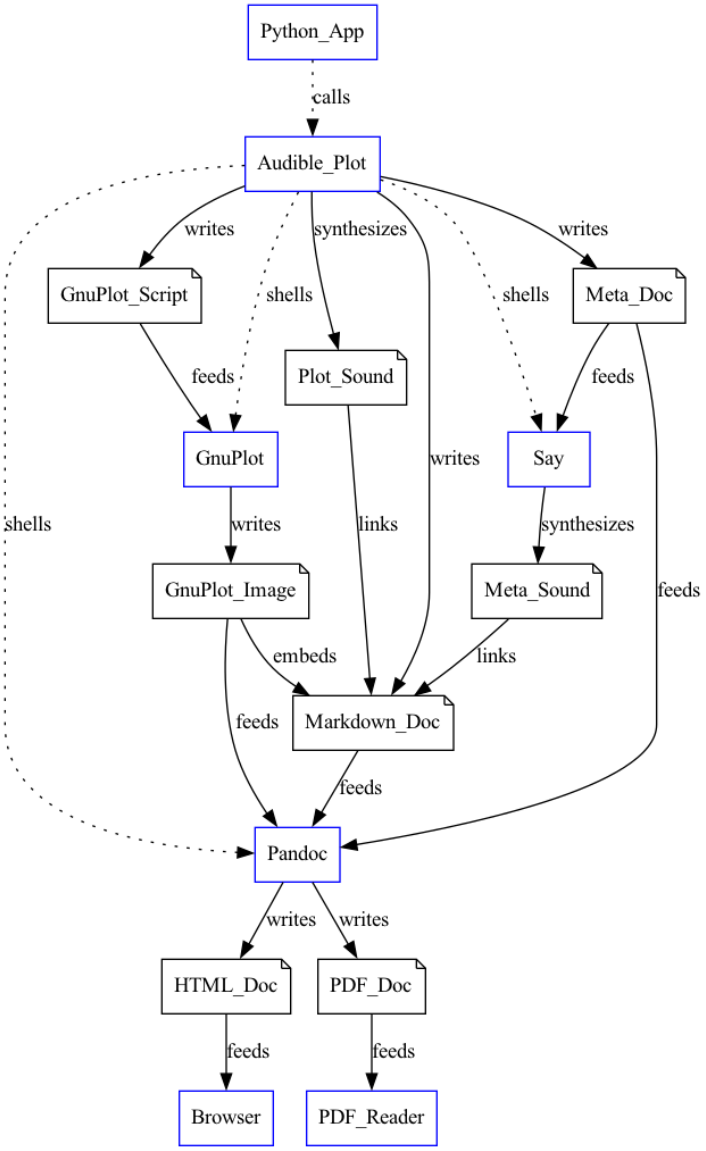
The ‘play‘ method synthesizes a collection of speech and audio tone samples to present to a listening user. The information presented at the moment is controlled by pressing keys. For example: ’t’ to speak the title, ’a’, ’b’, or ’c’ to select the 1st, 2nd or 3rd data series and repeat the last tone or speech output with the new selection, ’p’ to speak the current x and y position, ’x’ and ’y’ to describe the x-axis and y-axis, ’d’ to speak the plot description. back-quote, ’1’, ’2’, through ’0’ to move to the 0%, 10%, 20% through 100% positions on the x-axis. Left and Right arrow keys to

move the current x-axis position higher or lower. Down arrow to play the current position again. 'q' to quit the player. These keys are tabulated below.

The 'plot' method synthesizes complete speech description and audio tone sound files to represent the entire plot. It also creates a visual plot using 'gnuplot' and a HTML or PDF file using markdown and 'pandoc'. The resulting file includes both a visual and audible representation of the plot.

The application may call the plot command before the play command to both 'plot' the data to a document and then 'play' the data to the user.

The 'plot' method operates as described below and illustrated in the following diagram:



- The programmer/user writes a small python app to describe and output the desired plot. The app may be created for a single use or may be more general purpose and prompt the app's user for the plot data and info.
- The app calls the audible-plot module and passes the data and information describing the

desired plot, then calls the plot command. The plot command executes the following steps:

- Writes a ‘gnuplot‘ script containing the plot data, meta-data (title and axis labels). (This may be a bash script with an embedded ‘gnuplot‘ file to allow one script to include all the plot data including the command-line options. This allows for inspection, modification and reuse.)
- Shells out to ‘gnuplot‘ to create the plot image.
- Writes the plot meta-data to a file for speech synthesis.
- Shells out to ‘say‘ or ‘espeak‘ to convert the text to a speech audio file.
- Synthesizes the tones used to represent the plot. (These are the same as the ‘play‘ method, concatenated together.)
- Writes a markdown file that combines the plot meta-data, plot image, meta-data speech audio and plot audio into one document.
- Shells out to ‘pandoc‘ to create either an HTML document or a PDF[1] document, as requested.

[1]PDF documents require that LaTeX be installed.