ABSTRACT

Biomedical Engineering is one of the fastest evolving fields in engineering today. LabVIEW, a graphical programming tool from National Instruments has been used across multiple classes to teach Bioinstrumentation, circuit design, biological signal processing and image processing concepts in biomedical engineering. However, with the increase in number of functions in LabVIEW, it is difficult for a novice biomedical engineering student to find the functions that most apply to their course and hence a better organization is needed to help teach and understand concepts. In this paper, we will explore a new Startup kit that has been developed to address this concern. We will explore the current environment and the areas that can be improved upon and present the free biomedical startup kit and discuss the pros and cons of this approach.

1. INTRODUCTION

Biomedical Engineering education has evolved significantly in the recent years to encompass advanced areas from the life sciences, as well as electrical and mechanical engineering such as advanced signal and image processing, data acquisition and instrumentation. With the inclusion of such areas in the curriculum comes the challenge of being able to teach biomedical engineering concepts with the help of intuitive, powerful tools that do not require a steep learning curve from the students. National Instruments LabVIEW[1], a graphical programming language, has been used by biomedical engineers for a variety of tasks from data acquisition to instrumentation [2,3,4,5]. Additionally, LabVIEW has evolved to offer educators powerful extensions for signal processing, image processing and control[6,7]. However, since LabVIEW is also a general purpose programming language, the functions and palettes are not necessarily organized in a “biomedical-friendly” way. As a result, students and educators have to navigate the sophisticated menus to get to the most appropriate functions for the tasks that they would like to tackle leading to confusion and a higher learning curve that diminishes the usability that a graphical programming environment provides.

In order to address this concern, we present a biomedical engineering startup kit that we developed that takes advantage of the customizability that the LabVIEW environment offers. By using this startup kit, which is available as a free download online, we add a customized “Biomedical Applications” palette to the LabVIEW environment that has been customized to address a wide variety of tasks in Biomedical engineering including, instrumentation, imaging, signal processing and data acquisition. Additionally, since this startup kit is based on LabVIEW, it can be further extended and customized and we have created supporting documents that can guide educators and students to help customized the biomedical startup kit. We have also taken advantage of some of the lesser known but powerful features in the LabVIEW environment to create pedagogy-oriented Virtual Instruments or VIs as it is commonly referred to in the
LabVIEW environment, what we call as “Shell VIs”. But first, we will begin by quickly summarizing some of the changes that LabVIEW has undergone that has made it both powerful for Biomedical engineering at the same time, complicated to navigate for biomedical engineers.

2. EVOLUTION of LabVIEW

LabVIEW has evolved over time to include features specifically for academia and educators. Specifically, LabVIEW started as a tool for instrumentation and data acquisition. In the recent times, we have seen various extensions that have been added to LabVIEW that make it a true programming language, including features that enable educators and students to design digital filters[8], program FPGAs[9], acquire and analyze images from a variety of cameras[10] and build medical devices[11] and adding script code to existing LabVIEW VIs[12].

A direct consequence of these developments has been the sophisticated menu structure that a user ends up with in LabVIEW as shown in figure 1.

![Function Palette in LabVIEW 8.5](image)

As it can be expected, finding a VI or a function block in this menu system can present a significant challenge. For example, if a student wants to detect the peak of a waveform, the navigation route would be something as shown in figure 2.
Figure 2. Accessing a Waveform Peak Detect VI/Function in LabVIEW Today

Accessing a waveform peak detect VI requires roughly navigating through four sub-palettes and about 30 other VIs. While this may be acceptable to a familiar user of LabVIEW, it provides a hurdle in teaching concepts to new biomedical engineering students who want to just detect the peak of a waveform. In order to address this concern, we created the biomedical startup kit.

3. BIOMEDICAL STARTUP KIT

The biomedical startup kit is a wrapper that is built using the LabVIEW customizability options which when loaded, will add a section in the functions palette titled “Biomedical Applications”. In this section, based on feedback from various educators as outlined in the next section, we have attempted to organize the most commonly used functions for biomedical engineering classes organized into distinct categories that align with biomedical engineering labs and homeworks. A point to note is that this is our best attempt and may not be the perfect fit for all classes. But we hope to provide a starting point for all educators to further customize this kit for their own purpose and reuse this in their classes and labs. The Biomedical Startup Kit is shown in figure 3 as it will appear in the functions palette.
Inside the biomedical applications sub-palette, the user will find biomedical specific entries. Figure 4 shows this sub-palette in detail.

As it can be seen in figure 4, the terms chosen to denote each sub-palette have close affinity to biomedical curricula – such as Medical Imaging or Biosignals. Terms that were self explanatory were left as is in order to help usability. Examples include “Data Acquisition”, “Data Logging” and “Instrument Control”. The entry “Mathscript” enables educators to integrated existing
textual code from other text based languages into LabVIEW enabling reuse of work already done to create labs and exercises.

Figure 5 shows one level down into each of these palettes. For details on specific functions included in the palette, we refer the reader online where they can read and download this kit[13].

4. THE FEEDBACK PROCESS

Building a toolkit that would help educators in any field poses a lot of challenges. Concerns that came up when we created this toolkit included applicability to a wide variety of courses, the optimum number of functions/VIs per palette, missing functions/VIs in LabVIEW that need to be created and distribution mechanism

In order to answer some of these concerns effectively, we created a network of professors from various institutions to gather feedback from. The Universities that we enlisted for feedback were University of Texas at Austin, Texas A & M University, Rice University and Bucknell University. The universities were chosen primarily because of their proximity to the authors location which would facilitate feedback.

In order to receive effective feedback, we had several meetings and conference calls with the professors from these institutions and based on those meetings, came up with a first version of the toolkit that we sent for reviews and gave the professors two weeks to give us feedback. After two weeks, we got varying degrees of feedback. Educators that focused more on the teaching side generally liked the startup kit, and requested for some specific biomedical functions like the
ECG Express VI which enables professors to simulate ECG signals without any hardware needed as shown in figure 6.

Figure 6. “Simulate ECG Signal Express VI” created based on feedback from BME Community

5. SHELL VIs
Another consequence of creating the biomedical startup kit was the creation of Shell VIs. Shell VIs, a term given by the authors, leverage the ability of LabVIEW to drop whole blocks of code when the user drops one VI. In order to illustrate the effectiveness of a Shell VI, let us start with a simple example, teaching acquiring finite number of samples from an external sensor via the data acquisition device. There are three ways to do this in LabVIEW:

- Using Express VIs – a configurable way to acquire data
- Using standalone VIs – good to teach concepts of DAQ but lots of connections needed
- Using Shell VIs – good to teach concepts but easier to create

Figure 7 shows the Express VI approach. This is the simplest and fastest way to acquire data, but it hides a lot of details that the educator may want to teach students.
Another way to teach DAQ concepts is using lower level VIs. While this is great to teach concepts, it can be cumbersome for someone not familiar with LabVIEW to know exactly which VIs are wired and what connections need to be made. This approach is shown in figure 8.

- **Total # of VIs/Ctrls/Ind to create**: 10
- **Total # of Wires to connect**: 15
As part of this startup kit, the authors have created a much simpler way to build such step-by-step VIs that teach different concepts including timing and synchronization at the same time, eliminate the confusion of which VIs to connect. This feature in the startup kit, called “Shell VIs” encompasses a set of VIs and their connections together with comments that make it applicable for pedagogical purposes. This example is shown in figure 9.

![Shell VI example](image)

- **Total # of VIs/Controls/Ind to create**: 1
- **Total # of Wires to connect**: 0

As it can be observed in figure 9, using Shell VIs eliminate the cumbersome steps of figuring out what VIs are connected with each other and what inputs need to be specified. In addition, because these are ultimately lower level VIs, educators can still let the students customize these VIs in order to teach different tasks.

The biomedical startup kit includes Shell VIs for several tasks including:
- Data acquisition, finite
- Data acquisition, continuous
- Image acquisition, from a file
- Image acquisition, one frame
- Image acquisition, multi frame

In addition, the readers are directed to the web[14] for extended tutorials on how they can create their own Shell VIs with the biomedical startup kit.

6. CONCLUSION

LabVIEW has evolved as a tool that has been used for data acquisition and instrumentation in Biomedical Engineering to performing advanced signal and image processing and designing complex control algorithms. With these added features, comes added complexity in navigating through the various palettes to find the right function which presents a steep learning curve for
unfamiliar users of LabVIEW, like several newly enrolled biomedical engineers. In this paper, we present a biomedical startup kit for LabVIEW that we have developed with feedback from academia. While we do not foresee this biomedical startup kit to serve the need of every educator/class in biomedical engineering, we believe that this startup kit will serve as a good starting point for educators to extend the startup kit for their own needs. We have also included some elements that we believe will benefit pedagogy such as Shell VIs. Again, we hope that educators will take advantage of the initial work done and extend it to suit their own classes.

7. REFERENCES