Using LabVIEW in a Localizable Medical Device

Author(s):
Richard M. Brueggman, Founder and CEO, Data Science Automation, Inc.
Christopher S. Mullin, Ph.D., Senior Automation Systems Engineer, Data Science Automation, Inc.

NI Product(s) Used:
LabVIEW 2012
USB-6001

Category:
Advanced Manufacturing and Control
Advanced Research

The Challenge

A medical device for treating cancer controls several motors and valves, and relies on heavy interaction with the caregiver to safely treat the patient.

The Solution

The flexibility of LabVIEW's structures and patterns allowed DSA to create a module with high cohesion that makes translating captions, warnings, and user instructions easy to scale and maintain.

Introduction

Data Science Automation (DSA) is a premier National Instruments (NI) Alliance Partner that specializes in automating and educating the world leading companies. Clients choose DSA because of DSA's deep knowledge of National Instruments products, disciplined process of developing adaptive project solutions, staff of skilled Certified LabVIEW Architects and Certified Professional Instructors, and unique focus on empowerment through education and co-development.

Cancer Treatment with Immunotherapy

Immunotherapy is a rapidly developing area of cancer treatment and provides an alternative to patients that are untreatable by traditional methods. There are a variety of immunotherapy techniques but what they have in common is that each attempts to modify the human immune system in ways that allow it to combat cancer cells. While this treatment offers an alternative to individuals who can tolerate traditional treatments but whose cancers cannot be targeted, it is a much bigger game-changer for the millions of individuals who cannot tolerate chemotherapy. In fact, many of the inhabitants of sub-equatorial regions carry many more parasites in their blood than in other regions, and chemotherapy is not an option for them because it simply allows the parasites to take over.

Our client is developing a system that performs extracorporeal treatment on blood, which means that blood is removed from a patient, treated outside of the body, and returned to the patient. This particular method of immunotherapy is designed to remove certain molecules generated by cancer cells that actually suppress the body's own immune response to those cells. By releasing these molecules, cancer cells are trying to protect themselves. By removing these molecules from the bloodstream, our client intends to strip away the cancer cells' protection.

Extracorporeal blood treatment is very invasive and potentially dangerous, so it is very important that the caregiver use the machine correctly. Since the machine is being first deployed in Europe, numerous languages need to be handled easily, and many aspects of the user interface need to be translated, including captions, instructions, and warnings.

The challenge was met by using a database to store a table of translations with several types: enum, captions, Booleans, radio, and string. Identifiers for each type were kept in an enum, which indexed a table kept in a functional global to determine the text corresponding to the desired item on the user interface. Setting a different language in the functional global read a different column from the table out of the database. The database could be easily updated by using a utility VI that read a translation table kept in an Excel workbook, so that it could easily be maintained and updated by users unfamiliar with the database or LabVIEW.

It is somewhat ironic that because a single subVI was used to translate many different enums (which enumerated different warnings, for instance), the input to the VI was chosen to be a string, and the enum passed through a "Format Into String" function before being fed to the Translate Enum subVI. On the other hand, strings were used the most often (for displaying different instructions in a single indicator at different times during treatment, for example), and were often quite long, so each was given a unique label. To assure proper selection of what translation to use, the labels were listed in an enum, which could be fed directly into the Translate String subVI. Thus, the subVI to translate enums used a string input, while the subVI to translate strings used an enum input! A portion of the code used for this localization effort is shown in Figure 1.

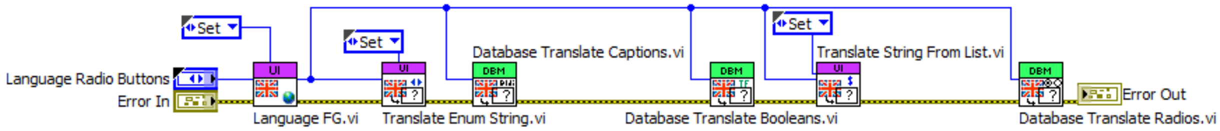
Set
Set
Set
Language Radio Buttons
Error In
UI
Language FG.vi
UI
Translate Enum String.vi
DBM
Database Translate Captions.vi
DBM
Database Translate Booleans.vi
UI
Translate String From List.vi
DBM
Database Translate Radios.vi
Error Out

Figure 1. A Portion of the Framework Used to Support the Localization Software.

Contact Information

Richard M. Brueggman, info@DSAutomation.com
Data Science Automation, 375 Valley Brook Road, McMurray, PA 15317-3370
www.DSAutomation.com