A GUI BUILDER ENVIRONMENT BASED ON LABVIEW FOR THE VIRGO PROJECT

F. Carbognani, B. Lopez, D. Sentenac

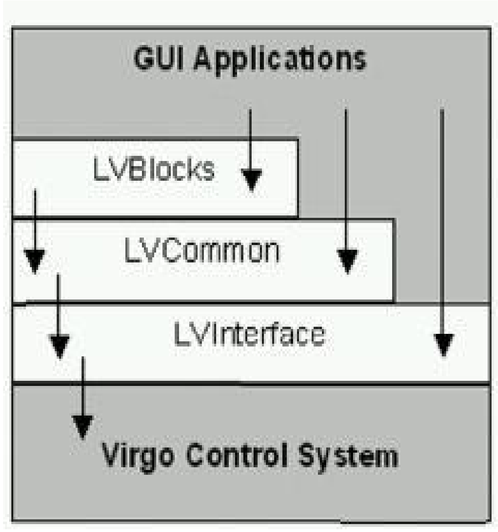
European Gravitational Observatory, Pisa, Italy

The Challenge

The VIRGO project consist of a suspended Michelson Interferometer with two 3 Km arms aimed at the detection of gravitational wave signals from cosmic sources. In order to support the VIRGO ongoing commissioning activity and facilitate the transition to full operational mode, the need for new, quickly built, flexible and graphically rich Graphical User Interfaces (GUIs) arose. The challenge was to set up a GUI building environment able to deal with those requirements and to smoothly integrate it with the existing distributed control software framework (C/C++ based) and associated software management procedures.

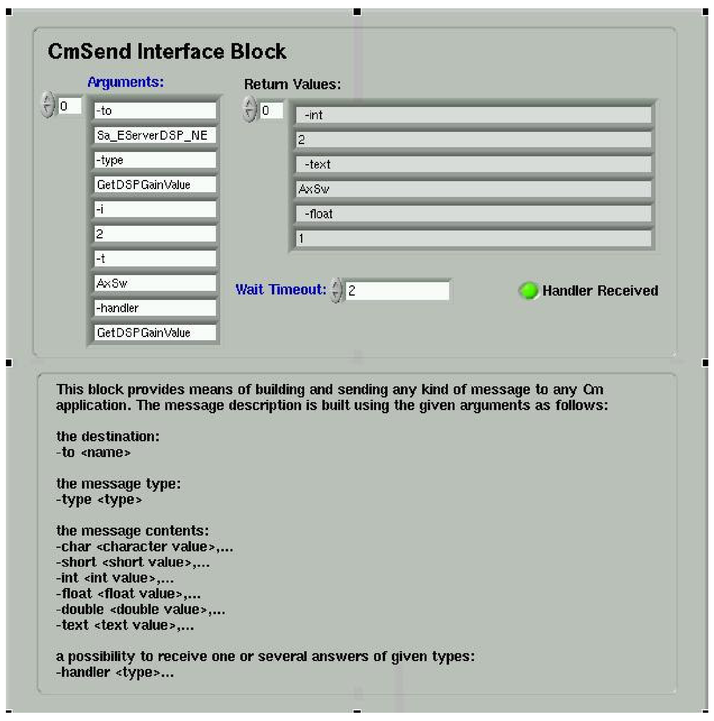
The Architecture

Three basic components have been found to be necessary in order to fulfil our requirements: a LabVIEW interface to our control framework, a common function library and a common building block library respectively called LVInterface, LVCommon and LVBlocks. Altogether, they provide the necessary structure and implementation needed for building user interfaces on top of them.



The Interface

The interface contains the implementation that allows communicating from LabVIEW applications to our control framework. In particular what was needed was a binding to the Virgo specific interprocess communication layer (Cm). Due to the specificities of the Cm communication layer it has been considered appropriate to develop the interface by using a Code Interface Node (CIN). This CIN permits the sending of control commands to the distributed control system via the Cm communication service, and handles several command parameter data types such as integers, floats and strings.

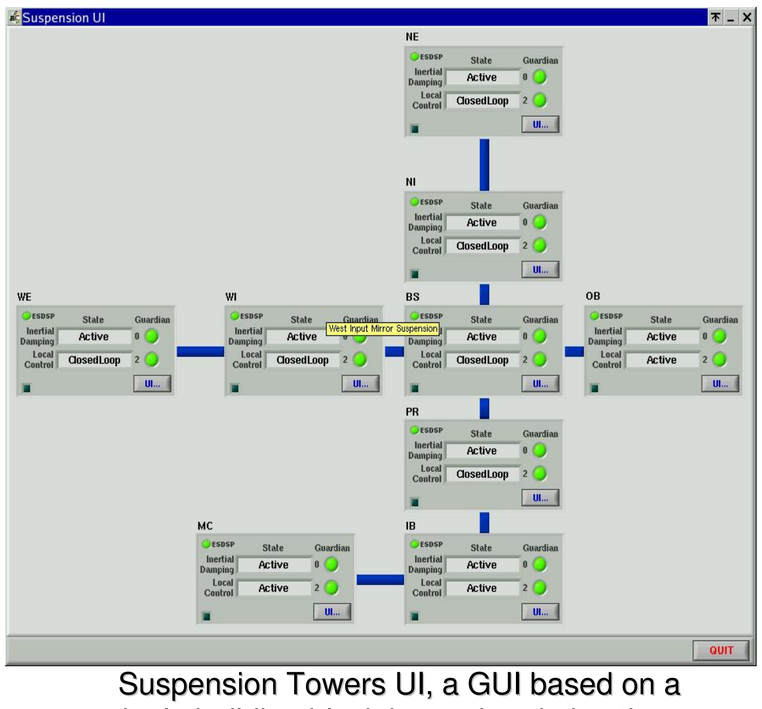


The CmSend Block able to build and send a message to a Cm application.

The Building Block Library

Building blocks: completely autonomous, configurable sub-GUIs that can be inserted into higher-level GUIs as needed.

Methodology: place sub-frame containers in the GUI wherever a desired block should be located, and then load and configure the respective sub-VI dynamically. The latter functionality is contained in a load-block-VI that receives the specific configuration as an input and hides the rest of the implementation to the user. The main effort was spent in the efficient implementation of this mechanism leading to fully modular GUI development, maximized code reuse and clear and maintainable block diagrams



Suspension Towers UI, a GUI based on a single building block instantiated nine times

Templates

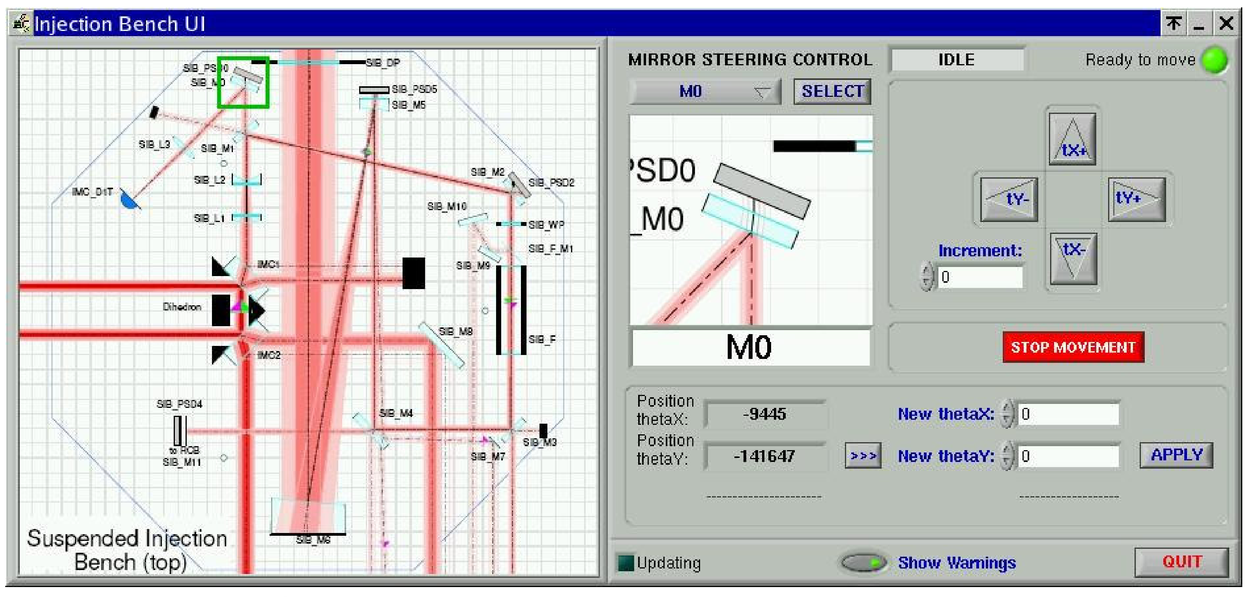
As soon as the first high-level GUIs had been built by using the new methodology we generalized their structure and then prepared templates that already contained implementations or structures for status updating, commanding, synchronization, constant definitions and sub-UI execution. This way a common look-and-feel across the different GUIs could be ensured from the beginning. As the functional development had been reduced, we had more time available for the graphical enhancement of the GUIs

Results

The presented GUI building environment based on LabVIEW required an initial effort of customisation by establishing the right methodology and implementing the basic components, but it is now allowing to build new GUIs in a very short time with a high level of flexibility and maintainability and so fulfilling the increasing amount of GUI-related user demands.

The Common Function Library

This library uses the described bindings and implements higher level common functionality like e.g. standard control process commands, utilities, etc. It is exclusively made up by LabVIEW VIs and does not contain graphical components.



Mirror Steering UI, less time needed for the functional part allowed to concentrate on the graphical enhancement