A STRUCTURED APPROACH TO CONTROL SYSTEM GUI DESIGN FOR THE SOLARIS LIGHT SOURCE

V. Juvan, I. Dolinšek, T. Humar, M.Pavleski, Cosylab, Ljubljana, Slovenia
P. P. Goryl, Solaris, Krakow, Poland

Abstract

In the framework of delivering control system services to the Solaris synchrotron light source, Krakow, Poland, Cosylab realized a comprehensive set of controls GUIs, using a structured approach. The goals of using this architecture are threefold. The first is to achieve reliable, predictable and consistent behaviour of the controls software. The second is that it is easy to deploy and maintain through scripting. The third is that it is future-proof by providing extensibility, using dedicated templates. The system is based on a configuration database, populated with devices, device specifics and device groups (clusters of devices performing specific operations). The GUIs are dynamically generated from this configuration. For the synoptic views, TANGO-standard JDraw and its configuration are integrated into the framework. Existing GUIs, written in PyTango can be easily adapted to function as part of this system. The compelling user benefits are high usability and life-time management through controlled upgrade and extension. For new big physics projects this GUI control program offers a customizable solution for any TANGO based control system.

CHALLANGES

When realizing the controls GUIs, we were presented with a series of challenges that we had to overcome.

Firstly, we had to develop a full machine control and state overview program. It was designed as a single entry point program for performing all operations, from every day operator to expert use. Moreover, a broad set of controls GUIs was required for specific operations.

Multiple instances of the control program were foreseen, each intended for a separate control system at the facility.

During the development, we expected to adjust the functionality and features to user feedback, which would be acquired in steps.

Existing tools were not sufficient since they did not provide the desired functionality. With a limited budget, we undertook a development of the Solaris Synchrotron Control Program.

GOALS

The Solaris Synchrotron Control Program had to meet a variety of goals.

We wanted to achieve a predictable and consistent behaviour of the control room software. The control program had to provide a required set of functionalities and features for operation. A transparent and convenient use of the software was requested.

A very important aspect of the control program was that it had to be configuration driven. The goal was to maintain one source of information from which the control program would gather all required information. We would evade hardcoding at all costs. Moreover, we made use of the dynamic GUI generation and generic panels to enable the control program to be adjustable by only modifying its configuration source.

A significant goal was to guarantee easy extensibility. This was achieved by providing dedicated templates and support for integrating external applications in the control program itself.

Last but not least, we wanted to provide easy deployment and maintenance through scripting.

SOFTWARE ELEMENTS

The final deliverable consists of four components, namely:

- Configuration database:
  - Device configuration
  - Device group configuration.
- Control Program.
- Custom GUI library
- GUIrunner

Configuration

The configuration was split into two parts, the device configuration and the device group configuration. The first holds all the information regarding the Tango devices that were to be managed by the control program. The latter holds the information in respect to the groups of devices, to which they may belong to. It heavily relates to the custom GUI library.

The device information incorporates the following parameters:

- Element name
- Type
- L[m], S[m], X[m], Y[m], Z[m],
- Section
- Subsystem
- Managed in CS
- Device Server name
- Device Server instance
- Device class
- Full TANGO device name
- Alias
- Triggered by TTL
- Custom GUI
- Aggregate GUI

ISBN 978-3-95450-148-9

- Description
- Comment

The device group information incorporates the following parameters:

- Group GUI Name
- Previous Group GUIs
- Next Group GUIs
- Additional argument

For more information about the configuration and its parameters please refer to the Solaris Synchrotron Control Program documentation [1].

We should mention here that the same device configuration file was used for the Solaris Import Program (a utility for populating the Tango database). Note that the Solaris Import Program also used a second source of information, regarded as device property configuration. The latter however exceeds the scope of the paper.

ControlProgram

The Control Program is the heart of the controls GUIs. It serves as a main application for accessing all the features and GUIs for operation purposes.

The Control Program parses the configuration and populates the content accordingly. It exposes the devices and device groups and provides access to the control GUIs.

The devices are presented in two different views:

- Device Tree
- Device List

The devices groups are exposed in a separate view:

- Device Groups

Moreover, the following features are implemented:

- Standard/custom device panels
- Custom panel generation
- Device filtering
- Device display management tools
- Device group panels
- Device state monitoring
- Profile management

Device overview

The main functionality of the Control Program is to present all the devices in the system. For every device, it provides a default device panel, exposing all the Tango attributes and command of the device to the user. Additionally, it is possible to use a custom implementation of the device screen for every particular type of the device instead.

The first two views of the Control Program (Device Tree and Device List) expose all the devices in the system. In the first view, user can access devices in a tree, structured in three layers. In the first layer, the devices are split according to the section to which they belong. In the second layer, the devices are split by their subsystems. In the third layer, actual devices are displayed, along with their descriptions. The name, section, subsystem and the description for every device are all read from the device configuration. In the Device List view, the devices are displayed in a list. The order of the devices in the list is read from the configuration, and is foreseen to match the actual order of the hardware in the facility.

For every device, a default device panel or a custom device panel can be opened to access the attributes and commands of the device and to execute required operations.

When working with the first two views, user is presented with additional functions for easier operation:

- Device filtering
- Tools for managing the displayed devices
  - Expand, collapse the tree
  - Select all devices

Attribute overview

Generating a custom panel was supported. Using standard selection techniques, user can select a desired subset of devices in any of the devices views. When requesting to generate a custom, the Control Program will open an additional popup window, prompting the user to select any subset of attributes that the selected devices expose. By confirming the attribute selection, a custom generated panel will appear, presenting all the selected devices with the selected attributes in a structured and flexible form.

Device groups

Devices can be organized into any number of groups. Each device group has a name, a type, any number of preceding device groups, any number of succeeding device groups and a desired additional parameter.

Affiliation of the devices to the device groups, the device group names and device group type are read from the device configuration. The preceding/succeeding instances and the additional parameter are read from the device group configuration.

For every type of the device group, a dedicated GUI must be developed. They are generally developed from a template that provides the layout and the form of the GUI, a background thread for heavy operations, the basic functionality for communication with the Control Program and the functionality for opening the preceding and succeeding instances. The actual content is independent from the template and is developed separately for each device group type. Moreover, the interpretation of the additional parameter for each instance of the device group is also independent from the template.

All device group instances are presented to the user in the third, Device Group view.

When opening a device group GUI, a Control Program looks for the implementation of the GUI for the device

ISBN 978-3-95450-148-9

group type of the selected device group instance, and retrieves a GUI panel. Doing so, the Control Program passes all the information, required by the GUI implementation to generate the panel, as an input argument. The input argument consist of a device group name, a list of all the devices belonging to the selected device group instance along with their ID and description, and an additional parameter of the selected device group instance. The GUI implementation for the device group type can upon this information build the panel for the selected instance.

Note that the implementation of the device group GUI differs for every device group type. The input to the device group GUI however differs for every device group instance.

A variety of device group panels were developed, intended for different operations:

- Beam diagnostic panel
- Modulator conditioning panel
- Power supply ramping panel
- Vacuum section overview panel
- Synoptic panel for GUN section of linac
- …

State monitoring

The Control Program provides full or partial monitoring of the device states. With this, a full control system state overview is provided. A user can also take advantage of this to monitor a certain subsystem, or to monitor the states of any device subset. The states of the devices are displayed on the left side of the device name, in both device views, and are represented with standard Tango colours. Additionally, a colour legend is displayed when this feature is requested.

Profile management

The Control Program supports the basic functionality for saving and loading a profile. With a profile, we refer to a set of opened GUIs, consisting of Device Panels, Custom Generated Panels and Device Group Panels. When monitoring a commonly used set of panels within the Control Program, it is useful to save a profile, to spare time when trying to open the same set of panels. When a profile is loaded, all the panels, with the same configuration and content are displayed, on the same position on the screen, as they were at the time when a profile was saved.

Custom GUI library

The Custom GUI library is a library that provides the custom device panels and device group panels that can be used within the Control Program. The structure, organization and naming conventions must be insured. For more information refer to the Solaris Synchrotron Control Program documentation [1].

GUIrunner

GUIrunner is an instance manager, an application for running the Control Program. Multiple instances of the control systems were foreseen, therefore multiple configuration sets and multiple custom GUI libraries were developed, each for one control system. The GUIrunner was developed to resolve the problems of feeding the correct configuration set and GUI sources to the Control Program. Moreover, GUIrunner also serves as an update manager. Before opening the Control Program, it checks for any possible updates and applies them beforehand.

ADAPTATION

The adaptation possibilities of the software were crucial in order to be able to adapt it to multiple control systems at Solaris. A configuration based design enabled such a degree of adaptation. Moreover, extendibility of the control program had to be guaranteed. Templates for developing new device group panels were provided to ensure easier future development. Later, a support for integrating external application into the Control Program was introduced.

The software is generic, future-proof and can be adapted to any Tango based facility.

CONCLUSION

Single entry point software for all control room operations was developed. Providing controlled content, a transparent usability was achieved.

The software is easily maintained; any updates or transitions are easily handled.

The required features and functionalities for the operation were developed; nonetheless the software additionally guarantees a great deal of extensibility of features.

A configuration based design was realized, and a required level of extensibility has been reached.

Future development of the software was also foreseen. A new layer of data aggregation was envisioned. Archiving support could be integrated, etc.

REFERENCES

[1] Solaris Synchrotron Control Program: https://github.com/synchrotron-solaris/app-cosylab-controlprogram

ISBN 978-3-95450-148-9