INNOVATIVE GRAPHICAL USER INTERFACES DEVELOPMENT: GIVE THE POWER BACK TO USERS

Gustavo Segura Millan, Adrien Ledeul,
Alexandru Savulescu, Bartlomiej Styczen, Daniel Vasques Ribeira
CERN, Geneva, Switzerland

Abstract

Graphical User Interfaces (GUIs) for Supervision, Control and Data Acquisition Systems (SCADA) are usually oriented to specialist users. In big organizations like CERN, where different teams play the roles of operators, scientists and instrumentation specialists, providing a unique or static user interface usually results in a situation of dissatisfaction for everyone. Providing distinct user interfaces for each type of user increases the development and maintenance effort and makes the software evolution heavier. The approach taken in the design and development of GUIs for radiation and environment protection at CERN addresses these issues by integrating user interface changes as an embedded software functionality. Key users were provided with a tool to build, deploy and maintain their own tailor-made user interfaces, in a graphical way and without the necessity of learning any kind of programming or scripting languages. Other benefits observed from this solution includes the reduction of the resources spent on the support and maintenance and the increase of the frequency of GUIs updates, executed without compromising the underlying control system. This paper describes the innovative design that was implemented.

INTRODUCTION

The Health Safety and Environment Unit at CERN provides a SCADA system for the radiation protection and environment monitoring of particle accelerators, experiments and the environment. The system called REMUS [1] (Radiation and Environment Monitoring Unified Supervision) runs 24/7 hours 365 days a year.

REMUS is a large system in its class (600.000 tags) interfacing 550 monitoring stations built convening 75 device types and 3000 measurement channels. Its GUI has 600 different synoptic views used by around 50 concurrent users from a total population of 200 authorised users. Those users have different roles, needs and expectations of the system.

Naturally, the GUIs require frequent updates to adapt to the ever changing CERN environment (installation of new instrumentation or regular changes of the instrumentation type or location).

The design of REMUS addresses two problems by design:

- Evolution and extension of the GUIs impact many users with different needs, sometimes conflicting.
- Frequent GUI updates are resource demanding tasks.

The REMUS approach to deal with these challenges is to provide the final users with the possibility of building their own GUIs.

BUILDING USER INTERFACES FOR SCADA SYSTEMS

The usual approach to design SCADA GUIs consists of making a schematic representation of the target process, installation or technical facility (synoptic views). Then placing widgets that represent devices, providing animated readings for supervision purposes and commands to control the underlying process. If the system is large, it may be necessary to present it in smaller parts or sub-systems in order to reach a lower level of details and to add a navigation mechanism that facilitates visualization and operation for users.

The attained level of detail is often the one of individual equipment, sensors and actuators under supervision and control from which the system is doing data acquisition. In order to complete the SCADA GUI it is necessary to add an access control mechanism to in order to grant the proper level of access rights to the authorised users.

BUILDING REMUS GUI

REMUS users have differing needs and expectations from the system. REMUS provides interfaces for operators of accelerators and experiments, physicists, instrumentation specialists, radiation protection engineers, environmental engineers, CERN Fire Brigade, instrumentation maintenance teams and the software support team. Providing a unique GUI for all these users regardless of their role is not a good approach as there is no interface that can suit everybody.

User interfaces need to adapt to the diverse user roles and require frequent modifications to handle the continuous changes in the supervised instrumentation.

The REMUS interface is organized in a tree-structure of synoptic views displaying CERN surface and underground areas. Widgets representing hardware devices and measurement channels are displayed on the synoptic views. There are different versions of widgets for each device type, addressing specific levels of detail required by different types of users.

REMUS Applications

In order to provide users with the most suitable interfaces, REMUS has a special kind of users named *Application Administrators*, who takes care of building interfaces for all other users. Administrators are neither part of the REMUS support team nor software developers.

Administrators can build *REMUS Applications* composed of synoptic views, widgets and a customisable alarm screen. These views are built to supervise an infrastructure or process (e.g.: accelerator or experiment radiation, water or air release to the environment, etc.). Applications include a sub-set of the supervised instrumentation and an access control mechanism.

Users Build the User Interfaces

Application administrators are provided with a tool named *REMUS Application Editor* that allows them to create new applications or modify existing ones (Fig. 1).

The application editor is a user friendly graphical tool that does not require any knowledge of programming languages or scripting to be able to build or modify user interfaces. The editor user composes the interface using drag-and-drop, without writing a single line of source code.

The application editor functionality include:

- Building synoptic views and placing background images (e.g.: schema of process, experiment plan, etc.)
- Adding navigation among the synoptic views
- Selecting widgets (devices) from a catalogue and positioning them on the synoptic views
- Linking widgets to existing devices declared in REMUS database

Applications are stored in the form of xml files.

In order to modify a running Application, REMUS provides administrators with a *download* function that writes the application as an .xml file that can be edited and *uploaded* back to REMUS Servers. REMUS servers then update all the clients running the uploaded application in real-time, without the need to restart the clients.

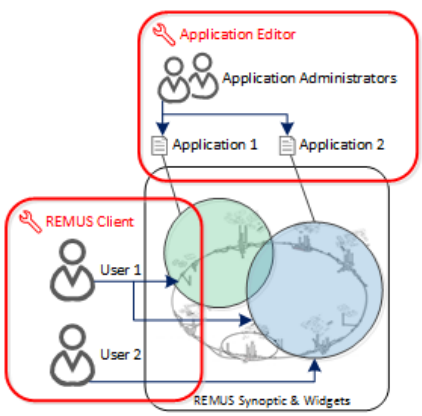

Figure 1: REMUS Application definition and scope.

REMUS automatically stores the successive versions of each application and makes all the versions available to administrators who can reuse them if necessary.

Access Control

Each application has its access control mechanism. Administrators can grant other users one of the following access levels:

- Visualize: real-time and archived data visualization.
- Acknowledge: visualize rights + the right to acknowledge alarms and system faults.
- Parametrize: acknowledge rights + the right to change device parameters.
- Configure: parametrize rights + the right of installing new devices.
- Admin: configure rights + the right to edit the application and giving access rights to other users.

Other Benefits

The outcomes of the REMUS strategy for building GUI are very positive:

- GUI support and maintenance efforts are distributed among Application Administrators that are close to the needs of the final users. Administrators are just final users with additional responsibilities.
- The GUI evolution and growth become the responsibility of the final users (administrators are users).
- Applications are tailor-made for specific kind of users and focus on the devices of their interest (only a sub-set of the total instrumentation) such as a specific accelerator or experimental area.
- The performance of the REMUS client software is improved, as the traffic between clients and server is reduced to the one generated by the devices displayed in the loaded client application.
- The REMUS support team focuses its effort on integrating new devices, developing widgets and improving core functionalities.

GUI FOR SAFETY SYSTEMS

Implementing changes in software used in safety systems differs from other type of systems in the process of test, acceptance and validation of the changes that affect safety functions. The processes for such systems are stricter, longer, more formal and tedious than for non-critical software systems.

GUI interfaces are not part of safety functions but they are very important especially when operators take safety-related decisions based on the information displayed by those interfaces.

REMUS as a Safety-related System

REMUS software runs in the CERN control rooms and provides operators with real-time information of the radiation conditions and of the release of effluents to the environment.

Based on the information provided by REMUS applications, operators can decide to stop accelerators and experiments for safety reasons or call for an intervention to stop or mitigate the pollution of the environment.

One of the particularities of REMUS is that the very frequent modifications of the user interfaces do not require extensive testing.

Modifying the GUIs in REMUS is, in most cases, just adding or modifying applications.

Changing or modifying a REMUS application is just using a function of the system that administrators can operate whenever they need to.

When a user needs to change a REMUS application, the user downloads the application from REMUS, edits and changes it with the Application Editor tool. The Application Editor software ensures that the application is valid and well-formed and writes it in the form of an .xml file. When the user uploads the modified application, REMUS parses it to construct the UI that becomes automatically available to all authorised user. The parsing process verifies that the application is coherent and well-formed and detects errors beforehand.

No further acceptance procedures are required for the validation of new applications or for implementing changes in existing ones. The benefits of this mechanism are:

- Reduction of the resources needed by the software development team in formal test and validation process.
- Increased user involvement in the process as they take the responsibility of the GUI construction.
- Increased user confidence in the system.
- Not extra software development required despite the dramatically increase of GUI modifications and updates.

CONTINUOUS OPERATION

At CERN, accelerators and experiments work 24/7 during run periods that last for months only interrupted by short technical stops taking place every few weeks and lasting for few hours. Long shutdown periods have a duration of a few months and take place between run periods.

Control systems remain unchanged during long run periods.

Generally, the deployment of new control software, upgrades and modifications takes place mainly during shutdowns in order to get the time necessary for extensive test and validation on quasi-real operational conditions.

The previous radiation and environment SCADA systems operated at CERN years ago (RAMSES [2], ARCON [3]) used to follow the same development and deployment cycle in which software upgrades used to take place during long shutdown periods. Only small modifications, typically related to GUI, were deployed in short technical stops.

REMUS allows modifying and deploying applications at run-time during accelerators operation without compromising the operation. The mechanism implemented to build and upload the applications ensures the integrity of the REMUS System at run-time (Fig. 2).

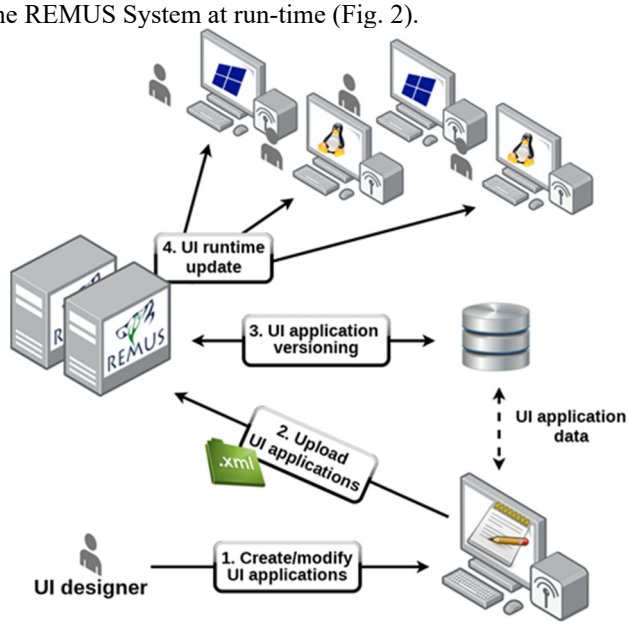

Figure 2: REMUS Applications edition.

CONCLUSION

The REMUS approach for the management of GUI has proven to be very versatile bringing many benefits to the system such as:

- Reduced software development and maintenance costs.
- Increased user satisfaction.
- Fast user adoption.
- Decreased "time to market" of new GUIs releases.

REFERENCES

[1] A. Ledeul, G. Segura, R.P.I. Silvola, B. Styczen, and D. Vasques Ribeira, "REMUS: The new CERN Radiation and Environment Monitoring Unified Supervision", in *Proc. 15th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'15)*, Melbourne, Australia, October 2015, paper TUD3O03, pp. 574-577, ISBN: 978-3-95450-148-9, doi:10.18429/JACoW-ICALEPCS2015-TUD3O03

[2] G. Segura Millan, D. Perrin, L. Scibile, "RAMSES: The LHC Radiation Monitoring System for the Environment and Safety", in *Proc. ICALEPCS'05*, Geneva, Switzerland, Oct. 2005, paper TH3B.1-3O.

[3] D. Forkel-Wirth, G. Segura Milan, D. Perrin, S. Roesler, C. Theis, H. Vincke, Hz. Vincke, P. Vojtyla, M. Widorski, "ARCON/RAMSES: Current Status and Operational Risk", Workshop on LHC performance 2010, Chamonix, France.