

# MEDICIS HIGH LEVEL CONTROL APPLICATION

C. Charrondière, K. Develle, T. Stora, CERN, Geneva, Switzerland

## Abstract

CERN MEDICIS is a research facility that will produce radioisotopes for medical research using the primary proton beam at ISOLDE and ISOLDE-like targets. It will start operating later in 2017. In this framework, the high-level application for the new MEDICIS beam line is responsible for the control of various equipment, such as power supplies, faraday cups and scanners, as well as the monitoring of environmental parameters such as the vacuum level. It is characterized by a single user-friendly interface to facilitate the operators' tasks.

In this paper, we provide arguments for the chosen solution and give the latest update on the status of the high-level application.

## INTRODUCTION

CERN-MEDICIS [1] is a facility to produce radioactive isotopes for medical research. Radioactive isotopes are produced in a dedicated ISOLDE-like target placed between the ISOLDE primary target and the proton beam dump: protons non-interacting in the primary target have the chance to interact in the second one. Once irradiated, the target is removed by a robot and placed in the MEDICIS laboratory, where isotopes are extracted by diffusion in a high-voltage front-end and selected by a mass spectrometer before delivery to the medical research centre.

## APPLICATION REQUIREMENTS

As any other beam lines at CERN, MEDICIS (Figure 1) one is equipped with several beam instruments.

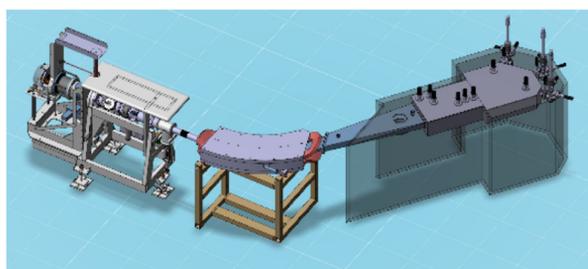


Figure 1: MEDICIS beam line.

Each instrument is provided by a CERN equipment group together with a specific expert user interface. A unified graphical user interface, to control or monitor all the hardware is required to simplify the facility operation and to reduce to a minimum the necessity of continuous monitoring of experts.

Since the MEDICIS facility is composed of a very heterogeneous set of hardware, one of the key point to be taken into account for the general control application is the flexibility: the software architecture should allow the integration in the Graphical User Interface (GUI) of any new

devices in an easy way without re-programming the software.

In the following, the overview of the devices to be controlled/monitored is given to focus later on the software architecture chosen to ensure the requested flexibility.

The MEDICIS application represents a use case of an architecture that could be totally generic and followed in the future developments of GUI for physics experiments.

## HARDWARE

The main hardware to control and monitor are the power supplies, faraday cup, wire scanner vacuum and collection chamber. These devices will be shortly introduced below.

### Power Supplies

Several types of power supplies (Figure 2) and converters are installed in the MEDICIS facility depending if there are aiming at beam steering, magnet powering for the beam line spectrometer or target warming for the front end.

A CERN protocol, CMW [2] is used as communication layer between the hardware and the user interface. The CERN power converter team is providing communication layers linked to the equipment [3].



Figure 2: Example of power supply installation.

### Faraday Cup

A Faraday cup [4] is a conductive cup designed to intercept charged particles in vacuum. The resulting current can be measured and used to determine the number of ions or electrons hitting the cup.

The CERN beam instrumentation group handles the hardware and low-level control including safety. The CMW is also used to communicate with the higher control level.

### Wire Scanner

A wire scanner [5] is installed in a diagnostic box just after the output of the separator magnet. This device is measuring the transverse beam density profile in a particle accelerator by means of a thin moving wire. As the wire passes through the beam the interaction generates a cascade of secondary particles. These are intercepted by a scintillator, coupled with a photomultiplier, which measures the intensity of the light produced. Alternatively, if a conducting wire material is used, the secondary emission electrons created can also be used to measure the beam

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

profile through the current flow generated on the wire (Figure 3).

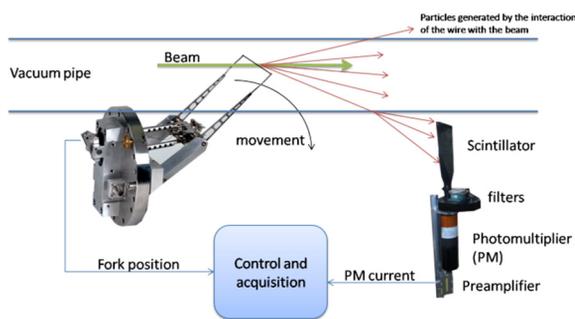


Figure 3: Working principle of a wire scanner.

Like the Faraday Cup, the CERN beam instrumentation group handles the hardware and CMW will be used for the high-level communication.

### Vacuum

At CERN, the TE-VSC group is in charge of the design, construction, operation, maintenance and upgrade of high and ultra-high vacuum systems for accelerators and detectors.

The vacuum team provides the expert interface to be able to control the installation and the relevant information (vacuum levels, pumps status, etc..) are published [6] for the unified graphical user interface through CMW.

### Collection Chamber [7]

The radioactive isotope beam from the front-end, after the mass selection done by the spectrometer, is cumulated in a collection chamber. The beam will be focused on samples where ions will implant.

EN/STI group is in charge of developing the mechanical parts (Figure 4) as well as the PLC programming to ensure the basic movements and the safety interlocks. Data will be exchanged with the high-level control interface using the Fetch/Write protocol from Siemens™ [8].

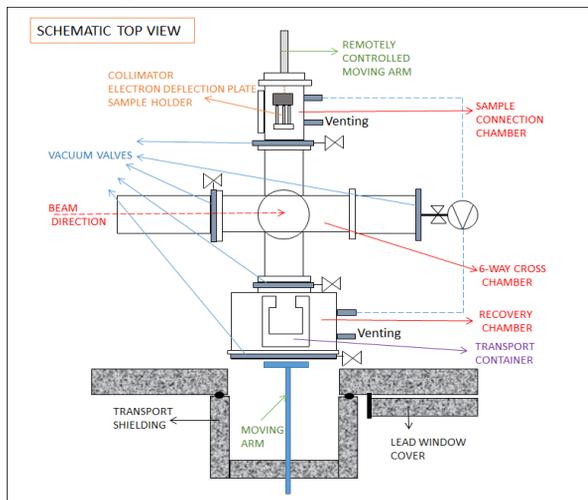


Figure 4: Collection chamber schematic top view.

## SOFTWARE

From the requirements and the variety of equipment to be controlled, it is clear that one has to develop a flexible user interface. On top of this, each hardware element should remain controllable by the expert interface at any time. Using the experience from the ISOLDE [9] offline separator, the decision of developing a single user interface was taken in order to be able to drive any hardware available on the MEDICIS facility.

To speed up the development process, LabVIEW™ has been chosen as programming language. Making use of configuration files, LabVIEW™ classes [10] and RADE [11] framework made it easy to add new modules and user interfaces.

### Architecture

**Class architecture** It has been decided to design the software with an object-oriented [12] program approach. As the software has to be flexible, and simplify future development, the design of the class hierarchy was really important. For this project, there are three main hierarchies:

#### 1. The CMWproperties hierarchy (Figure 5):

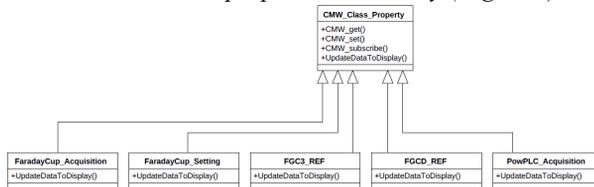


Figure 5: CMW skeleton, Class-properties hierarchy.

This class tree has been designed to be able to reuse for every CMW related device of the MEDICIS project. Every CMW get, set and subscription must work with all the devices, then some more specific implementation of function like the “UpdateDataToDisplay” will be adapted in each case thanks to dynamic dispatch.

#### 2. The Front Panel hierarchy (Figure 6)

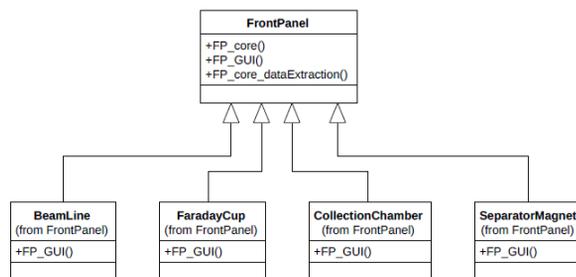


Figure 6: Skeleton of the Front Panel hierarchy.

As the final application has a lot of different graphical user interfaces, and they all have a lot of functionalities to share: run in parallel, get data from subscription, set a value, open additional function, the Front Panel class has been created to share all this functionality (with the

FP\_core vi and all of its subVIs) and specific User Interface can be developed with dynamic dispatch in FP\_GUI.

### 3. The Additional GUI hierarchy (Figure 7)

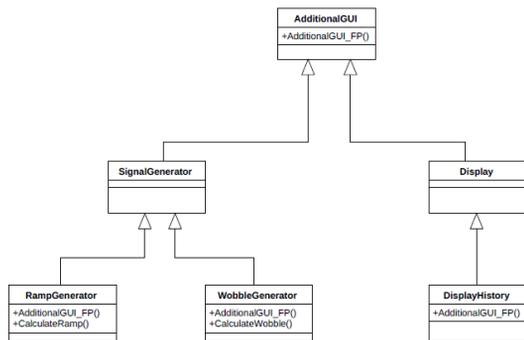


Figure 7: Additional GUI hierarchy.

This class has been developed for all the additional functions to be added to a device: trace a graph history for a value measured by a faraday cup, generate a ramp with a power supply or even some evolved shaped signals. The dynamic dispatch on AdditionGUI\_FP allows to run all these additional VI with a common interface, but with a specific function depending on the desired functionality.

There is also a minor class tree, not as important in terms of general flexibility of the application.

**Software architecture** This application has been designed to be as flexible as possible in order to react quickly to future request. For that purpose, the main VI starts a lot of daemons running in parallel (Figure 8).

The most important daemons are all the instance of CMWSubscription.vi. There is one instance per property per device. They collect all the data coming from all the devices. The second most important daemons are the FP\_core.vi. They extract data from a list of device coming from the configuration file, and they display them in the FP\_GUI.vi attached. Each FP\_core has a different FP\_GUI to display (Figure 9 and 10) the data thanks to the dynamic dispatch. With this approach, it is very simple to create a new subscription and the new Graphical User Interface by reusing all the common functionality of communication between a front panel and a device and expand only the feature corresponding to the new request.

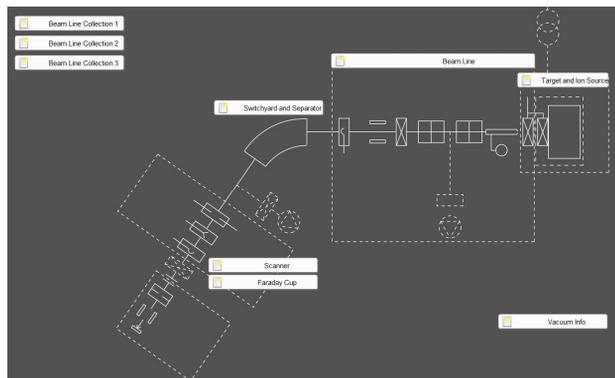


Figure 9: The main MEDICIS GUI panel.

With the synoptic view of the facility the user can select different panels to open.

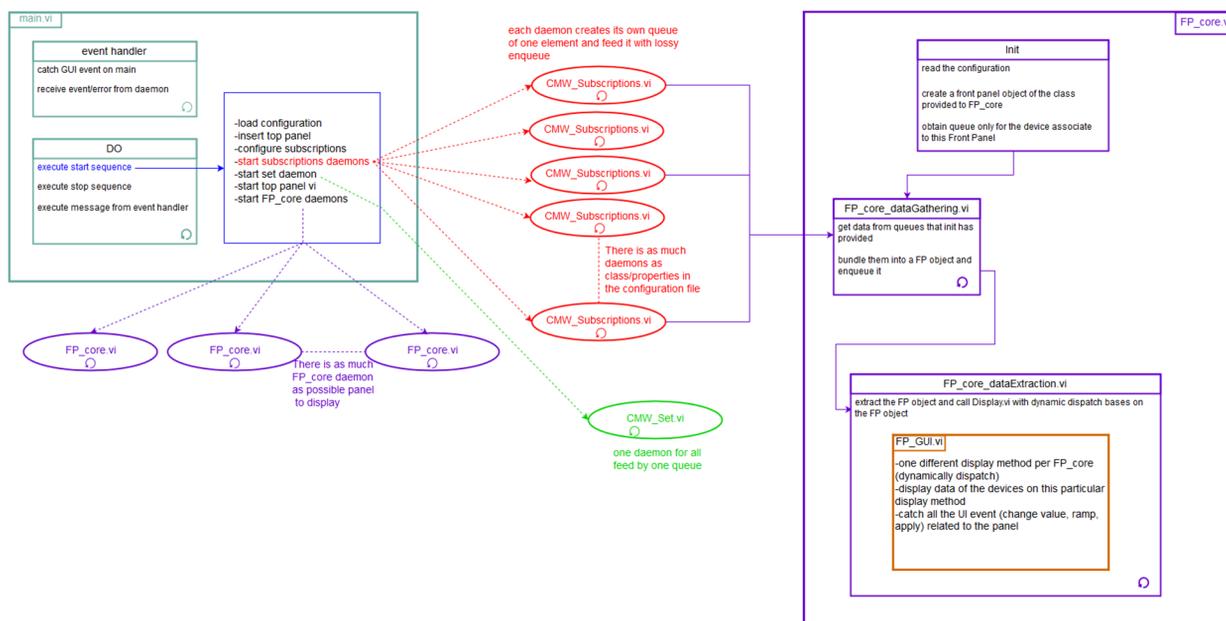


Figure 8: Software architecture.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

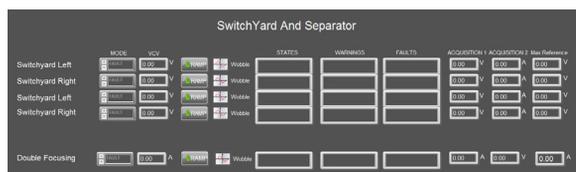


Figure 10: Example of a front panel GUI, the “SwitchYard and separator”.

### Configuration Files

The configuration file contains all front panel declaration associated with a name. For example:

- FP List.<size(s)> = "10"
- FP List 0 = "MedicisSynoptic"
- FP List 1 = "BeamLine"
- ...

This file also has the entire device list with all properties needed for each device. Below one can see that we are expecting data from 3 different properties to be displayed on a panel called “TargetAndIonSourceUnit”.

- device list 0.Properties.<size(s)> = "3"
- device list 0.Properties 0 = "/Acquisition"
- device list 0.Properties 1 = "/Setting"
- device list 0.Properties 2 = "/ConfigInPLC"
- device list 0.FP to be display on = "TargetAndIonSourceUnit"
- ...

Thanks to that, the application is very flexible for the developer because each device can be configured to be displayed in each Front Panel.

### Adding new hardware to display and control

Using the LabVIEW™ class architecture combined with the configuration file makes it easy to quickly react to new requests. It takes a few minutes to fill in the file with the data source information. This information is used to get or set data to the newly declared device. The fast development environment, ie LabVIEW™, allows the programmer to provide a new Graphical User Interface within a day. The acquisition data source is linked to a display in the configuration file.

## VALIDATION PROCESS

A series of validation tests has been done on already existing equipment in the ISOLDE facility (the ISOLDE of-line separator and the ISOLDE Faraday cup) plus a test bench reproducing the same conditions as in MEDICIS.

### Isolde Offline Separator

There are 3 kinds of CMW subscription for the power supplies. For one of them, it was not possible to have a test device, so the Isolde offline separator having exactly the

same type has been used to develop and test the communication. It has been also the opportunity to renew the old application by profiting of the flexibility of the MEDICIS software design.

Some required features such as mass scan or being able to set ramp parameters for power supplies have been developed for MEDICIS and validated first on the Isolde of-line separator installation.

### Isolde

There was no Faraday Cup available at the development time for test, so the development has been done based on a JAVA application controlling an older type of Faraday Cup in the ISOLDE facility. Once the communication was ready, a test has been done there to validate the behaviour until being able to commission in the MEDICIS facility.

### Test Bench

It has been possible to test individually the other kind of device on a test bench; the next step will be to commission everything in the MEDICIS facility.

## CONCLUSION

The MEDICIS high level application has been developed leaving the maximum flexibility for future developments and always considering the possibility of hardware modification during the construction of the facility. The use of LabVIEW™, the CERN common middleware and the RADE framework have sped up the development process. It is foreseen that the available user interfaces can be enhanced and/or replaced during the MEDICIS exploitation. Thanks to the availability of equipment in the CERN accelerator complex, it has been possible to validate drivers as well as communication layers before the operation of the machine.

A new system has been requested to appear in the high-level control application and we are gathering the communication information in order to quickly provide a new interface with summary information.

## REFERENCES

- [1] T. Stora, “CERN-MEDICIS (Medical Isotopes Collected from ISOLDE): A new Facility”, Applied Sciences, 2014, 4, 265-281
- [2] K. Kostro, “The controls middleware (CMW) at CERN status and usage”, WE201, ICALEPCS 2003, Gyeongju, Korea
- [3] Power supplies, <http://epc.web.cern.ch/>
- [4] E. D. Cantero, “Performance tests of a short faraday cup designed for HIE-Isolde”, IPAC 2013, Shanghai, China
- [5] R. Veness, “design of a high-precision fast wire scanner for the SPS at CERN”, IBIC 2012, Tsukuba, Japan
- [6] R. Gavaggio, “Examples of Vacuum systems controlled by PLCs”, THAP093, ICALEPCS 2001, San Jose, California
- [7] A. Brown, “Design of the CERN MEDICIS collection and sample extraction system”, CERN-THESIS-2015-251
- [8] Fetch/Write protocol from Siemens™, <https://www.siemens.com/global/en/home/products/automation/systems/industrial/plc.html>

- [9] ISOLDE, <http://isolde.web.cern.ch/>
- [10] LabVIEW™ classes, <http://www.ni.com/white-paper/3573/en/>
- [11] O. Andreassen, “Turn-key applications for accelerators with LabVIEW RADE”, WEMAU007, ICALEPCS 2011, Grenoble, France
- [12] LabVIEW object-oriented, <http://www.ni.com/white-paper/3574/en/>