

# EXTENDING WINCC OA FOR USE AS ACCELERATOR CONTROL SYSTEM CORE

M. Marchhart, A. Brett, M. Hager, M. Janulis, R. Moser, H. Pavetits, C. Torcato De Matos  
 EBG MedAustron, Wr.Neustadt, Austria  
 J. Gutleber, CERN, Geneva, Switzerland  
 J. Dedic, Cosylab, Ljubljana, Slovenia

## Abstract

The accelerator control system for the MedAustron light-ion medical particle accelerator has been designed under the guidance of CERN in the scope of an EBG MedAustron/CERN collaboration agreement. The core is based on the SIMATIC WinCC OA SCADA tool. Its open API and modular architecture permitted CERN & MedAustron to extend the product with features that go beyond traditional supervisory control and that are vital for directly operating a particle accelerator. Several extensions have been introduced to make WinCC OA fit for accelerator control: (1) Near real-time data visualization, (2) external application launch and monitoring, (3) accelerator settings snapshot and consistent restore, (4) generic panel navigation supporting role based permission handling, (5) native integration with interactive 3D engineering visualization, (6) integration with National Instruments based front-end controllers. The major drawback identified is the lack of support of callbacks from C++ extensions. This prevents asynchronous functions, multithreaded implementations and soft real-time behaviour. We are therefore striving to search for support in the user community to trigger the implementation of this function.

## INTRODUCTION

MedAustron [1] [2] is an ion therapy and research centre in Wr. Neustadt, Austria, presently starting injector beam-commissioning. The facility features a synchrotron-based accelerator (Figure 1) with up to 5 ion sources for protons, carbon ions and possibly other light ions. It will

provide ion beams with energies up to 800MeV to 5 beam lines, one of which is a rotating proton gantry. MedAustron chose the commercial SCADA tool SIMATIC WinCC OA from ETM professional control as core operating system for controlling the particle accelerator [3]. The tool has been designed following an open, distributed system architecture to accommodate traditional supervisory control and data acquisition tasks. It offers the possibility to extend the system for tasks that go beyond those traditional tasks.

## CONTROL SYSTEM

The MedAustron control system is based on a 4-tier architecture:

- **Presentation Tier** (Tier 1) contains all user interfaces for visualization and control: (1) WinCC OA based generic user interfaces, (2) LabView based expert panels and (3) procedures.
- **Processing Tier** (Tier 2) contains basic control system services such as the WinCC OA as the main operating system, publisher subscriber services and logging services.
- **Equipment Tier** (Tier 3) provides front-end applications primarily implemented in LabView that provide a unified interface to the upper tiers.
- **Frontend Tier** (Tier 4) contains all devices that perform local real-time control of specific subsystems.

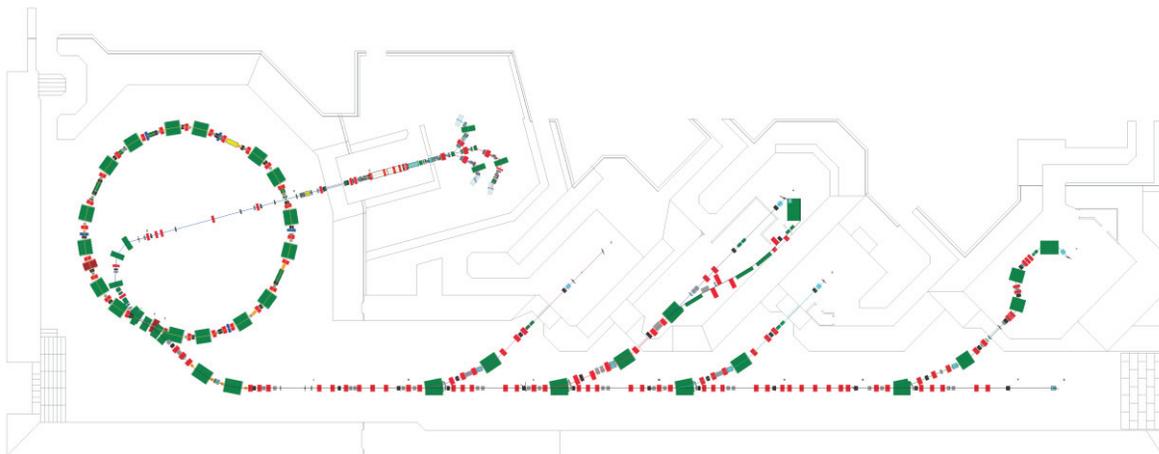


Figure 1: MedAustron accelerator layout.

## NAVIGATION AND PERMISSIONS

For panel navigation and handling different user permissions (Figure 2) the MedAustron introduced the so called Navigator.

The Navigator is the main entry point used during operation. It is a container that manages all user interfaces through a general panel framework into which individual graphical user interface panels can be plugged in. This framework gives the possibility to organize panels in a hierarchical tree so that users can navigate from one panel to a number of other panels. Panels may be reached by navigating along different paths in the tree.

Access permissions [4] can be assigned (1) for each panel and (2) for each element within a panel. Thus providing the same user interface for operators with different permissions. In addition each panel can be marked for exclusive access to prohibit opening the panel twice at the same time.

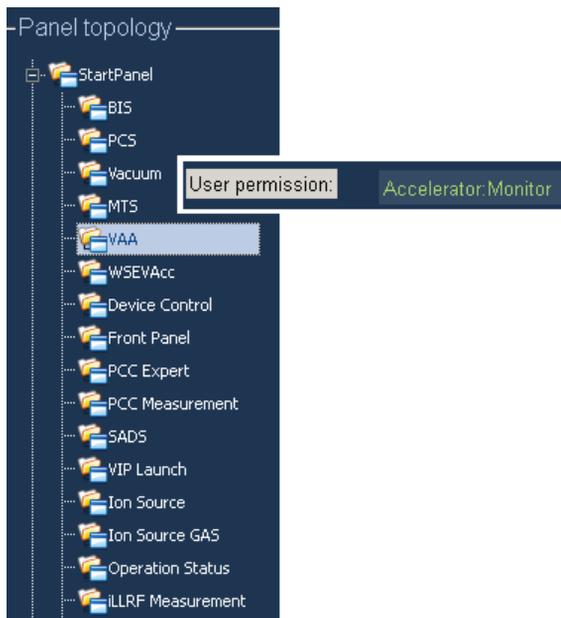


Figure 2: Definition of panel permissions.

## EXTERNAL APPLICATIONS

The VIP (Virtual Instrument Panel) launcher (Figure 3) is used to start external, device specific interactive applications such as Labview panel applications. The VIP launcher was introduced to handle the start and stop of external applications directly from the Navigator to keep the Navigator as main entry point and the role based permission handling.

Each device that offers a VIP must implement a data-point element called `panel` which specifies an executable program that is used to interact with the device directly. When such an external application is started, the VIP launcher registers the locally run application process in a dedicated WinCC OA datapoint.

A WinCC OA extension (.dll) was introduced that gives the possibility to keep track of started external applications by monitoring the process number assigned

by the operating system. The started application can always be stopped by the operator directly from WinCC OA. Due to the lack of callbacks from C++ extensions an additional task (WinCC OA panel script) checks periodically, about once per minute, if the launched applications are still running. If an application has terminated unexpectedly or upon an explicit application close operation in the application without that WinCC OA has been notified, VIP launcher removes the process from the registered list.

Device	VIP type	User	Start time [utc]	VIP start	VIP stop
S1_01_000_FCN	lvexe	mk1@CS-01-006-CPU	2013-09-16 08:04:43		stop
S1_01_000_SLX	lvexe			start	
S1_01_000_WSX	lvexe			start	
S1_01_001_WSX	lvexe	mk1@CS-01-006-CPU	2013-09-16 08:04:47		stop

Figure 3: VIP launch table.

## SHADOW AND RESTORE

During machine commissioning an essential functionality is the saving and restoring of a consistent set of accelerator configuration settings. The MedAustron control system distinguishes between three types of accelerator settings:

- Non-cycle dependent configuration values
- Cycle dependent simple values (e.g. a current set-point of a power converter)
- Cycle dependent complex values (e.g. a timing sequence file of the main timing system)

The shadow and restore mechanism permits the operator to take a snapshot of the currently applied accelerator settings in the SCADA system and allows the commissioning team to restore those setting at a later point in time. This mechanism is entirely implemented in WinCC OA panel scripts. The definition of device properties that should be taken into account during the shadow and the restore process takes place in the xml device description documents (see example below) which are used to configure the entire control system. After configuration of the SCADS system this information is stored using the WinCC OA user bits of the individual data-point elements. Only those properties that are explicitly marked will be saved and restored.

```

...
<property>
  <propertyName>singleSetPoint_ccv
</propertyName>
  <description> holds the list of cycle dependent single set
points </description>
  <datatype> float </datatype>
  <unit> A </unit>
  <ccDependent> true </ccDependent>
  <whereCreate>tier1,tier2,tier3,tier4
</whereCreate>
  <shadow> true </shadow>
</property>
...
    
```

Code 1: Example property definition in XML device description.

## DATA ACQUISITION AND VISUALIZATION

Visualization and correlation of measurement data is a key requirement during accelerator hardware and beam commissioning. Measurements taken by the front end controllers are published using the MAPS (MedAustron Publisher Subscriber) protocol.

The near real-time visualisation of these measurements is realized by the Qt based plot ewo (external widget object). This ewo can be placed in a WinCC OA panel and is configurable via WinCC OA scripts to subscribe to the required measurements (Figure 6) such as beam spectrum measurements for the ion source, measurements from the injector system or current and voltage measurements from the PCC (Figure 4).

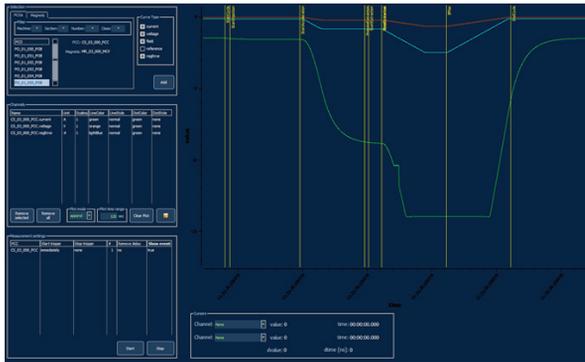


Figure 4: Measurement visualization in the power converter measurement panel.

## INTERACTIVE 3D VISUALIZATION

Quickly identifying misbehaving components within the plant is the key for maintaining a high level of safety and for keeping maintenance costs under control.

The synoptics widget for WinCC OA panels based on Qt provides interactive 3D process visualization of the accelerator complex. The accelerator layout is loaded from a step file (.stp) and the 3D objects can be connected to their corresponding device data-points using WinCC OA scripts to display the status of the devices (see Code 2).



Figure 5: 3D Visualization.

```

main()
{
  synoptics.loadStep("synoptics.stp");
  dpConnect("statusCB","device1.status")
}

statusCB(string device,string status)
{
  if(status=="OK")
    synoptics.setElementColor("device1","white");
  else if(status=="NOTOK")
    synoptics.setElementColor("device1","red");
}
    
```

Code 2: Example connecting 3D object to DPE.

## FRONTEND CONTROLER INTEGRATION

The MedAustron control system features in total 30 FECs (Front End Controllers) that are based on National Instruments PXI and PXIe platforms. Notable FECs are the Power Converter Controller (PCC) [5], the REDNet Main Timing System (MTS) [6], the ion source, the injector LLRF and the beam diagnostic FECs.

The communication between this FECs and the WinCC OA SCADS system is realized using the National Instruments Shared Variable Engine as intermediate communication layer. The Shared Variable Engine (SVE) features on one hand the NI shared variable protocol and on the other hand an OPC interface to which the WinCC OA connects via an OPC driver.

In addition an in-house developed FTP library (.dll) is used to upload cycle dependant complex data such as timing and set-point sequence files directly from WinCC OA panels to the FECs.

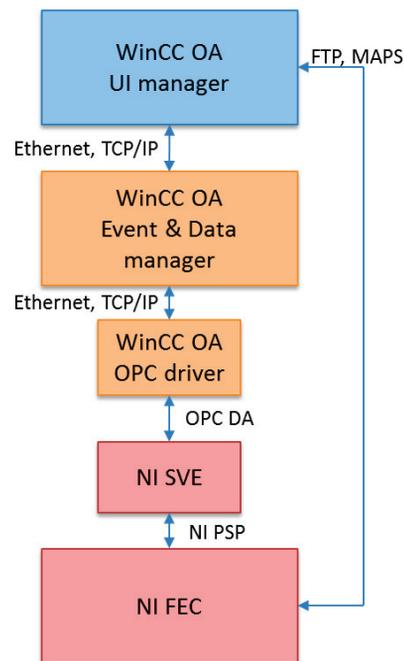


Figure 6: NI FEC communication with WinCC OA.

Copyright © 2014 CC-BY-3.0 and by the respective authors

## SUMMARY

The described system is deployed on site since December 2012. Since January 2013 the system is used to test and commission the low energy beam transfer line. Elements tested so far include ion sources and power converters such as set-point driven Heinzinger power converters, MaxiDiscCap and the LEBT fast deflector. The key was the choice to use WinCC OA as core operating system and the usage of its open API to extend the core functionality with the needs for commissioning and operating a particle accelerator.

The current integration of NI front-end controllers using the NI shared variable engine and OPC protocol is sub-optimal since it was discovered that the shared variable engine loses values after a restart which leads to inconsistencies with the values seen in the WinCC OA system. We intend to upgrade to LabView 2012 which provides the advantage of OPC/UA supports for non-realtime and realtime systems. This allows us to avoid the inconsistency by removing the shared variable engines and connect the FECs directly to WinCC OA using the OPC/UA protocol.

## REFERENCES

- [1] M. Benedikt, A. Wrulich, “MedAustron—Project overview and status”, *Eur. Phys. J. Plus* (2011) 126: 69.
- [2] M. Benedikt, A. Fabich, “MedAustron—Austrian hadron therapy centre”, *Nuclear Science Symposium Conference Record 2008. NSS '08. IEEE*, pp.5597-5599, 19-25 Oct. 2008.
- [3] J. Gutleber, “The MedAustron Accelerator Control System”, *Proc. ICALEPCS*, 2011.
- [4] P. Golonka, M. Gonzales-Berges, “Integrated Access Control For PVSS-Based SCADA Systems at CERN”, in *Proc. ICALEPCS 2009*.
- [5] J. Gutleber, “High-Density Power Converter Real-Time Control for the MedAustron Synchrotron”, in *Proc. in ICALEPCS 2013* (submitted).
- [6] J. Gutleber, “A Timing System For Cycle Based Accelerators”, in *Proc. In ICALEPCS 2013* (submitted).