THE *CS* FRAMEWORK AS A CONTROL SYSTEM FOR THE HITRAP FACILITY AT GSI

D. Neidherr[*], D. Beck, H. Brand, F. Herfurth, GSI Helmholtzzentrum für Schwerionenforschung GmbH, 64291 Darmstadt, Germany

Abstract

At GSI Darmstadt the linear decelerator HITRAP is currently under commissioning. The aim is to decelerate heavy, highly charged ions up to U^{92+}-, which come from the experimental storage ring (ESR) at an energy of 4 MeV/u, in several distinct steps down to less than one meV. In the end, bunches of up to 10^5 ions are captured and stored in a Penning trap from where they can be delivered to different experiments, which perform for example studies for ion-gas and ion-surface interactions, precision mass measurements, laser spectroscopy, or g-factor measurements. The low-energy part of HITRAP is controlled by an adapted version of the *CS* framework developed at GSI which is an object-oriented, event-driven and multithreaded framework, based on **N**ational **I**nstruments LabVIEW 2009 and designed for small and medium size experiments. The object-oriented approach of the *CS* framework makes the system very flexible with respect to changes in the configuration. The DIM protocol used as communication layer makes it possible to distribute it across a large network. For HITRAP new classes were implemented in order to read out a new type of detector and to fulfil the specific needs of this facility. This paper will give a short overview about those developments.

THE HITRAP FACILITY

The HITRAP [1] project at the GSI Helmholtzzentrum für Schwerionenforschung in Darmstadt/Germany is dedicated to make heavy, highly-charged ions accessible to precision experiments. The bare U^{92+} core represents the strongest electric field which can be produced in the laboratory. Therefore experiments with such ions at rest can test a number of physical theories, like the QED with an enormous precision [2].

HITRAP is installed behind the experimental storage ring ESR and is composed of three main parts: (i) a linear deceleration stage consisting of an IH-structure and a 4-rod RFQ which decelerates the beam coming from the ESR at 4 MeV/u down to 6 keV/u, (ii) a cylindrical Penning trap to accumulate and cool down the ions via electron and resistive cooling to 4 K, and (iii) a section which transfers the cooled ions to the different experiments. A sketch of the setup is shown in Fig. 1. Parts (ii) and (iii) are controlled with a control system based on the *CS* framework.

The ions from the ESR reach the HITRAP facility with a repetition rate of not larger than one shot every 30 seconds. Once decelerated in the linear section they are injected in the Penning trap where different cooling techniques are applied [3]. For beam monitoring at HITRAP in general a micro-channel plate (MCP) detector together with a phosphor screen and a CCD camera is used. These detector assemblies are mounted on a moveable feedthrough together with a faraday cup for an absolute current measurement. Because all with the setup connected devices are controllable within the control system, automated parameter scans can be performed, which improves the optimization procedure a lot.

THE CS FRAMEWORK

The *CS* framework [4,5] was developed at GSI/Darmstadt in order to provide a common platform for different experiments. Especially setups with a lot of ion optical elements and detectors can very easily benefit from the work already put in the *CS*. It is based on native LabVIEW from NI and uses object-oriented techniques in order to archive the needed flexibility for a general framework. The different objects use an event-driven mechanism to communicate with each other making it even possible to reconfigure the system on the fly. A class can for example represent a certain type of device and is used as a pattern to create objects, which are initialized with the help of a configuration database. The Data Logging and Supervisory Control (DSC) module of LabVIEW adds also an intrinsic possibility for alarming and historical trending.

Today, mainly trap experiments like SHIPTRAP [6] at GSI, ISOLTRAP [7] and WITCH [8] at CERN, TRIGATRAP [9] at the University of Mainz, LEBIT [10] at NSCL/MSU, PENTATRAP [11] at MPIK/Heidelberg, but also experiments like PHELIX [12] at GSI or Polaris [13] in Jena trigger further developments of the framework. In general, the similarities between these experiments make it easy to reuse code from one experiment at another one. This means that adding a new device to a setup is very often not linked with additional programmers work, an advantage since at many of these experiments PhD students have to maintain the control system additionally to their scientific work. Already available are classes for many different power supplies, frequency generators but also other devices such as a pulsed pattern generator on an FPGA card [14], capable of producing pulses with a precision of several ns, or a sequencer class to control the behaviour of an automated measurement.

Comparable to the situations at the other experiments, several specific add-ons had to be implemented for HITRAP.

[*] D.Neidherr@gsi.de

ISBN 978-3-95450-124-3

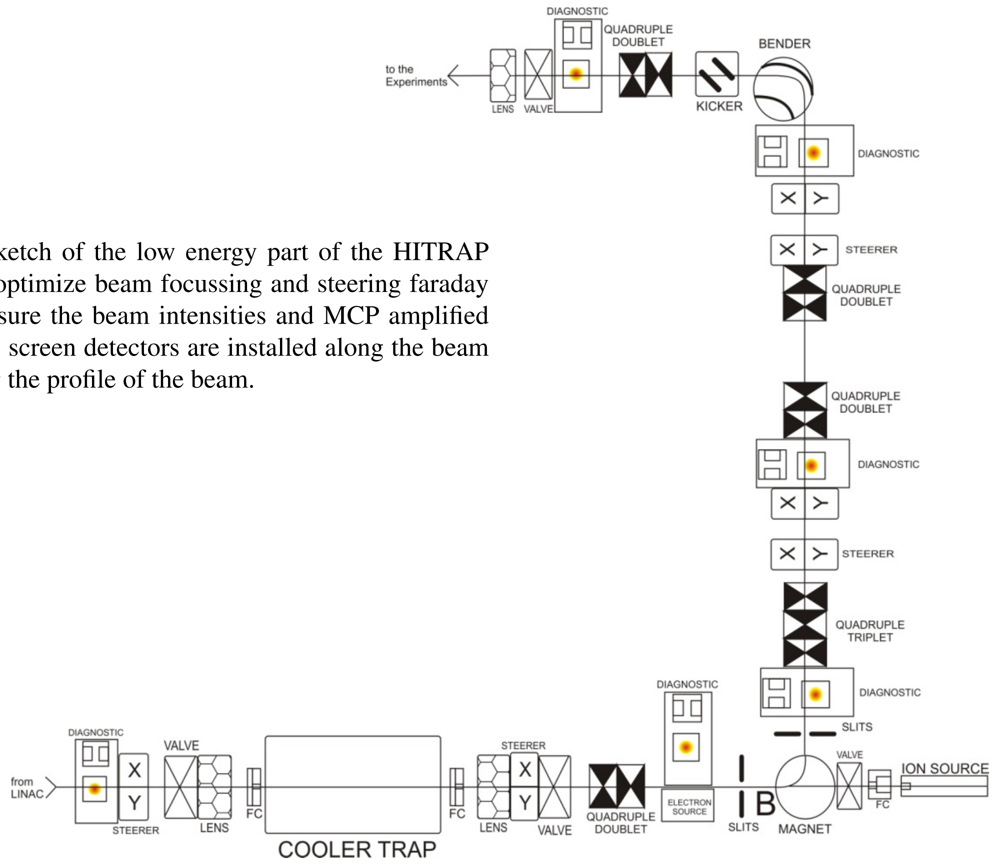

Figure 1: Sketch of the low energy part of the HITRAP facility. To optimize beam focussing and steering faraday cups to measure the beam intensities and MCP amplified phosphorous screen detectors are installed along the beam line showing the profile of the beam.

HITRAP SPECIFC ADD-ONS FOR THE *CS* FRAMEWORK

A lot of smaller improvements and extensions were made for the control system of the HITRAP setup. For example, the 90 degree bending magnet connecting the lower horizontal beam line with the vertical beam line works now with a P-regulation so that fluctuations in temperature and humidity should not affect anymore the transmission efficiency and resolution.

Each diagnostic chamber consists of a faraday cup and an MCP detector combined with a phosphorous screen and a CCD camera. The detectors are mounted on movable feedthroughs connected to step motors, which can be controlled within the control system.

The two main developments for the HITRAP control system were a diagnostic program to read out the camera information and link them to the control of beam line elements and a new, flexible graphical user interface (GUI). Both improve the usability of the control system and can also be used at other experiments if needed.

IMAQ Cameras Within the CS *framework*

With NI-IMAQdx National Instruments provides already a driver library for image acquisition. Supported are for example IEEE 1394, Gigabit Ethernet, or network IP cameras (Firewire as well as GigE cameras are used at HITRAP). This library has been already used at GSI in a feasibility study [15] to explore the possibilities of native LabVIEW OOP. For this, the so-called HGF base classes were developed. The main application for this study were small standalone tasks, but since it is object oriented it is very easy to incorporate the HGF_IMAQ classes into the *CS*. Only a few adjustments have to be made. One of the major differences between LVOOP and the object oriented concept used in the *CS* framework is that LVOOP does not treat objects as entities, whereas this is necessary in the *CS* since there, an object is in general directly connected with a hardware device. To use passive LVOOP classes in the *CS* one just has to create a wrapper *CS* class where the desired LVOOP base class object needs to be copied into the *CS* attribute data. This means that this one *CS* class can in principle call all functions of the LVOOP base class including their derived classes by calling the dynamic dispatch VI's (C++/Java: virtual function) of the base class. As a configuration parameter the *CS* class just needs the name of the LVOOP class (which is in case of IMAQ classes an identifier for the used camera type) in order execute the correct overwrite VI of the derived class.

The GUI developed together with the camera object is shown in Fig. 2. It also represents an independent object in the *CS* and communicates with the different camera objects over DIM [16] which means that the device and the operation layer can be separated onto two different PCs.

The GUI provides all the necessary features to support the optimization procedure at HITRAP. The aim is to transport the ions available from the ESR through the different preparation steps to the experiments with high efficiency. Especially because of the high charge state this is a very difficult task, but at least the losses due to the

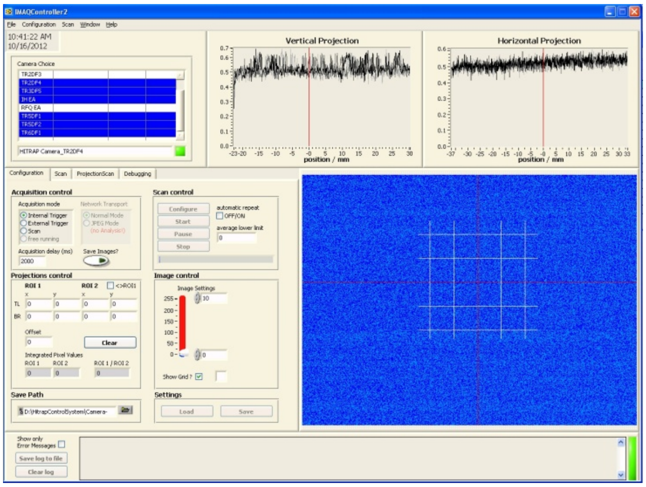

Figure 2: Layout of the Camera readout program. On the right side the current camera picture is shown, whereas the configuration parameters can be entered on the left side. On the upper part of the GUI the horizontal and vertical projections are shown.

transport should be minimized. For this, the cameras are calibrated in order to know when the beam is not only well focussed but also centred at specific focal points in the beamline.
The grid lines visible on the image are drawn with a distance of 5 mm between each other. It is also possible to mark different regions of interest where for example the beam is expected to occur. Parameters, like voltages or timings can be scanned and the resulting intensities can be plotted against the parameter values. Also two-dimensional scans are so possible which should make the optimization procedure much more convenient.

A General GUI for HITRAP

Figure 3 shows the general GUI for one part of the HITRAP beamline. Up to now, the *CS* framework has no built-in GUI. At all experiments either self-made GUIs or the so-called UserGOGs, a scripting method within the *CS* are used. The first solution poses a rather large work for the programmer, whereas the second one is very easy to use but has some severe disadvantages. In fact, the UserGOGs were implemented mainly for debugging purposes so they lack features which are mandatory for SCADA GUIs nowadays, like a proper alarm handling already at the panel of the GUI.

So the idea of the HITRAP GUI is not only to provide a proper GUI for the HITRAP facility but also to make a basis available for other experiments which are at least at the end of the commissioning stage. The responsible programmer just has to prepare a drawing of the setup, copy some controls to the front panel and configure the connection of the GUI with the device objects within the *CS*. In total, this should be much less work than starting from scratch.

SUMMARY

The *CS* framework is used at many different experiments worldwide. All these experiments create add-ons necessary to adopt the core of the *CS* to their needs. For the HITRAP facility at GSI, dedicated to produce heavy, highly-charged ions at very low energies, two new software components were developed: An image acquisition system and a GUI for the facility. Both can easily be integrated in the control system of other experiments so that the highly flexible basis of the *CS* framework is not affected.

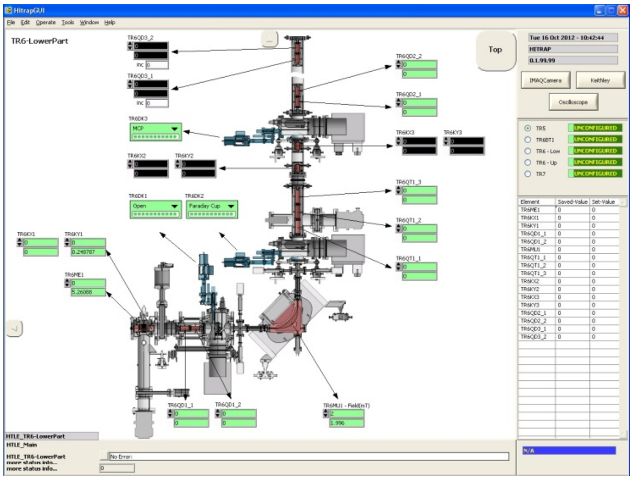
HitrapGUI
File Edit Operate Tools Window Help
TR6-LowerPart
Top
Tue 16 Oct 2012 - 10:42:44
HITRAP
IMAQCamera
Keithley
Oscilloscope
Element Saved-Value Set-Value
No Error

Figure 3: A first version of the general GUI of HITRAP. Shown are controls mainly for power supplies and step motors. It is not the intention of this figure to show details.

REFERENCES

[1] F. Herfurth et al., Hyp. Int. **173** (2006) 93.
[2] H.-J. Kluge et al., Adv. Quant. Chem. **53** (2008), 83-98.
[3] G. Maero et al., `WEM2C06`, Proceedings of COOL2007, Bad Kreuznach, Germany, 130-133.
[4] D. Beck et al., Nucl. Instr. Meth. A **527** (2004), 567-579.
[5] `http://wiki.gsi.de/CSframework`
[6] M. Dworschak et al., Phys. Rev. C **83** (2010) 064312.
[7] M. Mukherjee et al., Eur. Phys. J. A **35** (2008) 1-29.
[8] M. Tandecki et al., Nucl. Instr. Meth. A **629** (2011) 396-405.
[9] J. Ketelear et al., Nucl. Instr. Meth. A **594** (2008) 162-177.
[10] S. Schwarz et al., Nucl. Instr. Meth. B **204** (2003) 507.
[11] J. Repp et al., Appl. Phys. B **107** (2012) 983-996.
[12] H. Brand et al., `TPPA01`, Proceedings of ICALEPCS07, Knoxville, USA, 163-165.
[13] J. Hein et al., Laser and Nuclei, Lecture Notes in Physics 694, Springer, Berlin (2006).
[14] F. Ziegler et al., Nucl. Instr. Meth. A **679** (2012) 1.
[15] H. Brand and D. Beck, GSI Scientific report (2008) 260.
[16] C. Gaspar and M. Dönszelmann, Proceedings of RT'93, 8th Real-Time Computer Applications in Nuclear, Particle and Plasma Physics, Vancouver, Canada (1993).