INTEGRATING A FAST DATA ACQUISITION SYSTEM INTO THE DOOCS CONTROL SYSTEM

A. Agababyan[1], A. Asova[1], G. Dimitrov[1], G. Grygiel[1], B. Fominykh[1], O. Hensler[1], R. Kammering[1], L. Petrosyan[1], K. Rehlich[1], V. Rybnikov[1], G. Trowitzsch[1], M. Winde[1], T. Wilksen[2]
[1] DESY, Hamburg and Zeuthen, Germany, [2] LEPP, Cornell University, Ithaca, NY, USA

ABSTRACT

The VUV-FEL (Vacuum-Ultraviolet Free-Electron Laser) at DESY started recently its operation. As part of the Distributed Object-Oriented Control System (DOOCS) a novel Data AcQuisition (DAQ) System has been developed for this complex facility. To provide full diagnostics of all electron bunches, a fast ADC-based acquisition is used for collecting data from more than 800 channels at sampling rates of 1 to 9 MHz. The system allows data transfer rates of up to 100 MBytes per second to a central server. On this server data from all front end sources is synchronized and used by high level feedback and calculation programs. The data is stored for off-line analysis on a disk cache and eventually written to a tape server. A data volume of up to 70 TBytes per year is expected. The system implements technologies borrowed from High-Energy Physics experiments and integrate them fully into the existing DOOCS control system architecture.

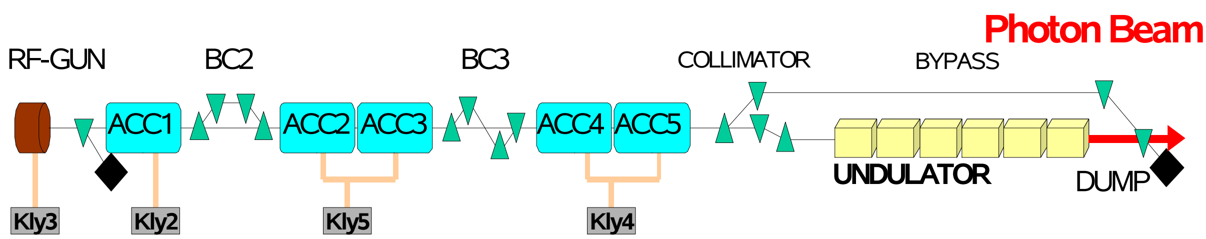

Figure 1: The TTF VUV-FEL: Shown is a schematic layout of the TTF VUV-FEL beamline with its laser-driven RF gun, superconducting accelerator sections ACC1-5 housing 9-cell TESLA cavities and the FEL undulator section providing photon beam for user experiments.

INTRODUCTION

The TTF VUV-FEL[1] is a R&D study exploring superconducting cavity technologies to be used by a future linear collider and a user facility for free-electron laser experiments. During first runs the linac was operated at 0.4 GeV to produce free-electron laser light at a wavelength of 32 nm. Wavelengths down to 6 nm are foreseen in the future. An electron beam of 1 nC charge is accelerated by five cryogenic modules with eight cavities each. The bunch length is shortened by two bunch compressors to provide the required high peak currents for SASE (Self Amplified Spontaneous Emission) in the undulator section that produces the electron laser light. At the end of the linac the emitted light from the undulators is send to experiments while the electron beam is deflected into a beam dump.

The same technologies will be used for the next, larger project, the X-FEL, which is currently under design. The X-FEL will be running at beam energies of 20 GeV and with light wavelengths down to 0.1 nm hence an order of magnitude larger compared to the current implementation.

The TTF VUV-FEL accelerator is operating as a user facility for synchrotron-light experiments as well as a test bed for future accelerator developments including technologies used for the ILC (International Linear Collider) project. Hence an implementation of a control and data acquisition system has to provide reliable operations of both the accelerator and the FEL. Furthermore it needs to be scalable and flexible enough to accommodate any changes in the test setup or additions to accelerator components, diagnostics and user experiments.

FELs require high quality diagnostics. Since the machine is operated in a pulsed mode every single bunch has to be recorded with all beam-based diagnostics and the RF systems. One shot can consist of up to 800 bunches. In future even 7000 bunches will be possible. With a repetition rate of 1 to 10 Hz data from all 800 fast ADC channels of the diagnostics has to be recorded. This would lead to several PBytes of data per year. Be means of filters it is foreseen to limit the total of yearly data to about 70 TBytes. This amount is of the same order as it is produced by a HERA experiment.

DOOCS ARCHITECTURE

The TTF VUV-FEL facility is controlled by the Distributed Object-Oriented Control System (DOOCS) [1]. DOOCS was designed 12 years ago and utilizes an object-oriented representation of control and diagnostic components. Three layers – the device layer, a middle layer and an application layer separate the functions in the control system. The application layer features a variety of DOOCS-own, commercial or free-ware tools as shown in figure 2.

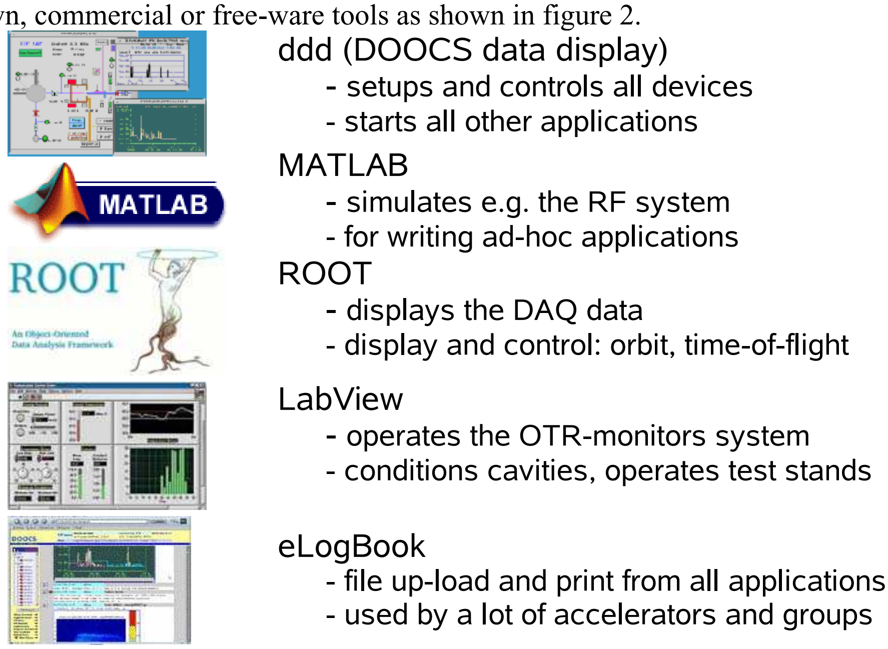

Figure 2: The main application programs of the DOOCS application layer.

Figure 3 shows an example of a ddd (DOOCS Data Display) application. The injector part of TTF is shown with all beam line elements. All elements are animated and can be clicked at to bring up detailed windows as shown for a magnet and a toroid. A click on displayed values pops-up a window with the archived data of the corresponding channel. Further channels can be added to this plot by “drag&drop” operations. One of the displays shows data over hours, the other one a time domain plot with micro second resolution (single bunches).

The device layer comprises all device servers connected to hardware. This includes VME crates with ADCs and other electronics, PLCs controlled via Ethernet and ProfiBus, camera image servers, and several systems provided by collaboration partners. All servers are built with the help of a standard server library. This object-oriented library provides support for various fieldbuses, error and alarm handling, configuration database, archiving of floats and status words and the support for many different data formats to be transported to applications on the network. Devices servers are self-contained and hence fully functional in case of network outages e.g. Interruptions in any of these server operations are met by auto-restart functionality.

Global functions are implemented in the middle layer. Examples are the name server, Finite State Machines (FSM), web servers for e.g. alarm and eLogBook, file servers and utilities. The fast data acquisition system is also a middle layer service.

DOOCS including the data acquisition system is supported on two platforms, Solaris 2.x and Linux as well as to some extent on Windows 2000 and higher.

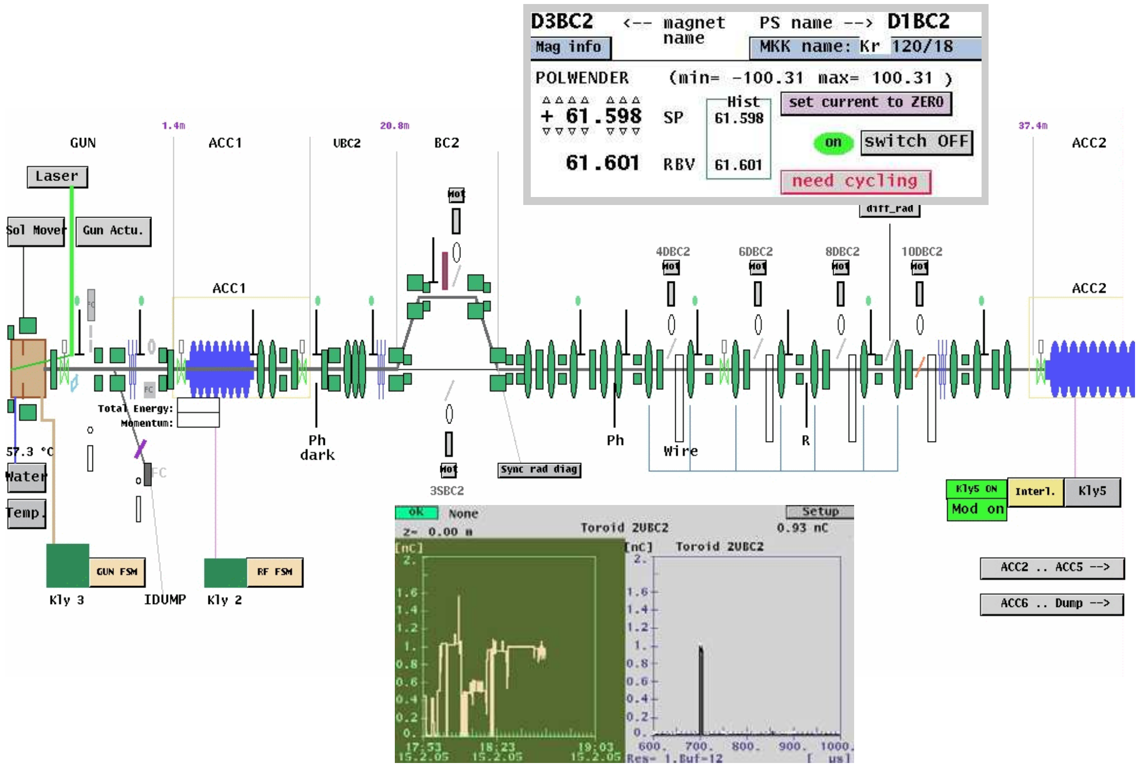

Figure 3: Synoptic display of the TTF injector section: All machine and diagnostic components are clickable and bring up detailed information on the corresponding device as shown here for a magnet (top) and for a toroid (below).

REQUIREMENTS AND GOALS OF THE DATA ACQUISITION SYSTEM

Three main goals are driving the development of the DAQ system:

- To improve the stability and reliability of the linear accelerator operation,
- To facilitate a central data collection for accelerator feedback purposes and other applications requiring immediate access to the data,
- To allow an easy and simple approach for analyzing and specifically correlate data of various components of the machine, diagnostics and user experiments.

The main idea is to collect all relevant measurements from multiple front-end computers into one central shared memory. The shared memory should be managed so that all information is synchronized on a bunch level. The type and amount of data to be stored on disks and later permanently on tape has to be configurable to accommodate higher repetition rates and special operation modes of the accelerator and experiments. The configuration could for example select a fixed rate of machine shots plus all interrupted shots and all wire scanner measurements. Furthermore tools for retrieval and analysis of the data have to be developed. These tools will allow a user to select from certain machine conditions or run periods data records and apply cuts on the data.

INTEGRATION OF A FAST DATA ACQUISITION SYSTEM

A smooth integration of the DAQ system into the DOOCS control system was required since the machine had to be in operation during the test phase of the DAQ. It was a goal to use existing tools and libraries of the control system whenever possible. All servers of the DAQ system are actually based on the DOOCS server library and can therefore be accessed via the network by the same API as all other servers.

Data is pushed from the front-end servers to the collectors within the DAQ server with a total rate of 10 to 100 MBytes per second. About 800 ADC channels for beam position monitors, toroids, beam loss monitors, RF signals and interlocks from distributed CPUs are send by a multicast protocol via the standard network infrastructure. The network interfaces and switches are capable for 1000 Mbytes per second. Special effort was put into the design and implementation of an efficient protocol to transmit the data over the network. The protocol keeps track of lost packets and asks for retransmission of only lost packets. Images from cameras are transmitted in a similar way. Only slow data channels like magnets and temperatures are collected by standard DOOCS API calls with the advantage that all data sources can be included into the DAQ.

On the DAQ sever a buffer manager exploiting UNIX shared memory facility takes care of synchronizing all data delivered by collectors. It buffers the data until all device servers have provided their information for the corresponding linac shot or time slice. This set of data is called an event (record) where the name and concept was borrowed form High-Energy Physics experiments. Even though this “event” is of quite different nature compared to HEP collisions the concept turned out to be quite useful for the upstream processing of the overall data.

At the output of the central shared memory a so called event builder is used to pack the data in different streams of eventually ROOT [4] files. These streams can be configured e.g. for the linac and experiments or special measurements. The ROOT files are stored for a few hours on disk and then transferred to the dCache [5] system provided by the DESY IT group. This system manages the transfer and buffering of the data on huge disk arrays and a tape robot.

A further part in the DAQ system is the run controller. This instance is responsible for the configuration and sequencing of the whole DAQ system. The run controller reads its configurations from a run control database according to the selected run modes and experiments at the VUV-FEL. Commands are sent as XML streams via standard DOOCS API to all contributing servers. One example is the configuration of the devices included in a certain event stream.

After configuration the servers are brought to the proper state by the help of a build-in state machine - the DAQ FSM - which is separated from the linac FSM used for the controls.

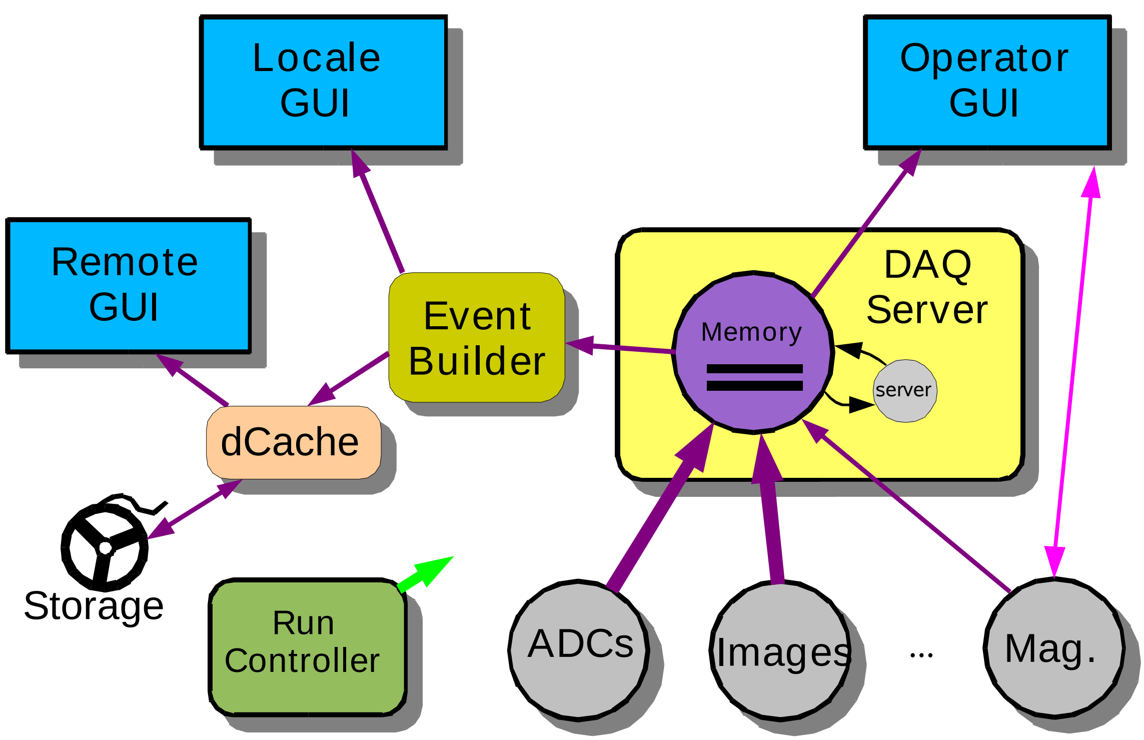

Figure 4: DAQ architecture: Shown are the device servers at the bottom (ADCs, Images, ...) feeding the data into the central shared memory on the DAQ server. The event builder (left) is receiving the data from there, storing it on the dCache system and eventually on tape. The GUIs are standard DOOCS displays.

The architectural design of the DAQ was chosen to be highly configurable and scalable. It is possible to have several DAQ systems by running the corresponding number of instances of DAQ servers, run controllers and event builders simultaneously. Directly connected to the central shared memory are middle layer servers that require reliable and synchronized data from several channels of the e.g. beam diagnostics. Examples for these servers are feedbacks, calibrations and measurement services. Recently a framework was developed to allow MATLAB programs to retrieve data directly from the central shared memory. This framework provides the methods to access data by names to read single values, vectors or images. Data can also be selected to be read by API calls to the control system or as passive properties of the control system to be written in by other applications. Outputs are provided in a similar way. Before the data is delivered to the MATLAB routines filters and exception handling can be applied. All data is seen in MATLAB as a vector of structures.

FIRST RESULTS

During the last months the DAQ system was running to record the diagnostic data of the VUV-FEL. Data was written to ROOT files on disk and has been analysed by ROOT tools. With the help of these tools one can select data from runs and channels and already apply cuts while retrieving the data. The example in figure 5 shows the data of a single toroid with single bunch resolution (1µs), a selected bunch over long time together with a beam position monitor and the correlation of these two channels. In the correlation plots a cut is applied on a range of bunches and a minimum of charge. The ROOT system provides a C++ command interpreter to manipulate data, to analyse it and to create plots. To help users in creating their plots without any programming effort a tool is under development. This so called data browser will be able to select the required data sets and will do the filtering and plotting.

CONCLUSIONS

The TTF VUV-FEL started its first lasing in January 2005. In August this year the first user experiments were taking data with the support of the DAQ system. Fast ADC data and images from a camera server were stored on disk and tape. The software to deliver the data from the front-ends to the central server is ready and runs stable. First middle layer servers are ready and using the data directly from the shared memory. Meanwhile two instances of the DAQ server and run controller are in operation in parallel, one is used for the linac data acquisition, the other one for a FEL experiment. So far no performance limitations have been observed. With the current machine setup data rates up to 30 MBytes/s and image rates of 10 MBytes/s were tested without problems. The additional network load had no negative influence on the TTF operation.

In summary, this is a novel combination and integration of an accelerator controls system and a HEP DAQ system that provides the required infrastructure for the next generation of FEL facilities. First results (e.g. FEL user experiments, high level middle layer servers) have proven that the overall architecture and design is capable to support and sustain the reliable operation of such a complex machine like the TTF VUV-FEL. Furthermore did the chosen design show the scalability needed for the upcoming generation of FEL facilities.

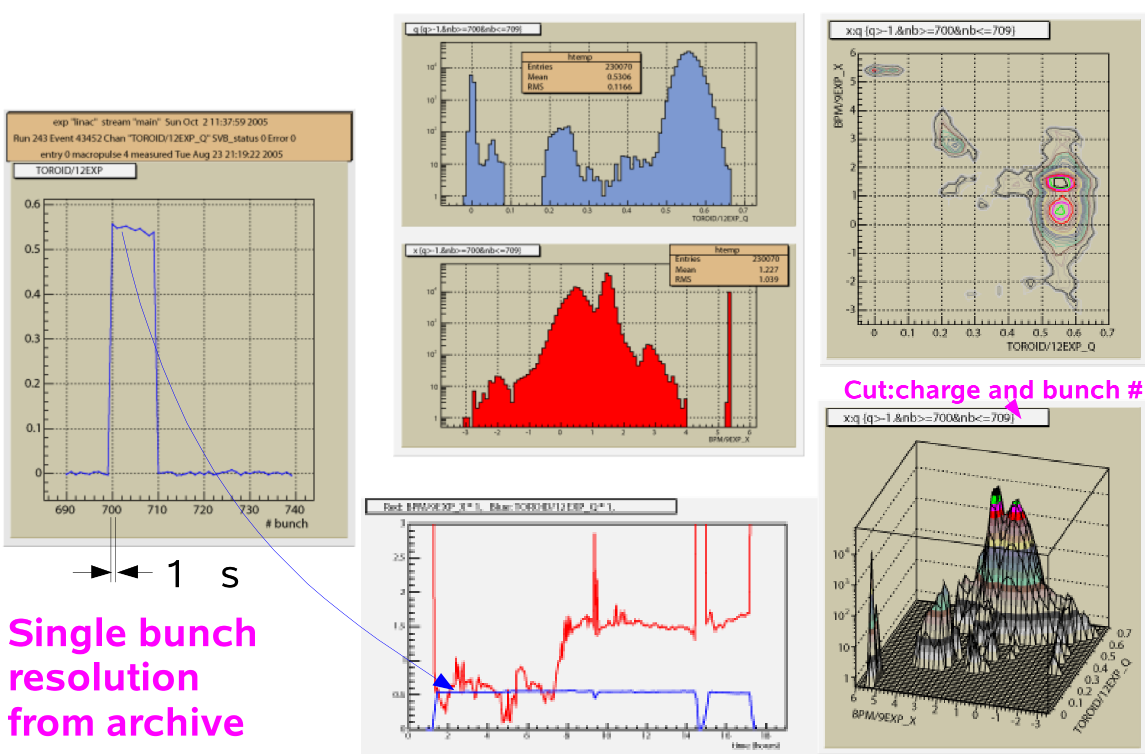

Figure 5: Examples of ROOT displays of the stored data.

REFERENCES

[1] *Rossbach et al, Generation of GW radiation pulses from a VUV free-electron laser operating in the femtosecond regime, Phys. Rev. Lett. 88, 104802 (2002).*

[2] K. Rehlich et al, DOOCS: an Object Oriented Control System as the integrating part for the TTF Linac, Proceedings ICALEPCS '97, Beijing, China. See also http://doocs.desy.de.

[3] A. Agababyan et al, Data Acquisition System for a VUV-FEL Linac, Proceedings PCaPAC2005, 22-25 March, SOKENDAI Hayama, Japan. http://conference.kek.jp/PCaPAC2005/paperindex.html#T2

[4] R. Brun and F. Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings of the AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch.

[5] P. Fuhrmann et al, dCache, a distributed Storage data caching system, Proceedings of CHEP 2001, Beijing, China. See also http://dcache.desy.de.