Image Acquisition and Visualisation in DOOCS and EPICS Environments

Piotr Perek, Jan Wychowaniak, Dariusz Makowski, *Member, IEEE*, Mariusz Orlikowski, *Member, IEEE*, and Andrzej Napieralski, *Senior Member, IEEE*

***Abstract*—The High Energy Physics (HEP) experiments, due to their large scale, required performance and precision, have to be controlled by complex, distributed control systems. The systems are responsible for processing thousands of signals from various sensors of different types. Very often, one of the data sources applied in such systems are visible light/infrared cameras or other imaging sensors, which provide substantial information about studied phenomena. High data throughput for camera systems require dedicated mechanisms for data collecting and processing. Moreover, the images from cameras should be also available to system operator. It needs the support from both operator panels interface and control application which should provide data in the dedicated format.**

The paper presents two different approaches to image distribution, processing and visualisation applied in distributed control systems. Discussed is the issue of support for cameras and image data implemented in the Distributed Object Oriented Control System (DOOCS) and an example control system designed to the needs of image acquisition system on the base of the Experimental Physics and Industrial Control System (EPICS) environment.

***Index Terms*—DOOCS, EPICS, HEP, MicroTCA, distributed control system, image acquisition system, visualisation, cameras, data processing**

I. Introduction

IN today's industrial and scientific environments, processes and investigations require integrated control and supervision solutions. Such means would be responsible for ensuring reliability of the processes, maintaining efficiency, increasing productivity or results, which would also have a reducing impact on costs. Distributed Control Systems (DCS), which answer to this need, provide production plants and research machinery with a net of controlling units of a particular kind distributed throughout. This puts various parts of a supervised system under control of such units, which are connected in a way enabling communication.

A DCS performs control over a system e.g. by accessing sensors, raising alarms on occurrence of abnormal conditions, triggering actions or altering properties of the supervised subsystem devices etc. In this context the advantage of DCS comes from the peer-to-peer-like communication between the controlling units and also the workstations or operator terminals.

P. Perek, J. Wychowaniak, D. Makowski, M. Orlikowski and A. Napieralski are with the Department of Microelectronics and Computer Science, Faculty of Electrical, Electronic, Computer and Control Engineering, Lodz University of Technology, Wólczańska 221/223, 90-924 Łódź, Poland, e-mail: pperek@dmcs.p.lodz.pl

Among various sensors, which the DCS controlling unit can be connected to, dedicated video cameras can be distinguished. This is particularly important in the High Energy Physics (HEP) science context. Data from cameras can be useful in determining parameters of investigated phenomena (e.g. electron beams) or supporting early abnormalities detection.

The paper discusses the experience of the authors with regard to diagnostic image acquisition in HEP experiments at Deutsches Elektronen Synchrotron (DESY) and ITER. The DESY is a German research center for particle physics, with main purposes focused on fundamental research in particle physics and synchrotron radiation. The ITER is international organization appointed to construct an experimental fusion reactor based on the "tokamak" concept. The ITER tokamak will be the world's largest and the most technologically advanced reactor. It was designed in order to produce more power than it consumes what has never been achieved. It is assumed that the ITER will be able to produce 500 MW of energy for 50 MW of input power [1].

Cameras used for the diagnostic image acquisition purpose work under control of the Distributed Object Oriented Control System (DOOCS) and Experimental Physics and Industrial Control System (EPICS) in the above mentioned centers, respectively.

DOOCS is a distributed control system developed by DESY to satisfy the needs of operating the HERA and FLASH experiments hardware. This object-oriented entity is a complex solution for all along the way from the physical hardware up to operator terminals. The system follows the server-client architecture model [2].

The EPICS environment is a set of software tools dedicated for creation of distributed soft-real time control systems for large-scale scientific experiments. It is open source, well-proved and reliable solution, developed by the most significant organizations involved with HEP experiments on the world e.g. ANL, FNAL, KEK, DESY, SLAC etc. [3] For this reason, the EPICS was chosen by ITER organization as the baseline for development of the ITER tokamak control system [4].

A comparison of approaches to cameras-based diagnostics in both control systems is also drawn.

II. Key Elements of DOOCS Architecture

The structure of DOOCS is coined around the notion of reflecting and representing the physical hardware devices in a form of software objects. This results in a model, in which such object - a device server - associated with a given device,

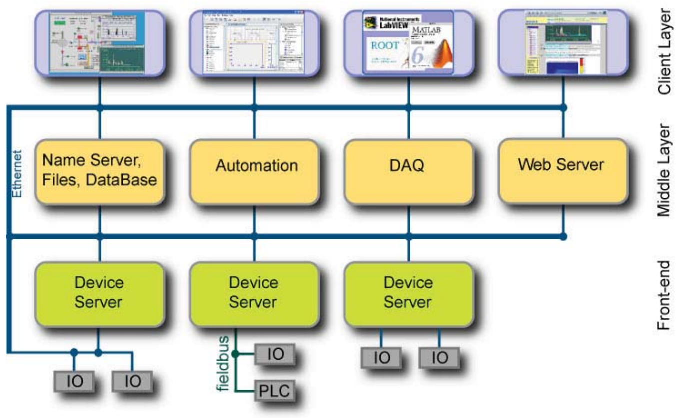


Fig. 1. The DOOCS architecture overview

controls its operation and exposes its properties to other parts of the system [5].

The system is written in the C++ programming language. The aim in the design process was to provide support for creating a device server as an autonomous software entity that would perform complete control over a group of devices of a given type and expose certain sets of its properties to the network. A server, upon start-up, creates an instance of the device representation for each supervised device. Then the properties of each device are made visible in the network [5], [6].

A collection of independent device servers, instead of a central software entity or database, is what conveys the notion of a distributed system. The system is composed of three main layers (Fig. 1) [7]:

- The layer of front-end servers (also referred to as device servers). These servers are what provides the software representation of the devices and makes their properties visible to the network.
- The middle-layer servers. This layer is responsible for facilitating resource-demanding computations, hosting the data processing and acquisition (DAQ) mechanism, databases and other services.
- The client applications layer.

A. Front-end Servers

The front-end servers, composing the lowest layer in the control system, perform direct communication with the hardware. For this purpose, different buses may be used, which include VME, CAN, PROFIBus, GPIB, or Ethernet-based solutions [5].

A device server in DOOCS operates as a UNIX process and represents and communicates with one or more hardware devices of a particular type. All the server classes in the server source code are a specialization of a base class. This base class provides a definition of a standard set of functions and properties that are later a common part of all the servers. It also implements the routines for communication with the network. Furthermore, maintaining the local configuration of a server in a form of a configuration file, as well as data archiving, is also implemented. Access to the local configuration makes it convenient for the server to smoothly recover from possible erroneous operation situations (like crashes etc.). The information exchange between the network and a server properties is provided with dedicated data objects.

B. Middle-layer Servers

The middle-layer, responsible for data processing, but also referred to as the service tier, is where the global services are based. These include the Equipment Name Server (ENS), the Data AcQuisition system (DAQ), the web services and the databases. This layer aims to implement all the complex functionality required in the system. The purpose of this approach is to possibly relieve the client layer applications from stepping beyond the responsibility to present data and interact with operators [5].

The mentioned ENS is responsible for responding to name queries by resolving protocols and host names. The responses from the ENS after requests initiating communication within the system help routing the subsequent calls to the proper destinations and with using proper data exchange protocols (RPC, shared memory, TINE etc.).

The DAQ is an approach for data integration derived from solutions in accelerator controls of experiments from the HEP domain. The intentions towards, and gains from, employing such structure include:

- providing correlation and synchronization of readings from the supervised hardware
- aggregating data obtained from the hardware in a single repository for the use of feedback and further processing
- preserving data for further investigation performed offline
- archiving

The DAQ provides a common communication interface shared by all the services and servers in the middle layer. Device servers are provided with the capability to route the collected data to the DAQ unit. There the data is later available in the central shared memory. Processes like feedback can then take advantage of the availability of this data and operate basing on using the stored values. This way the DAQ may be seen as an efficient data access path for the services and middle layer processes to the device servers. Synchronized data flow coming from the instruments via the device servers and data availability can be taken advantage of by advanced measurement or further processing activities.

C. Client Applications

The client applications layer is what is responsible for representation of the data passing through the control system to the operators. Client programs do not require any particularly sophisticated logic, as the middle layer relieves them from data processing. Thanks to this, it is possible to create the user interfaces with help of graphical editors. Such a tool is provided with the control system. It is a lightweight DOOCS Data Display (DDD) editor. It facilitates creating graphical panel, via which an operator is able to access data, read devices properties and issue commands.

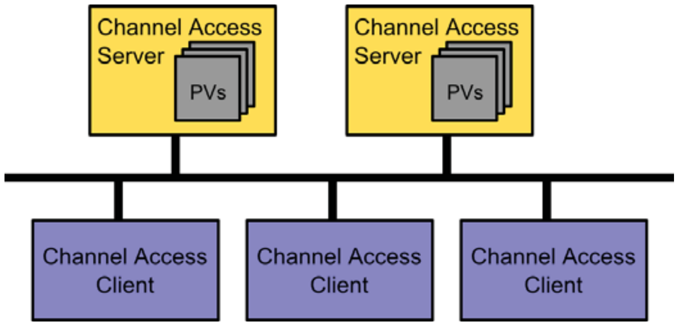


Fig. 2. EPICS architecture

The control system libraries for the client applications also provide integration with engineering and scientific software like LabVIEW or Matlab. What is more, an application programming interface (API) is provided to make possible creating separate applications for cooperation with the control system, in a selection of programming language [5].

III. EPICS-based Control System Infrastructure

The EPICS architecture is based on the server-client model (see Fig. 2). The model allows for existence of several clients and servers in a single control system. Data between servers and clients are exchanged using network-based protocol called Channel Access (CA) Protocol. The fundamental piece of data in EPICS is called the Process Variable (PV). And thus, the servers are programs that define and provide access to the PVs. On the other hand, the clients are programs that require access to the PVs. The entire set of Process Variables provided by all servers in the system are treated as distributed real-time database of information and control parameters.

The special type of EPICS server is Input/Output Controller (IOC). It implements real-time control algorithms and ensures interface to hardware [3]. The IOC provides a set of specific structures that store hardware status and control parameters. These data structures, referred to as records, have particular functionality associated with them (inputs, outputs, calculations, alarm detection, etc.). Different records types have different functions and applications. The data within a records is accessible via PVs. The records are frequently associated with I/O equipment which require unique 'device support'. The EPICS defines dozens of records which allow to exchange data of various types. However, in case of hardware which is not supported it is necessary to reimplement a set of records which allow for communication with the hardware.

The EPICS can also be viewed as a set of tools and applications. Each program which meets the requirements of CA Protocol can be described as compliant with EPICS. Therefore, many applications for various purposes are developed by EPICS collaborators. Each user can select the tools appropriate to their needs or develop their own.

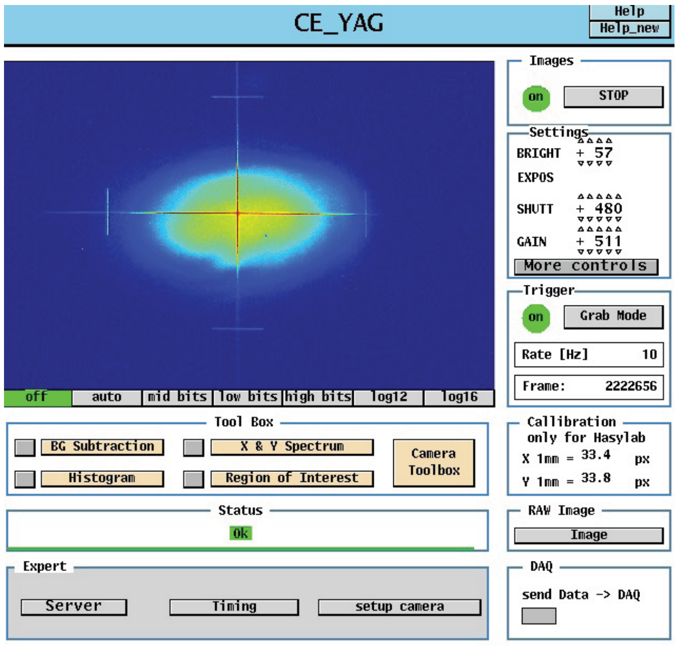
The data that is made available by an EPICS application by the means of records has to be presented to system operator in a graphical form. Furthermore, operator should have possibility to control the system using friendly graphical interface. For this purpose, a variety of display managers for EPICS (e.g. MEDM, EDM, EDD/DM, dm2k) have been developed. One of the most recent graphical user interfaces is BOY (Best OPI, Yet). The BOY is Operator Interface (OPI) environment, as EPICS, recommended by ITER and supported by the Control System Studio (CSS) – tool for control applications development [8], [9].

The BOY environment is a modern user interface which enables fast development of complex user interfaces. Operator panels prepared using BOY connect to an EPICS application and allow for displaying collected data in user-friendly way. They also provide interface for data input.

IV. Cameras Support and Image Data Processing in DOOCS

The Free Electron Laser in Hamburg (FLASH) is a 315 meter-long linear accelerator, currently operating at the DESY research center. It is capable of generating femtosecond-long laser pulses with a repetition rate counted in thousands per second. It is also regarded a test and technology development facility for hardware and software solutions planned to be implemented during construction of its larger version - the European X-ray Free Electron Laser (XFEL).

The operation principle of FLASH is derived from a physical phenomena known as the Self-Amplified Spontaneous Emission (SASE), which occurs as a result of bunch of accelerated particles (electrons) passing through an undulator section, the essential part of the accelerator tunnel. In order for SASE to actually take place, it is required to achieve electron bunches of low energy spread and high peak current. Dedicated equipment for electron bunch diagnostics needs to be employed in order to ensure that certain critical beam parameters remain in their acceptable boundaries. The required shape of an electron bunch is formed by the means of applying longitudinal compression, carried out



BRIGHT + 57
EXPOS
SHUTT + 480
GAIN + 511
Rate [Hz] 10
Frame: 2222656
Callibration only for Hasylab
X 1mm = 33.4 px
Y 1mm = 33.8 px

Fig. 3. An exemplary DOOCS camera GUI panel

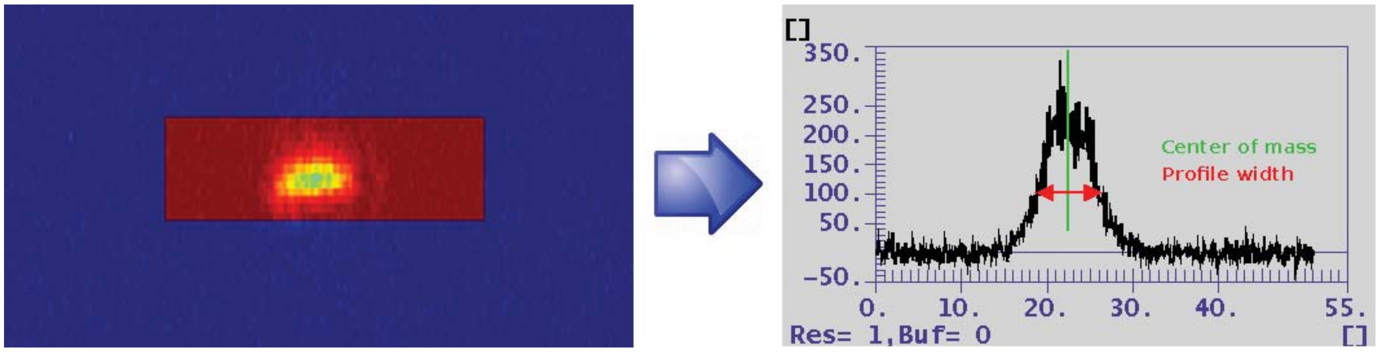


Fig. 4. Image data at the front-end server (left) and operations performed by the middle layer servers (right)

by dedicated magnetic elements. This process occurs after electron bunches are accelerated to gain high energies. The longitudinal charge distribution that a bunch is characterised by after this compression process reflects the quality of the process. Via this, the energy spread and the peak current of a bunch are also described. It is therefore absolutely crucial for these parameters to meet certain requirements. They can be measured and supervised by specialised diagnostic setups operating with the use of charge-coupled device (CCD) cameras. Data, which is gathered from the cameras, produces the possibility to examine the longitudinal profile of electron bunches running through the laser.

The selection of hardware devices that the DOOCS control system communicates with may include cameras of diverse flavours. Their different types and hardware connection standards are also handled by the system with the object-oriented approach. The system provides a base class that implements all the functions that are common to various camera types. This class also contains implementation of parameters that are common to all camera types (like exposure, brightness, etc). From this base class derive specialized classes, that are dedicated and specific to particular cameras. They define interfaces for communication with the hardware via USB, Ethernet, FireWire etc., as well as other camera-specific features (functions, parameters) that do not exist in the base class [10].

The server classes provide complete control over the camera properties and settings. The acquired images can be read directly by the client programs, preserved inn files or stored in the DAQ. There is also support for selected simple operations on images, like background subtraction, histogram calculation, etc. More advanced calculations/transformations can be implemented in middle layer servers.

The camera servers have the possibility to be connected to timing systems, which enables the support for synchronization of the image taking process with the events in the observed phenomenon. Therefore if a camera supports external triggering, images can be taken at moments specified by the timing pulses. The system can be configured in a way that timing-triggered images can contain additional data, like timestamps or event numbers. This enables for correlation of such images with other sorts of diagnostic data [11].

DOOCS provides the IMAGE data type, dedicated to use with cameras. The property of a camera server of the type IMAGE allows to access the most recently taken image. The IMAGE type is composed of a header and a byte array storing the image itself. The header contains image metadata, including resolution, image format, rotation and other. It can also contain a timestamp and event number. The IMAGE property of the camera servers can be directly used by the client applications, but also more advanced processing (like offline investigation or extracting information for feedback actions) can take place. For the latter purpose the DAQ system is a suitable environment.

As mentioned earlier, front end servers for the cameras are sometimes capable of performing simple initial processing of the acquired data. For more advanced or specialised tasks dedicated middle layer servers can be used. The operation of such servers may yield results including image prepared in a specific way for display with using the DOOCS Data Display (ddd) panels. Another output may be specific values, or other information, extracted from the image data.

An exemplary middle layer server operating by the means of utilising image data acquired by a camera concerns processing a longitudinal profile of an electron beam in the DESY FLASH facility. The server pulls data from a front end server communicating with an ICCD sensor-based camera observing diagnostic laser emitted in pulses synchronised with the electron beam. The longitudinal charge distribution of the electron beam is reflected by the laser [12]. The longitudinal profile of the beam is produced by the front end server and made available for further use in the DOOCS system in a form of a data array representing the beam charge against time range. The discussed middle layer server acquires the data and performs processing aimed at removing the background noise from the profile, computing its width or centre of mass (Fig. 4). The obtained values are stored in order to track changes of these beam parameters in time.

V. Epics-related Approach to Image Acquisition

The following section of the paper presents a control application for image acquisition system, which was prepared with using the EPICS framework. Due to the fact, that EPICS does not provide any native mechanisms for image processing, the authors have developed a custom solution for the needs of the described system.

The image acquisition system is a prototype solution developed as part of cooperation with the ITER organization. It was designed as a first step to built complex diagnostic system based on visible light/infrared cameras for ITER tokamak. About 20 cameras comprising the system will observe plasma and internal surfaces of the reactor [13], [14]. Images collected from the cameras will be useful for recognition and classification of thermal events occurring inside the tokamak [15].

The system was designed in order to investigate the performance of the Micro Telecommunications Computing Architecture (MicroTCA) system in applications of high-speed image acquisition and processing. It consists of a central processing unit, an image acquisition module and a high speed camera. The system structure is presented in Fig. 5.

The high speed, high resolution camera has been applied in the system. It is distinguished by following features:

- 3 Megapixel high speed CMOS sensor
- Resolution: 1696x1710 pixel
- 8 bit resolution
- Up to 285 frames/s at full resolution
- Up to 180 000 frames/s with reduced resolution
- CameraLink interface

Data from the camera is collected by Advanced Mezzanine Card (AMC) equipped with frame grabber made in FMC format. The FPGA device on AMC module receives data from the camera and transmits them to the CPU via Gigabit Ethernet (GbE) interface. According to the ITER requirements, images collected from the camera are to be sent to the database as well as presented on operator control panel. For this purpose, a data server application for CPU has been prepared. It is responsible for receiving data from the acquisition module and preparing separate streams for a database and the control application.

Data acquisition server application was designed to operate as a stand-alone daemon application. It communicates with one or more AMC modules using 1 Gbps links. As the database output link speed is 10 Gbps, the system theoretically can handle up to ten modules. The data flow and thread organisation in the software is presented in Fig. 6. Each AMC module data stream processing has been divided into 3 threads, what allows all data can be processed on the fly without buffering. The incoming packets are processed by the receiving thread. It organises communication with the module using dedicated reliable protocol, verifies data checksums, and collects data into ordered data blocks. Next the data are passed to the image frame extracting process, which identifies the individual frames in the data stream. The frame data are then passed to output thread which is responsible for sending images to the database. The thread also decimates the frame stream and passes selected images to another thread responsible for EPICS application communication.

Multithreaded environment with such a high data throughout required some special programming techniques to be used:

- No memory allocation on run-time (all buffers are preallocated during initialisation and reused)
- No memory copy (data copying is minimised by using shared buffers with reference counting)
- Non-blocking operations (data containers are designed to avoid synchronisation objects, where it is possible)
- Custom data containers for interprocess data exchange (FIFOs, lists, vectors)
- Hardware specific solutions (e.g. processor-specific special instruction set)

As a previous work shows [16] they play important role in the improvement of data processing efficiency.

The control application for the system has been prepared using the EPICS framework. The application is an IOC which gives the possibility to acquire data from the camera. Communication with the image acquisition module takes place through the agency of the data server. The EPICS IOC connects to the data server via local Ethernet connection. Information between them is exchanged using higher-layer protocol based on self-defined commands. The commands sent from the IOC to the server allow to control data acquisition parameters e.g. video source, resolution, refresh rate etc. On the other hand, the commands sent from the server inform about the current status of the acquisition.

As it was mentioned, to support new hardware it is necessary to develop 'device support'. This includes records which provide input/output operations. For the described acquisition system, a set of dedicated records was prepared. It includes Binary Input (BI), Binary Output (BO), Long Input (longin), Multi-bit Binary Output (MBBO), String Input (stringin) and Waveform record. The input records allow to read data from the server. On the other hand, the data can be written to the server, and then to the module, using the output records.

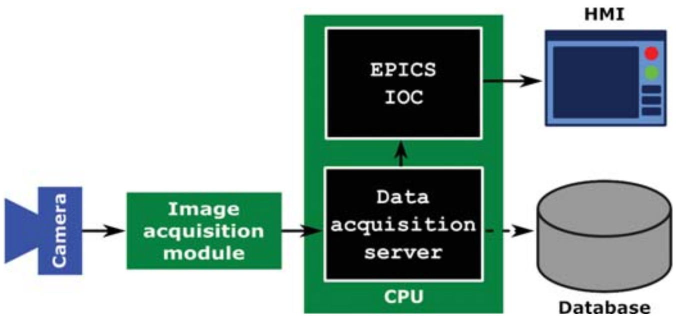


Fig. 5. Structure of image acquisition system

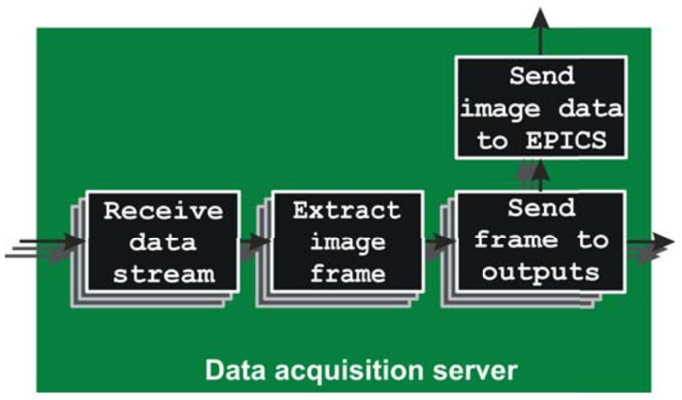


Fig. 6. Data flow in data acquisition server

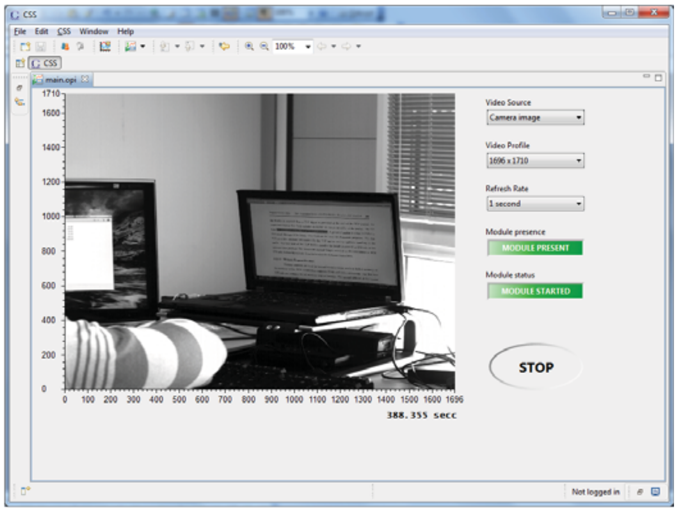
Besides the EPICS application, also a graphical interface for the system operator has been prepared. The operator panel prepared for the image acquisition system using the BOY environment is shown in Fig. 7.

The panel presents the image which was collected from the camera and information about its resolution. Below the image the timestamp is shown. The timestamp is generated by the data acquisition module and written into the image header. The EPICS application extracts this information and makes it available in the form of input record. Furthermore, on the panel there are two status LED indicators. The first one informs whether the data acquisition module is connected to the data server, while the other one indicates the current status of the acquisition process (running/stopped). The remaining components on the panel are responsible for the system control. First of all, the operator has the possibility to start/stop data acquisition. Moreover, on the panel there are also three drop down lists which allow the operator to modify following properties:

- video source (camera images, camera test images, test images generated by acquisition module),
- video profile (image resolution and camera frame rate),
- refresh rate (frequency of sending images from the data server to the EPICS application).

The raw image data in the EPICS application is presented in the form of waveform record. The waveform record can store data of any type supported by EPICS. The data is stored in an array. In case of images required from the camera the waveform record contains unsigned 8-bit values. Each cell of the array represents particular pixels of the image. In OPI environment the images are represented using the Intensity Graph component. It displays the numeric array in a form of 2D plot using colors defined in color map. For images presented in the described system the values from the waveform record are mapped to grayscale. Height and width of the intensity graph is adjusted to the actual image size on the basis of dedicated records ensured by EPICS application.

Due to the fact, that the camera that was used in the system



Video Source: Camera image
Video Profile: 1696 x 1710
Refresh Rate: 1 second
Module presence: MODULE PRESENT
Module status: MODULE STARTED
388.355 secc

Fig. 7. Operator panel for image acquisition system

has bayer pattern filter on the sensor glass lid, it is possible to get color information from the acquired data. Values of red, green and blue components of the pixel color are calculated on the base of neighbouring pixels. For this reason, there has been also prepared an EPICS application that converts data collected from the camera from grayscale to RGB format. In this case, the visualization also differs. The RGB data is provided by the EPICS application using waveform which components are 32-bit wide. The Intensity Graph component on the operator panel presents the image using self-defined color map. It is noteworthy, that this solution of color image presentation is not efficient, because 8 bits from every 32-bits value are lost. Therefore, the authors intend to examine another components both for EPICS application and BOY operator panels, which will be more accurate for the presented system.

VI. Conclusions

The above discussion upon two approaches to the question of taking advantage of the distributed control systems draws significant differences between them. Both the solutions have been present for more than 17 years now, therefore they managed to establish solid foundations for the operation principles they implement. They are, however, not the same in terms of the adopted model. DOOCS can be seen as a concrete example of how a DCS may operate within a complex scientific environment. This is due to the fact that it was developed for the needs of particular facilities, therefore it can directly consider their requirements. EPICS, in contrast, focuses more on providing a generic, not location-bounded framework, enabling to create control systems that would then adhere to specific environment conditions. One more consequence of this is that DOOCS was able to develop and incorporate direct support for image data processing and visualisation, from the hardware level up to software for operator terminals. At the same time a new DCS created from EPICS needs customisation that comprises providing hardware support, appropriate data records or components for operator panels. However, such approach has also advantages, because the users can exchange their solutions with others. Tools developed by one user can be easily applied by all members of community involved with EPICS development.

These essential differences between the discussed solutions affects also the topic of using cameras in diagnostic image acquisition. The DOOCS DCS, developed from the very start with the intention of adhering to particular hardware base, was able to incorporate a generic form of direct support for cameras. Appropriate software structures (e.g data types) are present, ready to be taken advantage of. This yields a basic unified and succinctly defined application interface for communicating with selection of camera types directly at the control system level, ready for its user to utilise.

On the other hand EPICS, as a framework ensuring the generic components for DCS development, does not provide support for camera handling and image processing. For this reason, the user is forced to implement their own 'record support' or to use existing records for image representation, as it has been done in the described image acquisition system. Another approach is application of modules for EPICS developed by other users and dedicated for operation on images. An example of such module is the areaDetector, which supports image acquisition and processing and ensures interface for cameras control.

Using such modules provides a partial remedy to the camera support lack in EPICS-based DCS'es, and it can substantially limit the user experience differences in both systems, when it comes to diagnostic image processing. Also, despite the generic interface being available in DOOCS, the implementation of hardware support, encompassing issues like communication (buses, data transfer protocols) is a requirement in both systems.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Commission under the FP7 Research Infrastructures project EuCARD, grant agreement no. 227579.

REFERENCES

[1] Y. Shimomura and W. Spears, "Review of the ITER Project," *IEEE Transactions on Applied Superconductivity*, vol. 14, no. 2, June 2004.

[2] TESLA Report 2008-03, "LLRF System Components Development (I)," Editor: R.Romaniuk, ISE, WUT.

[3] "EPICS Main Page." [Online]. Available: http://www.aps.anl.gov/epics/

[4] A. Wallander, F. D. Maio, J.-Y. Journeaux, W.-D. Klotz, P. Makijarvi, and I. Yonekawa, "Baseline Architecture of ITER Control System," *IEEE Transactions on Nuclear Science*, vol. 58, no. 4, August 2011.

[5] K. Rehlich, "Status Of The Ttf VUV-FEL Control System," *PCaPAC 2005 conference talk*.

[6] G. Grygiel, O. Hensler, and K. Rehlich, "DOOCS: A Distributed Object-Oriented Control System on PC's and Workstations," *PCaPAC conference*, 1996.

[7] S. Goloboroko, G. Grygiel, O. Hensler, V. Kocharyan, K. Rehlich, and P. Shevtsov, "DOOCS: an Object-Oriented Control System as the Integrating Part for the TTF Linac," *ICALEPCS 97*, Beijing.

[8] "BOY - Control System Studio." [Online]. Available: http://sourceforge.net/apps/trac/cs-studio/wiki/BOY

[9] X. Chen and K. Kasemir, "Boy, a modern graphical operator interface editor and runtime," *Particle Accelerator Conference*, 2011.

[10] K. Rehlich, "Status of the FLASH Free Electron Laser Control System," *Proceedings of ICALEPCS'07*, 2007, Knoxville, Tennessee, USA.

[11] G. Grygiel and V. Rybnikov, "DOOCS Camera System," *Proceedings of ICALEPCS'07*, 2007, Knoxville, Tennessee, USA.

[12] B. Steffen, et al., "A Compact Single Shot Electro-Optical Bunch Length Monitor for the SwissFEL," *Proceedings DIPAC09*, 2009, Basel.

[13] I. Yonekawa, "Data acquisition and management requirement for ITER," *Fusion Engineering and Design*, vol. 43, no. 3-4, pp. 321–325, January 1999.

[14] M. Walsh, et al., "ITER Diagnostic Challenges," *IEEEINPSS 24th Symposium on Fusion Engineering*, 2011.

[15] V. Martin, J.-M. Travere, V. Moncada, and F. Brémond, "Towards Intelligent Video Understanding Applied to Plasma Facing Component Monitoring," *Contrib. Plasma Phys. 51*, no. 2-3, pp. 152–255, 2011.

[16] A. Piotrowski, M. Orlikowski, T. Kozak, P. Prędki, G. Jabłoński, D. Makowski, and A. Napieralski, "Software Optimisation in High Efficiency Data Acquisition Systems," *Elektronika*, no. 12, 2011.

Piotr Perek received the MSc degree in the field of Electronics and Telecommunications at the Technical University of Lodz in 2010. He continues his education as a PhD student at the Department of Microelectronics and Computer Science TUL. His interests include embedded systems, programmable devices and software development. His recent research concerns the development of control and data acquisition systems based on ATCA, μTCA and AMC standards.

Jan Wychowaniak graduated from the Technical University of Lodz in 2010, earning MSc in Electronics and Telecommunications, in the field of microprocessors systems and reprogrammable devices. He is a PhD student at TUL, with the main fields of interests in software development and digital systems. He is involved in development of diagnostics and management facilities for systems based on the ATCA and uTCA architectures.

Dariusz Makowski received Ph.D. degree in electrical engineering at the Department of Microelectronics and Computer Science Technical University of Łódź in 2006. His main areas of interests are digital electronics, embedded systems and programmable devices. He is involved in the development of xTCA standards for High Energy Physics. He is engaged in the design of distributed data acquisition and control systems based on ATCA, μTCA and AMC standards.

Mariusz Orlikowski was born in 1971. He received MSc and PhD degrees in electrical engineering from Technical University of Łódź (TUL) in 1995 and 2000 respectively. He is currently a Associate Professor in the Department of Microelectronics and Computer Science Technical University of Łódź (TUL). His research interests include behavioral modelling using Hardware Description Languages, object oriented programming, distributed programming of data acquisition and processing systems.

Andrzej Napieralski received the M.Sc. and Ph.D. degrees from the Technical University of Łódź (TUL) in 1973 and 1977, respectively, and a D.Sc. degree in electronics from the Warsaw University of Technology (Poland) and in microelectronics from the Université de Paul Sabatié (France) in 1989. Since 1996 he has been a Director of the Department of Microelectronics and Computer Science. Between 2002 and 2008 he held a position of the Vice-President of TUL. He is an author or co-author of over 900 publications and editor of 18 conference proceedings and 12 scientific Journals. He supervised 44 PhD theses; five of them received the price of the Prime Minister of Poland. In 2008 he received the Degree of Honorary Doctor of Yaroslaw the Wise Novgorod State University (Russia).