EUROFUSION WPS1-CP(16) 15512

T Szabolics et al.

Software development for the simultaneous control of ten intelligent overview video cameras at W7-X

Preprint of Paper to be submitted for publication in Proceedings of 29th Symposium on Fusion Technology (SOFT 2016)

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Software development for the simultaneous control of ten intelligent overview video cameras at W7-X

Tamás Szabolics[a], Gábor Cseh[a], Gábor Kocsis[a], Tamás Szepesi[a], Sándor Zoletnik[a]
Christoph Biedermann[b], Ralf König[b] and the W7-X team[b]

[a]Wigner RCP, RMI, Konkoly-Thege 29-33, H-1121 Budapest, Hungary
[b]Max-Planck-Institut für Plasmaphysik, Wendelsteinstraße 1, D-17491, Greifswald, Germany

In the past few years a ten channel video diagnostics system was developed, built and installed for Wendestein 7-X stellarator (W7-X). The system is based on EDICAM (Event Detection and Intelligent Camera) CMOS cameras (400 fps @ 1.3 Mpixel). In the first W7-X experimental campaign (OP1.1) the video diagnostic system was not integrated into the central control and data acquisition system of W7-X, therefore the development of a user friendly interface was necessary to fulfill the following complex requirements: manage the configuration of all cameras (each with four indenpendent region of interests), the measurement cycle and the real-time storage of the collected data to SSDs, and it has to provide a live video stream of all the ten cameras in the control room. The full bandwidth movie stream yields a data rate of approximately 1GB/s for a single camera, therefore downsampled data was used for the live view. The software also has to cope with the huge amount of data when W7-X will operate for up-to 30 minutes. The software package is organized into two stand-alone pieces: VIDACS (Video Diagnostics Data Acquisition and Control Software) being the user interface and EDIDAQ (EDICAM Data Acquistion Software) controlling the individual EDICAMs. This paper will present the detailed design, implementation, testing and the first operation experiences of this software package.

Keywords: EDICAM, Event Detection Intelligent Camera, Software, Video Diagnostics, W7-X

1. Introduction

Wendelstein 7-X (W7-X) is the world's newest and most complex fusion device, therefore it needs high performance diagnostics to be able to meet today's requirements for high precision measurements [1]. One of these diagnostics is the video diagnostics [2] which was one of the few measurements operational from day 1. The ultimate purpose of the video diagnostics system is to monitor the plasma in millisecond or shorter timescales, thus it can provide huge amount of the data (up to Terabytes) with high data rate (950 MB/s) which have to be processed and intelligent decisions have to be made on-the-fly without human interaction for plasma feedback control.

The Event Detection Intelligent Camera (EDICAM) was developed to fulful the above requirements. The observation system has 10 equatorial ports to monitor the whole interior of the machine and in Op1.1 8 of these were equipped with EDICAMs.

In this paper the software solutions of the video diagnostics system, test results and first campaign experiences are described in detail. In the previous works the camera concept and specification was published. [3][4]

2. Requirements

The video diagnostics system had to meet quite a few requirements and these - have to be taken into account during the design process.

The system has to handle 10 EDICAM cameras in a user friendly way, keeping the possibility of integrating other types of cameras if necessary.

All camera data have to be streamed to SSD disks in real-time and have to be displayed in W7-X control room in parallel. Video playback functionality is also essential.

After the measurements all data have to be uploaded to the W7-X archive to provide access for the whole W7-X community. And of course the software has to be reliable and operational from the first operational day of W7-X.

3. Design

Scientific software development is different from commercial software development because it usually has a longer development time with continuously changing design and requirements during the development. Notwithstanding, the software has to be reliable all the time therefore the architecture has to be very well designed. For this purpose the architecture was designed using the Cleanroom development model [5]. This model aims to design the most critical components first thus they can be tested the most.

author's email: szabolics.tamas@wigner.mta.hu

3.1 Components design

Component design's aim is to define the main parts of the software package. In our case we have 10 cameras which have to be controlled by scientists not taking part in the programming. For this reason it has to be user friendly and easily manageable from one place. The best solution is to use server-client topology (see Fig. 1) to allow the users to control the system from one single PC.

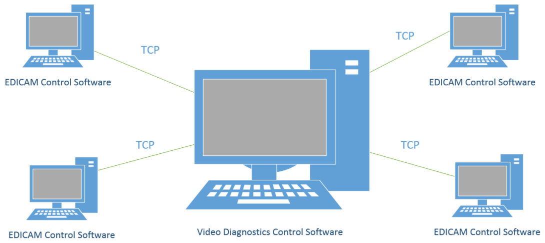


Fig. 1. Topology of the Video Diagnostics Software System

To fulfill the above mentioned requirements the software package was split into two main parts, one is called VIDACS (Video Diagnostics Data Acquisition and Control Software) acting as the central control software with a graphical user interface (GUI), managing all the 10 cameras. The other part is called EDIDAQ (EDICAM Data Acquisition and Control Software) which is responsible for controlling the individual cameras in separate PCs.

3.1.1 VIDACS

VIDACS was designed for two full HD monitors with the aim to provide an overview about all the 10 EDICAMs, showing real-time status and live image preview on one of the screens, and a more detailed (live) image view for four selected cameras on the other screen. For this VIDACS uses more than 40 individual threads in parallel in the background.

3.1.2 EDIDAQ

EDIDAQ was designed to control a single EDICAM camera and send/receive control command, status information, live image preview and measurement data to and from VIDACS. It streams the measurement data to SSD disks in real-time also. It is a console application which runs on every camera control PC, and needs no further user interactions after it was started.

3.2 Interface design

The design process's next step was the interface design which was split into four parts:

- Graphical User Interface (GUI)
- Network interface
- Data Storage Interface
- EDICAM Application Programming Interface

3.2.1 Graphical User Interface (GUI)

The GUI design is only applied for VIDACS because EDIDAQ needs no user interactions. The aim here was to allow the user to control the whole measurement from one place in a user friendly way. The GUI has two full HD windows able to display all the necessary information about the measurement.

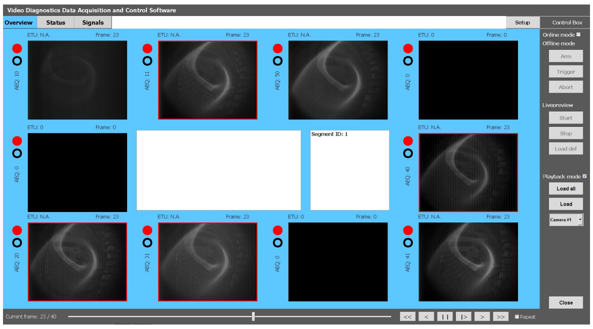
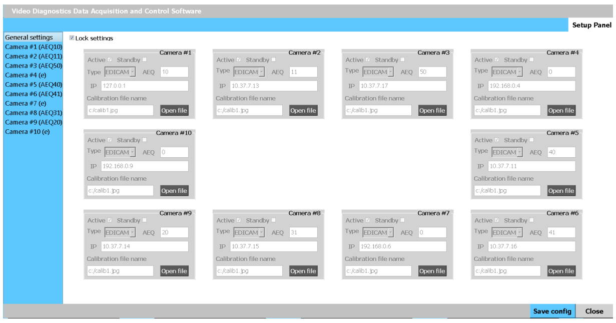


Fig. 2. The main window of VIDACS

As we can see on Fig.2, the VIDACS main window has quite a few different parts. On the top there are tabs for switching between different sets of information such as overview images, camera status and time signals generated from own and other measurements (not yet implemented). The 10 overview frames display camera images in 320x256 resolution, arranged in the order mimicking the experimental layout on W7-X. This makes it easier to identify what can be seen on the images. The status information tab gives information about the cameras, for example electronics temperature or connection status.

The setup button on the top right opens a new window where the user can adjust the settings easily and individually for each camera, or apply them for all other cameras if necessary (Fig. 3). The user can also set which cameras should be active / make measurements in the next sequence.



Video Diagnostics Data Acquisition and Control Software

Setup Panel

General settings
Camera #1 (AEQ10)
Camera #2 (AEQ11)
Camera #3 (AEQ50)
Camera #4 (e)
Camera #5 (AEQ40)
Camera #6 (AEQ41)
Camera #7 (e)
Camera #8 (AEQ31)
Camera #9 (AEQ20)
Camera #10 (e)

☑ Lock settings

Camera	Active	Standby	Type	AEQ	IP	Calibration file name
Camera #1	☑	□	EDICAM	10	127.0.0.1	c:/calib1.jpg
Camera #2	☑	□	EDICAM	11	10.37.7.13	c:/calib1.jpg
Camera #3	☑	□	EDICAM	50	10.37.7.17	c:/calib1.jpg
Camera #4	☑	□	EDICAM	0	192.168.0.4	c:/calib1.jpg
Camera #10	☑	□	EDICAM	0	192.168.0.9	c:/calib1.jpg
Camera #5	☑	□	EDICAM	40	10.37.7.11	c:/calib1.jpg
Camera #9	☑	□	EDICAM	20	10.37.7.14	c:/calib1.jpg
Camera #8	☑	□	EDICAM	31	10.37.7.15	c:/calib1.jpg
Camera #7	☑	□	EDICAM	0	192.168.0.6	c:/calib1.jpg
Camera #6	☑	□	EDICAM	41	10.37.7.16	c:/calib1.jpg

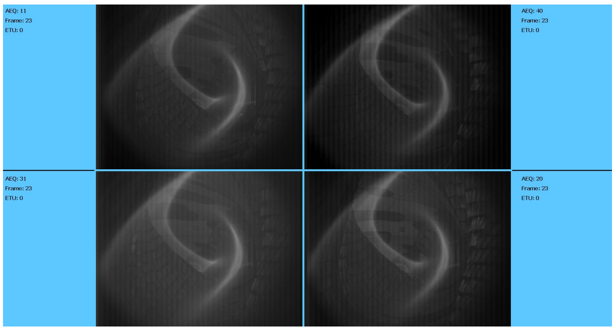
Fig. 3. VIDACS setup panel

author's email: szabolics.tamas@wigner.mta.hu

Buttons for controlling the measurement are located on the right. After pressing the Arm button the system is armed, and it can accept hardware triggers. With the trigger button the user can manually start the system, while the measurement can be stopped with the Abort button if some problem occurred. A live preview can be also started without saving the image data.

On the right under playback mode and on the bottom in the grey area there are the playback controls with which downsampled measurement videos can be loaded and displayed.

On the image display window (see Fig.4) the user can display 4 chosen camera images in a larger size, and this window is being displayed in W7-X control room for all the other scientists.



AEQ: 11
Frame: 23
ETU: 0

AEQ: 40
Frame: 23
ETU: 0

AEQ: 31
Frame: 23
ETU: 0

AEQ: 20
Frame: 23
ETU: 0

Fig 4. Image display window of VIDACS

3.2.2 Network Interface

The network interface is responsible for all kind of communications between VIDACS and EDIDAQ. As it can be seen on Fig. 5, the interface was split into individual channels for different communication purposes, out of which 4 are common. These are the control, status, video stream and data channels. The communication uses TCP protocol which ensures the reliable connection between the server (VIDACS) and the clients (EDIDAQ)

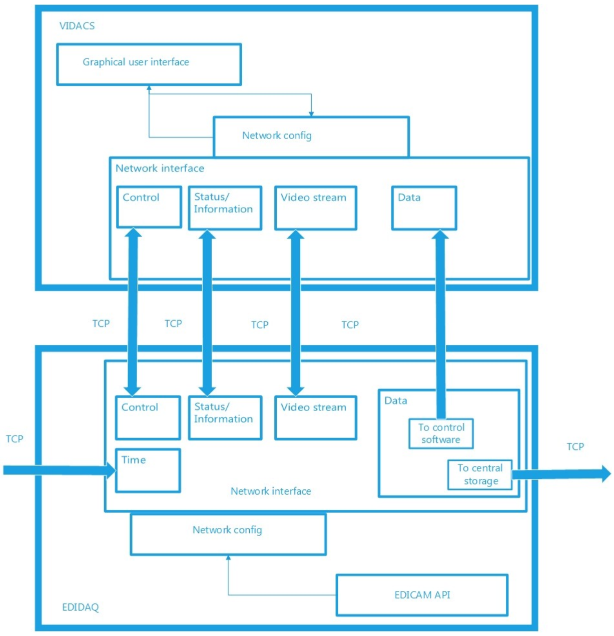


Fig. 5. Network interface block diagram

3.2.3 Data Storage Interface

For storing raw measurement data, the HDF5 file format was used. This file format is optimal for scientific data storage because beside raw image data other signals and measurement parameters can also be stored in the same file. The data storage interface has two sides. On EDIDAQ side the data is written into HDF5 file format and also read back at the end of measurement to extend it with post processed data and to produce downsampled image data in a separate file which is sent to VIDACS after every measurement. On VIDACS side, HDF5 files are only read for displaying downsampled image data.

3.2.4 EDICAM Application Programming Interface

The API was already developed a few years ago, but it is under continuous development as new camera features are being developed.

This interface is providing the basic functions to be able to control EDICAM cameras.

4. Implementation and testing

During the design process the development environment was chosen to be the Qt framework with VS 2013 compiler because we had the experience with Qt framework and MSVC 2013 C++ compiler from earlier development and also the target platform was Windows. Additionally, using the Qt framework the code can be easily migrated to Linux operating system which is part of the further plans.

Scientific software development projects are not like the commercial software development projects because during the design and even in implementation the

author's email: szabolics.tamas@wigner.mta.hu

requirements are continuously changing therefore an optimal model should be chosen to fit this kind of environment. For this purpose Evolution development was chosen after architectural design phase (see Fig. 6).

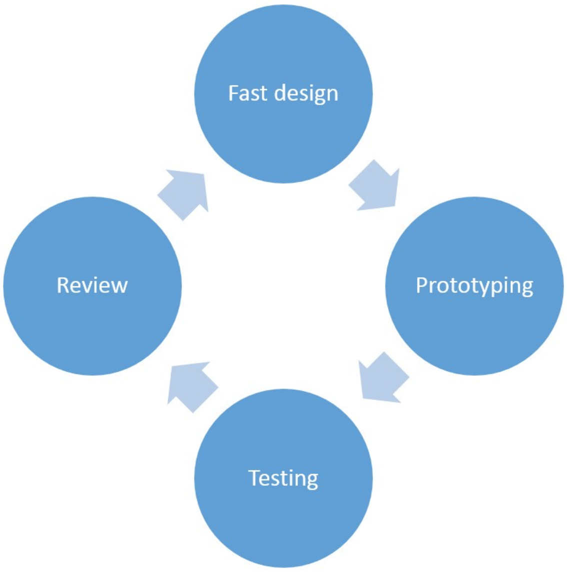


Fig. 6. The evolution development model

This model is taking into account that requirements are changing. After a fast design phase a rapid prototyping phase is coming which is followed by a testing phase to see if the result meets the users and other system parts requirements. After a review the modules can be changed and redesigned and the whole process can be started again until the required results.

After nearly two years of continuous design, development and testing the software package was ready for the first operation of W7-X.

5. First experience

As it was mentioned the software was ready for the first operation campaign, and measurements could be made continuously during the whole OP1.1 phase. [6]

The software package fulfilled all the prerequisites and new intelligent firmware functions could be also tested.

The software system could also capture the first helium and hydrogen plasmas from different views as well.

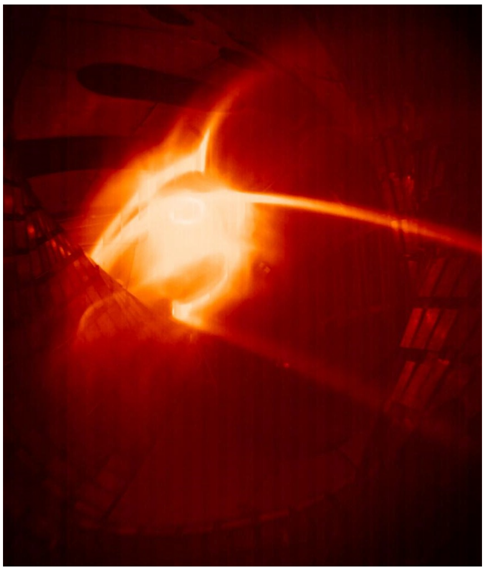


Fig. 7. First helium plasma caught by the video diagnostics system

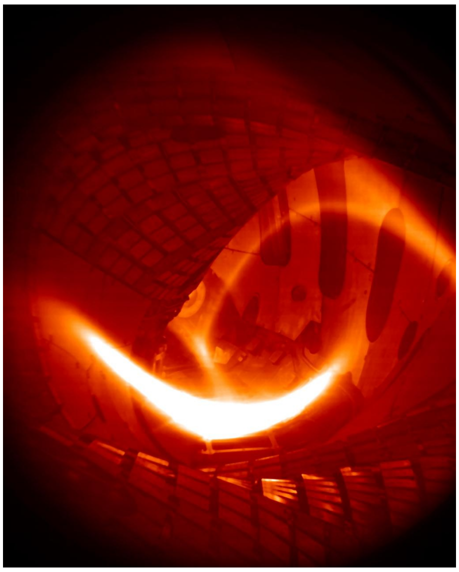


Fig. 8. First hydrogen plasma caught by the video diagnostics system

6. Further developments

The Video Diagnostics Software System has a great development potential because EDICAM cameras are continuously developing. The camera firmware upgrades require continuous software development as well and also there are plans for non-camera specific development such as automatic measurements, which means VIDACS can run automatically: the system can be initialized, armed and data can be saved and uploaded without user interaction. Automatic data upload to the W7-X's Archive was also mentioned which is also planned to be implemented to the next campaign. Preparations for this have already been started during the previous campaign. From the first experience there is a need for some minor user interface changes and upgrades, for example more status information should be displayed from the cameras

author's email: szabolics.tamas@wigner.mta.hu

and there is also headroom for improving the data display after measurements. For the next campaign there are plans for supporting other type of cameras for example Photron fast cameras. Of course continuous bug fixes are also part of the further developments / improvements.

7. Summary

The video diagnostics control software package was designed to control 10 overview cameras, provide live image streams in W7-X control room and save raw measurement data to SSD disks. The developed solution fulfilled the requirements and could be reliable operated during OP1.1. Further improvements and other type of camera support will be provided for later campaigns.

8. Acknowledgement

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

9. References

[1] O. Neubauer et al., I5.1 ,SOFT (2016)

[2] G. Kocsis et al., Fusion Eng. and Design 96–97 (2015) 808–811

[3] S. Zoletnik et al., Fusion Eng. and Design 88 (2013) 1405

[4] T. Szabolics et al., Fusion Eng. and Design 96-97 (2015) 980

[5] H. Mills, M. Dyer, R. Linger, IEEE Software 4 (1987) 19-25

[6] T. Szepesi et al, Europhysics Conference Abstracts Vol. 40A (2016), P4.004

author's email: szabolics.tamas@wigner.mta.hu