

Monitoring and Hardware Management for Critical Fusion Plasma Instrumentation

Paulo F. Carvalho, Bruno Santos, Miguel Correia, Álvaro M. Combo, António P. Rodrigues, Rita C. Pereira, Ana Fernandes, Nuno Cruz, Jorge Sousa, Bernardo B. Carvalho, António J.N. Batista, Carlos M.B.A. Correia and Bruno Gonçalves

Abstract — Controlled nuclear fusion aims to obtain energy by particles collision confined inside a nuclear reactor (Tokamak). These ionized particles, heavier isotopes of hydrogen, are the main elements inside of plasma that is kept at high temperatures (millions of Celsius degrees). Due to high temperatures and magnetic confinement, plasma is exposed to several sources of instabilities which require a set of procedures by the control and data acquisition systems throughout fusion experiments processes. Control and data acquisition systems often used in nuclear fusion experiments are based on the Advanced Telecommunication Computer Architecture (AdvancedTCA®) standard introduced by the Peripheral Component Interconnect Industrial Manufacturers Group (PICMG®), to meet the demands of telecommunications that require large amount of data (TB) transportation at high transfer rates (Gb/s), to ensure high availability including features such as reliability, serviceability and redundancy. For efficient plasma control, systems are required to collect large amounts of data, process it, store for later analysis, make critical decisions in real time and provide status reports either from the experience itself or the electronic instrumentation involved. Moreover, systems should also ensure the correct handling of detected anomalies and identified faults, notify the system operator of occurred events, decisions taken to acknowledge and implemented changes. Therefore, for everything to work in compliance with specifications it is required that the instrumentation includes hardware management and monitoring mechanisms for both hardware and software. These mechanisms should check the system status by reading sensors, manage events, update inventory databases with hardware system components in use and maintenance, store collected information, update firmware and installed software modules, configure and handle alarms to detect possible system failures and prevent emergency scenarios occurrences. The goal is to ensure high availability of the system and provide safety operation, experiment security and data validation for the fusion experiment. This work aims to contribute to the joint effort of the IPFN control and data acquisition group to develop a hardware management and monitoring application for control and data acquisition instrumentation especially designed for large scale tokamaks like ITER.

Index Terms — Fusion, Control, Data Acquisition, Monitoring, Management, Hardware, Software

I. INTRODUCTION

Control and data acquisition (C&DAQ) systems applied to nuclear experiments of large fusion devices play an

important key role to monitor and control the plasma parameters such as temperature, pressure, density, magnetic field, shape and position.

The instrumentation activities can help to prevent fusion experiments to terminate unexpectedly and without notice. Nuclear environment factors like radiation, dust, temperature, voltage and current peaks, distance between the systems and the reactor, signals routing of sensors and actuators, expose the instrumentation to several noise sources and electronic interferences that contribute to increase the hardware failure condition.

This scenario is not desirable since these systems are required to operate in continuous regimes without human local intervention over long time periods.

Instrumentation monitoring and hardware management capabilities are desirable to check the platform health status, mitigate system errors, provide platform inventory information, ease hardware replacement, update system firmware, handle alarms and send notification messages to operators.

A software architecture for monitoring and hardware management is being considered for development taking advantages of the xTCA® [1] high reliability, serviceability and redundancy key features to reduce the platform downtime, give support to data integrity validation and prevent hardware failure.

These features adds robustness to the platforms, reduces maintenance costs and provides instrumentation safe operation.

This paper describes the methodology behind the proposed software architecture aiming high availability to xTCA® C&DAQ instrumentation especially designed for fusion experiments of large nuclear devices like ITER.

II. SYSTEM OVERVIEW

The hardware components are commonly distributed through a computer network organized into clients, servers and xTCA® C&DAQ platforms as depicted in Fig. 1.

Clients communicate with the servers through TCP/IP to send commands and receive the corresponding data.

Servers connect to both clients and xTCA® C&DAQ platform controllers through IPMI over LAN, execute the received commands and respond with the requested data.

Paper submitted for review in July of 2017. IST activities received financial support from “Fundação para a Ciência e Tecnologia” through Project UID/FIS/50010/2013.

Paulo F. Carvalho, Bruno Santos, Miguel Correia, Álvaro M. Combo, António P. Rodrigues, Rita C. Pereira, Ana Fernandes, Nuno Cruz, Jorge Sousa,

Bernardo B. Carvalho, António J.N. Batista and Bruno Gonçalves are with the Instituto Plasmas e Fusão Nuclear, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001, Lisboa, Portugal (e-mail: pricardofc@ipfn.ist.utl.pt). Carlos M.B.A. Correia is with LibPhysUC, Departamento de Física, Universidade de Coimbra, 3004-516 Coimbra, Portugal.

C&DAQ platform controllers perform the required tasks to acknowledge the received instructions, report back in case of error occurrences and, if not, respond with the requested data towards monitoring or the task operation result regarding hardware management.

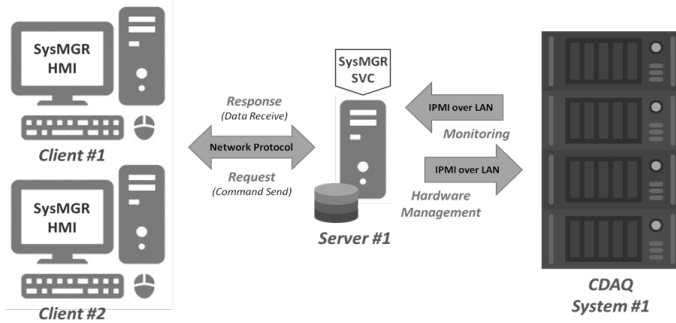


Fig. 1. System Overview.

III. SOFTWARE ARCHITECTURE

The software modules shall be organized according to the diagram of Fig. 2.

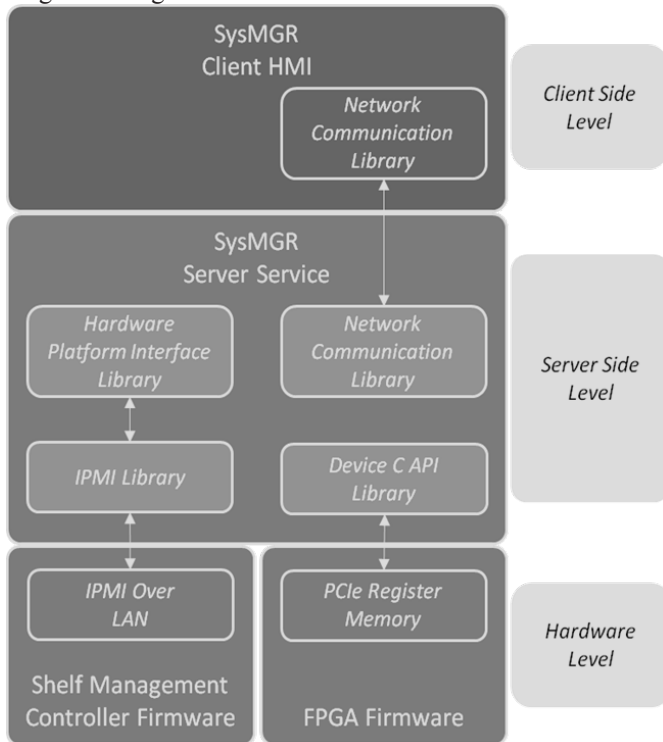


Fig. 2. Software Architecture and Communication Interfaces.

The high layer includes the System Manager (SysMGR) Human Machine Interfaces (HMI) running on the client computer side. The middle layer corresponds to the SysMGR service module running in the server computer side. It receives the requests from the SysMGR client HMI side and communicates with the platform management controllers and processor units through HPI, IPMI over LAN and Device C API libraries. The received data shall be published in the computer network and available to all connected SysMGR client HMIs. The low layer corresponds to the firmware of the platform management controllers and processor units.

This layer receives the instructions sent from the middle layer side, executes them and return the requested data or task operation result.

IV. TECHNOLOGY OVERVIEW

Radisys Platform Management, ScorpionWare® System Manager, FlowPilotTM, NATView and IPMIView are five currently available solutions to monitor and manage the hardware of Advanced TCA based control and data acquisition platforms. Compass System Manager and SpiderWare were also two available projects that were meantime discontinued. Common features offered by these solutions go far beyond sensor monitoring, alarm handling and some limited user, hardware and firmware management capabilities. Cross sensor data from different instrumentation, correlate information that can support data integrity and validation and enable hardware failure prediction implicates a different methodology. This should go through a centralized architecture to monitor environment and instrumentation sensors extended to include hardware failure prediction algorithms to help error mitigation and prevent malfunction scenarios before they could occur, identify possible error sources and notify the system operators in advance. Adding these capabilities to critical instrumentation exposed to several factors that increases the hardware failure condition might contribute to the high availability level required.

V. TARGETING ITER-CODAC COMPATIBILITY

Since the proposed architecture aims xTCA® C&DAQ instrumentation also compatible with ITER device it is desirable to study an architecture implementation that makes usage of the ITER-CODAC framework. This can reduce the time and expertise required to implement a future software solution based on the proposed architecture. The Instituto de Plasmas e Fusão Nuclear (IPFN) has developed a Fast Plant System Controller (FPSC) prototype [2] similar to the fast controllers found in ITER. The prototype is based on the AdvancedTCA® architecture.

The platform supports control and data acquisition boards up to 12 modules and provides allocation for 2 more HUB timing and AMC carrier boards already included in the ITER Catalogue. Using all the knowledge and expertise gathered with the described prototype, to ease the integration of the SysMGR application into the ITER CODAC environment, the software modules should be developed in compliance with the Plant Control Design Handbook (PCDH) requirements and implemented with ITER best programming practices. The HMI running in the client computer side can be developed using the Control System Studio (CSS) Integrated Development Environment (IDE) distributed with the ITER CODAC package.

The software module running in the server computer side can be an Enhanced Physics and Industrial Control System (EPICS) Device Support module developed in C++ programming language based in the Nominal Device Support (NDS) architecture.

This module should include communication protocols based on IPMI over LAN and optionally PCI/PCIe Device C API to provide access to the hardware for monitoring and hardware management.

VI. IPMI FUNCTION LIBRARY

Although IPMI is broadly used to manage server PC systems the majority of them use the IPMITool to perform the required tasks. Since this tool is a command line application and not a function library it is not a simple task to use the included source files into an application. A research study found that it is easier and straightforward to use OpenIPMI. This function library is currently under development and provides interesting examples to help developers to start using it.

VII. EPICS DEVICE SUPPORT

Using the ITER-CODAC MakeBaseApp.pl [epics] script to create an EPICS Device Support source code skeleton to start with, it possible now to include the OpenIPMI function library and use the provided IPMI examples to connect each dedicated EPICS sensor Process Variable (PV) to the corresponding IPMI read function. The device support st.cmd script was configured and prepared to read the server IP address, the username, password, authentication type and access privilege and run an initialization routine to collect and store all the system available information into an array of positions. This information will be useful to read each sensor value, store in the array and use the EPICS PV record udf field and sensor name to find the corresponding sensor value position. After this, the sensor PV is published in the computer network and can be seen by the CSS operator interface panel by pointing the widgets PV name field to the desired monitored system sensor.

VIII. FUTURE WORK

The work developed so far demonstrates that it is possible to integrate the described system architecture with the ITER CODAC framework. The next steps will be focused in the following aspects: (i); extend the described methodology to read all available system sensors and map them into the corresponding EPICS PVs; (ii) configure all sensor alarms and thresholds into EPICS PVs and provide this information to the system operator; (iii) implement task execution whenever an alarm event occurs; (iv) detect all system C&DAQ and timing boards status; (v) implement a system logger to display system messages; (vi) extend monitoring and hardware management capabilities to other xTCA® system; (vii) correlate acquired time stamped data with alarm occurrences to support data integrity and validation; (viii) demonstrate the possibility to implement hardware failure prediction/prevention algorithms; (ix) add, if required, management support to applications whenever a board is added/removed from system via PCIe hotplug and (x) ease the system firmware update procedures.

IX. CONCLUSION

The proposed software architecture aims to provide:

- A centralized software solution to monitor and manage de hardware of several xTCA® C&DAQ instrumentation

extended with environment sensor data can eliminate the need of additional software modules and provide cross data correlation between systems;

- Information about the health status of monitored xTCA® C&DAQ platforms to check if the hardware is operating as expected;
- Monitored sensor data that allows the implementation of algorithms to predict/prevent hardware failure [3] This can prevent occurrence of unexpected scenarios and reduce hardware maintenance costs;
- Correlation between timestamped archived data with event occurrences makes possible to support data integrity and validation algorithms;
- Hardware and firmware maintenance procedures such as replacement of damaged cards via PCIe hot-plug [4][5][6] and system firmware update [7] without powering off the entire platform can contribute to reduce the instrumentation downtime and therefore increase the systems availability;
- Compatibility with ITER CODAC environment can reduce time spent and expertise required to monitor and manage xTCA® C&DAQ instrumentation.

REFERENCES

- [1]. PICMG, PICMG® 3.0 Revision 3.0 AdvancedTCA® Base Specification, March 24, 2008.
- [2]. B. Gonçalves, *et al.*, “Engineering design of ITER prototype fast plant system”, Fusion Engineering and Design, Vol. 86, Issues 6–8, Pages 457-1574, 2011.
- [3]. Doug Turnbull, Neil Alldrin, “Failure Prediction-Hardware or Software”, IEEE Transaction on Aerospace, Vol. 2, Issue 2, Pages 550-557, 1964.
- [4]. PCI Express® Base Specification, Revision 3.0, November 10, 2010.
- [5]. Paulo F. Carvalho, *et al.*, “PCI express hotplug implementation for ATCA based Instrumentation”, Fusion Engineering and Design, Vol. 96-97, Pages 738-741, October 2015.
- [6]. Paulo F. Carvalho, *et al.*, “PCIe Hot-Plug Event Handling Tasks using PICMG Standard Interrupt Mechanism for ATCA based Instrumentation”, Fusion Engineering and Design, Vol. 96-97, Pages 738-741, May 2017.
- [7]. Ana Fernandes, *et al.*, “FPGA Remote Update for Nuclear Environments”, IEEE Transactions on Nuclear Science, Vol. 63, No. 3, June 2016.