Fusion Engineering and Design 127 (2018) 50–53

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

A portable control and data acquisition solution using EPICS, MARTe and MDSplus

G. Manduchi[a,*], A. Luchetta[a], C. Taliercio[a], A. Rigoni[a], F. Sartori[b], A. Neto[b], G. Carannante[b]

[a] Consorzio RFX (CNR, ENEA, INFN, Università di Padova, Acciaierie Venete SpA), Padova, Italy
[b] Fusion For Energy, c/Josep Pla, n° 2, Torres Diagonal Litoral, Edificio B3, 08019 Barcelona, Spain

ARTICLE INFO

Keywords:
Data acquisition
Real-time control
MDSplus
EPICS
MARTe

ABSTRACT

A modern Control and Data Acquisition System (CODAS) must provide an integrated management of supervision, data acquisition and real-time control and a data centric view of the system combining raw, elaborated and configuration data. These requirements have been addressed by the CODAS system of the SPIDER experiment in the ITER Neutral Beam Test Facility. The solution adopted in SPIDER CODAS integrates, by means of few interface components, three frameworks used in nuclear fusion and accelerator communities. A further refinement has recently started, aiming at providing a solution that is complete and generic, and, as such, easily portable to new applications, ranging from small systems to future fusion experiments. The refined system described in the paper will be deployed in the ECT-FALCON Test Facility, used to test the ITER EC launcher components and the European Gyrotron prototype. Its implementation in other test facilities is foreseen, in order to provide a fast track to the development of complete and integrated control and data acquisition.

1. Introduction

Quasi-continuous fusion experiments have introduced a new paradigm in data acquisition systems, moving from strict pulse-oriented acquisition towards a configuration where data produced during the discharge (i.e. the experiment) is streamed into the database and at the same time are made accessible during the discharge itself. This represents an important requirement for modern Control and Data Acquisition (CODAS) systems that must provide integrated management of supervision, data acquisition and real-time control. Another requirement is a data centric organization, where a single database view is used to represent both experiment configuration and experimental data. Using a unified view for experiment configuration is clearly desirable in order to manage the complexity of a fusion experiment (or of an associated facility, often representing a complex experiment in itself). An effective way of deriving information, via associated metadata, on the acquisition chain and configuration for acquired signals is of paramount importance for the proper interpretation of the signals and, in the end, to improve the quality of data analysis.

Two different classes of signals will be managed by the CODAS system: *pulse oriented* and *plant oriented* data. Pulse oriented data are generated during the discharge and are therefore typically associated with a pulse number. Plant oriented data describe the more conventional plant systems such as vacuum and cooling and are normally acquired continuously. In this case data samples will be tagged with absolute time. On the other side tagging pulse oriented data with absolute time would not reflect the normal use case in data analysis because what makes sense for the physical interpretation of signals acquired during plasma discharge is the relative time in respect of the initial pulse trigger. Time representation will then be the offset in seconds from time 0, i.e. the time assumed to be the starting time of the discharge. The relative representation of time permits also an easier comparison and visualization of pulse oriented data from different discharges.

In addition to data acquisition, the CODAS system must provide support for real-time control. It is therefore necessary to let a subset of the acquired signals be also used for real-time control. Other signals generated during real-time control must be recorded, too. These signals will describe the internal states of control and are important for a better understanding of the control behavior as well as for control system debugging [1]. It is clear that an integrated management of acquired data, both pulse and plant oriented, and of real-time control, including internally generated signals, is preferable over the usage of a set of different tools, each supervising a given component and offering different Application Programming Interfaces (API) and interfaces. Other important factors must be taken into account, such as development time and reliability. Building a CODAS system from scratch may not represent the best choice due to its complexity and, above all, the long

* Corresponding author.
E-mail address: gabriele.manduchi@igi.cnr.it (G. Manduchi).

https://doi.org/10.1016/j.fusengdes.2017.12.012
Received 9 June 2017; Received in revised form 8 November 2017; Accepted 11 December 2017

commissioning and debug time.

A possible solution is the use of existing frameworks used in fusion research and, more in general, in scientific experiments. Using a ready solution clearly reduces the development time and, since the framework has already been tested and used, the likelihood of bugs in the code is largely reduced. Moreover, as frameworks are intended to provide generic solutions to be adapted to the target system, they are naturally configurable, as opposed as importing a CODAS system developed and used for a specific experiment.

Three frameworks have been considered here: EPICS [2], MDSplus [3], and MARTe [4]. They cover different aspects of control and data acquisition with limited overlap in functionality. EPICS addresses system supervision but provides a limited support for data acquisition. Moreover, even if originally targeted at plant control, its architectural limits in performance prevent real-time applications [5]. MDSplus is a data management system largely used in fusion community that provides support for data management, acquisition and storage, as well as a multi-language API for data access. MDSplus provides some support for the supervision of the data acquisition process, but not for overall plant supervision. MARTe is a framework for real-time control used in several fusion devices. It addresses coordination of real-time tasks, exploiting multicore architectures and supervising real-time data flow.

In the rest of the paper, the coordinated usage of these frameworks to achieve a complete CODAS system will be described. A similar architecture has already been developed and is currently in use at the SPIDER experiment in the ITER Neutral Beam Test Facility [6], but the solutions adopted there were not general enough for a straight export to other applications. The system presented here is based on the experience gained in that development, but improves its portability, basically achieved by making the connection components more general and configurable than in SPIDER CODAS. The first example of application is the ETC-Falcon Test Facility [7] used to test the ITER EC launcher components and the European Gyrotron prototype and the usage of this CODAS system is foreseen in other facilities.

2. Frameworks

EPICS is used in this context to provide system supervision, including providing mimics for graphical user interface. The EPICS architecture is centered on the concept of Process Variable (PV). A PV is the container of basic information (a scalar, string or vector) that is hosted in an Input/Output Controller (IOC) and that can be read and written anywhere in the system. Within an IOC, PVs can be associated with a driver component that connects the PV to some data source, such as a hardware device, so that plant configuration is exported by a set of PVs that can be read or written for display and control. It is also possible to implement control schemas directly in IOCs, but the use of this feature is not foreseen here, except for the definition of State Machines. A global State Machine supervises the whole system and keeps every component, each supervised by a local State Machine, in step with the global one. A specific language, called State Notation Language (SNL), is defined in EPICS and is targeted at the development of state machines. SNL has been used in this system to develop a general purpose State Machine where both the states and the components are defined in a JSON configuration. A SNL program carries out the behavior of every component, and all the required EPICS support files are generated by a python program that parses the JSON description. It is worth noting that the use of EPICS to implement the general State Machine for system supervision is not mandatory and other approaches can be adopted as well, such as using a PLC to host the top level state machine, leaving EPICS PVs propagate state transitions to the other system components. Besides PV support, EPICS functions are provided by a set of tools in a suite called Control System Studio (CSS). In particular, the BOY and BEAST tools are used in the portable CODAS. BOY provides the constructions of mimic panels for displaying and interacting with PVs, i.e. with the plant configuration. BEAST is a graphical interface for alarm display, still based on the content of a selected PV subset, that may also reflect the occurrence of an alarm generated by a real-time event handled by MARTe. EPICS offers a limited support for the storage of the signals represented by the time evolution of the PV contents. In the presented system overall data management is left to MDSplus, that provides a rich data environment supporting a variety of data types. MDSplus data are centered on the “Expression” concept. Every data item in MDSplus is represented by an expression that is evaluated on the fly every time its value is required. Simple scalars and arrays are expressions as well as data including references to other data items or even requiring the activation of external functions when evaluated. By using expressions it is possible to describe data provenance within the datum itself, without the need of additional metadata. For example, the acquisition chain of an acquired signal can be described by an expression referring the raw ADC data and the way calibration has been performed, with reference to gain and offset data items. Whenever that signal is accessed, that expression will be evaluated returning the actual signal converted samples. MDSplus provides support for data storage on disk and a configurable remote data access layer supporting a variety of communication protocols. MDSplus provides also support for data streaming based on the concept of data segments, i.e. chunks of data that can be appended to existing stored signals during data acquisition. At any time, during streaming, different writer actors can concurrently store signals that can be accessed by reader actors, thus providing full data access support during the discharge itself. Multiple writer configuration is not supported by HDF5, the data system adopted by ITER. The current HDF5 distribution supports multiple reader configuration but poses constraints in the order writer and readers are activated and requires manual flush operations during data storage.

MDSplus does not provide support for real-time applications that are managed in this system by MARTe. Thanks to multiplatform interoperability and the high degree of modularity, MARTe allows a number of Generic Application Modules (GAM) be executed by a set of real-time threads that can be dispatched to the different core of a multicore system. New GAMs can be developed within MARTe in order to provide new hardware support or to implement new control algorithms, leaving the whole management of the data flow among GAMs to the framework itself. MARTe does not provide support for the storage of the data managed by GAMs, that is, input (sensors) and output (actuators) signals and the signals internally computed within or exchanged among GAMs. If acquired, these signals would be of great help in debugging and understanding the behavior of the real-time control system.

EPICS, MDSPlus and MARTe all together cover the requirements for a complete CODAS system and their integration requires the development of few adapter components, mostly addressing data flow among the frameworks. Both EPICS and MARTe handle data that should be stored in a single logical database, represented in our case by MDSplus pulse files. In EPICS data are represented by the evolution over time of the PV contents, stored in MDSplus via the Channel Archiver [8] interface component. This component listens for variations in PV content and every time a PV is changed the new value and the associated time is stored in the pulse file. In this way, the whole PV evolution, i.e. the state of the whole supervised system, is transparently stored in the pulse file. The resulting database is not related to any specific discharge as it reflects the state of plant systems that are always active such as vacuum and cooling. In this context it makes sense to tag every data sample with an absolute time representation. The use of absolute times presents however several drawbacks for pulse-oriented signals for the following main reasons:

- Information carried by signals acquired during plasma discharge in not related in any way to absolute time, rather it depends in the sequence of events that triggered the discharge. Therefore it is more convenient tagging data samples with time offset in respect of a particular trigger assumed as the start of the pulse and associated