USE OF ITER CODAC CORE SYSTEM IN SPIDER ION SOURCE*

A. Barbalace, M. Breda, R Capobianco, A. Luchetta, G. Manduchi, F. Molon, M. Moressa, P. Simionato, C. Taliercio#, Consorzio RFX, Associazione Euratom-ENEA sulla fusione, Corso Stati Uniti, 4 - 35127 Padova - Italy

Abstract

In February 2011 ITER released a new version (v2) of the CODAC Core System. The SPIDER Ion Source experiment is the first experiment planned in the ITER Neutral Beam Test Facility under construction at Consorzio RFX, Padova, Italy. As the final product of the Test Facility is the ITER Neutral Beam Injector, we decided to adhere from the beginning to the ITER CODAC guidelines. Therefore the EPICS system provided in the CODAC Core System will be used in SPIDER for plant control and supervision and, to some extent, for data acquisition. In this paper we report our experience in the usage of CODAC Core System v2 in the implementation of the control and data acquisition (CODAS) system of SPIDER and, in particular, we analyze the benefits and drawbacks of the Self Description Data (SDD) tools.

INTRODUCTION

The ITER plasma will be heated up to ignition by dedicated additional heating and current drive systems based on the injection of radio frequency power (ion cyclotron, electron cyclotron, lower hybrid) and neutral beams in H or D, which are produced by accelerating negative ions that are successively neutralized. As the performance requirements for the ITER Heating Neutral Beam Injectors (HNB) (beam power 16MW, energy up to 1MeV, beam-on time up to 3600s) are far beyond the parameters of existing injectors [1,2,3], ITER will support the development of the ITER HNBs by an ad-hoc ITER Neutral Beam test facility, to be constructed in Padova, Italy. The first test bed of the facility, referred to as SPIDER, will be used for the development of the ITER full-size ion source.

As the final product of the Test Facility is the ITER Neutral Beam Injector, we decided to adhere from the beginning to the ITER CODAC guidelines. Therefore the EPICS system provided in the CODAC Core System will be used in SPIDER for plant control and supervision and, to some extent, for data acquisition. ITER currently does not provide guidelines neither for the management and the storage of experiment data, nor for the implementation of fast control systems. For this reason, two other frameworks will be integrated in SPIDER: MDSplus and MARTe. MDSplus [4] is a framework for Data Acquisition and management largely used in the fusion community. MARTe [5] is a framework for fast real-time control that has been used in several control applications in JET and other fusion machines in Europe. Interfaces between EPICS, MDSplus and MARTe have been developed to provide seamless integration in a complete Data Acquisition and Control system.

An important concept in ITER CODAC is the Self Description Data (SDD), i.e. is the formal description of all the information related to CODAC. Such information includes the definition of the data items, the used hardware, and the software configuration. XML has been chosen to express self-description data, as it is application-independent and is purely data-centred. Even if in principle all self-description data can be expressed by XML text files, in practice more efficient storage strategies are chosen. In particular, the PostgreSQL relational database has been used in ITER to store SDD information. The current version of CODAC Core System includes also an SDD editor, which provides a user friendly interface for the definition of the data items in SDD. Not all aspects of Self Description are currently implemented because the technology of several CODAC components has not yet been chosen by ITER. Currently the SDD editor allows the definition of the functional breakdown, the signal list, and the variables and commands in controllers.

The information stored in the SDD database is used to generate the EPICS configuration files and the PLC Data blocks for data and command exchange between EPICS Input/Output Controllers (IOCs) and the PLCs. It is also possible to export the SDD in XML, and this feature will be used in SPIDER to add configuration generation tools for MDSplus, with the aim of making SDD the sole repository for the entire system configuration.

In the rest of the paper presents the software frameworks used in SPIDER, in particular those components that have been developed to achieve their integration. The applicability of the SDD approach in SPIDER CODAS will be then discussed, as well as the initial experience in the usage of the SDD editor and, more in general, the implications of the Self Description in the integration of different systems.

USED FRAMEWORKS AND THEIR INTEGRATION

Three open-source software frameworks will be used for the implementation of SPIDER Central CODAS and Plant System CODAS: EPICS, MDSplus and MARTe. The choice of three different frameworks is a consequence of the fact that no framework fully covers the requirements for SPIDER CODAS. EPICS addresses overall coordination and slow control, but does not provide a flexible management for data acquisition, nor is it designed for efficient, fast real-time control.

*This work was set up with financial support by Fusion for Energy.
#cesare.taliercio@igi.cnr.it

Sophisticated data management and storage are instead provided by MDSplus, while MARTe provides the functionality required for the implementation of hard real-time control systems.

EPICS

EPICS is a software framework developed to control particle accelerators and other physics experiments. The core of EPICS consists basically of two components: Process Variables (PV) management and Input/Output Controllers (IOC). Control and monitoring in EPICS is achieved by IOCs, where every IOC runs as an application and runs the operations specified by a set of records defined in one or more EPICS databases (EPICS DB). An EPICS DB defines the required actions for control and monitoring. A set of record types is available and can be used for operations such as reading and writing PLC data blocks. Several other tools are part of the EPICS package, including interactive generation of graphical interfaces, alarm handling and data archiving. These tools are built over Eclipse and form the Control System Studio [6] (CSS), available in the ITER MiniCODAC distribution

MDSplus

MDSplus is a software framework for the management of data and the supervision of data acquisition in fusion experiments. MDSplus is centred on the concept of *pulse file* that is a database containing all information which is pertinent to the experiment, including configuration data and experimental results. A large variety of data types is supported by MDSplus including the Signal data type, which describes the time evolution of signals, normally acquired by Analogue to Digital Converters (ADCs). In addition to data describing the experiment parameters and the experimental results, MDSplus provides the Device data abstraction. A Device in MDSplus is represented by a set of data items describing the current configuration of a given hardware component. Multiple instances of the same kind of hardware device will be reflected in the pulse file by multiple instances of data sets, with the same structure, each describing the current configuration of the associated device instance. A set of methods is associated with every device type, corresponding to a routine reading or writing device data items and interacting with the underlying hardware. A Dispatcher tool supervises the execution of all the required operations in the right order during the experiment sequence.

The foreseen usage of MDSplus in SPIDER CODAS is the management of all the configuration and experimental data, including data storage, data access, graphical visualization of acquired signals, and the supervision of the acquisition of data not coming from PLC. Whilst interaction with PLCs in SPIDER will be by means of EPICS IOCs, the supervision of the other hardware devices, such as ADCs and Camera devices will be handled by MDSplus devices, orchestrated by a set of dispatchers, each supervising the operation of a given control unit.

EPICS-MDSplus Integration

As all data dealt with by SPIDER CODAS resides in MDSplus, it is necessary to export configuration and setup parameters to EPICS. This is achieved by letting MDSplus data be exported as PVs via a Channel Access (CA) server. In SPIDER CODAS all the configuration data items held in the experiment model are exported as PV and therefore they are natively available whenever they are required (e.g. to send configuration data to a PLC via the corresponding PLC interface record in the IOC supervising the associated unit).

Exporting configuration data as PVs allows also the interactive development of graphical interfaces for operator displays, carried out by BOY, a component of the EPICS CSS toolset.

Data which are produced in EPICS IOCs will be stored in MDSplus pulse files, too. Such data will be stored in a pulse file by the EPICS – MDSplus Channel Archiver [6], a tool similar in functionality to the BEAUTY tool of the EPICS CSS. BEAUTY provides the monitoring of a selected set of PVs, storing PV values, acquired either at regular rates or whenever their values change, in a relational Database. The MDSplus Channel Archiver has the same interface towards the selected PVs (both tools are CA clients) but stores PV values in MDSplus pulse files. A PV is associated with every EPICS record, and therefore data values acquired from a given PLC, supervised by the corresponding EPICS records, are mapped onto the associated PV variable, and then automatically changed in the trend pulse file.

As EPICS is used in SPIDER CODAS for the overall coordination during the beam sequence, it is necessary to “trigger” the execution of actions supervised by MDSplus at the right time during the beam sequence. As stated before, the specification of the actions to be executed is embedded in the pulse file. The MDSplus Dispatcher will collect such information and will dispatch the corresponding actions whenever requested to execute a given phase in the beam sequence. Such a request must come from an EPICS IOC running the state machine for the SPIDER beam sequence, and will be sent to MDSplus via a dedicated EPICS record.

MARTe

MARTe is a framework for real-time control. MARTe orchestrates the execution of real-time threads, possibly running on different cores of a multi-core system, and the associated data flow. Every thread carries out the execution of a set of Generic Application Modules (GAMs), handling data acquisition from sensors (normally ADC devices), the computation of some control algorithm, and the generation of the reference signals for the plant. MARTe provides an abstraction of the underlying operating systems and therefore can be run on different platforms. New GAMs can be easily integrated in the system, provided they adhere to a given interface, and the whole system is dynamically configured by a configuration file, thus providing rapid integration of new control components in the system.

The functionality of MARTe is partially overlapping with EPICS IOCs, as real-time control configuration can defined as one or more EPICS DBs and run by one IOC. However the footprint of MARTe is much more reduced compared with that of EPICS, and MARTe can handle fast control with a number of inputs/outputs well beyond the limits of EPICS [8].

The foreseen usage of MARTe in SPIDER CODAS is the generation of the power supply reference signals, including the management of breakdowns, requiring the interruption of the voltage generation and its subsequent restart with a given dynamics whenever a breakdown is detected.

MARTe integration

The required functionality for real-time control in SPIDER CODAS (initially limited to the generation of the power supply reference signals and the management of breakdowns) will be carried out by a set of GAMs developed for this purpose. These GAMs will acquire in real-time one or more signals carrying information about breakdowns and will produce the reference signals for the power supplies, which have to be turned off and then gradually on in real-time. In addition to the Input/Output signals these GAMs will likely require some configuration parameters and will write some signals (e.g. describing the internal behaviour of the control algorithm) in the pulse file. The required data exchange will be achieved via MDSplus as all data dealt with by the experiment is hosted in MDSplus pulse files

INTEGRATION OF CODAC CORE SYSTEM

The current version (v2) of CODAC Core System basically consists in the EPICS core and CSS components, and the SDD editor.

The SDD editor provides a graphical interface for the definition of the Self Description database, forcing at the same time a set of rules for naming convention in order to adhere to the ITER standards. The current version of the SDD editor covers only a subset of the Self Description Data, represented by those components which are required for the generation of the configuration files for EPICS and the data blocks for the declared PLCs.

The SDD provides Schemas for Plant Systems and Plant Systems I&C. A plant system is an autonomous part of the ITER plant which implements a given technical function. Self Description for Plant Systems will define those physical components which are relevant for Instrumentation and Control (I&C). The list of signals handled by the system is part of this description. Every signal, characterized by a name adhering to the ITER naming conventions, is described by properties describing the component to which it refers, its physical characteristics, such as the physical units, and its I&C characteristics, such as its digital resolution.

Whilst the Plant System description provides a description of the physical organization of the described Plant, a different view is normally required when considering the logical organization of its I&C. Such organization is captured in the Plant System I&C part of the SDD schema. Plant System I&C description includes the I&C components, which may be represented by:

- Slow and fast controllers;
- I/O boards;
- PLCs

In addition, Plant System I&C Self Description includes all the EPICS Process Variables (PVs) which are involved in control and data acquisition.

In order to ease the development of SDD, the SDD editor provides a simplified interface which hides the internal distinction between Plant System and Plant System I&C. Basically, for every plant system, the SDD editor creates three folders:

- *Signals*: defined by a set of components, each containing zero or more signals. Components describe physical parts of the plant system. Signal definitions here refer to the physical description of the input or output systems handled by the component. All information defined under the Signal folder is saved in the Plant System part.
- *Variables & Commands*: describing the Process Variables used to exchange data with the PLCs and the I/O boards as well as the commands to be issued to the controllers. Variables and Commands are organized in functions, which describe the functional breakdown of the Plant I&C. This information is stored in the Plant System I&C part of Self Description.
- *Control Units*: describing the control components (Fast Controllers, I/O modules and PLCs) involved in the Plant System I&C.

An extensive consistency check is carried out by the SDD editor to ensure the consistency of the entered information, including the adherence to the ITER naming convention and the cross references. For example, a Process Variable (defined in Variables & Commands) must be deployed to an existing target (defined in Control Units), and a signal must be associated either with an I/O module channel or with a PLC.

Once the I&C configuration has been entered and saved in PostgreSQL, the required configuration files can be generated. Currently, the EPICS IOC Database description is generated, which defines the EPICS records corresponding to the declared Process Variables, and the CSS configuration files for the alarm interface (BEAST), the Operator Panel (BOY) and the channel archiver (BEAUTY). In addition the data block files for the PLC configuration are generated.

SDD USAGE IN SPIDER

The ITER approach in the definition of the plant I&C via the SDD has proved extremely useful in our experience, and the availability of a good SDD editor from the very beginning of the system development represents a key factor. Being forced to declare all I&C related information using a single tool simplifies the development process, preventing the scattering of information among different components, and avoids to a large extent errors related to the inconsistent mapping among subsystems. It forces at the same time a strict usage of the naming rules, a very important factor in the development of a large system. Moreover, such an approach allows the automatic generation of a set of useful support tools for system monitoring. As an example, besides the generation of the IOCs based on the current definition of the Process Variables and I/O channels, the CODAC Core system generates also two IOCs for the monitoring of several system parameters such as processor load and the state of the communication link with the PLCs.

We found however two main limits in the current tools for SDD management. The first one is that this approach tends to produce a "flat" model, where a set of records is generated, mapped to the I/O channels (PLCs of other boards) defined in the system. Additional semantics, such as the implementation of specific control in the generated IOCs, or the implementation of synchronizing State Machines can currently be achieved only by manually configuring the EPICS DBs to describe the additional functionality. The drawback of such an approach is that whenever SDD information is changed (e.g. a new signal is added), and the new configuration regenerated, all the additional user-provided configuration information is lost. This is a general problem in the automated generation of software components, which could be solved only by incorporating in the SDD definition also the control semantic. This is however a very ambitious approach because it is very hard to foresee all the possible control approaches, and in practice may lead to a system that is too rigid to be usable in practice.

The second limit is due to the fact that the current version is entirely EPICS centred. While the usage of EPICS has a large consensus for plant supervision and monitoring and, more in general to handle plant (slow) signals, the usage of EPICS for fast control and data acquisition may lead to inefficient solutions. Consider for example the acquisition and the storage in the experiment database of a set of signal acquired by an ADC board with a number of analogue inputs. When supervising data acquisition in EPICS, an Analog Input (ai) record would be defined for the ADC board, whose associated driver handles communication with the ADC hardware. The corresponding PV variable would then be monitored by a BOY Channel Archiver tool, saving every sample in a database. Whenever a new ADC sample is available, the corresponding record will be activated, and the value of the associated PV exported via the Channel Access (CA) protocol over the network and finally received and stored in the experiment database. If the sampling rate is of several kHz and many signals are involved, the system overhead may soon become unsustainable. The approach taken in MDSplus, the Data Acquisition system chosen for SPIDER, is completely different, and a data acquisition thread is created to supervise the acquisition and storage of all the channels of the ADC. Whenever a new set of data samples is available, the thread will directly store the data samples in the pulse file using optimized disk access and network communication, (in the case data access is remote). MDSplus has been in fact designed to handle very high data throughput and is routinely used in fusion experiment where data rates of hundreds of MB/s are achieved.

The SDD approach turns out to be useful even in the implementation of the non-EPICS parts of SPIDER CODAS, and the XML representation of SDD will be used as the source of the structure of the MDSplus pulse files. More in detail, information like the list of signals and the definition of the I/O modules can be retrieved and used to produce a standardized structure of the pulse files. This is achieved by using EXtensible Stylesheet Language [9] (XSL) to define the transformation of the Self Description XML files into the XML files used in MDSplus for the definition of the pulse file structure.

REFERENCES

[1] Y. Ikeda et al., "Present status of the negative ion based NBI system for long pulse operation on JT-60U", Nucl. Fusion 46 No 6 (2006) S211-S219.

[2] K. Ikeda et al., "Recent Progress of Neutral Beam Injector and Beam Emission Diagnosis in LHD", Plasma Sci. Technol. 11 No 4 (2009) 452-455.

[3] A. Staebler et al., "Development of a RF-driven ion source for the ITER NBI system", Fus. Eng. Des., 84 (2009) 265-268.

[4] MDSplus, web page http://www.MDSPlus.org/

[5] A.C. Neto, et al, MARTe: A Multiplatform Real-Time Framework, IEEE Transactions on Nuclear Science, No 57 (2010)479-486.

[6] CSS Home Page at http://css.desy.de/content/index_eng.html

[7] G. Manduchi, A. Luchetta, C. Taliercio, A. Soppelsa, A. Barbalace "New EPICS Channel Archiver based on MDSplus Data System" to appear in IEEE Transactions on Nuclear Science

[8] A. Barbalace, G. Manduchi, A. Neto, G. De Tommasi, F. Sartori, D. F. Valcarcel "Performance Comparison of EPICS IOC and MARTe in a Hard Real-Time Control Application" to appear in IEEE Transactions on Nuclear Science

[9] XSL Home page at http://www.w3.org/Style/XSL