



MDSplus integration at TCABR tokamak: Current status



W.P. de Sá*, G. Ronchi

Instituto de Física, Universidade de São Paulo, Rua do Matão, 1371, CEP 05508-090 Cidade Universitária, São Paulo, Brazil

HIGHLIGHTS

- The implementation of MDSplus in TCABR tokamak, current status.
- Interfaces between the system already installed and the MDSplus.
- Web MDSplus interface.

ARTICLE INFO

Article history:

Received 17 June 2015

Received in revised form 14 May 2016

Accepted 2 June 2016

Available online 10 June 2016

Keywords:

MDSplus

Remote participation

Data acquisition

Control

Tokamak

TCABR

ABSTRACT

Experimental data for the TCABR tokamak is currently stored in MDSplus (Model Driven System Plus) database. The access to the data recorded during the experiments is performed using tools and libraries available by MDSplus system. The MDSplus system is widely used in different physics experiments, especially in plasmas physics and nuclear fusion. This standardized environment enables easy interaction among scientists of different experiments in different countries without the need to understand the particular characteristics of control, data acquisition and analysis, and remote access (CODAS) customized in each laboratory. In the first phase of implementation, intermediate interfaces had been developed between the legacy proprietary system and the MDSplus. In a second phase, the new diagnostic systems had been directly included in the created MDSplus system in the laboratory. After three years of use, the system installed on TCABR proved extremely efficient and significantly increased productivity in data analysis by involved scientists, regardless of whether they are locally at the TCABR, or accessing the system remotely from their home laboratories. The third phase, and subject of this article, are the development and implementation of the following systems: (i) web tools for the visualization of data, integrated with the experiment logbook, (ii) integration of MDSplus with applications (LabVIEW + MDSplus) and newer data acquisition hardware.

© 2016 Published by Elsevier B.V.

1. Introduction

Scientific experiments in different countries, nowadays, are heavily focused on collaborations with exchange of scientists, data remote access and even remote control of the experiments [1]. It is customary that each plasma physics laboratory has a proprietary scheme of the control and data acquisition system. It means that each laboratory has its own way of controlling the experiment and retrieving data from the database. As fusion research relies, to great extent, on international cooperation, these private systems make it difficult to follow the work for visiting and overseas researchers. Thus, standardized computational tools are becoming extremely important.

In this context, the MDSplus (Model Driven System Plus) framework has been developed by several researchers across the globe for data acquisition and storage [2]. It is an open source software tool that allows all experimental or calculated data to be stored into a single, self-descriptive, hierarchical structure.

The TCABR (*Tokamak Chauffage Alfvén Brésilien*, $R=0.615$ m, $a=0.18$ m, $B_t=1.1$ T, $I_p \leq 100$ kA, $n_0 \leq 4 \times 10^{19} \text{ m}^{-3}$) is a small tokamak installed at the Plasma Physics Laboratory in the Institute of Physics of the University of São Paulo, Brazil [3]. This device has contributed with relevant works to the community of plasma physics and nuclear fusion, and the remote participation plays an important role there. In the TCABR, the control and data acquisition system have been updated to be inserted into an internationalized research program [4]. This has been done by employing the MDSplus, which is widely adopted in the fusion community. After three years of use, the system installed on TCABR has proved extremely efficient and significantly increased the productivity of the scientists involved in the experiments, regardless of whether they are locally at the

* Corresponding author.

E-mail addresses: pires@if.usp.br (W.P. de Sá), ronchi@if.usp.br (G. Ronchi).

TCABR or accessing the system remotely from their home laboratories.

The formerly TCABR CODAS system worked in a serialized way: once all signals were available, a main program formatted each one of them properly and sent them to the database. Currently, all diagnostics, or set of diagnostics, operate independently, even the legacy drivers (i.e., low bandwidth VME instrumentation). Each one receives its settings and complementary information from the main controller (or from another computer in the internal network) by messages over TCP/IP and, after the physical trigger, each diagnostics acquires and populates its own MDSplus tree with the signals, setup information, calibration, etc., and then they send the trees to the main database server.

Furthermore (subject of this article), with MDSplus, the web data can be viewed in any modern browser which support HTML5, just using JavaScript API. This approach is a fast way for programming and it is easier to maintain compared to dynamic content pages (with PHP or Python). It is particularly useful in the logbook webpage, where just a few signals are relevant.

The progress with the use and implementation of MDSplus in TCABR is described in this article. In the next section, we briefly describe the old control, data acquisition and analysis, and remote access system up to the first implementation phase of MDSplus. The current phase of the update TCABR CODAS is described in Section 3 (general description of the upgraded system) and Section 4, where we present the visualization system developed. Finally, we present what it is going to be done in the coming years (Section 5).

2. TCABR legacy CODAS

By 2010, the TCABR control system and data acquisition was based on the VME instrumentation standard with Motorola 68040 and Motorola 68060 CPU and OS9 operating system, an ATCA (Advanced Telecom Computing Architecture) crate with Intel Core 2 Duo CPU and Linux operating system, computers with PCI boards and oscilloscopes. ADC modules of 12-bit resolution were available for data acquisition [5]. The VMEs handled most of control and data acquisition, while the ATCA were dedicated for reflectometry diagnostics, and the computers with PCI boards were designed for ECE diagnostics and electrostatic probes acquisition.

The Linux operating system is used in the workstations, server and main controller in the Control Room. The TCABR Acquisition System was centered around a Database Management System (based in MySQL), that allowed to store, manipulate and retrieve data in table structures containing all necessary information for TCABR activities. All software needed for control and data acquisition have been developed locally.

In order to access the data and the information from the database, a set of softwares was written using the C/C++ and Perl languages before implementation of MDSplus. This system was behind the laboratory firewall and the access from outside this firewall was not allowed. Thus, most of the data analyze were performed by the user in the control and data acquisition room.

In 2010/2011 the system was redesigned by adding new features that would function as a bridge between the old system and an implementation using the features of MDSplus system [4]. It worked as a translator that takes the data stored in the old structure and stores in MDSplus pulse files. The MDSplus stores the information, as well as data, in a hierarchical tree structure, composed by nodes and subnodes [6]. In the first approach, the data access could be performed by a thin client (using MDSplus Remote Data Access Protocol on a TCP/IP communication) in any computer inside the university firewall.

3. Current system

In implementing new features and newer data acquisition systems (DAS), one main challenge is to keep backward compatibility, i.e., avoid compromise the legacy drivers still in use in the laboratory. To easier handle this issue, the centralized and serialized CODAS was converted in distributed system. Each diagnostic (or set of diagnostics) manages a local MDSplus server in its controller, where it builds and populates its own tree file. After the end of a plasma discharge and data acquisition, the local server synchronizes the files, via security copy, with the main MDSplus database and the mirrored database (outside the firewall). Although the MDSplus pulse files allow concurrent read and write operations, the adopted approach allows an easier management of data transfer and the read/write permissions across multiple machines and DAS. Furthermore, as they work independently, the latency time to made data available for scientific analysis is reduced, since it decreases the workload and dependency from the VME crate, which is still responsible for the main discharge signals but has a low-speed transfer rate. Thus, while one system may still processing its data, another that already had completed all tasks can be accessed by any scientist to retrieve and analyze such data.

The data structure of TCABR has a main file which works as a central index, linking all the other MDSplus tree files (other plasma diagnostics and post processed data). This central tree stores the basic plasma signals (plasma current, loop voltage, Mirnov oscillations). The data access can be made using signals alias or by the full path of each signal/data, and the user can recover the information from all trees by accessing the main tree, or directly the specific tree. In terms of efficiency, there is virtually no difference in accessing the data directly or through the main tree.

For the current CODAS, the signal database is mirrored via security copy to a server in an external network (internet), as schematically shown in Fig. 1. However, for security reasons, all diagnostics setup and discharge control is still done in the internal network. Inside the internal network, the configuration of each diagnostics can be set up physically at the specific controller, or remotely by means of TCP/IP messages.

Much of MDSplus evolution in the last 5 years was absorbed by the TCABR CODAS system. The main new features can be summarized as [2,6,7]:

- new LabVIEW interface using object-oriented programming (OOP) [8];
- update in the MATLAB interface (from MEX API to Java), allowing OOP also in the MATLAB;
- Python interface, and Web Server Gateway Interface (WSGI) implementations allow the request of data arrays inside any modern browser with JavaScript API.

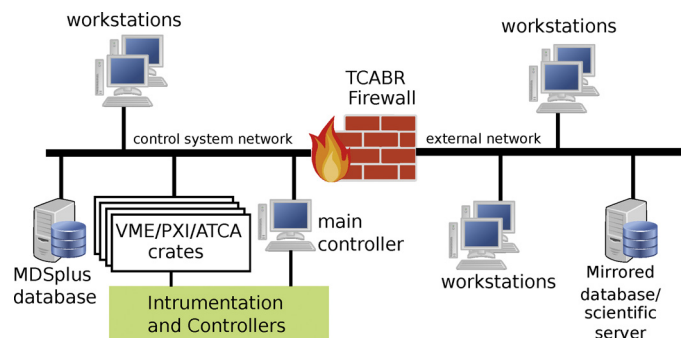


Fig. 1. Simplified TCABR data acquisition and control system.

There was an effort to make the MDSplus available for object-oriented programming since its approach allows to set and retrieve and change MDSplus Tree information in a more systematic way. This approach is used by each diagnostics (to populate the local trees) and in the remote scientific server where the mirror MDSplus database is located. This server runs Debian Linux and is available for scientist and users by secure shell (ssh) or remote desktop protocol (rdp), in a way that all processing can be done in this machine, avoiding larger data transfer, and thus reducing time analysis and network bandwidth.

Other important features developed in this period, such as segmented data for continuous or long discharge duration and the integration of MDSplus with EPICs, are not used in the TCABR.

In terms of hardware, the current CODAS uses most of the former system and it was expanded for more plasma diagnostics, particularly with National Instruments PXI crates, both running Windows or Real Time operating system. These new modules are mainly programmed with LabVIEW, including its MDSplus integration, as explained in detail in the subsection below.

3.1. LabVIEW

LabVIEW is a graphical programming language developed by the National Instruments that has been used for applications in a vast number of instrumentation fields, like optics and plasma physics. It brings together the simplicity to build user interfaces (usually denoted as front-panel) with sophisticated tools for data analysis and control.

In the laboratory, the LabVIEW is used both to interface the diagnostics configuration, for instance, wavelength, gain and sampling rate of the optical diagnostics, as well as to generate the MDSplus structure and to export it to the main database [9].

In the National Instruments Real Time crates running IntervalZero Phar Lap ETS operating system, the integration with MDSplus would be hard since this OS is not supported by MDSplus. In this case, it was chosen to store all data and configuration in the internal hard drive as NI TDMS (National Instruments Technical Data Management Streaming) files which are transferred via FTP to a separate server at the end of acquisition. This intermediate server runs a Python application that reads and build the MDSplus tree for those data. In the end, those new file are synchronized with the MDSplus databases. This same approach is also used in others diagnostics where the OS can not easily build the MDSplus trees.

4. Data visualization

New ways to visualize the MDSplus data have been also implemented, that are particularly useful for fast characterization of the plasma discharge. MDSplus now brings built in a Web Server Gateway Interface (WSGI) that allows querying information from the plasma discharge and signals recorded in the MDSplus trees with simple JavaScript on an HTML page [11]. More specifically, web pages using Asynchronous JavaScript and XML (AJAX) technology can use XMLHttpRequest API access data or events of MDSplus trees through the WSGI running in the database server. The major advantage of this approach is the simple and fast implementation, by the cost of signal formatting and plotting done in the client side, that may cause and impact in the performance for large dataset. This application was first show with the WebScope [11], where this new technology is explained in details. The WebScope is a web-based tool for interactive data visualization that recreates an environment similar to the jScope (a Java application for this purpose).

An application in the TCABR of this tool is the use of AJAX in the logbook page to show the main signals of each discharge, as presented in Fig. 2. In this case, the JavaScript makes XML request to the

WSGI, using the MDSplus API, to get the signal data, and then, plotting them using the Dygraph visualization library [11]. This graphic library allows fast zooming and the recovery of specific time and signal value highlighted by the mouse, that would not be possible in a static image. On the other hand, it has higher network throughput, since the dynamics graphics are generated from data, which may be large, and that are not cacheable in the browser. The time to render the whole page is of order of hundreds of milliseconds.

It should be that there are alternatives implementations that also focus on philosophy of internationalization of access to data. In terms of data visualizing, another approach was developed at the H1DS laboratory, using MDSplus and Python with a Django (a web framework written), to implement a Representational State Transfer (REST) architecture to access the plasma signals and to make them available in any browser, with SVG graphics [12].

5. Future enhancements

The use of MDSplus in the TCABR is now quite consolidated for storage and retrieval of tokamak discharge data, by using local file trees for each diagnostic, and then synchronized them with the main MDSplus database, or by an intermediate process, if the MDSplus is not available for the DAC operational system, as previously explained.

By now, the main concern is to define and apply a comprehensive and unified protocol to configure and control each diagnostic via HTTP. Although the current approach with TCP/IP messages allows remote set-up, in Python or LabVIEW, for example, one is required to have a specific software to handle this communication. On the other hand, any internet browser available today in most computers or tablets could be used for monitoring/configuring the diagnostics with an HTTP implementation. This approach can improve the diagnostics maintenance, and reduce eventual system downtime. For instance, we have two distinct case under development: interferometry diagnostics running Linux and controlled by Python software, and probe diagnostics, in NI Real Time DAC.

In the reflectometry diagnostics, a main Python software controls the acquisition setup and configuration. This software has one thread to handle TCP/IP communication to receive the configuration data. Another Python instance runs a Django web application to generate the HTML and to GET/POST JSON files with the diagnostics setup. This application, thus, connects and transfers this file to the main control software.

In the second case, as mentioned before, the Python program generates and populates the MDSplus tree for the data reacquired by the NI Real Time DAC, mainly from electrostatic probes. The DAC settings, though, are deployed by virtual instrumentation (VI) written in LabVIEW language that runs in the real time controller. In this case, we are working to create a similar web application of the reflectometry, to allow the set-up of this diagnostic also by HTTP protocol, so that the operator or the diagnostics leader can be able to configure properly the system in a user friendly way.

6. Summary

Remote participation and remote access to experimental data is the main factor in the collaborative work between scientists in different laboratories around the world. The implementation of MDSplus tools system has become a standard in plasma physics and nuclear fusion. Hence reducing significantly development tools for Control, Data Acquisition and Data Analysis and Remote Access.

Here we presented the current integration of MDSplus in TCABR, where we developed a modular approach for several distinct controllers and diagnostics. From the former system, we keep a central index (main tree) that allows list all signals and pre/post-processed

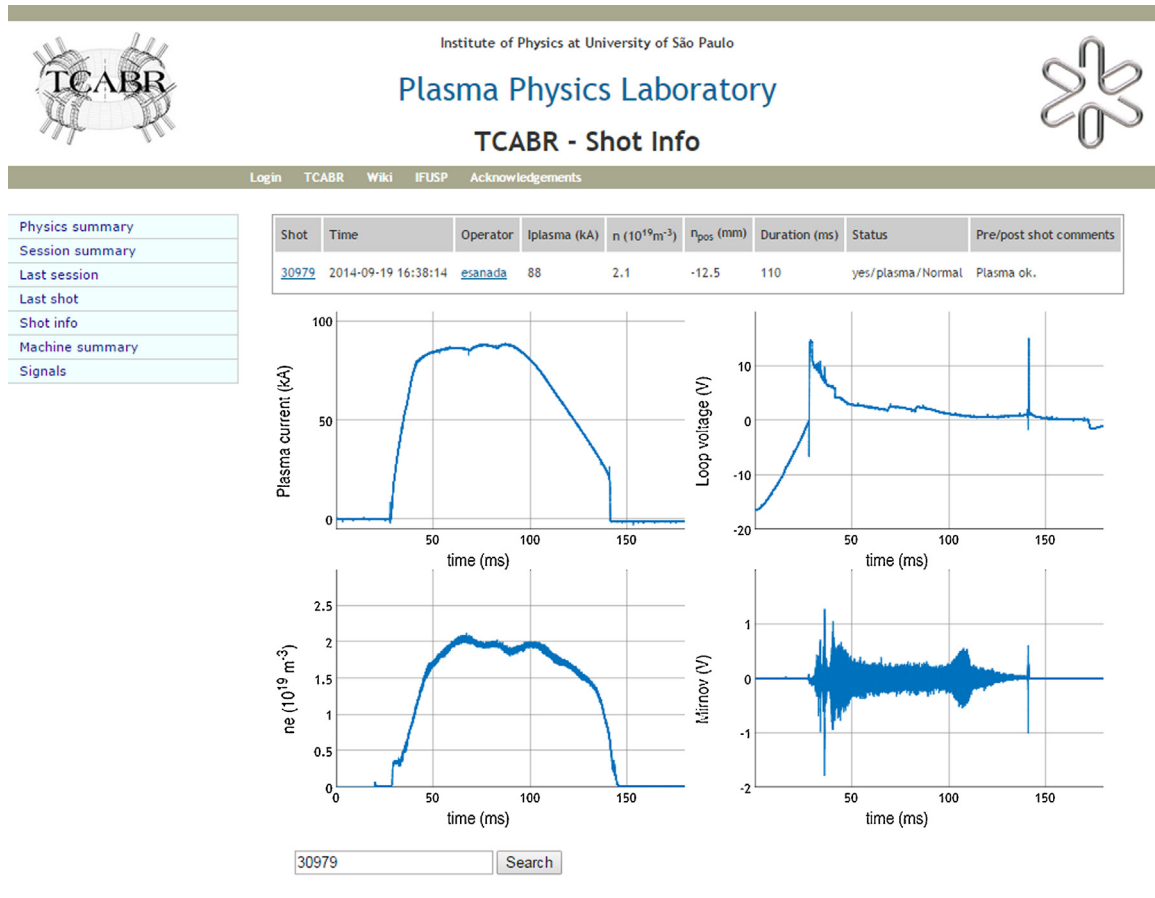


Fig. 2. Logbook visualization of 4 main signals (plasma current, electron line average density, loop voltage and Mirnov oscillation), build with AJAX.

data for a specific shot, but now it is composed also by subtrees related to the several diagnostics. When the diagnostic controller is capable of run the MDSplus tools, it handle its own tree. Otherwise, the data are stored in a native format and then transferred for another server, where another program generates the MDSplus tree and properly integrated it to the main system.

It was also explored new technologies, in particular in web technology that provides tools to simplify the data visualization that gives an important first response in data analysis for a researcher, which can reduce significantly the amount of time taken to find plasma discharges with a specific feature.

Following the same philosophy, the integration and interfacing of diagnostics with TCP/IP messages simplifies and reduces the time for setting up the diagnostics, avoiding the technician/researcher to go to the machine every time to changes its settings.

Acknowledgements

This work has been supported by The National Council for Scientific and Technological Development (CNPq), and by grant 11/50773-0 of São Paulo Research Foundation (FAPESP).

References

- [1] V. Svoboda, J. Kocman, O. Grover, J. Krbec, J. Stöckel, Remote operation of the vertical plasma stabilization @ the GOLEM tokamak for the plasma physics education, *Fusion Eng. Des.* 96–97 (2015) 974–979.
- [2] G. Manduchi, T. Fredian, J. Stillerman, Future directions of MDSplus, in: Proceedings of the 9th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, *Fusion Eng. Des.* 89 (5) (2014) 775–779, <http://dx.doi.org/10.1016/j.fusengdes.2014.02.012>.
- [3] R. Galvão, C. Amador, W. Baquero, F. Borges, I. Caldas, N. Cuevas, V. Duarte, A. Elfimov, J. Elizondo, A. Fonseca, T. Germano, G. Grenfell, Z. Guimarães-Filho, J. Jeronimo, Y. Kuznetsov, M. Manrique, I. Nascimento, C. Pires, P. Puglia, A. Reis, G. Ronchi, L. Ruchko, W. de Sá, R. Sgalla, E. Sanada, J. Severo, V. Theodoro, D. Toufen, Report on recent results obtained in TCABR, *J. Phys. Conf. Ser.* 591 (2015) 012001, <http://dx.doi.org/10.1088/1742-6596/591/1/012001>.
- [4] W. de Sá, Tokamak TCABR: Acquisition system, data analysis, and remote participation using {MDSplus}, in: Proceedings of the 8th {IAEA} Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, *Fusion Eng. Des.* 87 (12) (2012) 2199–2202, <http://dx.doi.org/10.1016/j.fusengdes.2012.04.022>.
- [5] A. Fagundes, W. Sá, P. Coelho, TCABR data acquisition system, *Fusion Eng. Des.* 48 (1–2) (2000) 213–218, [http://dx.doi.org/10.1016/s0920-3796\(00\)00129-0](http://dx.doi.org/10.1016/s0920-3796(00)00129-0).
- [6] G. Manduchi, Commonalities and differences between MDSplus and HDF5 data systems, *Fusion Eng. Des.* 85 (3–4) (2010) 583–590, <http://dx.doi.org/10.1016/j.fusengdes.2010.03.055>.
- [7] G. Manduchi, T. Fredian, J. Stillerman, MDSplus evolution continues, *Fusion Eng. Des.* 87 (12) (2012) 2095–2099, <http://dx.doi.org/10.1016/j.fusengdes.2012.01.032>.
- [8] G. Manduchi, E.D. Marchi, A. Mandelli, A new LabVIEW interface for MDSplus, *Fusion Eng. Des.* 89 (6–8) (2013) 1196–1199, <http://dx.doi.org/10.1016/j.fusengdes.2012.12.013>.
- [9] G. Ronchi, J.H.F. Severo, W.P. de Sá, R.M.O. Galvão, Data acquisition and automation for plasma rotation diagnostic in the TCABR tokamak, *J. Phys. Conf. Ser.* 591 (2015) 012007, <http://dx.doi.org/10.1088/1742-6596/591/1/012007>.
- [10] G. Manduchi, T. Fredian, J. Stillerman, A new web-based tool for data visualization in {MDSplus}, *Fusion Eng. Des.* 89 (5) (2014) 780–783, <http://dx.doi.org/10.1016/j.fusengdes.2014.02.026>.
- [11] Dygraphs Javascript Visualization Library, <http://dygraphs.com>.
- [12] D. Pretty, B. Blackwell, HIDS: a new web-based data access system, *Fusion Eng. Des.* 89 (5) (2014) 731–735, <http://dx.doi.org/10.1016/j.fusengdes.2014.03.025>.