Fusion Engineering and Design 89 (2014) 558–562

Contents lists available at ScienceDirect

Fusion Engineering and Design

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

The web-based user interface for EAST plasma control system

R.R. Zhang[a,*], B.J. Xiao[a,b], Q.P. Yuan[a], F. Yang[a,c], Y. Zhang[a], R.D. Johnson[d], B.G. Penaflor[d]

[a] Institute of Plasma Physics, Chinese Academy of Sciences, Anhui, China
[b] School of Nuclear Science and Technology, University of Science & Technology of China, Anhui, China
[c] Department of Computer Science, Anhui Medical University, Anhui, China
[d] General Atomics, DIII-D National Fusion Facility, San Diego, CA, USA

ARTICLE INFO

Article history:
Received 24 May 2013
Received in revised form 10 February 2014
Accepted 26 February 2014
Available online 19 March 2014

Keywords:
EAST plasma control system
Web-based user interface
Remote participation

ABSTRACT

The plasma control system (PCS) plays a vital role at EAST for fusion science experiments. Its software application consists of two main parts: an IDL graphical user interface for setting a large number of plasma parameters to specify each discharge, several programs for performing the real-time feedback control and managing the whole control system. The PCS user interface can be used from any X11 Windows client with privileged access to the PCS computer system. However, remote access to the PCS system via the IDL user interface becomes an extreme inconvenience due to the high network latency to draw or operate the interfaces. In order to realize lower latency for remote access to the PCS system, a web-based system has been developed for EAST recently. The setup data are retrieved from the PCS system and client-side JavaScript draws the interfaces into the user's browser. The user settings are also sent back to the PCS system for controlling discharges. These technologies allow the web-based user interface to be viewed by authorized users with a web browser and have it communicate with PCS server processes directly. It works together with the IDL interface and provides a new way to aid remote participation.

1. Introduction

EAST is an experimental device aiming at steady-state plasma operation for fusion research. Its plasma control system [1,2] (EAST PCS) is adapted from DIII-D plasma control system architecture [3–5] and continues to expand and improve its capabilities for supporting the increasing experimental needs. EAST PCS is a mini Linux based cluster with one host and three real-time computing nodes. Its software application is composed of two main parts: an IDL graphical user interface and several programs for performing the real time feedback control and managing the whole control system [6,7]. The two components interact with each other through utilizing TCP/IP protocol. The user interface can be invoked and used by several users simultaneously from any X11 Windows client. A large number of plasma parameters for controlling discharges can be viewed and set via the user interface. As shown in Fig. 1, the physicists can view and edit the target values of the first poloidal field coil current on the PCS graphical user interface. The user interface provides a fast and simple way for viewing and modifying hundreds of parameters required for each plasma discharge. A PCS server process that runs on the PCS host node, which called "waveform server", manages the inputs from those interfaces. The waveform server process is the central storage location for those PCS setup data. It manages and coordinates all changes to discharge parameters from each of the user interface clients. A set of PCS processes obtains the setup data from the waveform server, processes the information and performs realtime feedback control during a typical discharge cycle. However, it becomes an extreme inconvenience to run and operate the interfaces to specify the discharge on a distant host. It takes a few minutes to initialize the interface widgets and several seconds to view or edit a waveform data item due to the high network latency to draw or operate the interfaces. Since the PCS waveform server always stores the latest discharge setup parameters, the IDL graphical user interface only provides an easy way to view and set the parameters. Other interfaces can be used to interact with PCS server processes, for example, a simple non-GUI interface to the PCS for supporting remote control [8–10]. This paper describes a web-based user interface system, which called WebPCS, for the EAST tokamak for remote collaboration and control. The setup data are retrieved from the PCS system by server-side PHP programming language and client-side JavaScript draws the interfaces into the user's browser. Users can also edit or update the setup data to the PCS system through the WebPCS system. These technologies allow the web-based user interface to be viewed by authorized users with a web browser and have it communicate with PCS server processes directly. Since the interfaces can be rendered

* Corresponding author. Tel.: +86 551 65592779; fax: +86 551 65591310.
E-mail address: rrzhang@ipp.ac.cn (R.R. Zhang).

http://dx.doi.org/10.1016/j.fusengdes.2014.02.070
0920-3796/

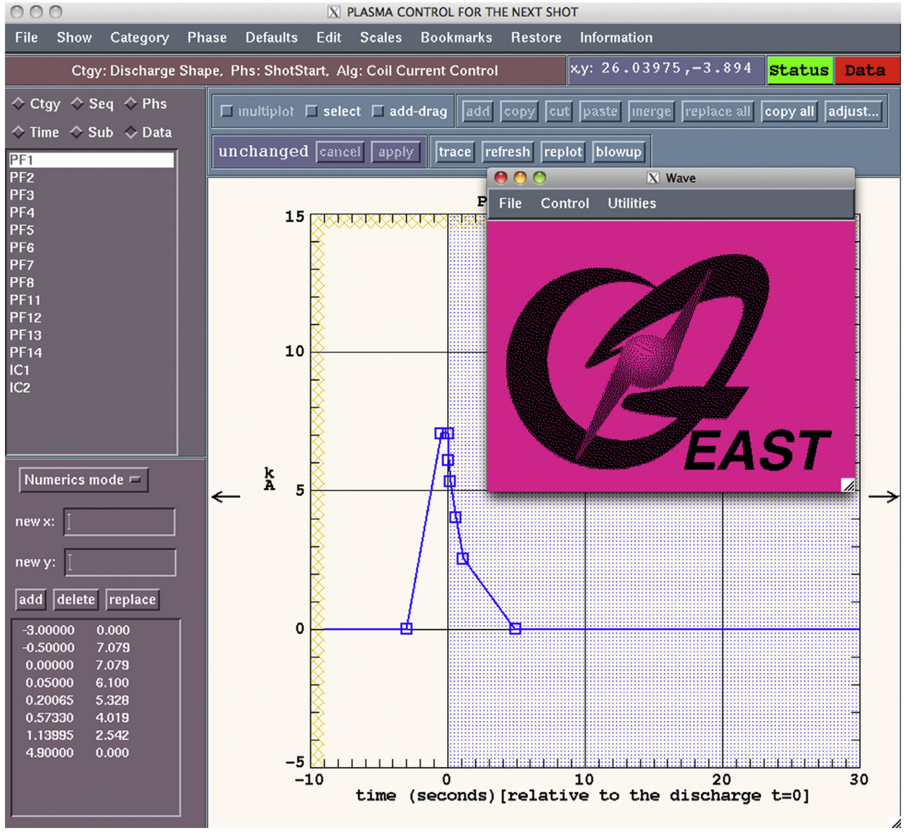

Fig. 1. The IDL user interface of EAST PCS.

on the client-side with dynamic data obtained from the server-side, most of the communication between the WebPCS and EAST PCS server processes involved very small packets of information, such as several floating-point vertex pairs for specifying waveform target values. It can provide better performance for remote operation. The software structure and implementation details are presented in Sections 2 and 3. The performance tests and analysis are given in Section 4. Finally, a summary is given in Section 5.

2. System overview

The WebPCS is an independent system that interacts with the server processes of EAST PCS, such as the waveform server process. Fig. 2 shows the pipeline of WebPCS system's functionalities.

The front end of the WebPCS system is designed as a Web service which can be browsed by users worldwide. It receives user requests and displays the data from PCS system and draws the control interfaces into the browser. On the back end, it processes user requests and communicates with PCS server processes through sockets. Based on the requests, it may return the result to the front end or send the parameter changes to waveform server process. NFS (Network File System) protocol is used to share PCS data files between the PCS host computer and WebPCS computer. Both the PCS host node (pcstest) and the WebPCS computer are connected through a local network with a bandwidth of 1 Gbps, thus the latency of exchanges between the two computers is very low, the WebPCS system can quickly obtain and process information from PCS host and then displays the changes in the user's browser. MySQL database is chosen to collect information from PCS server processes.

3. Architecture design and technologies used

The overall process of WebPCS is a sequence of three stages: (1) collecting PCS active sessions (2) drawing the interfaces, and (3) processing user requests.

3.1. Collecting PCS active sessions

Multiple PCS sessions can be run on the same host node at the same time for testing or normal operations. Each session has a waveform and other server processes. The PCS system uses a number of network ports for communication between its various processes. The IDL graphical user interface communicates with a waveform server process over the given port through socket technology. For example, the waveform server is listening on port 1234 for input messages in normal operations mode. Any changes which were made by the user interface would be sent to PCS host node's port 1234 and used for the real tokamak control. The local IDL user interface obtains the port numbers by fetching the Linux environment variables, which are defined in PCS startup script. However, the web interface cannot know those port numbers since it runs on a different computer. We collect the PCS active session information by adding codes to the PCS startup script to record the information into MySQL database, namely webpcs_east, which includes a PCS_SETTING table for storing the PCS server process port numbers, run mode and other useful information. The information becomes invalid once the PCS processes exit, codes are also added in the cleanup section of the startup script to delete the session record information from the MySQL database. Thus the WebPCS system's MySQL database always stores the active PCS sessions. The WebPCS system provides a web page to display the PCS session information. Users can select one session to join the discharge parameters setting.

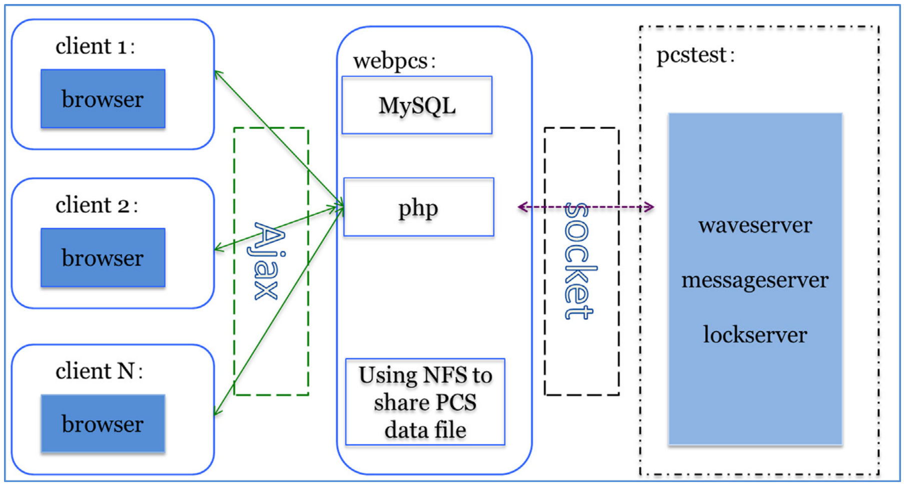

Fig. 2. The overall functional design of the WebPCS.

3.2. Drawing the user interface

The PCS Users are very familiar with and accustomed to use the IDL user interface, so much of the general interface design has been considered when deploying the interfaces on the web. As shown in Fig. 3, it is the waveform editor window page, much like the original IDL interface.

On the client side, we build the user interface utilizing jQuery UI [11], and make it communicate with the server-side PHP code using jQuery AJAX (Asynchronous JavaScript and XML) technology. The jQuery library, a fast and feature-rich JavaScript library, is used to simplify development of JavaScript code and solve many cross browser compatibility problems associated with JavaScript. jQuery also allows us to easily make AJAX requests to provide more responsive user experience, data transmission rates and performance. We use Flot [12], a pure JavaScript plotting library for jQuery, to plot the waveform data in the browser.

3.3. Processing user requests

Hypertext Preprocessor (PHP), a widely used general-purpose scripting language for Web development, is used to process a series of input requests such as “initwave” and “new_vertex”, and communicate with PCS server processes to fetch and organize the reply. Once the user selects one PCS active session, the front end of the WebPCS sends “initwave” AJAX request to the PHP to initialize the waveform editor window, the web UI would be created and PCS setup data would be obtained from the selected waveform server process.

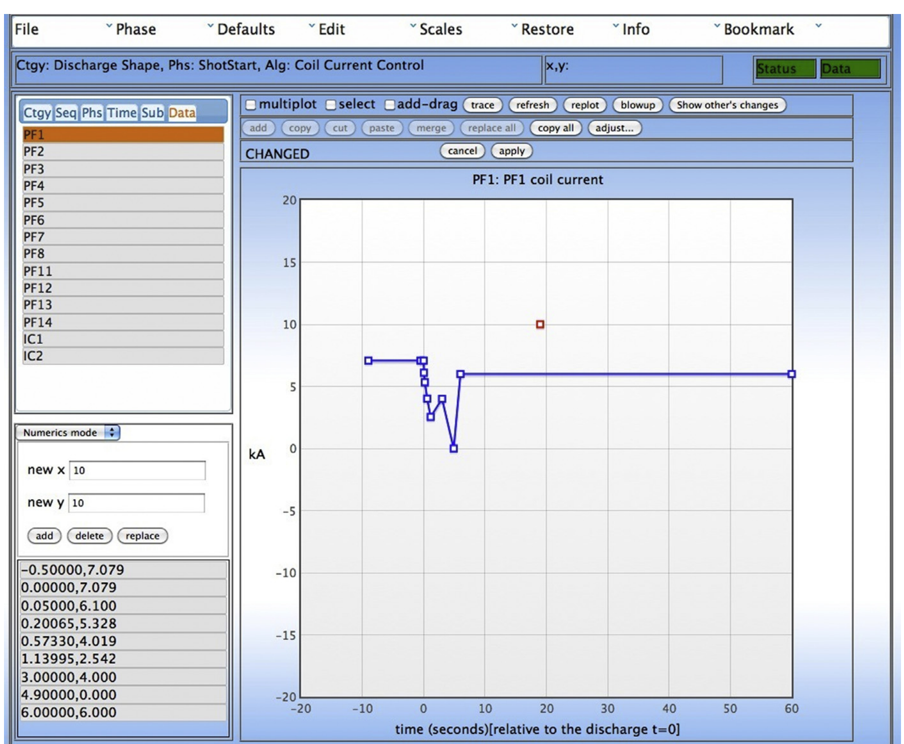

Fig. 3. The Web user interface of EAST PCS.

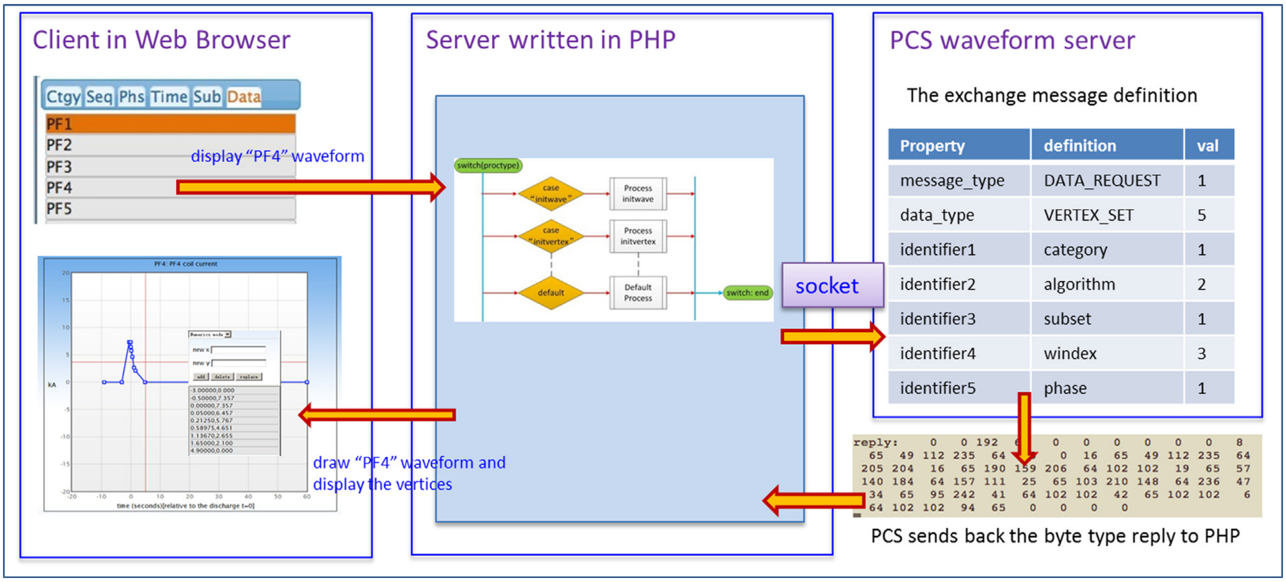

Fig. 4. An example of display PF4 waveform request showing the workflow of request process.

Here we take a simple example to demonstrate the workflow of a user request, as shown in Fig. 4. The user wants to obtain the target values setting of the fourth poloidal field coil current by clicking the “PF4” item in the Web user interface. The client side JavaScript code responds to the click event, and sends the “initvertex” request to the server side. The PHP code receives and processes the request. It extracts the information such as category index, algorithm index, and waveform index from the passed parameters of the jQuery AJAX request and exchanges the message with the PCS waveform server process. The waveform server handles the data request and returns the vertex pairs of the specific waveform. The PHP script extracts all the vertices from the waveform server's byte type reply and sends back the vertex information in JSON (JavaScript Object Notation) format to the WebPCS client side. The client side JavaScript gets the return vertices information, draws the PF4 waveform by using the Flot plotting library and displays the vertices on the interface.

4. Test and analysis

Timing tests were performed in different network environments. The WebPCS user interface was a little slower than the local IDL graphical user interface over a Local Area Network. The initialization of web user interface can be done in two or three seconds. But for remote accessing, for example, outside our country, the initialization process can be done in ten seconds, far less than running the IDL interface through SSH software which may take several minutes. In the most cases, the web user interface and remote PCS server processes exchange very small packets, such as some vertex pairs for specifying waveform target values, that can be done in 100–200 ms.

We found that the AJAX requests consume the most time by monitoring the network. Two methods were utilized to insure user interactivity with minimal latency. One is to reduce the number of AJAX requests and the other is to reduce the data amount from the server side. Taking the interface initialization as an example, the IDL interface communicates with the PCS waveform server 21 times to get all the required data. To speed up the display of Web user interface, the client side only sends one AJAX request, at the server side, PHP communicates with PCS server processes several times to get the data and sends back the combined data to JavaScript. In order to reduce the result data amount from the server side, the JSON data format was selected for it's a lightweight data-interchange format.

5. Summary

WebPCS has been developed as a user interface tool for remote access to the EAST plasma control system. Authorized users with a web browser can view and edit the waveforms for specifying discharges without installing any plug-ins. It can provide faster access for remote access control. As the preliminary test, WebPCS has realized basic functions of the IDL user interface and tested with EAST PCS in an off-line mode. Preparations are underway to test the system in a live plasma discharge. More functions need to be added for remote control. The system needs to provide support for other platform such as mobile devices.

Acknowledgments

This work is supported by the National 973 Project with number 2012GB105000, the National Natural Science Foundation of China with contract number 11205200 and 11205191.

References

[1] B.J. Xiao, D.A. Humphreys, M.L. Walker, A. Hyatt, J.A. Leuer, D. Mueller, et al., EAST plasma control system, Fusion Engineering and Design 83 (2008) 181–187.
[2] Q.P. Yuan, B.J. Xiao, B.G. Penaflor, D.A. Piglowski, L.Z. Liu, R.D. Johnson, et al., New achievements in the EAST plasma control system, Fusion Engineering and Design 85 (2010) 474–477.
[3] J.R. Ferron, A. Kellman, E. McKee, T. Osborne, P. Petrach, T.S. Taylor, et al., An advanced plasma control system for the DIII-D tokamak, in: Proceedings of the 14th IEEE/NPSS Symposium on Fusion Engineering, San Diego , 1991, Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ, 1992, p. 761.
[4] B.G. Penaflor, J.R. Ferron, M.L. Walker, D.A. Humphreys, J.A. Leuer, D.A. Piglowski, et al., Worldwide collaborative efforts in plasma control software development, Fusion Engineering and Design 83 (2008) 176–180.
[5] D.A. Humphreys, J.R. Ferron, A.W. Hyatt, R.J. La Haye, J.A. Leuer, B.G. Penaflor, et al., DIII-D integrated plasma control solutions for ITER and next-generation tokamaks, Fusion Engineering and Design 83 (2008) 193–197.
[6] B.G. Penaflor, D.A. Piglowski, J.R. Ferron, M.L. Walker, Current status of DIII-D real-time digital plasma control, IEEE Transactions on Nuclear Science 47 (2000) 201–204.

[7] B.G. Penaflor, J.R. Ferron, M.L. Walker, D.A. Humphreys, J.A. Leuer, D.A. Piglowski, et al., Extending the capabilities of the DIII-D plasma control system for worldwide fusion research collaborations, Fusion Engineering and Design 84 (2009) 1484–1487.
[8] D.A. Piglowski, D.A. Humphreys, M.L. Walker, J.R. Ferron, B.G. Penaflor, R.D. Johnson, et al., Accumulated experiences from implementations of the DIII-D plasma control system worldwide, Fusion Engineering and Design 85 (2010) 451–455.
[9] D.P. Schissel, G. Abla, D.A. Humphreys, B.G. Penaflor, B.S. Sammuli, M.L. Walker, Remote control of a fusion facility, Fusion Engineering and Design 84 (2009) 1729–1733.
[10] D.P. Schissel, G. Abla, T. Fredian, M. Greenwald, B.G. Penaflor, J. Stillerman, et al., An investigation of secure remote instrument control, Fusion Engineering and Design 85 (2010) 608–613.
[11] http://jqueryui.com/
[12] http://www.flotcharts.org/