§21. Development of Portal Web Pages for the LHD Experiment

Emoto, M., Iwata, C., Yoshida, M., Nagayama, Y.

Fig.1. Portal page of the LHD experiment

Because the LHD project has been operating with the cooperation of many institutes in Japan, the remote participation facility plays an important role. The authors regard Web services as essential tools for the current Internet communication, and have been developing Web services for remote participation, for example, experiment proposals, experiment schedulers, data viewers, and so on[1]. However, because these services are dispersed among several servers in NIFS, users cannot find the required services easily. Therefore, the authors developed a portal Web server to list the existing and new Web services for the LHD experiment[2-3].

Fig. 1 shows the portal web page. This portal page provides not only useful links of the LHD experiment for the participants, but also the real time information of the experiment. The calendar of in the top left corner is linked to the experimental scheduler, and the user can view the schedule or submit the experiment proposal from here. The following frames display the summary information, such as video file of plasma monitoring, NBI current graph, and summary graph. Fig.2 shows the flow diagram of the data. In the middle column, there is a daily log of the experiments. In the top area, comments of the experiment coordinator are displayed. In the middle, the configuration of the next plasma is displayed. At the bottom, there is the summary information of each shot. The right column displays the information of the following plasma discharges, such as the configuration of the next discharge, and the time table of the discharges. Also, there is a shot calculator in the middle. This is convenient to know how many shots can be done in the assigned machine time.

In order to realize an interactive GUI for the portal page, AJAX (Asynchronous JavaScript + XML) technique is used. Because most of the web browser supports the JavaScript, the user does not have to install extra software. Generally, it is difficult to implement AJAX because the developer has to use the multiple computer language at the same time; for example, Java for the server program, SQL to use a relational database (RDB), and JavaScript for the user interface. In order to develop AJAX application efficiently, the authors used Ruby on Rails (RoR). Because RoR encapsulates JavaScript and the access to RDB, the developer can use only Ruby to build AJAX application.

The above information must be updated as the experiment schedule progresses. To notify the computers of the experimental sequence, IP multicast packets are broadcast in the local network. The packets are sent when the sequence status or experimental number changes. When the server program receives the IP multicast packet, it updates the web contents.
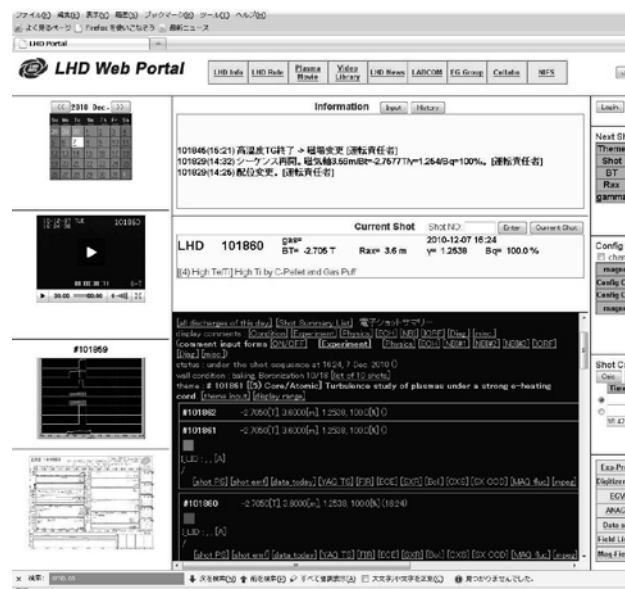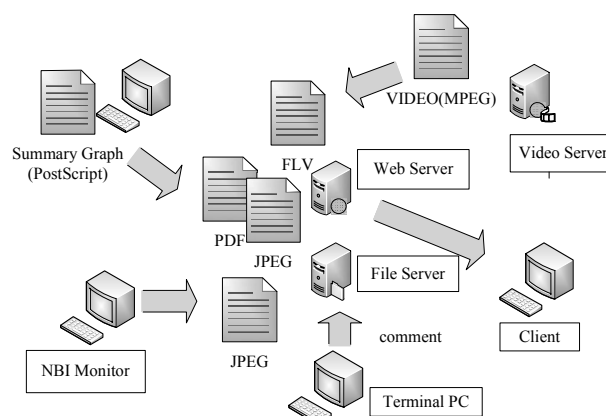

Fig.2. Data flow diagram

The portal web helped the user take part in the last experiment remotely, but the authors received several requests and suggestions. In reply the request, the authors have been modifying the portal page, and the improved version will be shown in the next experiment season.

1) Emoto, M., et al.: Fus. Sci. Tech., **58**, (2010) 458
2) Emoto, M., et al: 20th International Toki Conference, Toki, Japan (2010)
3) Nagayama, Y., et al: 8th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, San Francisco, USA (2011)