Web interface in DABC

S.Linev[1]
[1]GSI, Darmstadt, Germany

User interface for DABC

DABC [1] is general-purpose DAQ framework, developed in GSI since 2007 and used in different applications. Because it cooperates with many other components (like slow control systems, other DAQ systems, online analysis), DABC requires a simple and flexible user interface for monitoring and control. Decision was taken to evaluate web protocols for implementing a user interface via normal web browser. As a starting point for development, JSRootIO project was considered.

The JSRootIO project

The JSRootIO [2] project is a new development of ROOT [3] team. Objects like histograms or graphs, stored in binary ROOT files, can be read and displayed with all modern web browsers. Usage of JavaScript allows to build interactive and very informative graphical elements.

Several important improvements were done to increase flexibility and usability of the JSRootIO graphics. A context menu was implemented, where convenient commands are provided: switch for lin/log axis, changing of draw options, toggle statistic box. Drawing and update of histogram statistic box was implemented - it is especially important for the case when histogram content is regularly updated. Also a significant performance improve (by factor 10) was achieved. At its present state JSRootIO allows to insert JavaScript-base ROOT graphics in arbitrary HTML page and provides look-and-feel of the original ROOT graphics.

A main disadvantage of JSRootIO is that it works only with ROOT files. This makes it difficult to use it for online tasks, where many objects should be frequently updated.

HTTP server in DABC

Instead of creating temporary files for online monitoring, one could provide specialized HTTP server delivering objects data directly to the browser. Mongoose [4] embeddable web server was chosen as basis for implementing http server in DABC. Mongoose implements basic web protocols and via callback functions provides possibility to construct user-defined response on the HTTP requests.

Specialized hierarchy of objects was introduced in DABC, where different kinds of data can be registered. Main aim of hierarchical organization - provide structural access to user-defined data. One could compare this hierarchy with file system - different sub-folders correspond to different parts of a big system, and files represent some parameters or objects. As response on HTTP requests, DABC server returns hierarchy descriptions (in XML files) or binary data from registered elements.

JavaScript code on the browser side interprets this hierarchy description and creates tree view, seen on the left side of Figure 1. On the right side of the web page selected items are displayed. On the example figure these are histograms, displayed using JSRootIO graphics. Also ratemeters and simple text output are supported. When monitoring is enabled, object content (histograms, ratemeters) will be updated regularly. It is possible to extend code for displaying of any user-defined kind of data.

As a result, HTTP server in very generic way provides direct access to the information from arbitrary DABC-based application. Such information can be explored, displayed and monitored in any modern web browser. No any intermediate files are necessary.

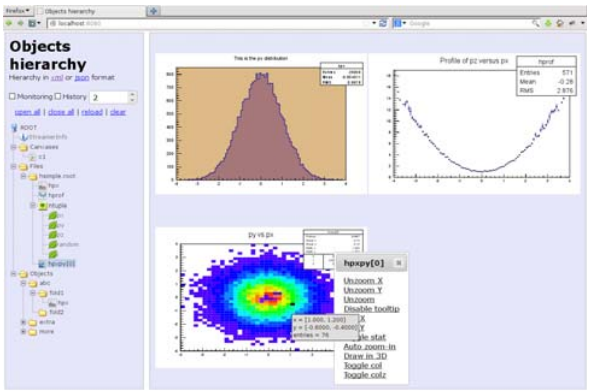

Figure 1: Web interface for ROOT application

FastCGI interface in DABC

FastCGI [5] is a protocol for interfacing interactive programs with standard web servers (like Apache or lighttpd or many others). Using FastCGI, one benefits from standard software and could integrate online applications into existing web infrastructure. FastCGI server was implemented for DABC and provides similar functionality as mongoose-based http server.

Monitoring of ROOT applications

The combination of DABC as web server and JavaScript code in web browser allows to implement live monitoring of arbitrary ROOT application. With a few function calls any ROOT-based application can start HTTP server and publish various objects like histograms or graphs. Figure 1 shows browser with several histograms, produced by running hsimple.C macro from ROOT tutorials.

In Go4 [6] production release 4.6.0, DABC-based HTTP server is also provided. Without modifications any existing go4-based analysis can start a web server, where all histograms, graphs, canvases and trees are available for display. One can also browse parameters and events objects members. Moreover, via a command interface one could suspend/resume analysis execution or clear histograms content by pressing button in the browser window.

Unified interface for many components

DABC provides flexible mechanisms to integrate different kinds of data into the framework - it could be raw data from front-ends, but also information from slow-control systems. For instance, plugin for MBS [7] can retrieve both raw lmd data and statistic information from arbitrary number of MBS nodes. Or FESA [8] plugin can acquire preselected subset of records from accelerator control system. Via unified HTTP interface all these kinds of information can be provided to the user.

With HTTP server one could also control DABC applications and all its components. The user can define commands, which are published via web interface in the browser. Typical use-cases: start/stop file writing, toggle logging mode, reset counters, and so on. Of course, all these commands can be protected from unauthorized use.

DABC allows to run many agents (slave applications), collecting information from different sources. HTTP server, running on the master node, will provides seamless access to data from all agents. This allows to build user interface for distributed (running on many nodes) and heterogeneous (acquiring different kinds of information) systems.

Conclusion

The web interface in DABC provides a unified view for data from many different frameworks like ROOT, Go4, MBS, FESA or DABC itself; support for EPICS and DIM is in development. Web interface could be used on many computing devices and typically does not require any additional software installations - just a normal web browser. The developed interface can be directly used with arbitrary Go4-based analysis; with minimal efforts it can be enabled for any ROOT-based application. Current DABC version 2.6 with the web interface was released in November 2013 and is available on the project home page [9].

References

[1] S. Linev, J. Adamczewski-Musch and P. Zumbruch, “Status of data acquisition software DABC”, GSI Scientific Report 2011 (2012) p. 64

[2] B.Bellenot, S.Linev, JSRootIO project, Journal of Physics: Conference Series, CHEP 2013 proceedings (submitted)

[3] ROOT project, http://root.cern.ch

[4] Mongoose embaddable web server, https://github.com/cesanta/mongoose

[5] FastCGI protocol, http://en.wikipedia.org/wiki/FastCGI

[6] Go4 project, http://go4.gsi.de

[7] MBS DAQ system, http://daq.gsi.de

[8] Front-End Software Architecture (FESA), https://www-acc.gsi.de/wiki/FESA

[9] DABC homepage, http://dabc.gsi.de

 doi:10.15120/GR-2014-1-FG-CS-12