CDF/DOC/COMP_UPG/PUBLIC/8467
Version 0.9
August 22, 2006

A Web based Histogram Viewer Application built using the AJAX Technology

Subir Sarkar

Abstract

We've developed a web-based Root histogram viewer as a Rich Internet Application (RIA). Asynchronous Javascript and XML (AJAX) is used as the presentation layer on the Client side to achieve a rich desktop like look-and-feel and interactivity. The Server side CGI program uses Root in combination with a CGI library known as cgicc. In this note we shall briefly present the AJAX processing model and describe the application we've developed for the SVT Online monitoring tasks in detail.

1 Introduction

Although the classic web application model played its role quite creditably, it always lacked in the richness of graphic elements and responsiveness of desktop applications.

Recent resurgence of Javascript as a high-level, cross-platform, cross-browser interpreted language in combination with the Asynchronous Javascript and XML (AJAX) [1] technology has finally created the much-awaited Rich Internet Application (RIA) platform. A Web application built as a RIA can match desktop applications in interactivity and simplicity with the added advantage that it is accessible to anyone, from anywhere, using nothing more than a modern Web browser and with no other client side software installation.

2 AJAX Processing Model

AJAX is a collaboration of several matured, open-source technologies like Extensible Hyper Text Markup Language (XHTML), Cascading Style Sheets (CSS), Document Object Model (DOM), Extensible Markup Language (XML) and Extensible Stylesheet Language Transformations (XSLT), XMLHttpRequest etc., all bound together by Javascript.

An AJAX supported web application provides,

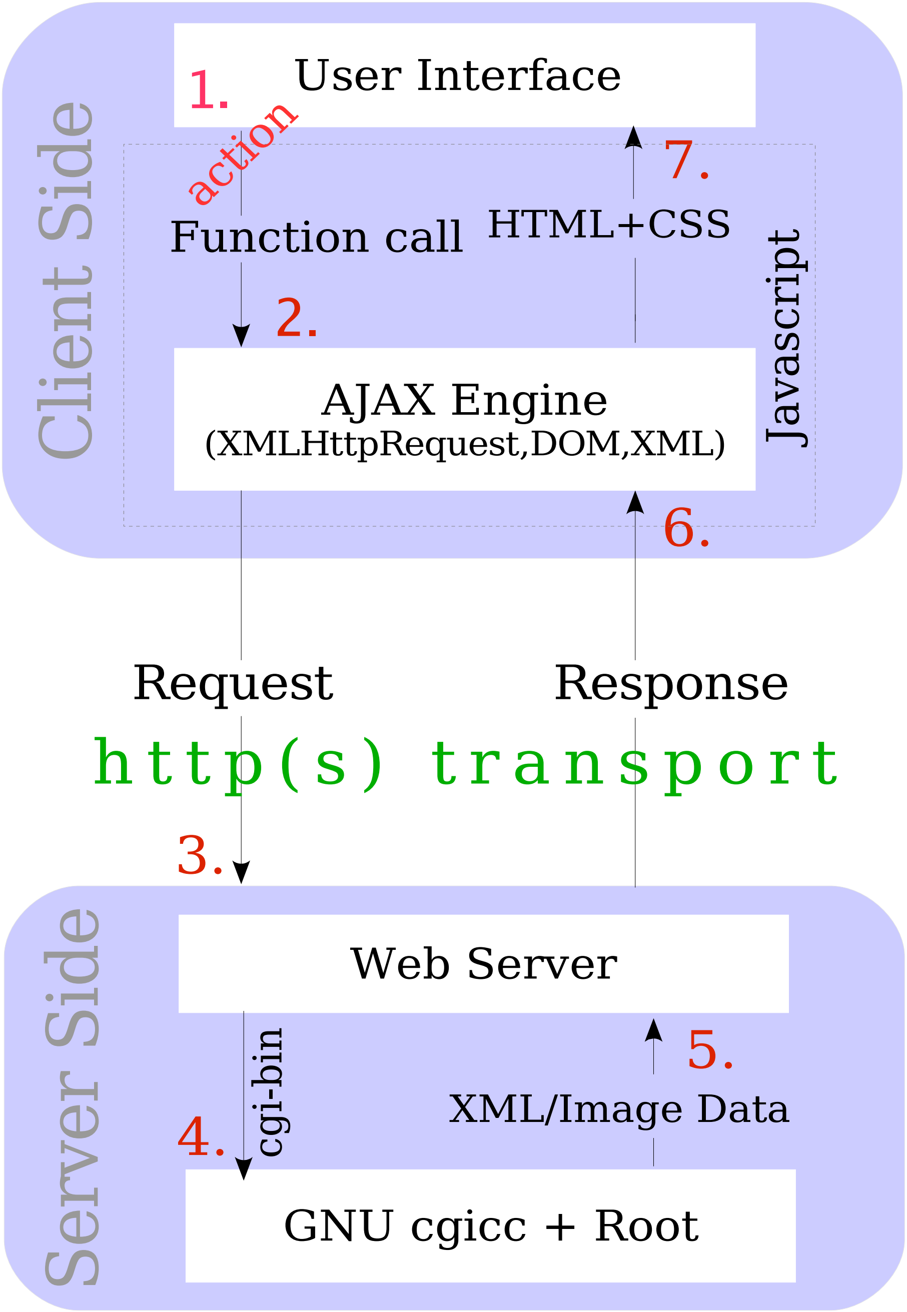


Figure 1: The AJAX Processing Model. In this model most user actions involve Javascript function calls through an AJAX engine. An XMLHttpRequest object is at the heart of the AJAX Engine which packs the request to be sent to the server and unpacks the server response automatically which is subsequently processed by the Javascript callback function set up earlier while sending the request.

- significantly reduced response time as the interface is loaded only at start-up. Subsequently, every time an application needs something from the server, the server returns precisely that. Javascript intercepts the response and updates only the relevant part of the interface at a time. In fact the user never even feels that the application goes to the server at all, which is the sign of a truly desktop like application.

- a clear separation of Client and Server sides.

The AJAX processing model is shown in detail in Figure 1. AJAX unleashes the full potential of Javascript which takes the centre stage in this scenario and processes most of the events and communication with the server side. When a client side event occurs, Javascript

- collects input data from various Form elements and validates them.

- encodes the parameters properly and sends the request to the server using the HTTP protocol with the help of an XMLHttpRequest object. A callback function is registered that will handle the server response. The client can freely perform other tasks at this point.

- The callback function triggers as soon as the server response arrives. Depending on the type of the server response, action is taken and the relevant part of the interface updated.

3 Implementation

We shall now discuss a Web based Root [2] histogram viewer which is an implementation of the above model. The application we are going to discuss has been developed in teh context of SVT Online monitoring tasks but the concept is very general and can easily be adapted to other needs.

As can be seen from Figure 2, which is itself a snapshot of a typical histogram viewer session, the user interface is divided into a (a) drawing canvas, and (b) drawing control. When the interface is loaded it automatically displays the available run numbers for which histograms are available. It also fills the list of reference run numbers for comparison, presently with the same values. One can now get the list of plots available for the selected run and plot them. There are several plotting options available, namely

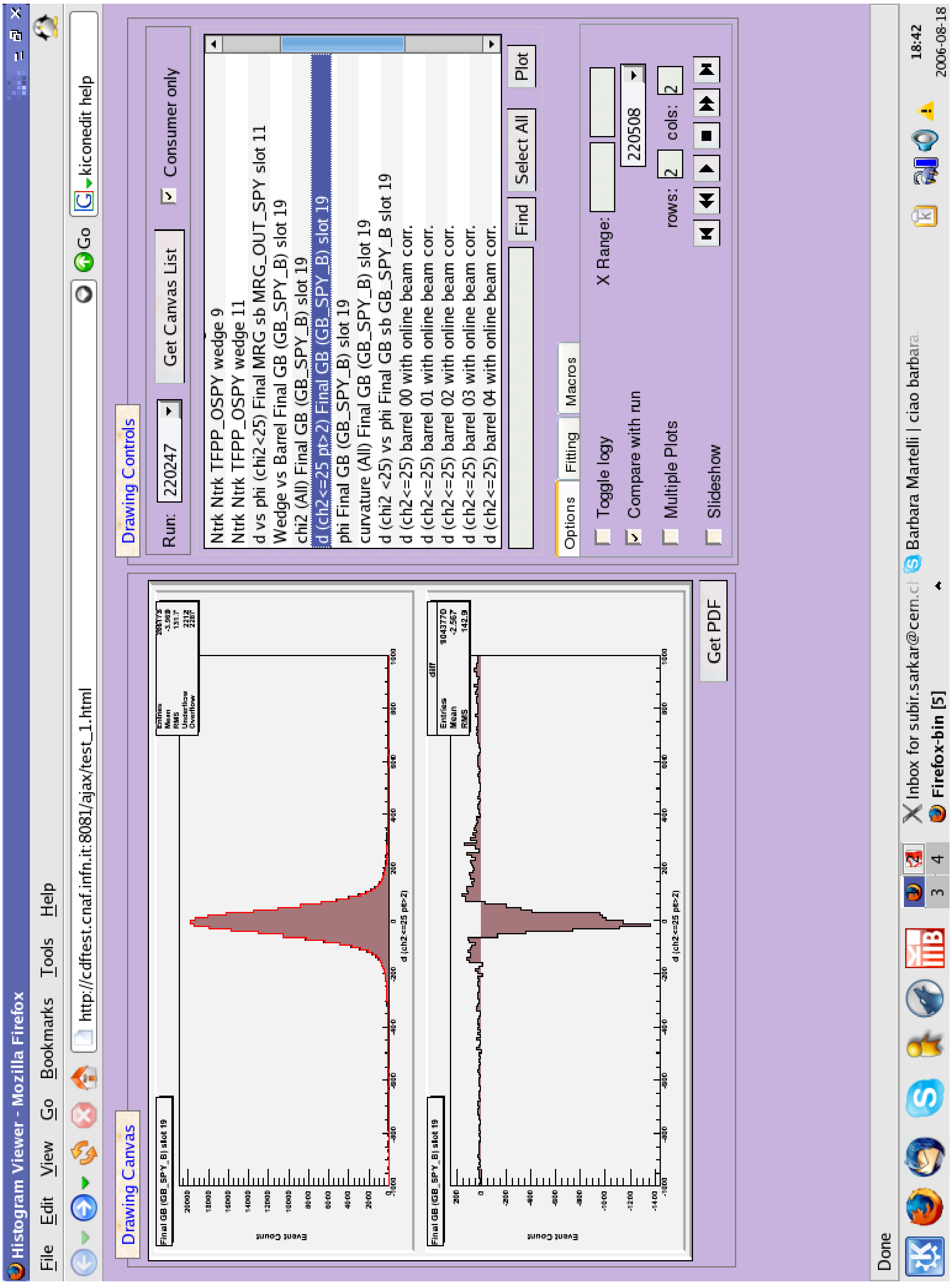
- toggling logy option

- plotting within an X range

- multiple plots on the same canvas

- comparison of a plot with a reference one

- slideshow of a set of selected plots



http://cdftest.cnaf.infn.it:8081/ajax/test_1.html

Run: 220247

☑ Consumer only

Ntrk Ntrk TFPP_OSPY wedge 9
Ntrk Ntrk TFPP_OSPY wedge 11
d vs phi (chi2<25) Final MRG sb MRG_OUT_SPY slot 11
Wedge vs Barrel Final GB (GB_SPY_B) slot 19
chi2 (All) Final GB (GB_SPY_B) slot 19
d (ch2<=25 pt>2) Final GB (GB_SPY_B) slot 19
phi Final GB (GB_SPY_B) slot 19
curvature (All) Final GB (GB_SPY_B) slot 19
d (chi2 <25) vs phi Final GB sb GB_SPY_B slot 19
d (ch2<=25) barrel 00 with online beam corr.
d (ch2<=25) barrel 01 with online beam corr.
d (ch2<=25) barrel 02 with online beam corr.
d (ch2<=25) barrel 03 with online beam corr.
d (ch2<=25) barrel 04 with online beam corr.

□ Toggle logy
☑ Compare with run 220508
□ Multiple Plots
□ Slideshow

X Range:

rows: 2 cols: 2

Figure 2: A snapshot of a histogram viewer session

which can be extended as new requirements arise.

The Server side is a simple CGI program written in C++. The GNU cgicc [3] library (version 3.2.2) is used to extract the parameters passed to the program while Root takes care of the rest. Root v5.12.00 is recommended as it supports png/xpm image creation from Root Canvases on the fly.

The viewer can be accessed at [4]. Currently the following browsers are supported,

- Firefox 1.x, Mozilla 1.x, SeaMonkey 1.x
- Safari 2.x
- Opera 9.x

Support for Internet Explorer 6 is being worked on.

The present application suffers from some limitations, namely, (a) conceptual problem with back and refresh buttons, (b) slow start-up of large and complex Javascript code etc. However, with careful design these limitations can be lifted.

4 Conclusion

AJAX based Rich Internet Applications show a lot of promise for High Energy Physics (HEP) since remote viewing of histograms and other important data is crucial. The present application shows how, with a very little extra effort, we can achieve responsiveness and interactivity of desktop applications. We hope more and more traditional HEP web applications will be converted to AJAX supported RIAs in the future.

References

[1] Ajax: A New Approach to Web Applications, Jesse James Garrett, 2005,
http://www.adaptivepath.com/publications/essays/archives/00038.php

AJAX Hacks, Bruce W. Perry, O'Reilly, 2005

[2] ROOT - An Object Oriented Data Analysis Framework,
Rene Brun and Fons Rademakers, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86

[3] http://www.cgicc.org/

[4] http://cdftest.cnaf.infn.it:8081/ajax/test_1.html