

# WEB BROWSER BASED APPLICATIONS FOR THE MIT-BATES ACCELERATOR

F. Wang, K. D. Jacobs, E. Bisson, X. Geng and, J. B. van der Laan<sup>†</sup>  
MIT-Bates Linear Accelerator Center, Middleton, MA 01949  
B. Yang, TeleGea, Inc. Waltham, MA 02451

## Abstract

A number of Web Browser based applications have been developed to serve in the challenging environment of the Bates storage-stretcher ring operation. Most of them are accelerator physics applications that require intensive data processing and analyzing efforts, with moderate tasks of interfacing with the control system. The basic idea of developing applications using a commercial Web Browser is to truly realize object orientated design and platform independence, thus making development efforts more affordable. To implement this concept, common graphic services and methods of interfacing with control systems and various modelling codes are the two essential pieces. As a start, a small package of Java applet classes is developed to provide basic graphic service, and Common Gateway Interface (CGI) is used for interfacing the control system, etc., with the Web server, as only light interfacing is required. For more dynamic graphic user interface approaches, Java servlets are used to replace CGI scripts and communicate with applets. Two accelerator physics applications (beam-based ring optics modelling and beam position monitor (BPM) center offset calibration), and an operational program for scaling the ring energy, are presented to illustrate this method. The merits, limitations and potential development of this approach are discussed.

## 1 INTRODUCTION

The initial motivation for developing web-based applications is to search for a flexible and efficient way to build comprehensive on-line accelerator physics applications in the dynamic evolution environment of the accelerator facility at the MIT-Bates Linear Accelerator Center [1-3]. There are several achievable features of this approach:

- It is possible to organize complicated tasks in an orderly manner based on object-orientated design.
- The applications can be platform independent.
- The applications are always based on widely used commercial standards.

These features are crucial to minimize development and maintenance efforts. Careful programming methodologies are important to keep these features, when dealing with the two key issues: graphic representation, and interfaces to local control systems and modelling codes.

\* Work supported by the U.S. Department of Energy cooperative agreement DE-FC02-94ER40818.A000.

<sup>†</sup> on leave from NIKHEF, Amsterdam, The Netherlands.

## 2 PROGRAMMING METHOD

### 2.1 Graphical Display

Data visualization is essential to any complicated problem solving process, especially for on-line applications. Large quantities of graphics should be generated conveniently and available for display during the application process. To realize platform independence, Java applets are used for graphic displays. A Java applet package was built to be a common building block for basic graphic service. For common use purposes, standard format input files are required by the basic graphic service. These input files are text files with simple data structures, and text "commands" to describe how to draw. The "Top-Drawer" command format was adopted here for convenience. This applet package provides single or multi-graphic display applets. It satisfies most of our needs. Data can be provided for other plot tools to meet special graphic needs. Applets can be made with more graphic user interface (GUI) capacities using the Java Abstract Window Toolkit (AWT). The simple text format can still be used inside these applets if desired. Real-time GUI with local systems requires special interface arrangements. It will be discussed in the next section.

### 2.2 Interfacing

Two typical types of interfaces are required. One is to communicate with various modelling codes, and the other is to interact with the local accelerator control system. The context of the modelling codes and the control systems covers a broad range. Most parts of our control system are under EPICS [4]. The EPICS channel access programs are designed to be written in C. The control codes from the previous homemade control system are also in C. The modelling codes are written in FORTRAN, C++, Perl or anything else. There are always security restrictions for the network communication between the web browser (applet, or the web page) and the above mentioned server-side applications. The easy part of the security concern in our situation is that the applications, which affect real facility operations, are always put on the lab controls Intranet. So the real problem becomes how to get around the security constraints imposed by the network languages themselves, like the sand box role for Java applets.

The first method for interfacing is to use CGI. CGI is a more or less platform-independent way to produce

dynamic web content, and CGI programs can be in many languages. The applets in this case are used merely for graphic display. This approach satisfied many of our requirements. CGI programs launched from a web page have certain security restrictions, such as the inability to write a local file without pre-creating such files with wide-open write permissions, etc. This brings some “inconvenience”. But the real problem with CGI interfacing is that it is not suited for high performance dynamic applications because each CGI script fired from the web server starts a new separate process.

The second approach is to use Java Servlets. A Java Servlet is widely used Java server-side technology. Without going into the advantages of using servlets (see Ref. [5] for example), the goal here is to replace CGI scripts for running dynamic web applications. A servlet engine is needed in the web server, whether it is built-in or an add-on, to deploy servlets. The idea is to run the GUI from applets (or a web page) to communicate with servlets which run server side applications. There are very few security restrictions for servlets to do things on the server machine if properly set up, such as writing files, executing system commands, etc. However there is a catch in using servlets: Java servlets are written in Java, yet usually must work with native languages (C, C++, etc.) to get things done. Java servlets do run Java Native Interface (JNI) to interface with applications written in native languages. Therefore they can do EPICS channel access with JNI. Directly launching process executive commands from a Java servlet program also works. This makes running other codes and processes as easy as running a CGI program. It is true that executing a process is different than running a thread inside Java. However except for high speed, multi-thread applications, the back and forth communications can be concealed well inside a servlet program. The speed and efficiency of an applet-servlet network application may not be able to compare with a local platform dependent X-window application, but the overall programming efforts could be reduced for many of the applications with moderate interfacing needs.

### 3 WEB BASED APPLICATIONS AT BATES

#### 3.1 Beam-based Optics Modelling in the SHR

This was the first web browser based application at the Bates Center. The Bates South Hall Ring has two basic operational modes: storage and resonant extraction. A discrepancy between the design optics and the measured optics was observed. The beam-based modelling method [6,7] was used to set the quadrupole strengths correctly. The method requires measurement of the beam response matrix. The measured response matrix is then compared to the one calculated using the model optics. The model parameters are then iteratively adjusted until the difference between the modelled and measured matrix is minimized. The linear matrix equation used in our fit had a matrix size of  $3720 \times 187$ , half filled. It took a workstation with a 450 MHz Pentium-II processor

roughly two minutes to do one iteration. However the actual time consuming tasks were to figure out what was wrong if the fit did not work. Problems arose from instrumentation, hardware, improper operation, and even unexpected beam physics phenomenon. The application is written as a step-by-step procedure. It includes how to set reference optics, how to make measurements, and how to analyze data. Large quantities of raw data can be analyzed and displayed on-line. This plays an essential role in troubleshooting. After correction, the measured lattice parameters (betatron tunes, amplitude functions, etc.) were close to the expected optics. The corrected lattice has been used for the resonantly extracted beam experiment during the past year.

#### 3.2 BPM Zero Offset Calibration

This application is intended for on-line dynamic BPM zero offset calibration, using the quadrupole  $k$  (strength) modulation method. The reference orbit is taken as the magnetic centers of the quadrupoles. If the beam closed orbit deviates from a quadrupole center, making small variations in the quadrupole strength will shift the closed orbit linearly proportionally to the original closed orbit deviation, and to the fractional quadrupole strength variation [8]. In the actual calibration, the coefficient matrix of this linear dependency (BPM to Q- $k$  modulation response matrix) is measured and compared with that derived from reference optics. By doing so, errors from hardware problems, incorrect beam conditions, etc., can be detected promptly. Once the closed orbit to quadrupole centers are determined, the offset of each BPM to the reference orbit can be deduced from the known local geometrical layout and steering parameters.

The structure of this application's web page is similar to the optics modeling web page. It starts by verifying the ring optics mode, then giving instructions for performing the measurement and on-line data analysis. The display applets for the measurement result are the same as those for the optics modelling application. Figure 1 shows the deviation of the closed orbit from a quadrupole center measured by those BPMs which have suitable phase relations to the quadrupole and also to the steering kickers used in the BPM response coefficient matrix calibration. This applet features quick switching between a number of graphics. The “Closed Orbit Display” applet shown in Figure 2 is somewhat different. It has more GUI capacities to display detailed information on any selected section of the ring. The applet still uses the same graphic package but internally converts data to standard format for plotting. This applet also uses an applet-servlet connection to write BPM zero-offset results to local files. Figure 2 depicts the BPM zero-offsets calculated from a recent test run with a few independently powered quadrupoles. Once the BPM zero offset data are available, the closed orbit correction application will have a much better reference.

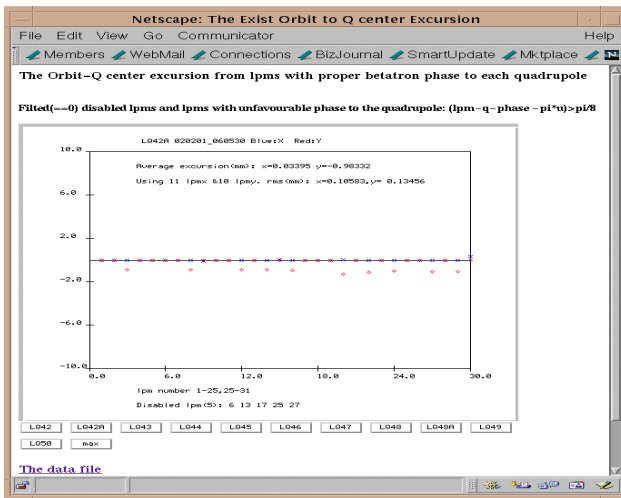


Figure 1: Orbit-Q center excursion (Multi-Graphic Display Applet).

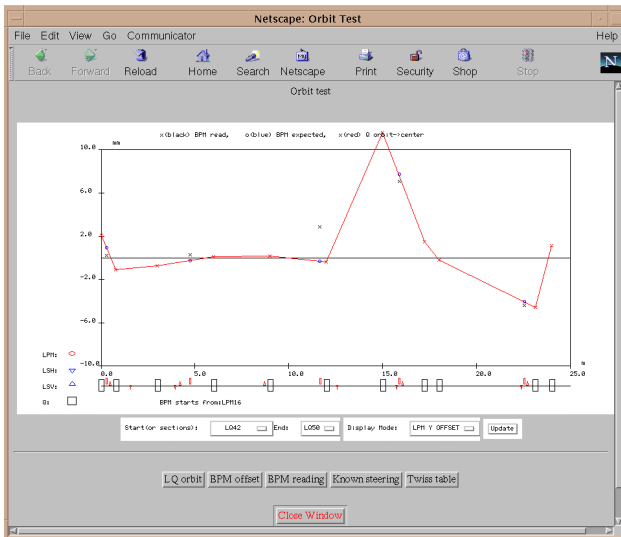


Figure 2: Closed Orbit Display (Interactive Applet).

### 3.3 Ring Energy Scaling

The above applications involved complicated data analysis. The ring energy scaling application, on the other hand, is a typical operation program. For the resonantly extracted beam experiment run last year, frequent energy changes were required. There are several technical issues which require beam-based adjustments at a few locations. This prevents the energy scaling process from being very straightforward. For one, the linac-recirculator magnet settings and RF setting are beam current dependent. Also, the trajectory and energy match of the linac-recirculator to the energy compressing system have to be verified. Finally, there are a number of magnets in the injection system which are not accurately calibrated. Command line programs or setting up manually are workable but somewhat tedious and can easily lead to slightly different approaches by different operators. The energy scaling application organizes all the tasks in a single web page: to choose and set reference files, preview scaled settings,

and scale magnets by sections. It turns a tedious task into an enjoyable one, and helps to standardize the procedure.

The application uses CGI only and the efforts to build and maintain such an application are minimal. Due to magnet calibration inaccuracies, beamline scaling was not perfect, but very close, and scaling for the ring was accurate. Having a clear procedure helps to identify real magnet setting issues. An example is the saturation effect of the fringe field of the 180° recirculator dipoles. At 950 MeV the strength of the first (and only the first) vertically focussing quadrupole in the recirculator back-leg had to be doubled to compensate for the vertical focussing reduction from the dipole, to keep the recirculator optics close to operational values.

## 4 DISSCUSION

Internet technology has now become part of everyone's lives. This is a strong reason to try to make use of this technical environment. The focus here is on programming methods instead of building new software tools. To understand the limitations of this kind of "network" application, we need to look at our existing local approaches: (A) MEDM [4] for EPICS. This GUI manager, also run in a browser, is convenient, efficient and reliable for routine operation, but it is not easy to work with complicated tasks nor to fit to special needs. (B) X-Motif applications are viable for some of our routine applications with high performance, but considerable development efforts are required. (C) Java swing applications can be comparable with X-Motif programs, but the Java to native language interface has to be addressed.

The web browser based applications described here lack high performance due to the use of a high level network language such as Java, and the inherent extra network communication. However this approach is a significant addition to our problem solving tools. The combination of object-orientated design, platform independence, and the all-popular commercial software based building method, makes the development and maintenance of applications very efficient and adaptable to new Internet technology evolution. It could greatly enhance the ability of individuals and teams to realize more creative work with less intimidation from dealing with time consuming programming efforts.

## 5 REFERENCES

- [1] K. Jacobs, et al., WPPH060, these proceedings.
- [2] G. T. Zwart, RPPH151, these proceedings.
- [3] F. Wang, et al., "New lattice for the South Hall Ring", Bates Internal Report B/SHR/2001/01.
- [4] <http://www.aps.anl.gov/epics/docs/index.php>
- [5] J. Hunter with W. Crawford, "Java Servlet Programming" published by O'Reilly & Associate, Inc., 1998.
- [6] J. Safranek, Nucl. Inst. Meth., **A 338**, p. 27 (1997).
- [7] F. Wang, et al., PAC 1999, p. 3101.
- [8] P. Kuske, EPAC 1996, p. 887.