A Java Program for the Viewing of Ambient Data

Department of Energy

Science Undergraduate Laboratory Internship (SULI) Program

Andrew Fisher

California State University Chico
Stanford Linear Accelerator Center
Palo Alto, CA 94305

August 9, 2004

Prepared in partial fulfillment of the requirement of the Office of Science, DOE
SULI under the direction of Victor Serbo at Stanford Linear Accelerator Center.

Participant:

Signature

Research Advisor:

Signature

Abstract

The BaBar detector system has a large number of sensors and data feeds, called ambient feeds. These feeds are vital to the operation and monitoring of the Pep rings and the BaBar system. In order to more easily monitor these systems a simple web interface to display graphical representations of the data is needed. Towards this end it was decided that using a Java Web Servlet (Sun Microsystems) would be an effective and simple way to achieve this effect. By combining Web Servlets with the Corba Technology (OMG) this provides a way for many people to access data from anywhere in the world. Using this type of program in conjunction with the HEP AIDA systems for graphing makes a powerful tool for the monitoring of the BaBar system. An overview of the BaBar system, and how the Ambient data is given. The uses and limitations of this method for viewing the data as well as examples of other ways to access the data and potential other uses for the servlet are also discussed.

Introduction

Although the most interesting part of the data coming off of the BaBar detector is the data about the particles being gathered for the experiments by the detector itself, there is a wealth of ambient data which is generated continuously, and is useful for the operation of the detector. Although for detailed analysis a dedicated viewing and fitting program like JAS3 is most likely to be used, many times one would just want to look over data for the past several hours as a kind of easy check on the status of the BaBar system. Also to show a summary of data for a presentation something easier to call on is useful. This is where the Corba system and web servlets come into play.

Our work is focused on creating a simple web servlet program designed to output a simple histogram when called on from any website. Basically, once the servlet is set up, all a person would have to do would be to click on a hyperlink and they would immediately have a histogram of the data they want to view. This will allow people to access the ambient data from any computer in the world provided it has an internet

connection and web browser. Another advantage of the servlet technology is that just by using a simple query string, the additions after a question mark on a URL, a user can specify a number of variables allowing them to decide what data they want to get, how the data is displayed and other formatting decisions. This allows a fairly large number of variables to be specified by a simple hyperlink.

The data which we will be working with is the BaBar ambient data in the AIDA format, which is accessed through a Corba server. AIDA, which stands for Abstract Interfaces for Data Analysis, is a universal data analysis system which gives you a number of functions to call on from many different programming languages. The Java implementation, called JAIDA, is what we will use to create the servlets which will output the ambient data.

There are many types of data coming off of the ambient server which can tell users things about the operations of the BaBar detector. You can get anything from radiation levels in the detector, currents on the beams, beam sizes, coolant temperatures, electronics rack temperatures, currents in the drift chamber, and any data you could want to monitor the operation of the BaBar detector.

For a system like this to work you have to make certain choices about what decisions you want to make at the user side and what things will be specified at the server side. What must be decided is which of the choices it makes more sense for the server to specify, for example path of the histogram must of course be decided user side, but what about the axis labels? On the one hand, some users might want to specify the labels, but on the other hand having the server specify the units is a quick way to have accurate

labels. The result of this is that one must pick and choose which options for users to control and which ones to make automatic.

Once created though, this will be an easy to access; versatile way for people to make overviews of the ambient data off of the BaBar server as well as something that could be adapted to anything that has a data in something like the AIDA format.

What is Ambient Data?

Ambient data is just a quick way of saying all of the data being pulled from the different sources that isn't being output by the detector. In other words, it is all of the data coming out of the BaBar system which isn't actually the data being used for the experiments. This data can take many different forms, and has many different uses. For example, one of the primary uses for the data is in the tuning of the beams by the control center. With accelerators, there are two things that you would like to get as high as possible, intensity of the beam, and energy of the beam. Unfortunately there are few fine tuning methods to get a higher energy beam making ambient data less than useful for this. However there are many different small ways to get higher intensities. By focusing the beam into the smallest possible area, the intensity at that point is higher for the number of electrons being accelerated. By continuous monitoring of the beam height width and position using the ambient sensors, the conditions which make for a very tightly focused beam can be determined and recreated which will increase the efficiency of the beam. Another use is that of monitoring and spotting. If an operator can just look at a quick graph and see what the currents have been on a beam line for a day, they can very quickly see if anything is wrong, thus speeding up the process of problem solving. Which is how

this program comes in; it provides a quick way of looking at sets of data graphically, allowing for quick interpretation of the data.

Reasons for Java

The main reason for using Java can be stated simply: it is the only programming language which is functional on many different operating systems. One of the problems with most programming languages is that it is very difficult to write a program that can be easily ported over to a different operating system. With a language like C++ one has to be very careful when writing a program in order to get it to work on both a PC and a Mac for example. Now, since the two processor schemes are very different, the compilers don't quite interpret commands in the same way, so when porting a piece of software over from a PC to a Mac you have to carefully write libraries for the compiler to use. With Java however you don't have to worry about this. The reason for that is Java does not compile to machine code. Instead, Java compiles to a pseudo machine code to be used by the Java Virtual Machine. This JVM then live converts the code that it gets from the compiled software into commands which can be understood by the machine one is running it on. Another of the major advantages to the Java system is the ability to run the program on the server rather than on the computer of the User who is trying to access the data. Using what is called a Servlet or Java Server Pages, you can write a program that will run itself on the server and then output data directly to the user's web browser, thus eliminating the need for the user to install new software on his computer in order to run the program. Finally there is java's versatility. By utilizing libraries that already exist one can easily create a program which includes the functionality of a pre-existing open source

program with the features and functions that users are looking for. This greatly simplifies matters as one can include functionality without having to rewrite large bodies of code which is possible with some other languages. The last reason for Java is its ability to seamlessly include HTML code into a Java page. This is useful because it allows for simple formatting and the output of the program using a markup language that can be read by any web browser. All these things combine to make java a very powerful tool and lead to the choosing of it for this project.

The piece of Java that was utilized for this particular package of programs was the Java Server Page, or JSP. A JSP is a type of code which combines both HTML and Java as well as XML and custom tags into a single program. As one site puts it:

A server-side technology, Java Server pages are an extension to the Java servlet technology that was developed by Sun. JSPs have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the static elements—the actual design and display of the page. Embedded in the HTML page, the Java source code and its extensions help make the HTML more functional, being used in dynamic database queries, for example. JSPs are not restricted to any specific platform or server. [6]

In other words it allows you to seamlessly include both HTML markup language and Java code into a program with ease. This was utilized in this particular program because of the need to create simple HTML interfaces for the user, including scripts and other elements, but still include Java code to perform certain functions like getting data from query strings, defining variables and performing if else statements. JSPs also give you the ability to write tag objects that will execute blocks of code to achieve some effect without cluttering the program. The combination of these powerful aspects made JSPs the perfect choice for writing the program.

Purpose of this Program

It was determined that there was a need for a program to view the ambient data from the BaBar server in a simple graphical format which would allow a quick overview of said data on computers around the world without installing software on the users computer. To this end a Java program was conceived that could take in the data, along with style choices, and output a graph, and allow you to save it in a variety of graphical formats. This program has a variety of uses from beam monitoring, experiment monitoring and maintenance data, to presentations on the accelerator. Since it is a general purpose utility, it can be used for a variety of purposes. Hopefully it will become a powerful and versatile tool for the viewing of BaBar ambient data.

Steps of Creation

The first step in creating the software was learning the interfaces available in the area. The code I was working came in two forms, one was the Abstract Interface for Data Analysis, and the other was the FreeHEP library. The FreeHEP library is a large collection of software, and information on that software, which is available and designed to be useful for high energy Physics [1]. AIDA, which is part of the FreeHEP library, is a set of interfaces for several different programming languages which makes it easier to analyze physics and other types of data [2]. Utilizing these tools, we began the process of creating a useful piece of software. The first difficulty that encountered was the fact that many of the different packages which we used were not quite what we were looking for with this system. For example the main plugin which was used to retrieve the data from the server was originally designed for use with a program called JAS3 [3]. So modifying this was one of the first steps to creating the program. Once past the technical details of

the project there was the major problem of determining which style options would be useful as well as how to specify the date. Part of this was accomplished by asking a large number of people what style choices they would like in the graphs. For the date it was just decided to give the user the choice of either specifying a time interval back from a given date, or just two dates and times. Finally we went through and determined a number of defaults to be set so that data would be in an easily understandable format even without specifying large numbers of choices. After completing this it was just a matter of cleaning up the code and adding in comments.

Discussion

This program will be a useful and viable tool for the quick summarizing and analysis of BaBar ambient data. Not only that, but with the large number of style choices available to the user as well as the capacity to save in multiple different formats, it will also be useful for the inclusion of data into presentations and web sites for the BaBar collaboration. Also, due to the modular nature of the program itself, it would be simple and possibly valuable to adapt it over to use for different applications as well. As it stands it is able to access data from any CORBA data server and could be fairly easily adapted to work with other systems that use the AIDA data format. So with both modularity and formatting options this program should be a useful tool for data analysis in the future.

*example of versatile formatting and color choice

*Example of histogram bars

*The Interface itself

Ambient Plot Styles - Microsoft Internet Explorer

Connect Points:
☒ Connected ☐ Not Connected

Show Histogram Bars:
☒ Hide ☐ Show

Fill Histogram Bars:
☒ Empty ☐ Filled

Show Statistics Box:
☒ Show ☐ Hide

Marker Size:
Marker Type:
Line Color:
Marker Color:
ForeGround Color:
BackGround Color:
Histogram Box Color:
Histogram Fill Color:

*Style Options

Acknowledgements

Finally I would just like to thank a few people that made this possible. First off my mentor Victor Serbo, who put up with a large number of programming questions and took the time to help me put my program together. Also thanks to the SULI summer program for making this possible. My advisor Dr. Gaffney for helping me out with references and support. My good friend Matt Muelenaer without who's help I would have been a little lost for programming references. And Finally Andre Bach, my office mate, who put up with me for the whole summer.

- [1] (2004 , August 3). FreeHEP [Online]. Available: <http://www.slac.stanford.edu/FIND/FHMAIN.html>
- [2] (2003, October). AIDA [Online]. Available: <http://aida.freehep.org/index.shtml>
- [3] (2004). JAS3 [Online]. Available: <http://jas.freehep.org/jas3/index.html>
- [4] David Flanagan, “Java in a Nutshell,” 3rd ed. Paula Ferguson Ed. California: O’Reilly and Associates, 1999
- [5] Ian Graham ,“HTML 4.0 Sourcebook” 4th ed. Cary Sullivan Ed. Wiley Computer Publishing, 1998
- [6] “White Paper Glossary” [Online]. Available: http://www.greensuite.com/whitepapers/whitepapers_03.html#jsp