History Data Collection, Retrieving and Display in the NSLS Control System *

Y.N. Tang and J.D. Smith, National Synchrotron Light Source, Brookhaven National Laboratory,
Upton, NY 11973

Abstract

This paper presents the history data collection, retrieve and presentation subsystem of our control system. We made every effort to to make the software friendly, convenient and robust. It is widely used and has become a very valuable diagnostic tool. Anyone may start and stop his own history at any data rate up to 20 Hz. The data can be periodically written to disk files or stored in a cyclic ring buffer and dumped to a disk file upon request. All the history data files use the self-contained and self-explanatory standard format. One graphical display program displays all the history data files because of their standard format. In addition to the standard features of a plot program, this display program can display up to 16 devices on both axis, take differences with a reference point, browse the data file by pages or a scroll bar, zoom-in and zoom-out any portion of the plot, do many types of analyses, etc.

I. INTRODUCTION

The history data archiving and retrieving is a very important subsystem in every accelerator control system. When we started to upgrade our control system several years ago, we decided to revamp this subsystem and we paid most of our attention to:

- It should be very flexible and easy to use to make this subsystem available and accessible to everyone;
- It should have a convenient and powerful program to display and browse the history data graphically.
- The history data should be readily available to other programs such as the orbit analysis program set.

In our old DG-based control system, we also had the history data collection processes running all the time. However, in order to look at the huge data set, one had to print it out. It was very time-consuming and tedious to check a thick stack of printed text papers. The limited display capability severely limited its use.

With the successful completion of our control system upgrade, the newly designed and developed history data subsystem has realized our design goals and become a very valuable tool in the daily operations and diagnostics. It plays an important role in the determination of the causes of beam dumps and other unexpected beam movements. The people in the RF group, power supply group, radiation and safety group, interlock group, Vacuum group and other groups use the history data to do different kinds of analysis, get the daily radiation doses for many locations, evaluate equipment performance, and detect equipment problems etc.

The following are the key elements of the history data system:

- a self-contained and self-explanatory standard file subsystem;
- two data collection programs: one for the normal history and the other for the fast history;

- a general graphical display program to browse and plot the history data. It also converts the data files of standard format to other formats used by commonly-used packages and data analysis programs.

II. TWO TYPES OF HISTORY DATA COLLECTION PROCESSES

Generally speaking, the history data collection processes produce huge data sets. In order to save disk space, we have two types of history data collections.
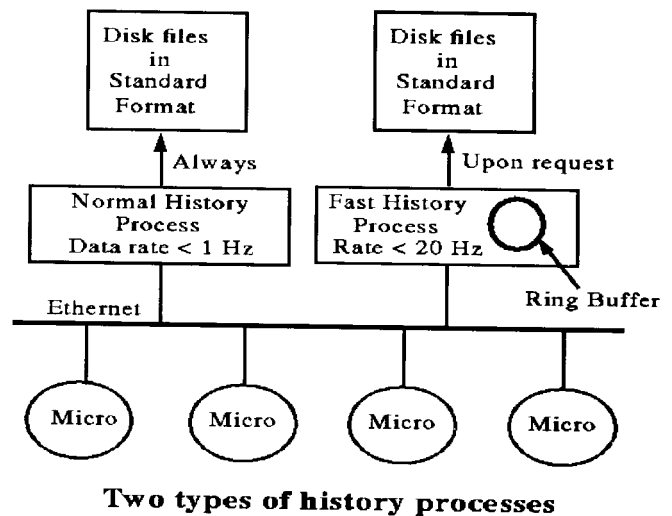

Figure. 1. Two types of history data collections: the slow history and the fast history.

- The fast history collects data in a ring buffer, which is then dumped to a disk file upon request. Its collection rate can be as fast as 30 Hz. For example, the X-ray and VUV orbit is tracked by two fast histories at 5 to 10 Hz rate.
- The slow history collects and writes data to disk files at the required rate. The max rate is 1 Hz. Each slow history writes data to a specific subdirectory in the history disk, generates one file per day and uses the date as file names such as Mar1595.

All the data files generated by both fast and slow histories use the standard file format.

III. THE STANDARD FILE FORMAT

When we started to upgrade our control system, we designed and developed a so-called "self-contained and self-explanatory NSLS standard file subsystem" (we call it STF in the following text). At the beginning, we planned to apply this file format to all the applications. If done so, every application could read/write any files produced by other applications without the prior knowledge of the file format and structure. It has been proved that our

plan was a little bit too ambitious. Yes, it could be done. But it needs too much effort. However, the STF has made our history data collection and display subsystem a real success. Without the STF, it would not be possible.

The STF subsystem contains:

- a set of careful designed formats and rules for the file;
- a library containing functions to open/read/write/search the file. There should be functions to read/write anywhere in the file.

An STF file has a file header followed by several segments. Each segment has a segment header followed by the data. The file header consists of several sections and segment headers consist of a parameter section and several blocks. Some blocks are mandatory and some are not. Programmer may add their own blocks into these headers. Generally speaking, these headers have all the information needed to read/write the STF files.

As a matter of fact, the STF files are similar to database tables with the exception that an STF file may have many segments and each segment has its own table. Usually the numbers of rows and columns are different form segment to segment in an STF file. In the header of each segment, one may assign a label, display format, data physical type such as setpoints, limits etc for each column in the segment. Any data types including structures are allowed to form a column.

In the STF function library, one could

- get all the information in the file header and segment headers. For example, one may call *stfgetlabel* to get the labels of the columns in a data segment. These labels could be device names.
- read/write any columns by column index, by label, by physical type etc.
- read any portion of a segment. This is very useful when an application is browsing the file back and forth and zooming a section of data.
- update any portion of an existing segment or append data to the end of a segment.
- insert columns and/or rows to an existing segment.
- delete columns and/or rows from a segment.

Though it sounds that this is a very complicated system, it in fact is not. We paid most of our attention to its efficiency. Because the history data files are generally large files (most are of several megabytes, some of them more than 20 megabytes), the display and browser program has to search back and forth many times through these files to display them. Therefore, the efficiency is one of the most important considerations.

We use one program *hbrowser* to display and plot all the history data. When hbrowser opens a history data file, it fetches all the device names (column labels) whose data are recorded in the file from the segment header. Then hbrowser presents these devices and asks the user to select devices to display. In addition to it, the program hbrowser fetches all the necessary information from the file and segment headers to read them correctly. Therefore, one program can display all the history data files.

IV. THE HISTORY DATA COLLECTION

We have two programs to start the history data collection. One is for the slow history and the other for the fast history. To start a history is a very simple matter: the major task is to make the input data file to the history program. The file contains all the needed information including the data rate, device names, the data fields of interest etc.

Now we have about 20 history processes running at different data rates (from 0.01 Hz to 20 Hz). Most of them are started by ring managers, physicists, technicians and engineers.

V. THE HISTORY DATA BROWSE AND DISPLAY

There are two keys to the success of the history data subsystem. One is the standard file format, the other is the general graphical display program hbrowser.

History processes produce big sets of data. In order to make the history system really useful, how to display and present these huge data sets is essential.

In addition to all the common features of a plot program, the program hbrowser

- has two y-axises. Each y-axis can display up to 8 devices.
- uses time or any other device data such as the beam current as the x-axis.
- zooms any portion of the display by click and drag.
- displays any portion of the data set including the whole data set. To select the desired portion of data, use the mouse to move the line-marker on the picture.
- selects any point as the reference point to display the difference.
- uses push buttons or slider to browse through the whole file.
- provides filters and statistics calculation facilities.
- translates and rescales the picture.
- converts the history data files to other formats used by other programs and packages. For example, it generates data files for c-plot and orbit correction and analysis programs.

There are many other features not listed here. We have three sample displays. Figure 2 is the radiation history display One may see the radiation increases during injections and whenever the booster is on. The top-right text panel gives the total radiation counts of the day. Note that it plots the difference with the reference point at the start of the day. Figure 3 presents a picture of a portion of a fast history, from which we detected the cause of a beam dump in the early morning of Jan 31, 1995. The cause is that X17 (wmainamp) tripped. The data of wmainamp, the X-ray current and three PUEs are plotted. Figure 4 is another format to display the radiation data.

VI. ACKNOWLEDGEMENTS

References

[1] Y.N. Tang and J.D. Smith, NSLS Technotes, W-7, W-35, W-36 and W-37, 1990-1994.

[2] .D. Smith, S. Ramamoothy and Y.N. Tang, Proceedings, ICALEPCS 1993.
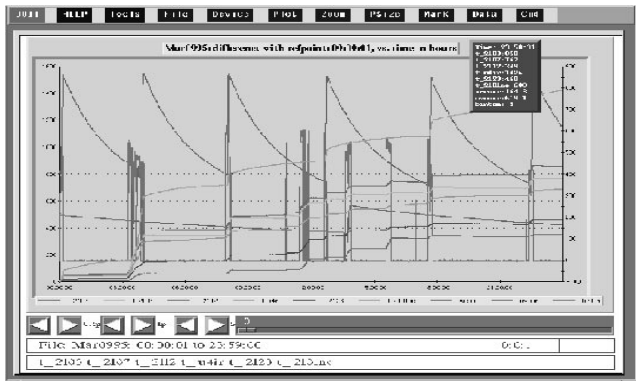
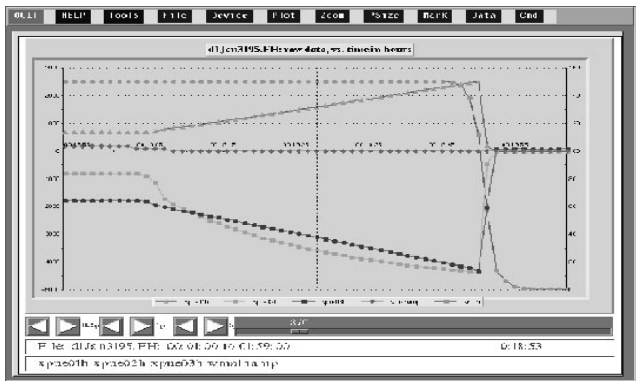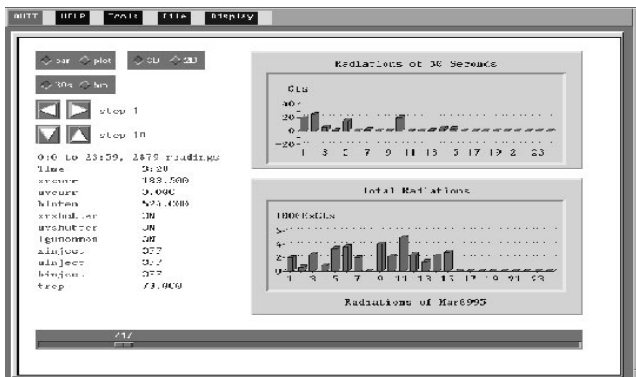Figure. 2. The radiation history display

Figure. 3. The beam dump caused by x17 trip

Figure. 4. Display history data in a bar chart