DATABASE USE IN APPLICATION PROGRAMMING AT SNS

J. Galambos[1], C. Chu[1], J. Patton[1], T. Pelaia[1], A. Shishlo[1]
[1]*SNS, Oak Ridge National Laboratory, Oak Ridge TN*

ABSTRACT

The Spallation Neutron Source (SNS) application programming effort uses database storage in several areas. A primary use is in the configuration of the accelerator hierarchy used in our Java based application programming framework. This task involves considerable effort in data population, but facilitates the ability to write general purpose applications. Configuring the machine definition in this approach for the entire application programming framework also eases code maintenance. Another area of database use is in the storage of measured machine values. This later usage case typically involves working with collected sets of data, and the data population tends to be automatic. Several tools have been written to hide the intricacies of the database connections to programmers. Experience in both of these areas is discussed.

ACCELERATOR HEIRARCHY

The Java based high level application programming framework at SNS [1, 2] includes a hierarchal view of the accelerator. In this view, shown schematically in Figure 1, the accelerator is composed of sequences, sequences are composed of devices, and devices can have their own characteristics. This is a typical object-oriented view of the accelerator, and in our case the hierarchy is configured from database tables. One advantage of this structure is the ability to write an application once and have it reusable anywhere in the accelerator. Another advantage of this approach is that single point changes in the database configuration are propagated to many applications, easing maintenance. A consequence of this approach is the necessity of careful population and maintenance of the database.

The tables are part of a global SNS relational database [3]. As this global database is used by other groups at SNS, the table structure is not a one-to-one match with the class structures used in XAL. Much of the information needed for the setup of the physics applications is “mined” from tables maintained by engineers. We have written a program that queries the database to configure the XAL structure. The overall organization of these tables is discussed below, as well as the approach used to setup the XAL accelerator hierarchy.

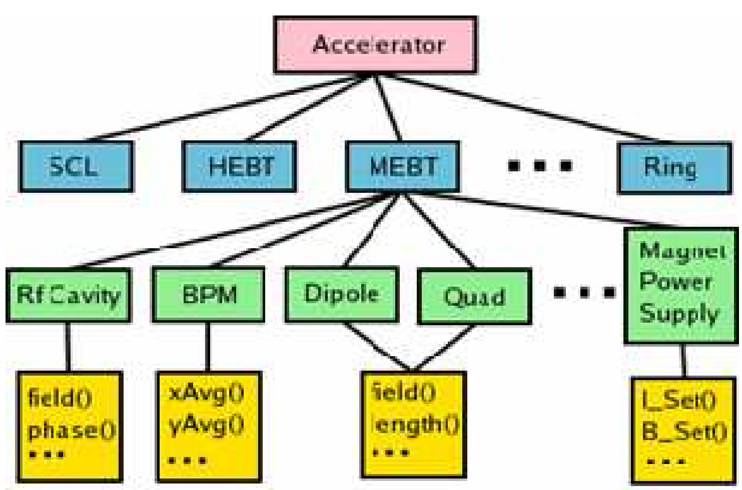


Figure 1: XAL Accelerator hierarchy indicating an accelerator composed of sequences (blue), which are composed of devices (green), which have methods and attributes (yellow).

Devices and Equipment

There is a high level delineation of data storage between devices and equipment in the SNS database tables. Devices are the abstract placeholders for objects that a beamline is composed of (magnets, RF, diagnostics devices etc.). The devices all have prescribed global coordinate positions (used by alignment personnel), a distance along the design beam trajectory (used by beam modellers), and are given names which generally follow a naming convention that allows easy understanding of what and where the device is.

Each device has a specific installed piece of hardware at any given time, which is the equipment. All pieces of equipment use a bar-code number, for their identification. There is an association table that relates which piece of equipment is installed at each device. The information specific to the hardware is associated with the equipment. For example, magnet measurement data is associated with the equipment, not the device. If a magnet fails and is swapped out, a new piece of equipment (i.e. the new magnet) is associated with the device. Much of the data associated with the equipments is held in tables maintained by various groups (magnet measurement, power supply, diagnostics, etc.) but is critical for the setup of the high-level applications. Sometimes information that is characteristic of a class of equipment is stored in tables for an equipment model type (e.g. power supply limits are all the same for individual power supplies of a given type).

We use data from about 25 tables to setup the XAL accelerator hierarchy. This is done via a programmed set of queries which presently takes 10’s of seconds, but has not been optimized. Rather than have every application perform this query on start-up, we create an XML file containing the accelerator configuration, in the structure used by XAL. Presently this XML file is over 20,000 lines long for the SNS linac alone. The accelerator configuration does not change often within a beam run, so the occasional updating of the XML configuration file has not been problematic. Also, storage of these files provides an easy way to reconstruct the accelerator’s setup for a given time in the past. Importantly, the device has types of signals associated with it, and the signal names which are used by the EPICS control system [4] for the real machine communication are specified in this file. It is this link that allows XAL to provide generic device methods that hide the details of the control system from the user (e.g. a magnet method to get the applied field, of a Beam, Position Monitor (BPM) method to get the measured beam location).

MEASURED DATA STORAGE

The EPICS control system provides tools for archiving and retrieving large amounts of data, and is the primary mechanism used for data archiving at SNS. Mirroring the EPICS control system, the standard method of archiving information is oriented around individual signals. This is useful and works well when the different combinations of signals to be retrieved are not known a priori. However, there are occasions when collection of prescribed collections of signals is anticipated. An example is the collection of all information needed to set up a model from machine settings and compare to measured beam, e.g. the magnet settings, RF settings, and the BPM measurements (“referred to as “physics” data). While it may be possible to collect the hundreds of signals needed to for this task from the EPICS archiver, this would be tedious. Moreover, there would be no guarantee that all the values retrieved were from the same beam pulse (SNS is a pulsed machine that will run at 60 Hz). A PVLogger tool has been written to facilitate logging groups of data (PV stands for Process Variable in EPICS, which is like a signal). The important distinction between this tool and the standard EPICS archiver, is that this tool collects data as a group, and can guarantee that all data within a group are collected from the same pulse, by requiring signals to have timestamps within a certain range of each other.

This capability is available as either a background service that runs continuously in the background, or as a pushbutton tool. We log the physics data every fifteen minutes in the background service, and other applications that take measurements, such as wire profile measurements and scanning applications take snapshots whenever measurements are taken. There is a separate application to browse for a group snapshot by time, and to view the value of a logged signal vs. time (from one snapshot to another), and is shown in Figure 2. Also another application provides capability for on-the-fly setup of a new group of signals to be logged, as well as specification of the frequency to log the data .

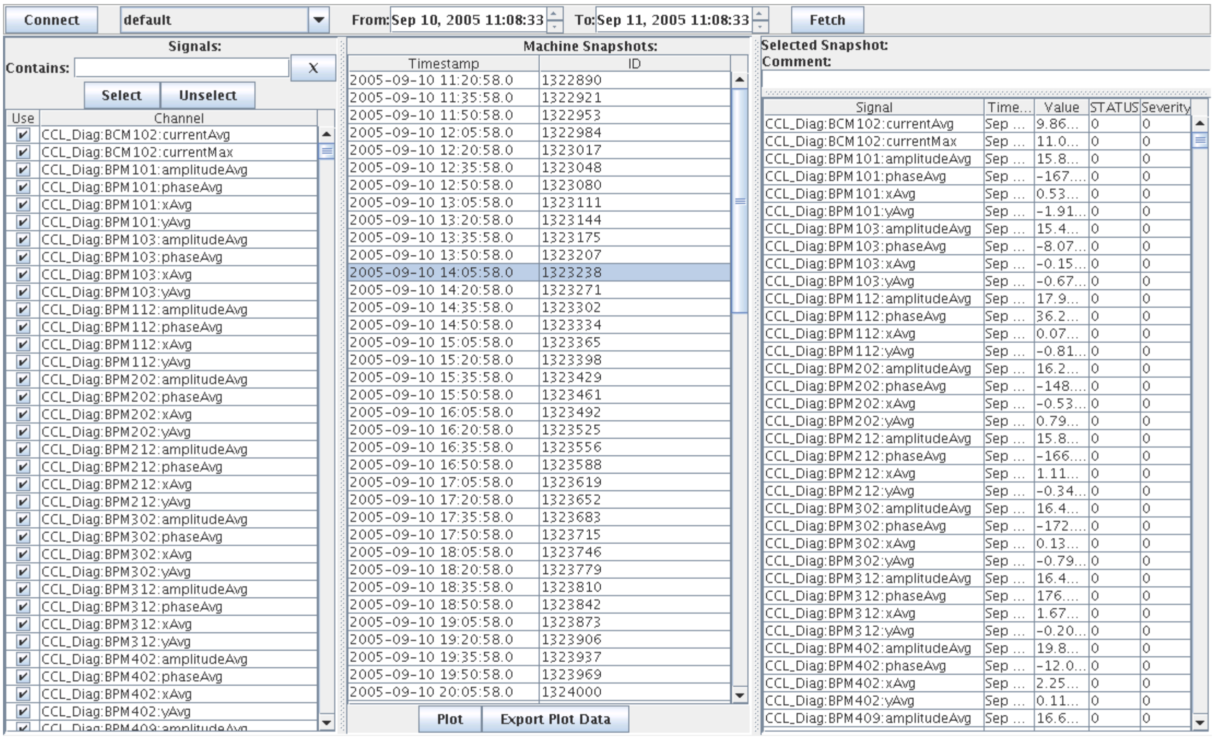
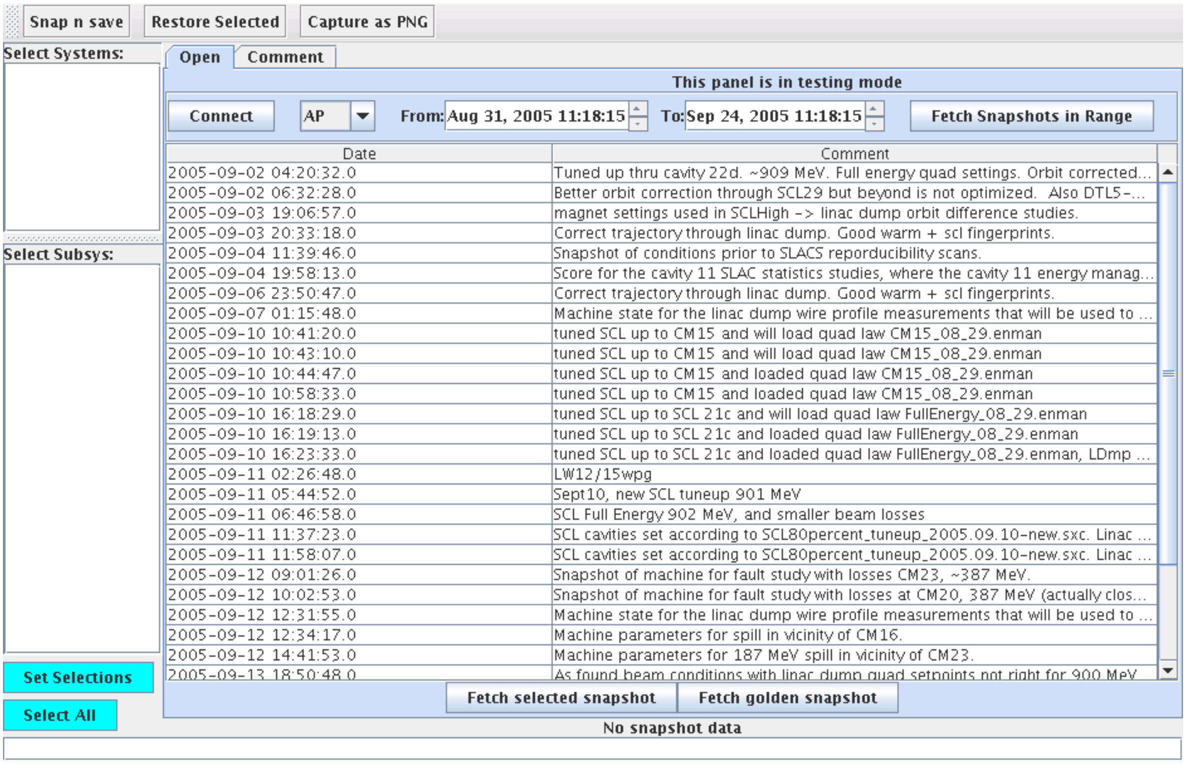


Figure 2: Browser application for retrieving PVLogger snapshots. The group to select (filterable) is shown in the left panel, The available snapshots in the specified time-range are shown in the middle panel, and the specific snapshot is shown in the right panel.

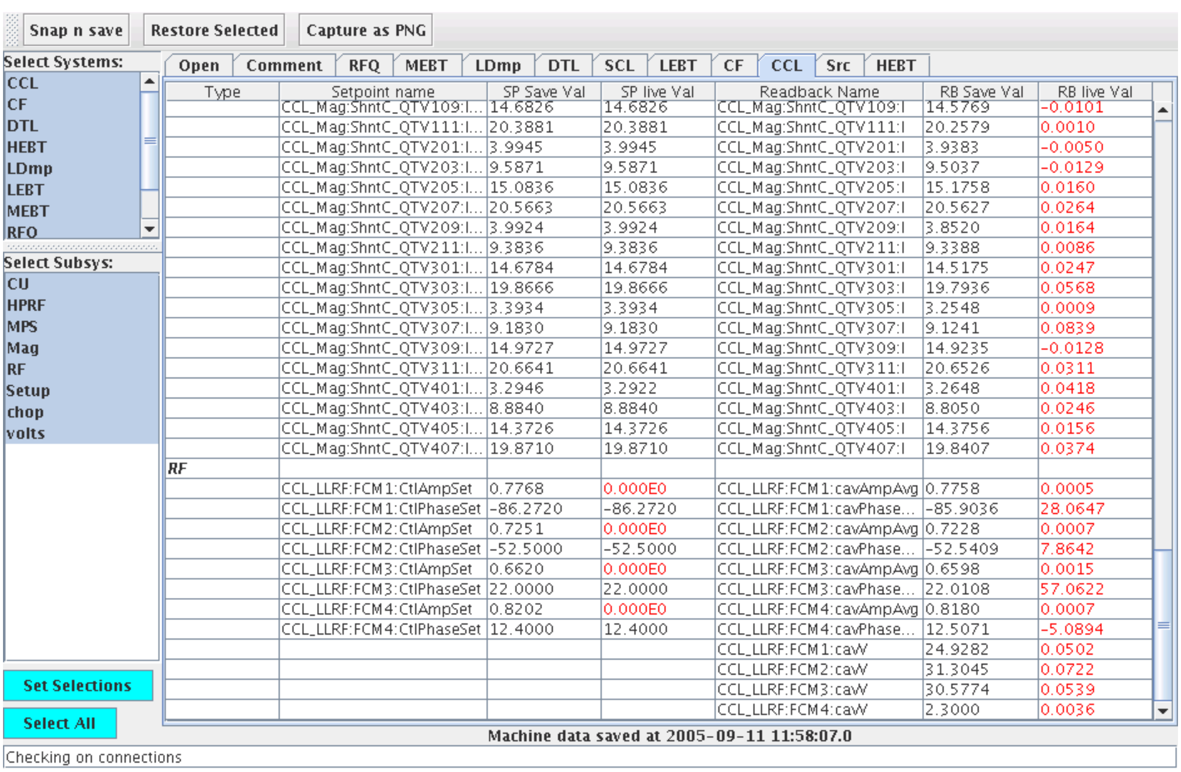
By far the most important application of the logged data is configuration of the online model [5] of the accelerator. The SNS online model is a simple envelope model used for transverse and longitudinal lattice calculations. It is fast enough to be used in the control room. While the basic lattice geometry is fixed and can be configured from the database as described above, specific setup of a large accelerator changes, especially during commissioning periods as with the SNS. For example, an operator may tweak a quadrupole. In order to analyze data it is important to know the machine setup when the data was collected, and the PVLogger provides this capability. As mentioned, applications that take measurements typically also request a PVLogger snapshot, and store an identification number of this snapshot with the measured data. The online model has a PVLogger data-source which sets adjustable model input (e.g. magnetic field levels) to those of a given PVLogger snapshot. This capability is shared by several applications that use the online model for data analysis and machine setup, as well as single-use scripts.

SAVE-COMPARE-RESTORE (SCORE)

All large accelerator systems have some sort of save, restore capability for tuning the machine. It is always desirable to be able to easily recover a good machine setup, after (perhaps unsuccessfully) attempting to improve things. At SNS an application called SCORE provides this capability, and it uses a database for all data storage (see Figure 3). Collections of data that are desired to be viewed together are setup as a group. Examples of groups that have been used so far are the general operator setup of the machine, diagnostic device configuration, and some RF equipment configuration. The groups consist of defining sets of setpoint and readback signals, each with a system and sub-system identifier. These identifications can be used to filter query results. A collection (snapshot) of each group can be taken, and is delineated by the timestamp of the snapshot.



(a)



(b)

Figure 3: Score application screenshots. (a) The panel used to select a group, and snapshot to fetch. (b) The compare feature for a specific system. Items in red indicate live values more than a specified amount different than the saved value. Individual rows can be selected restoring saved values.

Originally this application used an XML file based system for the storage of snapshots. Transition to the use a database has offered several benefits. First, the maintenance of the many files taken during a given run is eliminated. Also, previous snapshot retrieval was based on file browsing (using a file name protocol to identify when the data was taken). Use of a database to store information provides flexibility of offering the user more than just the timestamp of a machine snapshot for data retrieval, e.g. now we also offer a short text description in the snapshot browsing feature. To do this previously would have required opening and parsing of many files.

COMMENTS

The high level applications programming effort at SNS is using a database as the primary means of machine setup configuration. This design choice was made early on, and has had several ramifications. One impact is that a significant effort has gone into data-population. This effort has paid dividends already, in the ability to deploy general purpose applications. Also, critical machine information is stored in a central repository, instead of personal spreadsheets, or files on different computers. Regarding the use of a database to store measured machine data, this also has been a positive experience. In addition to the flexibility of using database storage of information, maintenance is easier – files in multiple directories no longer have to be managed. We anticipate increasing the database use with the high level applications.

ACKNOLEDGEMENT

SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy. SNS is a partnership of six national laboratories: Argonne, Brookhaven, Jefferson, Lawrence Berkeley, Los Alamos, and Oak Ridge.

REFERENCES

[1] http://www.sns.gov/APGroup/appProg/xal/xal.htm.

[2] J. Galambos, *et al*., “XAL Application Programming Structure”, Particle Accelerator Conference, Knoxville, Tennessee, USA, May 2005.

[3] http://www-ssrl.slac.stanford.edu/lcls/epics/agenda.php

[4] http://www.aps.anl.gov/epics/

[5] C. Allen, *et al.*, “A Novel Online Simulator for High-Level Control Applications Requiring A Model Reference”, ICALEPCS’2003, Gyeongiu, Korea, October 2003.