Australian Synchrotron Technical Note AS-CIT-200904-01

The Flange for controls system to Internet application

Richard Farnsworth, Andrew Starritt,
with assistance from Chris Myers, Mike D'Silva and Nicholas Hobbs

21 April 2009

Abstract

A technique for delivering EPICS Process Variables (PVs) to web enabled applications has been collaboratively developed by the Australian Synchrotron controls team and a local e-research initiative the Victorian eResearch Strategic Initiative (VeRSI). EPICS PVs are collated by a program called the "Flange" which acts as a gateway and passes the PV values and selected metadata to a MySQL database which contains current values and optional short term historical records which then allows standard web based applications, in our case AJAX (Asynchronous JavaScript and XML)) web applications. The flange supports a Historical Backfill from the EPICS archive, throttling (i.e. minimum updates and dead banding) and finally image processing (useful for taking arrays and waveforms) and converting to standard image formats stored in the MySQL database.

1 Introduction

As with most other similar facilities, the Australian Synchrotron (AS) has developed a Facility Status Monitor (FSM) application to monitor the most basic control system parameters, such as beam current, beam lifetime, operating status, beamline shutter status and other vital parameters. This is shown in figure 1 below.

It was originally a stand alone executable that could be run from the supported operating systems within the facility and used by Beamline scientists and Accelerator operators and any interested parties to quickly determine over all operational status. This paper describes the process and development used to take that original application, generalise it, convert it to a web enabled application and then use that technology in a variety of other ways.

The original need gave rise to a Flange program which in turn takes data from the EPICS gateway and passes the value, selected metadata such as enumeration type, connection state, update times, element count, data type, Alarm severity, precision, Engineering units and high/low operating range to a MySQL database which contains short term historical records which then allows standard web based applications to be created.

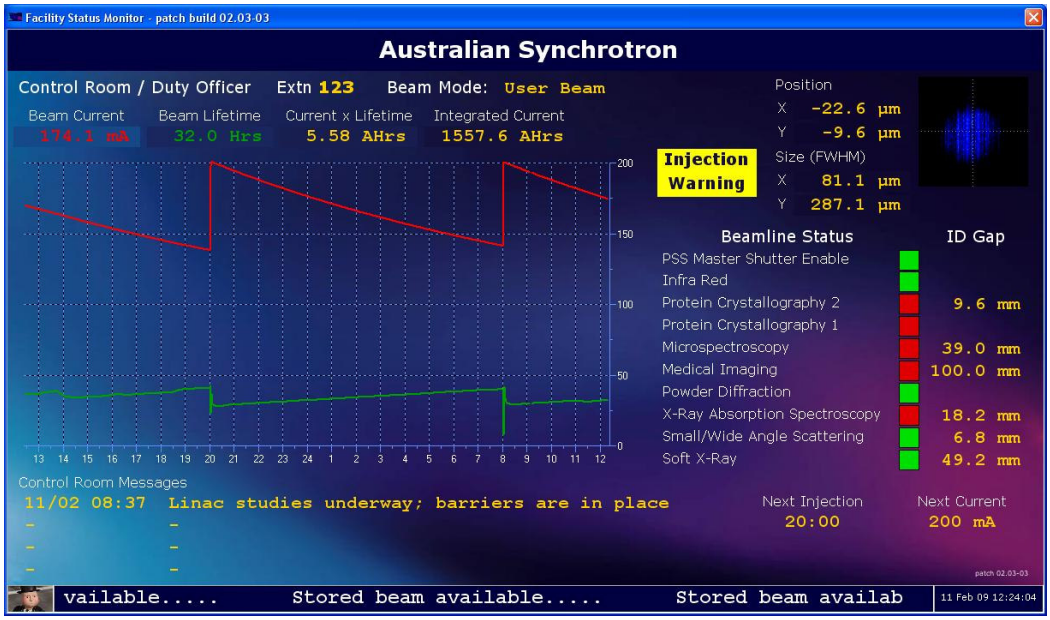

Figure 1 – the "original" FSM

2 Details

2.1 Requirements

When the FSM was created, the we wanted it to use the "corporate" colour scheme and contain a summary of the most important operational data that either an accelerator physicist/operator or a beamline scientist might wants to know at any instant in time.

The AS did not want to distribute a video signal around the facility – because of the cost of co-axial cable, the loss of definition and of the perceived "low tech" solution.

The AS also wanted to publish data on the Web, internally or externally, and wanted the ability to build arbitrary web applications in the future.

2.2 Prototype

The FSM program is (one of a number of applications) based on the AS OPI framework based around the Borland Delphi programming language. For the "first go" we "cut the head off" the FSM program and modified PV connect and update event handlers to issue MySQL update queries [1] rather than update visual components.

While this worked very well, it was not very generalised. A more flexible was required.

[1] We use the umy_sql software developed by Cristian Nicola, n_cristian@hotmail.com

2.3 Collecting Data

The modified FSM program was replaced by the EPICS to MySQL Flange (EMF) program. This used the same software to connect to, subscribe for PV data and to issue MySQL insert/update queries the modified FSM program. However, the PVs that are subscribed for are specified by configuration files rather than being hard coded into the program.

As well as current data table within the MySQL database, the EMF optionally supports arbitrary amounts of historical data for a PV as well. The configuration files specify the historical window and the interval (e.g. the configuration file specifies that we maintain a 24 hour window of beam lifetime data at 2 minutes intervals, i.e. 720 historical table data entries). The EMF program can also request historical data from the EPICS data archiver [2] so that we can backfill the historical data during initialisation.

The EMF program gets PV data via an EPICS gateway from the machine network. The gateway is read only, and therefore provides a certain measure of security. The program runs on the same box as the MySQL database which makes for efficient database updates.

We currently have 1551 PVs, although we have successfully tested up to about 20,000. As well as a basic PV update, we also have a number of extra functions available. This can be specified per PV and include:

a) Maintaining/backfilling historical data: We maintain 8 days of shutter data, and 1 day of beam current and beam lifetime data to support the web FSM;
b) Special processing for images. The X-Ray diagnostic beamline image is provides as 16K byte waveform record, which is converted to a JPEG image for insertion into the database as a single item;
c) Dead-band specification for numerical PVs (equivalent to the ADEL/MDEL field);
d) Minimum update interval specification. This specifies the minimum interval between MySQL database updates for the PV. Together with dead-band, this enables database update throttling to be applied if/when required; and
e) Database alias. Normally the PV name is used as the MySQL primary key. This allows the entry in the database to be given a different name.

2.4 Performance Issues

As stated, we have passed up to 20,000 PVs through to the database. This stressed the EMF program, but this has been mitigated by running three instances of the EMF program, each handling one third of the PVs.

As the EMF/MySQL programs run on a four core machine, this distributes the load over the four cores and improves performance. The specification of which instance and the total number of instances is specified by command line parameter, thereby ensuring flexibility and easy upgrade path (e.g. to an 8 core machine or two separate machines).

[2] We use the SNS maintained archiver http://ics-web.sns.ornl.gov/kasemir/archiver/

Also, there can be a large load at initialisation processing a multitude of PV connection events and (typically 720 or 5760 per PV) historical backing query updates. A mechanism for staggering the processing of these during program initialisation also spreads the load.

The performance of our web server is very good.

2.5 AJAX and Php

We access the MySQL database with a combination of Asynchronous Java Script and XML (AJAX) and php to create web applications – see figure 2.

The URL is http://vbl.synchrotron.org.au/fsm/

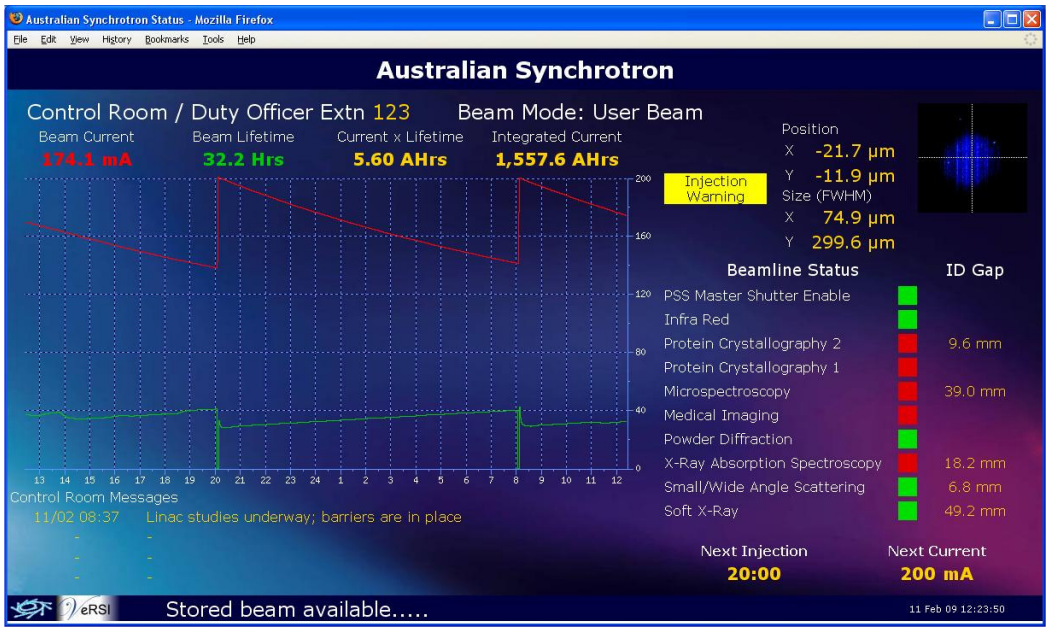

Figure 2 – the web FSM

AJAX allows the web browser to obtain updated information from the web server without re-loading the whole web page. Php is used by the web server to extract PV data from the MySQL database.

2.6 EDM Screens

A number of php programs have been auto created (using another php program) to process the EPICS Extendable Display Manager (EDM) edl files to created web based EDM screens and these are populated from the MySQL database.

Figure 3 shows an actual EDM screen shot while figure 4 shows the web based equivalent (the latter is still under development).

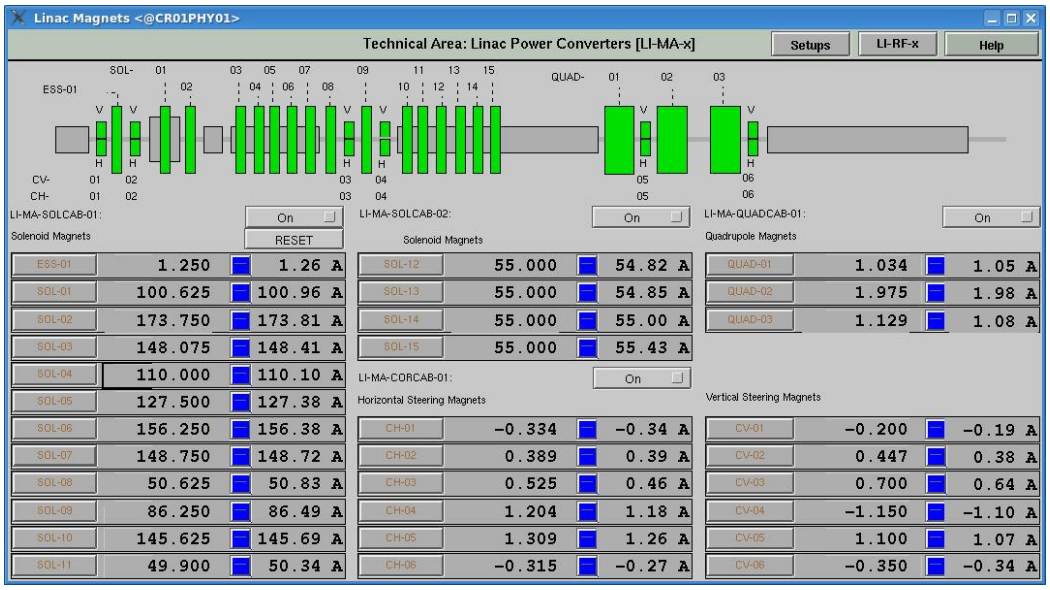

Figure 3 – EDM screen shot

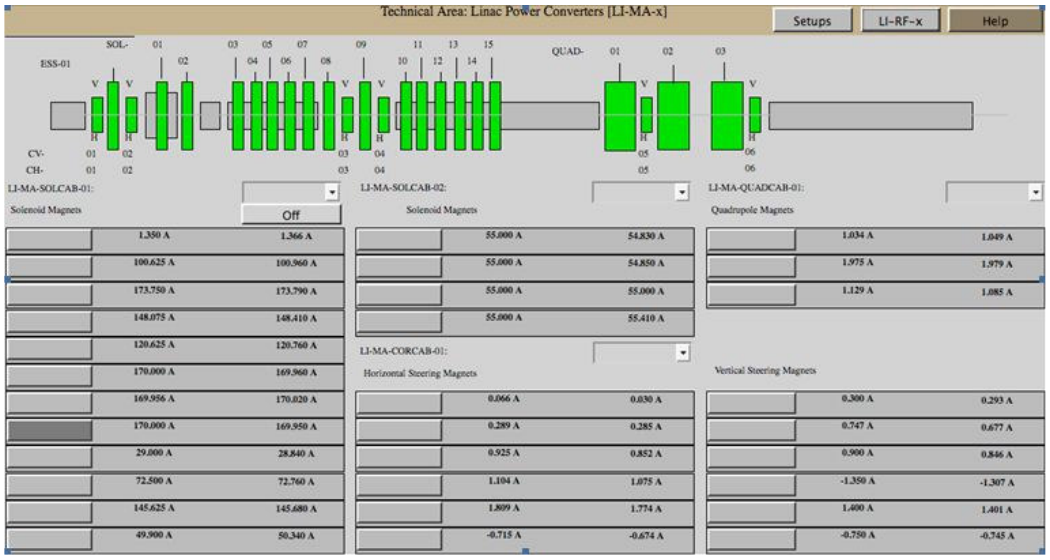

Figure 4 – web screen shot

2.7 Other Applications

Figure 5 shows the web FSM on an iPhone. Figure 6 show basic FSM status on an older style phone using WML format. There is no auto update suitable for low bandwidth access. The URL is http://vbl.synchrotron.org.au/fsm/index.wml.

Figure 5 – iPhone FSM

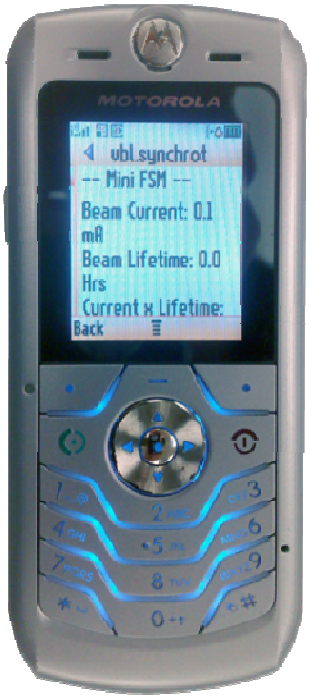

Figure 6 – WML format

2.8 MySQL Database Schema

The MySQL database schema is shown in figure 7 below.

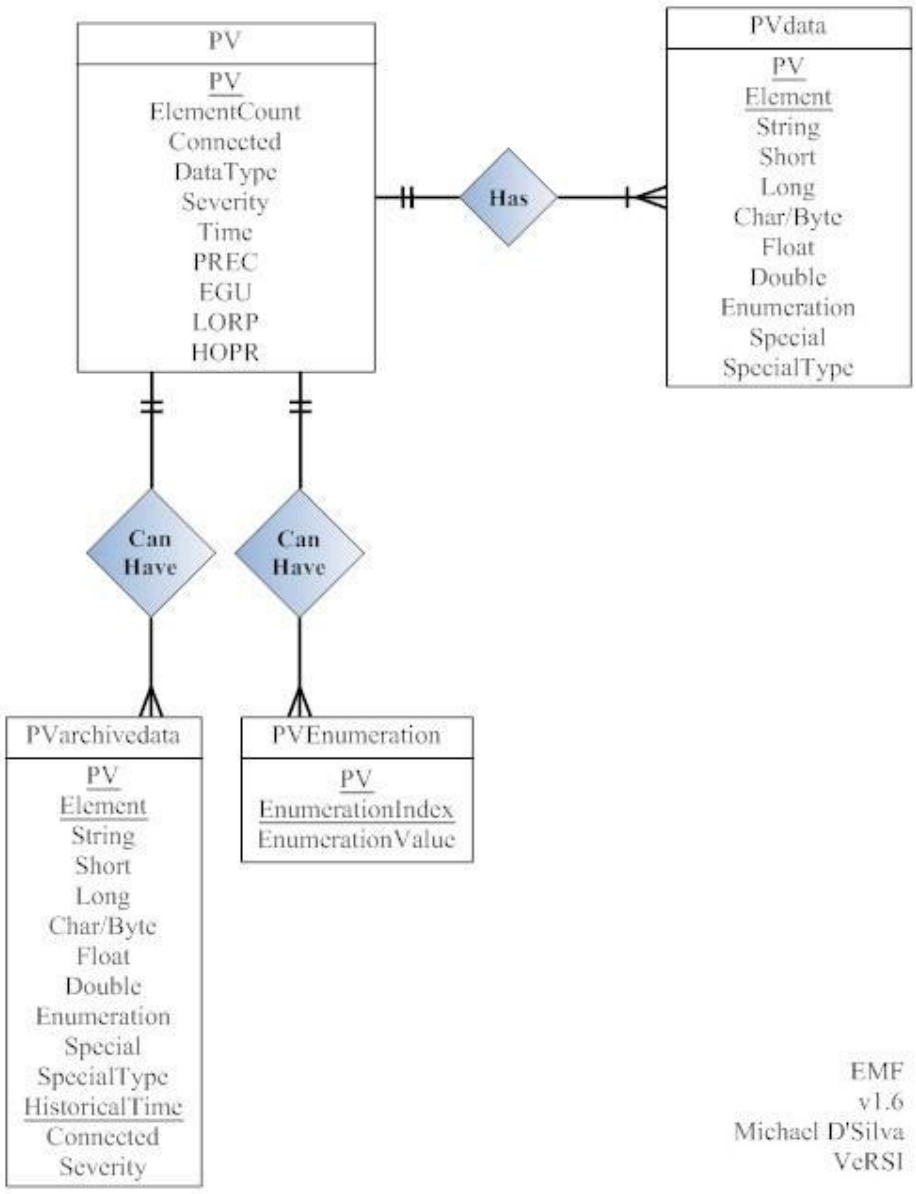

Figure 7 – MySQL database