PDV – a PVSS Data Viewer Application
D. Hoffmann, O. Pisano

To cite this version:

HAL Id: in2p3-00546068
http://hal.in2p3.fr/in2p3-00546068
Submitted on 13 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
The PVSS Data Viewer (PDV) has been developed to access environment and control data of the Pixel detector of the ATLAS experiment, with an effort to be sufficiently generic to provide access to data of other subdetectors and even data of other experiments or PVSS systems in general. Other important keys for the design were independency from any existing PVSS installation and universality regarding operation systems or user environments.

**Design Criteria**
- General/PVSS/PvssDb interface
- No detector or experiment specific features
- Universal (Java VM)
- No OS, environment or policy dependency
- WWW-able (Java Network Launch Protocol, JNLP)
- No dependency on PVSS installation
- Decompression and clever display
- Basic analysis functions
- “All possible” save-as options

**Technical Implementation**
A data flow diagram embedding the URL for the software design is shown below. The user interacts with the application mainly through the PDV dialog data parameter element name selector dialog and the display part of the currently selected display menu, which shows the time interval (period) of data that are being queried.

The internet of the data storage and caching of DPEs for schemes which have already been used in a previous session, are completely hidden from the user. A typical session from a user session shows the user selecting the date period for which data are going to be queried (normal Display 1). DPE definitions can vary with time and are taken into account in a way that is transparent for the user.

The DPE Selector window shows the schema and DPE structure in tree form. The relatively long DPE names are cut intelligently in order to arrange them in the tree. The DPE name display can be changed at any time (for display and selector) from DPE names to the PVSS alias convention.

**PDV – a PVSS Data Viewer Application**
Dirk Hoffmann, Olivier Pisan – Centre de Physique des Particules de Marseille, case 902, 163 avenue de Luminy, F-13288 Marseille CEDEX 09

The PDV application links user and database. The user interacts through the Display and DPE Selector components of the application with the PvssDb application.

**Context and Origin of Data**
The ATLAS experiment uses PVSS, an industrial SCADA application, for its detector control system (DCS). A custom driver establishes a connection to a central Oracle instance for the purpose of acquiring the recorded data in the PVSS Database (PvssDb).

The ATLAS DCS consists of 14 different subsystems, associated roughly to subdetectors, which contain a total of some hundred computing nodes in form of Windows and Linux PCs. The PDV application to the Oracle instance is presently filled by 11 of these subsystems with an allowed average rate of 3 GB per day, which translate into 5.10^6 values. This average daily data volume of 3 GB needs to be handled efficiently by any application that implements user queries.

**Data Decompression and (re)coding**
PVSS associates validity information with each value. In case of transient errors during data acquisition, data values are still recorded, but can be filtered according to a variety of error conditions. PDV allows to filter these data, i.e. include or exclude all or part of values with error conditions from the plots.

Depending on this level of understanding or needs, he has the choice between multiple options:
- raw data format (uncompressed from PvssDb, no reconstruction)
- decompressed format (sometimes time slice)
- optimized reconstructed format where duplicate lines (table values) are eliminated.

An interface to set filters, but difficult to implement due to the requirement of machine independency, which excludes PDV features in principle.

**Application Features**

**1: Invalid data**
Inval important data is tagged by PDV through a 32 bit mask for filtering of invalid data by PDV, fully configurable bit mask without any detector or experiment specific features.

**2: DPE search**
Selection of DPEs by specific algorithm or interface (possibly graphical)

**3: History**
Users typically verify same groups of DPEs routinely or, more generally, want to re-make a query that they have executed formerly. The PDV repeats the last executed queries for each user in the user-specific PVSS database (stashed in the Java ‘.history’).

**4: Export and Save Data**
The possibility of exporting displayed data after a query has been explored exhaustively and is probably the most complete feature of the PDV in its present state. Beside the standard graphical export:
- a printer (or PostScript as print-to-file) or
- a PvssDb graphics file

The user can export the underlying data of a display as compressed separated value (CSV).

**5: Plugins**
Accomodate subdetector/user specific requests
Selection of DPEs by specific algorithm or interface (possibly graphical)

**Data Export in user specific formats**
User provided code, stored in jar or class file in user directory

**Software management and frameworks**
The source code of the PDV application has been successfully managed with CVS in repository and Savannah for user feedback, bug tracking and task management. Migration from CVS to SVN has been performed by the Python script extension of its own utility framework without any impact on the project.

The Eclipse IDE has not been mandatory, but turned out to be the favored tool of all developers and contributors.

Source code is open, but property of CNRS/IN2P3.

The initially used graphics library JChart2D has been replaced by JFreeChart, as we expected it to provide better support and documentation, faster importation to bug reports and easier integration of bug fixes from our side.

**Experience and Future**
After the first year of development of the PDV application, many additional requests, in addition to the basic features have been made by the user community.

Some subdetector specific requirements have come up with the Pixel detector, and requirements for more user specific features are expected to expand beyond the four main limits. They have been consistently implemented as PDV plugins and are marked for full integration.

The PDV is now in a maintenance phase. ATLAS has decided to develop a web-based GUI for PDV requests, which uses part of the PDV code unchanged for the display of the data, which is considered a result of good encapsulation design.

The relatively low level access of the PDvssDb tables in Oracle is no be replaced by a common API provided by CERN EN-ICE. Our experience gained with the interface to the Oracle instance and the PvssDb utilities has left its mark on this central development.

**References**
http://cern.ch/twiki/bin/view/Atlas/ATLAS
http://pvss.at