Dcs Data Viewer

A Flexible Interface to the ATLAS DCS data

C. Tsarouchas[1], S. Schlenker[1], S. Roe[1], U. Bitenc[2], M.L. Fehling-Kaschek[2], S. Winkelmann[2], S. D'Auria[3], D. Hoffmann[4], O. Pisano[4]

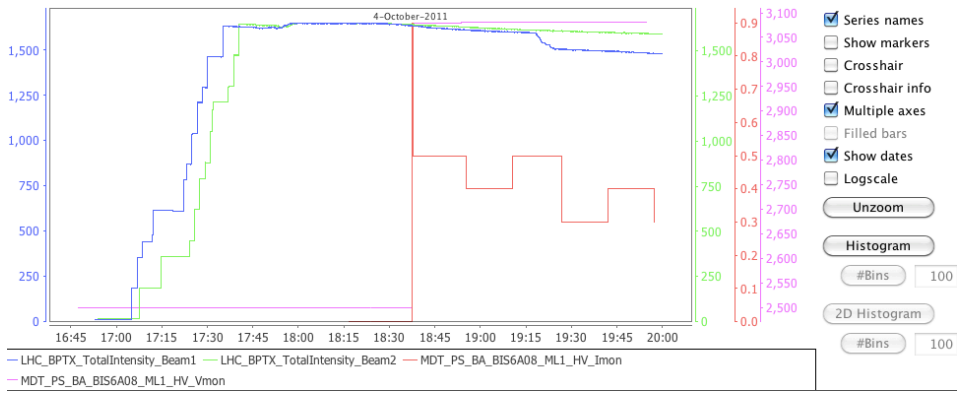
1) CERN, Geneva, Switzerland 2) Albert-Ludwig University Freiburg, Freiburg, Germany 3) CPPM, Marseille, France 4) University of Glasgow, United Kingdom

Aim And Specifications

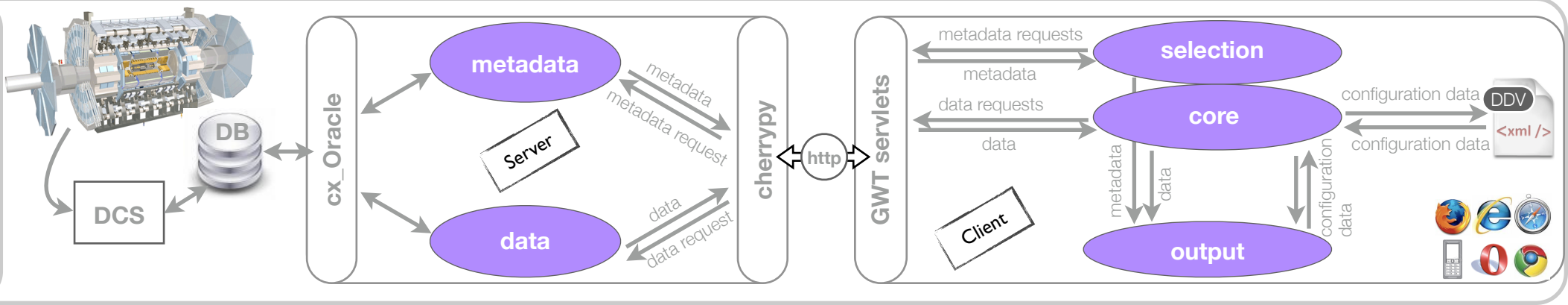
The DDV project aims for implementing a data viewing application for DCS data. The application targets users from the whole ATLAS collaboration allowing to access and display data from database archives. Below is the list of specifications.

- Platform and browser independent project
- Reasonable application startup time (less than 10sec)
- Small response time of queries (order of second)
- All possible navigation mode options (element name, alias, description)
- Multiple output formats (chart, tables, ascii, ROOT)
- Current configuration in XML format option
- Database protection mechanisms

chart output



Architecture



DDV Server

Metadata Organization

The ATLAS DCS metadata is information concerning archived data. The name of an archived parameter is called element name and it is a string that identifies that specific parameter in the oracle tables. Besides, each element name can have a description and an alias which are more user friendly definitions. A scheduled task connects to the DB and copies the metadata information from the database to an SQLite database disk file within the DDV server. The organization of metadata in SQLite file, serves two different purposes. Firstly the SQLite tables are configured to be stored in memory offering in this way the quickest possible response time of metadata queries. Secondly, all queries concerning metadata are performed exclusively inside DDV server keeping the database resources as avail- able as possible. Finally, DDV server takes into consideration that DCS metadata are not static by activating the metadata replication scheduled task once every 24 hours.

Data Retrieval

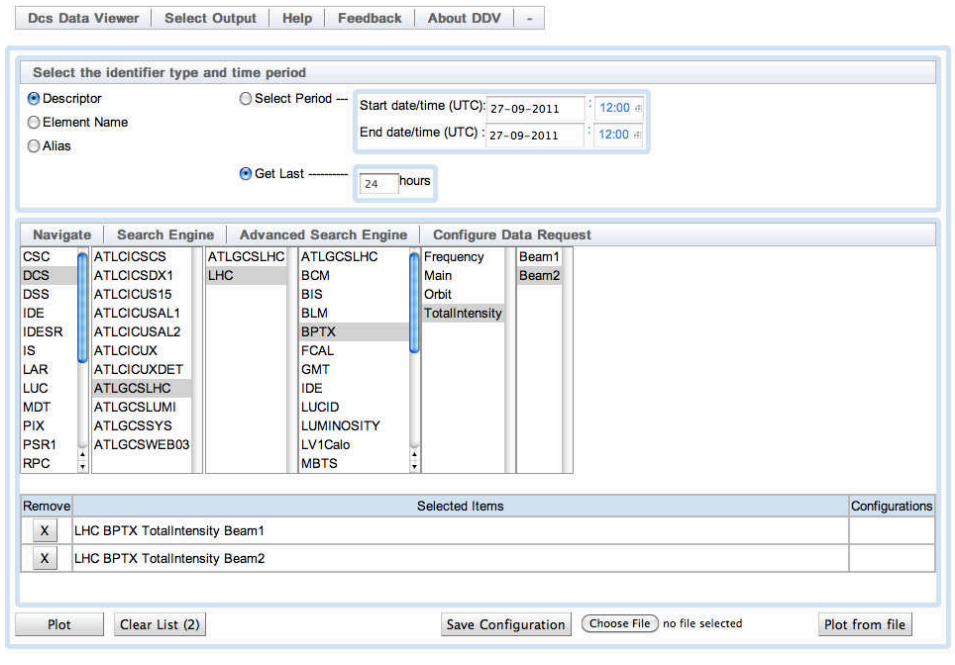
In the case of data requests, DDV server acquires data directly from DB. The server is written in python and communicates with the DB using the cxOracle extension module. On the other side DDV server can communicates with the @@@ Requests are formed under URL protocol using the POST method whenever it is needed.

DDV Client And Outputs

The client part of DDV is developed using the Google Web Tool kit (GWT) framework for web applications. The developing is done in Java while the result after compilation is an AJAX based web application with browser-independent Java script. The client interface is an internet browser page designed in a way to help user perform requests in an intuitive way. The page is organized in two main parts. The top part is the main page, it is written in GWT and includes the various selections. The bottom part is independent of the GWT framework and it hosts the output plugins (Java Applets or Java Scripts). The current outputs are listed below.

- **Chart Applet.** The default output for graphical representation of data. It is implemented in a form of a Java Applet based on the JfreeChart libraries.
- **Chart Java Script**. Another output for graphical representation of the data. It can be visualized in environments that do not support java applets (e.g. smart phones).
- **Table Java Script.** Output that displays the database response in a table holding the information of requested item, date, time and recorded value.
- **ROOT**. Output in ROOT format. Aimed for users that want to do elaborate manipulations with the data.
- **ASCII.** Output as ASCII file simple format and highly universal.

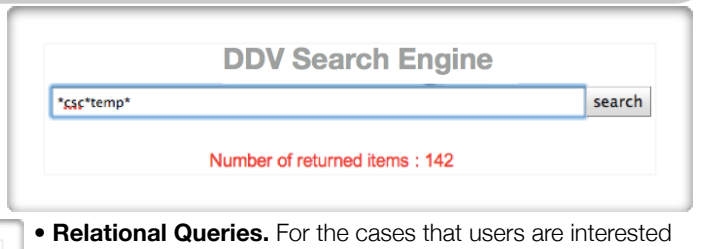
Browser Interface

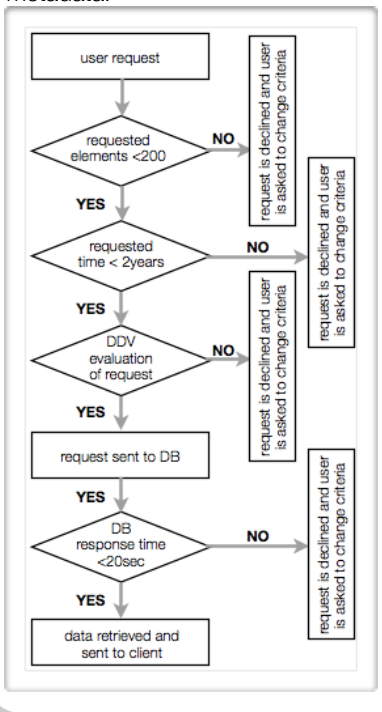


The main browser window interface of DDV. The top part deals with the selection of the navigation mode and the start/end date and time. The middle part offers an easy navigation among the metadata with a column-tree view. The bottom part holds the action buttons like plot for a graphical display of data and save configuration for saving the current configuration in XML format.

DDV features

- **Search Engine.** DDV offers an easy navigation mode through *column lists* but since the vast number of archived parameters can create difficulties to non DCS experts, DDV provides a *Search Engine* for DCS metadata.





- **Relational Queries.** For the cases that users are interested in a subset of data (e.g. case of a spike) DDV provides a *Relational Queries* mechanism where an accepted range of values can be specified for the selected items. Each item can be configured separately and have its own accepted value region.
- **External Requests.** With the use of *External Requests* the user can call DDV by pointing in the same time a configuration file that is pre-saved in the server. All the requested information is kept in the configuration file and in this way a single call is enough to pop up a new browser window with the DDV page and the output already populated.
- **Data Base Protection Mechanism.** Intending to serve a high number of users , DDV tool take into notice each request that comes from users, validates it with a minimum-response-time cost and finally propagate it to the DB. This powerful three level protection mechanism is schematically displayed on the left.
- **Configuration in XML.** DDV offers the possibility of saving the current configuration in a file. A *Save as* dialog is in place that prompts the user to save the current configuration in a XML file locally. On the other hand an *upload file* dialog prompts the user to upload the pre-saved XML file and quickly visualize the wanted information.