A Web based application to collect, manage and release of alignment and calibration configurations for data processing at CMS

Muhammad Hasib\textsuperscript{1}  Arun Kumar\textsuperscript{2}  Giacomo Govi\textsuperscript{3}  Luca Pernie\textsuperscript{4}  Giovanni Franzoni\textsuperscript{5}  Sverre Herland\textsuperscript{5}

\textsuperscript{1}University of Delhi  
\textsuperscript{2}National Taiwan University  
\textsuperscript{3}Fermi National Accelerator Lab  
\textsuperscript{4}Texas A&M University  
\textsuperscript{5}CERN  
On behalf of the CMS Collaboration
CMS detector is under very high solenoid magnetic field and high collision rate, which introduce misalignments in various subsystems, to maintain the physics performance of CMS detector, one has to continuously recalibrate and realign the detector.

The Conditions are non-event data which describe the evolving status and performance of the several detector components of CMS.

- In the High Level Trigger
- the processing of the recorded collisions
- the production of simulated events for data analysis and studies of detector upgrades

They record a given state of the detector, and measure alignment and calibration constants with dedicated algorithms.

Condition data are crucial for reconstruction of collision events coming from simulated or real data, as well as for physics analysis.

Beamspot is the position of interaction region between two colliding beams.
The CMS experiment makes a vast use of alignment and calibration measurements in several data processing workflows:

**Condition Data Model**

- **Payload** The "atom" of conditions data is the payload, it represents the set of parameters consumed in the workflows of the physics data processing.
- **Interval Of Validity (IOV)** time interval during which a given Payload is consumed.
- **Tag** history of a given calibration or alignment content comprising a set of IOVs and their associate Payloads.
- **Global Tag(GT)** a consistent set of Tags providing all the condition data needed by a given workflow.
CMS Condition Database Architecture

The CMS Condition Database Architecture

- ORACLE
- Master Condition Database (Online)
- ADG Copy (Offline, read-only)

CMS online protected network

CERN offline network

Condition DB Access (cmsDbAccess)
RESTful API for accessing the online condition database from the offline environment.

Condition Browser (cmsDbBrowser)
Web-based entry point for the CMS condition navigation, bookkeeping, and management.

CMS Collaborator
Physicists can easily navigate and browse all the conditions related data. Also, it is possible to draw various plots of deserialized Payload data.

Calibration Expert
Calibration experts have a simple way to request a record update for a specific production workflows as well as create a special Global Tag type called a Candidate to test their requested Payload changes in the CMSSW.

Condition Manager
Condition managers have the ability to accept or reject requests done by the calibration experts, manage all the condition metadata and to compose a complete calibration scenarios (Global Tags) to be consumed in a production workflows.
The Conditions involved in a typical production workflow are usually grouped into up to 300 Tags. This implies the existence of a large number of Tags and Global Tags in the database. A web based application called cmsDbBrowser was created in order to:

- **Navigation**
  - Intuitive way to inspect-navigate-search existing conditions data and metadata to any CMS member.

- **Manage**
  - Bookkeep the condition metadata by condition managers.
  - Handle Tag update requests submitted by detector experts.
  - compose a complete calibration scenarios like Global Tags.

- **Monitoring**
  - Provide a single entry point to monitor all condition related services.
Design and Implementation Choice of CMS Condition Browser

- The backend of cmsDbBrowser is implemented in Python programming language using the Flask web framework.
- As an Object Relational Mapper SQLAlchemy is used which handles all the database transactions.
- For the frontend the Bootstrap CSS framework is used together with the jQuery and Highcharts JavaScript libraries.
Service Users
- Few 100 CMS users, 50 frequent users of this service.

Search Service
- Users do more than 1000 search per month

Global Tag Request
- 1000 GT created over the year

We have 1000 of uploads every month to database that this service monitor there status.

This application has been proven Vital in catching the problems and there have been few incidents which we have solved because of quick notifications from this service which basically results in maintaining the good quality of data for physics analysis.
The Payload Inspector was created in order to let detector experts to inspect and monitor alignment and calibration measurements stored in the CMS conditions database.

The plots are developed in C++ by the experts of each system following a schema of templated C++ classes, and that the browser can discover the monitoring plots thus implemented discovering dynamically from CMS Software (CMSSW) releases.

The tool consists of two separate layers:

**Deserialization layer:**
- This consists of dedicated C++ plugins inside CMSSW which will load the payloads from DB and extract the relevant information for rendering the plots.

**Visualization layer:**
- Allows users to generate interactive historical plots for monitoring conditions.
Payload Inspector

plot_SiStripApvGainsMaximumTrackerMap

SiStrip APV Gain maximum per module (payload : d8a76fc9e5ca9272af550e7de31e55bc38e723f)

plot_EcalLaserAPDPNRatiosPlot

Ecal Laser APD/PN, IOV 2018-07-05 02:24:25
Monitoring section in the cmsDbBrowser displays various log pages:

- **ConditionUploader logs** - Condition upload is a tool to update one or more tags in the official database, here you can see the status of your upload in detail.

- **O2O logs** - Online to Offline (O2O) is a process that converts the online data format to a more meaningful data format to offline, so that other processes can understand and use it for physics data taking.

- **Tag logs** - Shows recent changes done on Tags.

- **Global Tag Logs** - Detail description about all the tag for a given GT.
Subscription

- **Goal**: Provide users with a customizable way to receive updates through email about modification and creation of data in the conditions database.

- **Users**
  - **Users**: Who wants to be notified about changes being made to relevant data.
  - **Expert**: Whose job is to validate and further process work done by others, who want to know when tags are appended to productions GTs.

- **Tag Log Events**
  - New tags created
  - IOVs inserted - When someone has attached a new payloads, containing alignment and calibrations to an existing tag

- **Global Tag Log**
  - **Global tags created**: When someone has created a new global tag
  - **Global tag map request**: When a tag is requested/rejected/approved to be included in a global tag
1-Create new subscription
2-Subscription activation toggle.
3-Send email
4-Edit subscription
5-Delete subscription
6-Add entities
7-Remove all entities
8-Remove entity
9-Toggle panel collapse
Summary

- Monitoring the CMS condition data.
- Recording the status of the experiment and of the ongoing data taking
- Accepting conditions data updates provided by the detector experts
- Aggregating and navigating the calibration scenarios
- Distributing conditions for consumption by the collaborators.

Thank You!
Do you have any questions?