A multi-dimensional view on information retrieval of CMS data

A. Dolgert, L. Gibbons, V. Kuznetsov, C. D. Jones, D. Riley

Cornell University, Ithaca, NY 14853, USA

E-mail: vkuznet@gmail.com

Abstract. The CMS Dataset Bookkeeping System (DBS) search page is a web-based application used by physicists and production managers to find data from the CMS experiment. The main challenge in the design of the system was to map the complex, distributed data model embodied in the DBS and the Data Location Service (DLS) to a simple, intuitive interface consistent with the mental model of physicists analysis the data. We used focus groups and user interviews to establish the required features. The resulting interface addresses the physicist and production manager roles separately, offering both a guided search structured for the common physics use cases as well as a dynamic advanced query interface.

1. Introduction

The LHC era is just around the corner. Two major CERN experiments, CMS and ATLAS, are ready to start taking data in 2007. Given that a single experiment will accumulate about a few petabytes per year, data management becomes one of the challenging tasks within the experiment. In the past, several systems have been developed to address this issue within the HEP community [1]. In CMS, the Data Bookkeeping System (DBS) [2] has been designed to catalog all CMS data, including Monte Carlo and Detector sources. It has already undergone several iterations in its design cycle to better address physicists' needs for data access. A clear understanding of data flow, data retrieval and usage were the key ingredients in its final design. A full discussion of the CMS DBS system can be found elsewhere [2]. Here we present the data retrieval aspect of the DBS within the CMS data model, emphasizing details of the Data, Mental and Presentation models.

2. CMS data

CMS data is a complex conglomerate of information from different stages of processing, from generation at the detector to reconstruction steps. Clear understanding of data workflow was a key issue in developing the data retrieval system. We concentrated on three models: the data model which defines objects and entities used in the production system and DBS, the mental model of our users, such as physicists or production managers who use informal language and definitions appropriate for their workflow and the presentation model which sbridges the two.

2.1. Data model

The CMS data model is a file-centric. The smallest entity for data management system is a file. The file itself is a holder of particular data objects, i.e. Analysis Object Data (AOD), while data management tools, due to their distributed nature, operate with blocks. The block holds a bunch of files, representing a specific task in the processing chain. Finally, the dataset represents a data sample produced or generated by experiment. In CMS the dataset notation is a path, which consists of three components */PrimaryDataset/ProcessedDataset/DataTier*:

- *PrimaryDataset* describes the physics channel, e.g. MC production chain or trigger stream;
- *ProcessedDataset* names the kind of processing applied to a dataset, e.g. release version and filtering process, such as CMSSW_1_1_1-OneMuFilter;
- *DataTier* describes the kind of information stored from each step of processing chain, e.g. RAW, SIM, DIGI, RECO.

Each of those entities were represented in the DBS via a set of tables whose relationships were established based on use case analysis of data workflow and data management tools. The DBS schema implementation can be found elsewhere [2] and is beyond the scope of this discussion. Data management in CMS is composed of the following systems:

- Data Bookkeeping System (DBS) is a metadata service which catalogs CMS data and keeps track of their definitions, descriptions, relationships and provenance;
- Data Location Service (DLS) maps file-blocks to sites holding replicas [1];
- Physics Experiment Data Export (PhEDEx) manages transfer of data among the sites;
- CMS Remote Analysis Builder (CRAB) is a set of tools for analysis job submission to the GRID;
- Production Request system (ProdRequest) is a tool to submit a workflow to the production machinery;
- Run quality and Condition DBs provide information about runs and detector constants;

Even though those systems represent specific tasks in data management, they were integrated with each other at different levels. The current situation shown in Fig. 1. As the most authoritative source of information about CMS data, the DBS is tightly integrated with other data management sub-systems. Its schema [2] accumulates all information for different groups of users, and the discovery service has become the main tool for data search. Its implementation has been discussed separately in [3].

It was obvious that such complex data management was not appropriate for users to fully understand and manipulate. Instead they operate in terms of mental models more appropriate to their individual goals.

2.2. Mental model

During analysis of different use cases we identified groups of users whose use of data fall in the same category. We outlined their scopes and roles and tried to map their view to data model.

- *Physicists*, a group of users which are interested in data analysis. They focus on finding analysis and processed datasets, with need for further details about involved data, for example, detector conditions and trigger description in a case of data or Monte Carlo generators and their details in a case of simulated data. This group of users usually ask the following question: “Is there any data appropriate to this physics, and how can I run my job with this data?”
- *Production managers* are interested in details of production workflow. They operate at the level of datasets, blocks and files. The main use case was to find out and monitor data production flow for a given process. For example, there are several production teams who

[1] Recently the DLS system was merged with the DBS to improve data lookup performance.

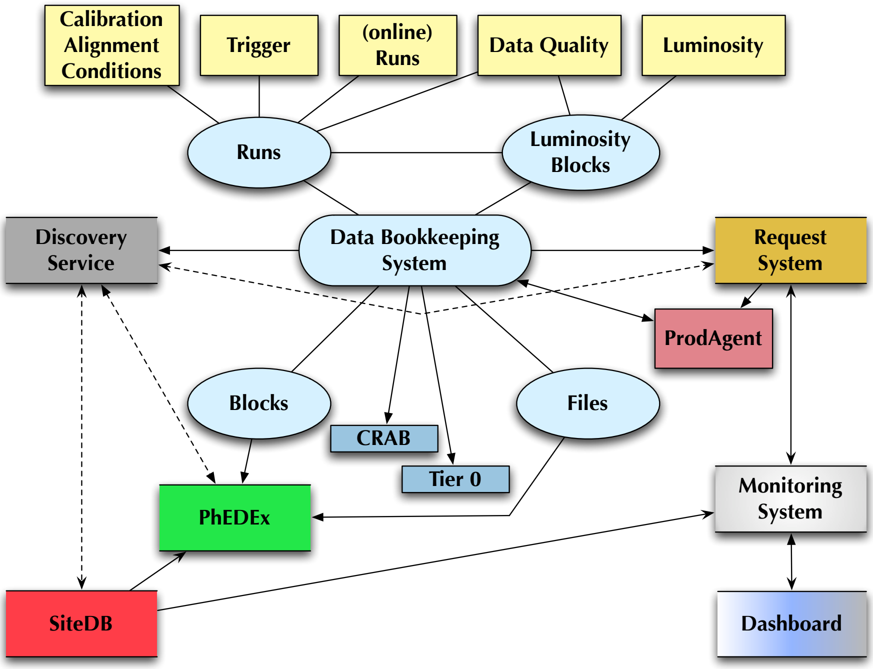

Figure 1. DBS role in CMS data workflow. The dashed lines represent web services between data discovery and other subsystems.

are responsible for Monte Carlo generation of specific data samples, e.g. Higgs data. For them we defined a local scope DBS where their data was located. Data managers were able to lookup their data in a local scope DBS and monitor progress of production flow.

- *Run managers* represent a group of users who are interested in online and DAQ-specific information about data. Their role is to identify the quality of the data and detector conditions, but the run information is spread among different databases, such as the run quality database or condition database, which are attached to the DAQ or geometry systems. Therefore, for this group of users, run summary information from different systems were critical.
- *Site administrators* are a small group of users whose role is to maintain data at a given site without necessarily knowing the meaning of the data itself. Their view was to locate data of a certain size, measure disk usage, and identify data owners.

We outlined the main use cases for each category of users and mapped them onto our data model. Although this was a trivial process, the actual presentation for data lookup as well as their appearance on web pages were the most cumbersome task.

2.3. Presentation model

Walking through the use case analysis [4], we identified patterns in data discovery which were implemented rapidly via web interfaces [3]. An iterative process of reviewing the use cases, implementing web interface prototype, conducting user interviews (via tutorial sessions, Hyper News forum discussions, etc.) and log analysis helped us to finalize the presentation layer of the data discovery tool. As can be seen from [5], it consists of the following groups of search interfaces:

- *Navigator* is a menu-driven search form which guides users through main entities of the processing chain, e.g. Physics Groups, Data Tier, Software Releases, Data Types, and Primary Dataset. The hierarchical grouping was applied here to help users navigate in their data search. The result of this search was a summary list of datasets which matched search criteria, with details of where data was located, their size and links to data details, such as run information, configuration files, etc.
- *Site/Run searches* answer questions such as, "which data are located at my site" or "what data conditions were applied for a given run?" For site searches, result output centers on details about files and blocks. For run searches, result output is just grouped by run.
- *Analysis search* is for physicists to find analysis datasets. Somewhere between the previous two types of searches, this gathers results output with a short summary for each analysis dataset, summarizing how it was produced, who created it and some details of what it contains. When available, there is a long view for each analysis dataset. We also provide a lookup for production datasets, with output like that of the *Navigator* page.
- *Finder* represents an opposite approach in data lookup based on a user's freedom of choice. Here, the DBS is treated as a black box where information is stored. Since the metadata are stored in relational database, we expose part of the database schema to users and allow them to place arbitrary queries against it. We used an intermediate layer to group database tables into categories, e.g. Algorithms, Storage, Datasets and hide all relational tables from the user's view. Once a user made a choice, for example "I want to see datasets and runs", we construct a query for them using shortest path between Table A (datasets) and Table D (runs) based on foreign-keys of the schema. The output results are accompanied by a list of processed and analysis datasets related to query results.

Based on a group's scope and roles we tried to identify the amount of information suitable for presentation on a web page, keeping in mind that information retrieval should be only a few clicks away. For that, we introduced the concept of the view, which determines which information details to show on a page. For example, viewing a processed dataset, the information about its owner was only available in the *Production* view, while site replicas were visible in all views. A positive outcome is that most users do not have their search cluttered with details of the production machinery. Finally, it provides access to different DBS instances, for example the local scope DBS's were only presented in *Production* view, while the global DBS instance stays in all views. The DBS was designed to support collaboration, group and user scopes. Similar design decisions were made in earlier frameworks [6]. In the local scope of a group or individual user the data registered in DBS should only visible to that group or a user, while in global scope all data available for whole collaboration are visible. The production team who generates Monte Carlo operates within the local scope in one of the DBS instances. Once data are ready, they are published to the global DBS. The Data Discovery service exposes those DBS instances only in production view, even though they are not restricted for users. That allows us to separate published data from development data. The same ideas were applied to Physics Groups and analysis datasets, but there an additional layer is applied to secure specific analysis from general viewing.

Modern AJAX techniques [3] were widely used in development to provide guidance and an interactive look and feel on a page. For example, in Navigator the menus were presented in hierarchical order suitable to the mental model of physicists when they lookup their data. Since most of the time users ask questions like "I would like to find Monte Carlo data produced with a certain release for a $t\bar{t}$ sample," we end up with the following order: *Physics groups, Data Tier, Software release, Data types, Primary Datasets/MC generators.* As the order of the menus narrows the search, asynchronous AJAX calls populate submenus. We also used wildcard searches with drop-down menus of first matches to help users quickly lookup data on their patterns.

We examined web logs to identify the most frequently viewed information on pages and adjusted the contents of results page several times. It is clear that data retrieval eventually will migrate from processed datasets to analysis datasets when CMS experiment matures. We hope, as well, that the most configurable data search service, the Finder, will gain popularity in time when the amount of stored data increases.

3. Data flow

To accommodate various users' needs, we also looked closely at CMS data flow. The current situation has been shown on Fig. 1 and Fig. 2. Production for each data tier used a different DBS instance. Data were inserted, validated and managed in local scope DBS instances. After validation steps, the data were merged from local instances to a global DBS instance using the DBS API [2]. Data then appeared for public usage. To accommodate this, the data discovery service has the ability to lookup data in every DBS instance. Since this service was read-only, it was opened up for all CMS collaborators, but the only concrete view of all this data was on the discovery page, i.e. *Production* view. We also discussed how several CMS web services should exchange information among each other and achieved this integration via AJAX requests. A good example was the ability to lookup transfer and run summary information on the discovery page's run search interface. When the run information, such as run number, number of events in a run, dataset it belongs to, and number of files, was retrieved from the DBS and shown on a page, separate AJAX calls were sent to the Run summary DB and PhEDEx systems to get run details and block transfer information. This allowed the run manager to have summary information from different sources for further monitoring. At the same time data retrieval was bidirectional. For example, the Production request system was able to request software releases and other information from the DBS to present it to end users who were then able to place their requests for produce Monte Carlo samples, and, on the discovery page, we were able to lookup Production requests for the dataset in question. It is worthwhile to mention that, in the end, the Production request, Data Discovery and SiteDB services were integrated into common web framework [7].

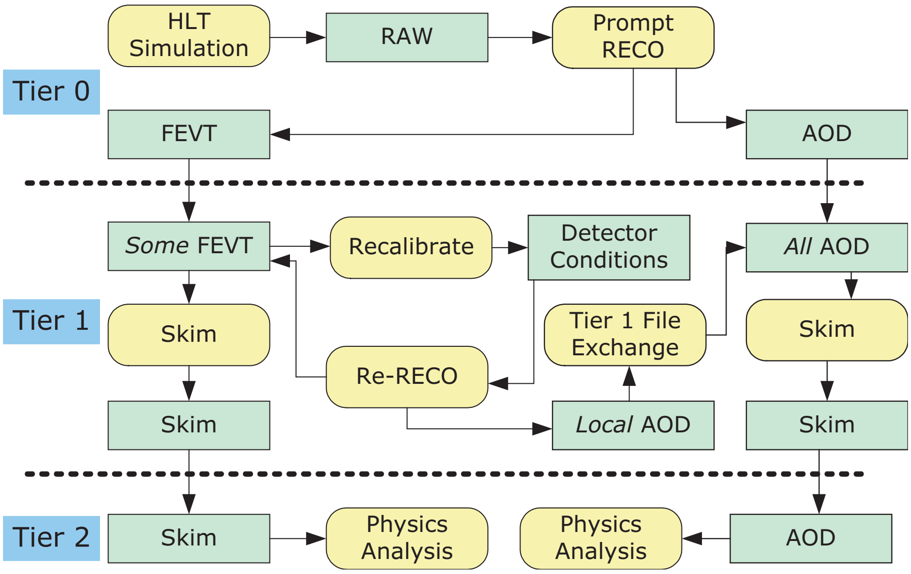

Figure 2. CMS CSA06 data flow diagram

4. Further plans

This work is still in progress and has not yet reached mature status. The data discovery prototype has been developed and deployed for almost a year but continuously evolves. One hot topic of interest is physics analysis at local sites. We are investigating how to deploy a local site with a DBS instance and data discovery service. It became clear that further user customization is required to fit user needs at local sites. For instance, users should be provided with a web interface to register, share, import and export their own data within the scope of a group of users or physics group. Due to the distributed nature of the CMS data model, it will require a certain level of authorization, authentication and data validation. A novel approach has been shown in Hilda project [8] where a system was designed to support all of these requirements.

5. Conclusions

We discussed information retrieval patterns within the CMS High Energy Physics collaboration. Three models were covered: a Data model which describe data produced in this experiment, a Mental model of users, physicists, who access their data and a Presentation model of the search service. Concepts of the view, scope and different search interfaces were shown.

6. Acknowledgements

We would like to thank all CMS collaborators for extensive discussion of various use cases involved in CMS data management. In particular we thank Lee Lueking, Anzar Afaq, Vijay Sekhri for valuable discussioins about DBS implementation.

This work was supported by National Science Foundation and Department Of Energy.

References

[1] FNAL SAM system, CHEP 2000; CLEO-c EventStore, CHEP 2004

[2] A. Afaq, et. al. “The CMS Dataset Bookkeeping Service”, see proceeding of this conference

[3] A. Dolgert, V. Kuznetsov “Rapid Web Development Using AJAX and Python”, see proceeding of this conference

[4] https://wiki.lepp.cornell.edu/hep/bin/view/SWIG/DbsCsa06UseCases

[5] http://cmsdbs.cern.ch/DBS2_discovery/

[6] C. D. Jones, V. Kuznetsov, D. Riley, G. Sharp “EventStore: Managing Event Versioning and Data Partitioning using Legacy Data Formats”, CHEP04.

[7] M. Simon et. al., “CMS Offline Web Tools”, see proceeding of this conference

[8] F. Yang, J. Shanmugasundaram, M. Riedewald, J. Gehrke, A. Demers, “Hilda: A High-Level Language for Data-Driven Web Applications”, ICDE Conference, April 2006