

## CMS Offline Web Tools

**S Metson<sup>1</sup> S Belforte<sup>2</sup> B Bockelman<sup>3</sup> K Dzedziniewicz<sup>4</sup> R Egeland<sup>5</sup>  
P Elmer<sup>6</sup> G Eulisse<sup>7</sup> D Evans<sup>8</sup> A Fanfani<sup>9</sup> D Feichtinger<sup>10</sup> C Kavka<sup>2</sup>  
V Kuznetsov<sup>11</sup> F van Lingen<sup>12</sup> D Newbold<sup>1</sup> L Tuura<sup>7</sup> S Wakefield<sup>13</sup>**

<sup>1</sup> H.H. Wills Physics Laboratory, University of Bristol, Tyndall Avenue, Bristol BS8 1TL, UK

<sup>2</sup> INFN, Sezione di Trieste

<sup>3</sup> University of Nebraska Lincoln, Lincoln, NE, USA

<sup>4</sup> CERN, Geneva, Switzerland

<sup>5</sup> University of Minnesota Twin Cities, Minneapolis, MN, USA

<sup>6</sup> Princeton USA

<sup>7</sup> Northeastern University, Boston, MA, USA

<sup>8</sup> Fermilab MS234, Batavia, IL, USA

<sup>9</sup> Università degli Studi di Bologna

<sup>10</sup> PSI, Villigen, Switzerland

<sup>11</sup> Cornell University, Ithaca, NY, USA

<sup>12</sup> California Institute of Technology, Pasadena, CA, USA

<sup>13</sup> Blackett Laboratory, Imperial College, London, UK

E-mail: [simon.metson@cern.ch](mailto:simon.metson@cern.ch)

**Abstract.** We describe a relatively new effort within CMS to converge on a set of web based tools, using state of the art industry techniques, to engage with the CMS offline computing system. CMS collaborators require tools to monitor various components of the computing system and interact with the system itself. The current state of the various CMS web tools is described along side current planned developments.

The CMS collaboration comprises of nearly 3000 people from all over the world. As well as its collaborators, its computing resources are spread all over globe and are accessed via the LHC grid to run analysis, large scale production and data transfer tasks. Due to the distributed nature of collaborators effective provision of collaborative tools is essential to maximise physics exploitation of the CMS experiment, especially when the size of the CMS data set is considered. CMS has chosen to provide such tools over the world wide web as a top level service, enabling all members of the collaboration to interact with the various offline computing components. Traditionally web interfaces have been added in HEP experiments as an afterthought. In the CMS offline we have decided to put web interfaces, and the development of a common CMS web framework, on an equal footing with the rest of the offline development.

Tools exist within CMS to transfer and catalogue data (PhEDEx and DBS/DLS), run Monte Carlo production (ProdAgent) and submit analysis (CRAB). Effective human interfaces to these systems are required for users with different agendas and practical knowledge of the systems to effectively use the CMS computing system. The CMS web tools project aims to provide a consistent interface to all these tools.

### 1. Introduction

The Compact Muon Solenoid (CMS) experiment on the Large Hadron Collider at CERN is expected to start running during 2008. There will be approximately 170 institutes involved in CMS and nearly 3000 physicists spread over 40 countries and 18 time zones. An extensive offline

computing system is being developed, based on the WLCG grid [2], to transfer, simulate and analyse data for the CMS collaboration.

This computing system is built around a number of loosely coupled components [3, 4, 5]. Providing effective interfaces to this machinery, for both expert operators and untrained physicist, is vital if the computing system is to deliver. It is important that access to the computing system is available to all and that using it doesn't represent an obstacle or significant overhead to analysis work. It is also important that sufficient information is provided to users, be they physicists doing analysis or expert operators managing the system, so that monitoring progress or debugging issues is possible.

It has been decided in CMS that the manner in which people interact with the offline computing system will be web based, providing uniform access to people across the globe. These tools are based on emerging techniques from within the industry, commonly referred to as "web 2.0".

## 2. Components

The web tools are generally written in python (the PhEDEx page is Perl, but migrating to the framework) and use a CMS developed framework called as Bonsai. This is built on the CherryPy[6] application server and Cheetah[7] templating engine. The framework also provides a consistent interface to logging and provides a simple manner to secure pages based on user and their associated roles within the collaboration (Physicist, Monte Carlo Production Manager, Site Manager, for instance).

All applications rely heavily on backend databases, either SQLite, MySQL or Oracle. The use of databases is such that it would be fair to say that the web tools applications are written in SQL, rather than perl or python. Database access is made using either Perl::DBI[8] or SQLAlchemy[9].

A set of reusable components, known as "widgets", are being made available. These are usually built using the Yahoo Interface Library (YUI)[10] and are written in CSS and JavaScript. Where possible these are reused to provide identical functionality across different components, so that a user learns a single interface to all web tools.

The services run on a fairly standard configuration: a pair of Apache servers working as a load balanced proxy in front of two application servers. The front end servers are accessible to the outside world, the back end machines are firewalled off from remote access.

## 3. The Present

The current PhEDEx [3] web site allows for any authenticated member of CMS to create a subscription request. This notifies a "Data Manager" who can approve or deny the subscription. Once approved the request is placed into the PhEDEx machinery and transfers begin. Through the web interface it is possible to request subscriptions for any of the official datasets; more than half a million files totalling almost 800TB of data.

In addition to the management of the PhEDEx system the main role of the PhEDEx web page is monitoring that system. It is possible to produce plots (examples of plots would be transfer rate, transfer quality or resident data volume) for any site in the PhEDEx topology for any period from PhEDEx's inception in 2004 to date. Errors are reported back to the central database from download agents and are displayed on the web. Component status is also reported. This enables much debugging to be done using only the web page.

ProdRequest became part of the CMS montecarlo production chain in early 2007 and it has been used ever since to submit over 1500 requests representing almost 500M events. It provides a three fold interface for general user, for operators and for managers which exploits group/role functionalities of WEBTOOLS security module to automatically show a role-optimized view of the system and its controls. The backend is based on WEBTOOLS mini-framework BONSAI

and can work as a standalone application using sqlite for database backend or as a full-blown server accessing an oracle database.

The client GUI tries to leverage as much as possible on YAHOO YUI [10] javascript library, both by using its default widgets for lists and dialogs and by exploiting it's API to create customized controls for request creation, building on the work done in [11]. In addition, ProdRequest interacts with other WEBTOOLS project like DBS, ProdMon and ProdManager to share information and particular views of the requested workflows and datasets.

The Data Bookkeeping System (DBS) [4] has been developed to catalog all CMS data, including raw data, Monte Carlo and detector sources. The DBS represents a service whose primary responsibility is to answer questions about data existence, their characteristics, description and location in CMS. A web based front end to this system has been developed, called data discovery. It serves as a specialized search engine to lookup CMS data.

During the discovery page's development cycle it was decided to split the functionality of DBS API and data discovery service. While former one is implemented as Java Servlets under tomcat server, the later was initially written in python using the CherryPy application framework and Cheetah template engine, later migrating to the webtools framework. The benefits of using these modern technologies has been discussed in [12]. The data discovery uses read-only access to underlying DB back-end via transparent SQL layer served by SQLAlchemy.

The user interface uses extensively several AJAX frameworks, such as YUI, openRICO[13], RSH[14] to improve site response and simplicity for end-users. We reduced latency on the user interface by optimizing user queries and building them dynamically upon user selection, as well as following guidelines for speeding up web sites from Yahoo [15]. The data discovery application provides several search interfaces to lookup data in DBS:

Conditions data may be accessed online via the CondDB web tool. This service allows a user to view the tags and Interval of Validity (IOV) information concerning the conditions data; effectively a web interface to CMSSW command line tools. In addition to this a user can upload or download an SQLite file with the conditions assigned to each detector component. Lastly the service allows a user to gather a set of production requests and their associated conditions data, and view this collection. This was implemented using an Oracle database provided by CERN. The service consists of 3 parts: database, application server (used by the IOV part, as an HTTP interface to command line tools and to file upload/download) and the web page itself.

- *Navigator* is a menu-driven approach which guides users through their selection of common search tags, e.g. Physics Group, Data tier, Software releases, Data types, Primary datasets and Sites.
- *Finder* is a represents an opposite search interface. Instead of guiding users through a set of selections we expose a relevant part of the DBS schema to them and provided an interface to make a data selection.
- *Run/Site* search interface concentrating on looking up runs and data on a given site
- *Analysis* search interface were built to serve analysis use case (tentative)

The DBS Data Discovery web application is integrated with several external services such as ProdRequest, PhEDEx and the run summary database via AJAX callback. This means a user can find detailed information from a number of sources through the same interface on demand. For example, information about runs found in DBS is complemented by information from the run summary database providing specific details about runs in question, such as magnetic field, etc.

#### 4. The Future

At present the PhEDEx page is very site orientated, the next stage of development will seek to address this by emphasising the request as the logical block a user of the site interacts with.

Current site functionality (what data is located at a site, how big is that data) is in the most part already available in the DBS browser. Future developments of both tools will seek to make the DBS browser the authoritative source of information on a datasets location, while streamlining the PhEDEx site to provide sufficient information to sites while making tracking subscriptions and debugging issues simpler.

Because of the way CMS production has been organised to date, the development of ProdRequest has been focussed on optimizing the production manager/production operator workflow. ProdRequest will evolve into a more user-centric tool in the future to improve the request submission interface, easing the task of submitting requests by the casual user.

Monitoring of the production system will be provided through ProdMon. Currently this is a selection of plots produced dynamically from data reported from individual ProdAgents and stored in the ARDA Dashboard, but with little focus on the user interface. In the future this monitoring will become more sophisticated, incorporating role based views of the production system and more complex queries and plots.

DBS Data Discovery will be further integrated with other CMS web applications, such as the Luminosity database and the DQM. As the use cases associated with analysis become clearer the tool will be simplified and streamlined to solve the primary tasks effectively. The DBS itself will soon run “local” instances, and a Data Discovery tool will be provided for these local instances, and for searching across multiple local sites.

The future development of CondDB will focus on integration with the TagCollector tool, used by CMS to organise the contents of releases of the CMSSW software. The part gathering production requests will be expanded in accordance with CMS policy concerning a condition object’s life-cycle. This expansion will require more functionality in the request viewer, including more queries and the introduction of some graphing tools.

The web tools framework has recently been adopted by the Data Quality Monitoring (DQM) group to provide both online and offline monitoring. The web based DQM tools have been used in monthly global runs since June 2007 and is rapidly developing into a tool to support the needs of the DQM community. The tool must quickly provide access to hundreds of thousands of histograms, and be tuned to a user’s needs (based either on their role or on the problem at hand).

Further integration and “cross-pollination” between the user interface components of the different projects are expected in the next development cycles. This has dual advantages. Complex UI components can be written once and reused, reducing development time. Likewise reusing components means a user only needs to learn one interface to interact with the systems, enabling users to rapidly realise the potential of the whole CMS computing infrastructure using just their web browser.

## References

- [1] The Compact Muon Solenoid Technical Proposal, (CERN/LHCC-94-38), The CMS Collaboration, CERN, 1994
- [2] LHC Computing Grid Technical Design Report, LHC Computing Project, CERN, 2005, <http://lcg.web.cern.ch/LCG/tdr/>
- [3] Rehn, Barrass, Bonacorsi, Hernandez, Semeniouk, Tuura and Wu, “PhEDEx high-throughput data transfer management system”
- [4] A. Afaq, et. al. “The CMS Dataset Bookkeeping Service”, see proceedings of this conference
- [5] D. Evans, A. Fanfani, et. al. “The CMS Monte Carlo Production System: Development and Design”, CMS CR-2007/057
- [6] <http://www.cherry.py.org/>
- [7] <http://www.cheetahtemplate.org/>
- [8] <http://dbi.perl.org/>
- [9] <http://www.sqlalchemy.org/>
- [10] <http://developer.yahoo.com/yui/>

- [11] G. Eulisse et. al. "Interactive Web-based Analysis Clients using AJAX: examples for CMS, ROOT and GEANT4", Proc. CHEP06, Computing in High Energy Physics, Mumbai, India, 2006
- [12] A. Dolgert, V. Kuznetsov "Rapid Web Development Using AJAX and Python", see proceedings of this conference
- [13] <http://www.openrico.org/>
- [14] <http://www.onjava.com/pub/a/onjava/2005/10/26/ajax-handling-bookmarks-and-back-button.htm>
- [15] <http://developer.yahoo.com/performance/rules.html>