Available on CMS information server

CMS CR -2012/082

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

11 May 2012 (v3, 20 June 2012)

PREP: Production and Reprocessing management tool for CMS

Fabio Cossutti for the CMS Collaboration

Abstract

The production of simulated samples for physics analysis at LHC represents a noticeable organization challenge, because it requires the management of several thousands different workflows. The submission of a workflow to the grid based computing infrastructure starts with the definition of the general characteristics of a given set of coherent samples (called a campaign), up to the definition of the physics settings to be used for each sample corresponding to a specific process to be simulated, both at hard event generation and detector simulation level. In order to have an organized control of the of the definition of the large number of MC samples needed by CMS, a dedicated management tool, called PREP, has been built. Its basic component is a database storing all the relevant information about the sample and the actions implied by the workflow definition, approval and production. A web based interface allows the database to be used from experts involved in production to trigger all the different actions needed, as well as by normal physicists involved in analyses to retrieve the relevant information. The tool is integrated through a set of dedicated APIs with the production agent and information storage utilities of CMS.

Presented at *CHEP 2012: International Conference on Computing in High Energy and Nuclear Physics*

1 Introduction

The needs of the simulation of physics samples for a general purpose LHC experiment like CMS are driven by the large number and complexity of the physics processes involved in pp collisions. This implies the generation of many different final states, with possible different phase space selections, and the usage of a variety of calculations and models based on different assumptions and levels of approximation in the physics description, both for the known Standard Model physics processes and for every possible new physics scenario which can be investigated. Since the simulated events are fully reconstructed up to analysis level objects, re-processing of data using improved reconstruction algorithms, detector calibrations and alignment usually implies a similar re-processing of the corresponding simulated data.

From the computational point of view all these activities translate into the need of managing several thousands of different workflows in order to satisfy the simulation necessities related to a single period of data taking. There are two concurrent problems to be solved in this process. One one side the managers of the production have to be provided with procedures and tools to handle every single step in the process in a robust way, automatizing as much as possible the various actions with the minimal need of manual manipulations, which are intrinsically time consuming and error prone. On the other hand the requesters and users of simulated samples should have a user friendly way to submit their sample request, specifying all the input parameters and settings, having the possibility to monitor the status of the production and having access to a standardized repository where to retrieve all the information about the sample which can be relevant for its usage in the physics analyses (cross section, phase space selections, input parameters for the model used, etc...).

Up to the beginning of the data taking the simulation production was managed essentially using twiki pages formatted in a standardized way as main source of information both for the production team and for the users. Requesters were responsible to fill in the needed input parameters according to predefined formats, and some semi-automatic procedure was processing the page content to actually build the needed workflows to be injected into the production agent. This procedure quickly proved not to scale with the increased necessities implied by large scale analyses, compared to limited computing challenges and preliminary analysis studies of the detector commissioning period.

The main deficiencies of this system were residing in the fragility of the underlying tools used, the need to heavily rely on requesters following information formatting policies and the lack of an automatized messaging system capable to trigger subsequent actions in the procedure. In order to overcome these drawbacks and realize a more effective management system, a project called PREP (Production and REProcessing management tool) was started in 2010.

2 Basic concepts and production policy definition

The pre-requisite for the development of PREP is the definition of the policy for production management, i.e. the logical flow of actions which the different actors in the process need to undertake in order to accomplish the task. Two basic concepts constitute the building blocks of this policy: the production (or reprocessing) campaign and the request in a given campaign. In detail:

Campaign : a collection of samples produced with the same release cycle and a homogeneous set of conditions. Simulation campaigns can be either production ones (starting from event generation and detector simulation) or re-processing ones (pile-up and electronics simulation and/or reconstruction). A campaign can allow to specify different kinds of conditions, pile-up scenarios, etc., but they have to be all available within a given release cycle and a default has to be anyway specified as basic choice;

Request : a single sample corresponding to a user defined physics process, event generator, model, set of conditions belonging to a specified campaign. The default settings of the campaign are used, unless explicitly modified by the requester choosing among those available for a given campaign.

The policy defines how a campaign is created, and how subsequent requests are managed in it from the initial proposal till the completion of the production and the publication of the result. The basic scheme in Figure 1 represents the key steps in this process. As explained in the figure caption, the color code used in the scheme distinguishes among the actions to be taken by proponents of one step (like campaign definition, request definition, request submission to the computing infrastructure, in yellow) and the intermediate milestones in the flow of actions (approval of the proposal, or denial of it sending it back for a next iteration, in green). Every action defines a status and triggers a subsequent action, and in order to make this flow as much automatic as possible and keep

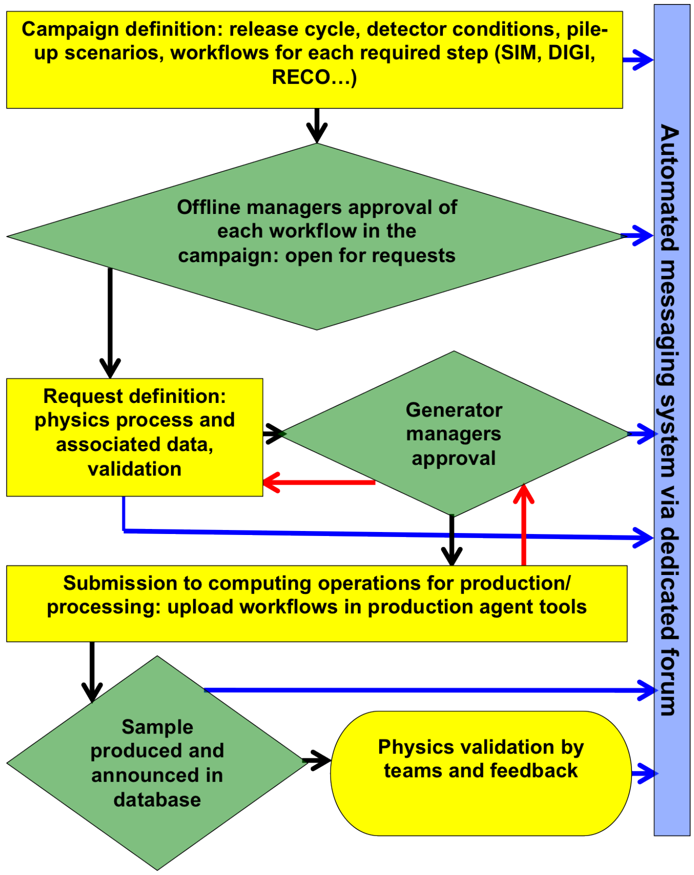

Figure 1: Schematic representation of the logical flow of actions needed to build a campaign and manage the requests in it. The yellow boxes represents actions by the proponents of different steps, the green ones indicates intermediate milestones in the process itself, black arrows show the normal flow of actions among them, red arrows represents the possibility of stopping and resting part of the process. The blue part shows the connection with the messaging system for the information change about the different actions and statuses.

track of the history of the whole process, at every step a messaging system is fed with details about what done (the blue part of the flux diagram). The details of the policy implementation can depend on the real nature of the campaign, whether it is a new production of a reprocessing of existing simulated data. From the campaign definition point of view the essential difference resides in the list of software managers who need to approve the workflows (event generators, detector simulation, electronics simulation, trigger emulation, event reconstruction, etc...). The main difference is in the submission of requests, where for new production the requesters from the different physics group play a key role in defining the generator physics settings and all the possible other special settings desired, with the event generators managers having an essential scrutiny and coordination function in this job. For a reprocessing the management of each request in generally more centralized, apart for special requests with alternative settings.

Since the policy implies the presence of different actors in the process, who playing different roles need also different kind of privileges to perform actions, the concept of user and users' group are introduced. Each person having access to the system (e.g. each CMS member with access to the software tools of the collaboration) is assigned with a list of privileges, authorizing him/her to access different blocks of information and performing different actions on the system (like creating a new campaign, a new request, deleting it, etc...). Normal users can just access the information, specialists can create requests and perform a limited subset of approval steps, while superusers can perform any allowed, even destructive, action on the system. This hierarchical organization is designed in order to protect the system from mis-management by unexperienced manipulations and limit the control of the key actions to a well defined set of responsible coordinators.

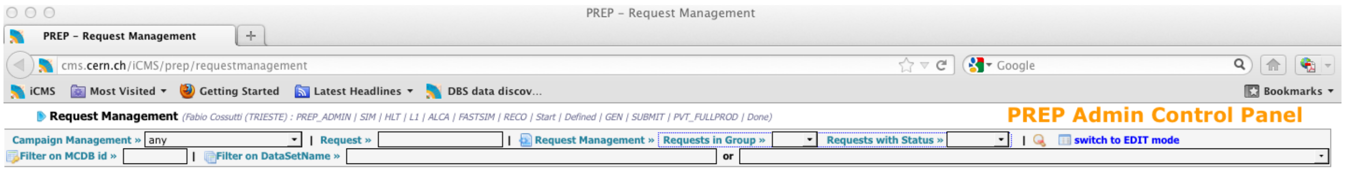

Figure 2: Main access view of the PREP web interface.

3 Software components

PREP is a collection of integrated software components which provide the backbone implementing the production policy described above.

The heart of the system is constituted by a mySQL database, which stores all the parameters needed to define a campaign, a request, the relationships among them, and keep track of the history of the actions performed on these objects. Its internal structure is largely independent on the other components.

A Javascript core implements the system engine which allows each user to interact with the database through some interface. It is the backbone of a web based interface accessible from any supported browser through which the database content can be visualized and modified. This backbone takes care of registering in the database a time stamp associated to each action performed and the user identity of who has made it. At the same time it sends predefined e-mail messages to configurable destinations, in order to ensure an automatic flow of information. A dedicated hyper-news forum collects by default all these messages, practically implementing a communication archive. A set of pre-defined actions are available for every object (campaign, request), type of user and status of the object. Some basic statistical information about the campaign status is internally kept for monitoring.

The interaction with the computing infrastructure presents a twofold problem. First of all it requires to build workflows in the context of the CMS collaboration software framework CMSSW [1], which means having access to a CMSSW working area to create python configurations for the framework executable application. On the other hand it requires to pass these configurations in some pre-defined format to the computing production agent, which is itself evolving over time. The different kind of expertise needed to manage the core of PREP and to manipulate configurations suggested to have a modular way to make the core to communicate with the computing system through dedicated python APIs developed by the software managers. In order to let the APIs accessing the database content in a transparent way, XML snapshots of it are made available through the interface and to python scripts for parsing and read-only manipulation. A dedicated python library connects the core with the CMSSW utilities needed to build the framework configurations from the information stored in the database, and provides the needed input to the computing infrastructure. During the last 12 months the CMS production agent was changed, moving from the old tool named ProdAgent to the new WMAgent [2]. The new production agent allows to inject the configurations directly into the production system interface without the need of a direct human interactions with the computing team experts, but through dedicated APIs. These APIs have been integrated into the python library discussed in the previous paragraph.

It is interesting to notice that the XML snapshots constitute a natural way of exporting in a modular way (per campaign and per request) the database content in a standardized format which can be processed with various external tools (from python scripts using the PREP python library utilities to spreadsheets for dedicated statistical analysis).

At present two additional components are under development. One is a command line interface to PREP, which appears to be particularly useful for the specialists who need to perform a large number of repetitive actions on the system (for instance submission of many minor variants of the same basic request), and could clearly benefit from scripts automatizing them. The other ongoing project is the realization of an APIs connecting PREP to the DAS Data Aggregation System [3], which constitutes the central portal to access various information repositories about the production system of CMS. While from the PREP interface a dedicated link to the sample catalogue record for each request as provided by DAS is already available, the PREP specific information can be exported into DAS for a query combining it with information coming from other underlying sources of information.

4 User Interfaces

The main view of the PREP web interface when accessing it is shown in Figure 2.

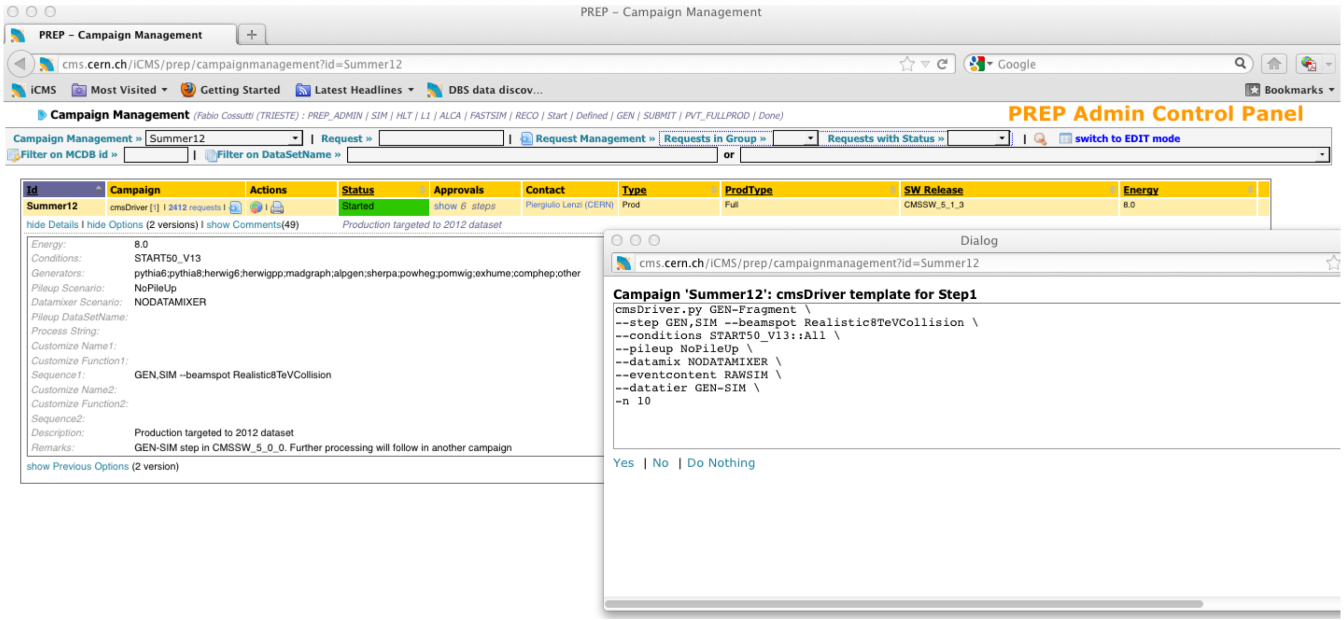

Figure 3: Campaign description view of the PREP web interface. Both the campaign available information and the definition of one available workflow are expanded.

The user and its privileges are specified in the first row, and a switch between the VIEW and EDIT mode is available, allowing to work both in a safe read-only view or in a mode which allows actions to be taken. As it can be seen, pull-down menus allow to access the list of the available items (campaigns, type of sample), and filters on different possible fields are available to help the user to quickly access the wanted record without browsing all the available ones (which has obviously also an impact on the PREP response speed). From this view the campaign one can be accessed, and its main features are shown in Figure 3.

The common menu bar on top is always present, while a field with the chosen campaign record is presented. Hyper-links allow to access the requests, the workflow definitions, the content of the database, and the list of the actions performed, as registered in the database itself. A sub-window with the campaign information is expanded for exemplification, as well as a pop-up window with the description of one of the workflows stored. A few icons allow to access to various functions: the XML snapshot of the database for this campaign record, the statistics status for the campaign (shown in Figure 4) and the complete view of the database content for the chosen campaign, a part of which is shown in Figure 5.

The hyper-link indicating the number of requests allow to access to their list for that given campaign. One example is shown in Figure 6. While the meaning of the shown columns changes, the basic format of the record presentation is the same of the campaign view. The status field allow to monitor the history of the production for the chosen request as known in PREP, where the information is imported by polling the DAS main aggregator.

5 Conclusion

The challenge represented by managing several thousands of simulated samples requested by the physics groups of CMS has required the development of dedicated policies and tools to accomplish the task. The PREP system concentrates in itself different functions: management tool for submission to the computing production agents, monitoring tool of the requests' status, database storing relevant information for physics analysis. After one year of operations during the LHC data taking it has proved an essential way of facing this challenge, and the natural entry point to the computing production infrastructure. The experience accumulated suggests to add some further functionalities, like a command line interface and an API to integrate its content in the CMS Data Aggregation System, to further expand its potential and facilitate its usage by an increasing community of physicists involved in data analysis.

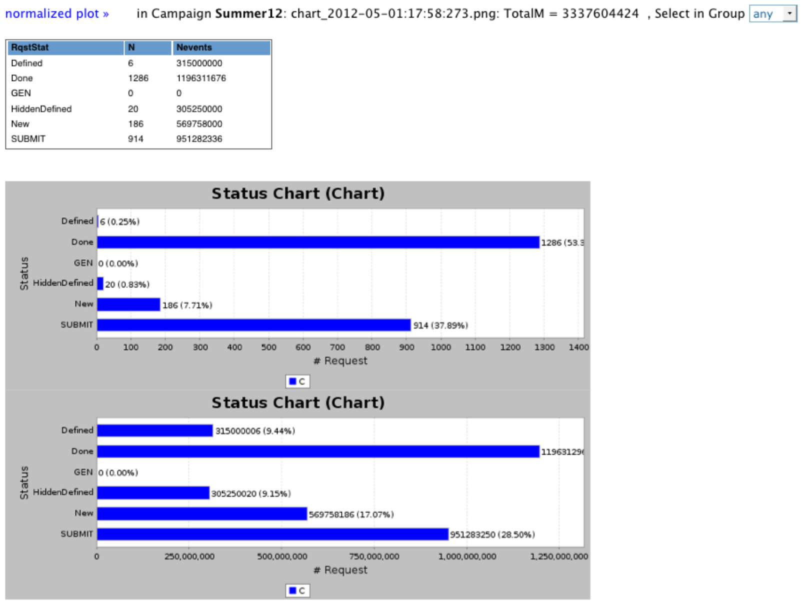

Figure 4: Campaign statistics window.

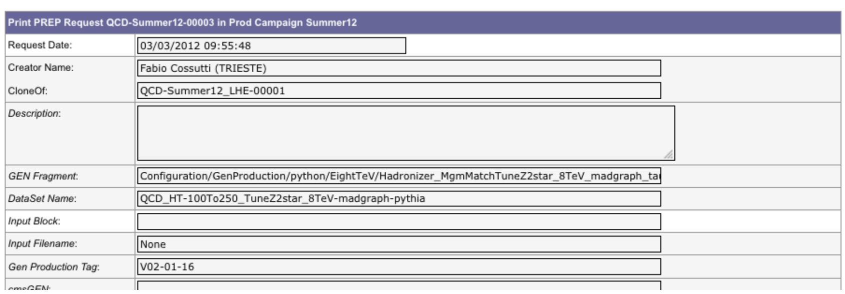

Figure 5: Detailed view of the database content for a chosen campaign.

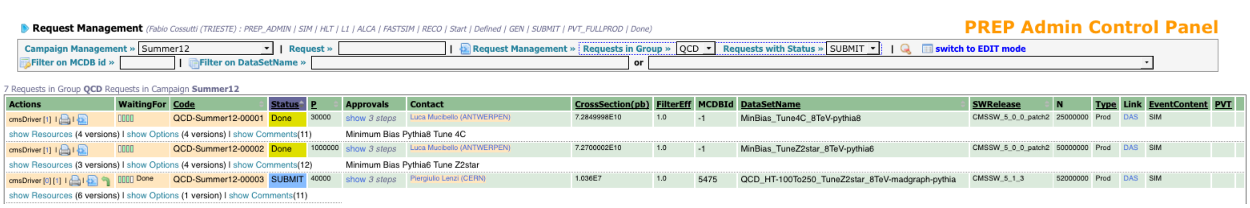

Figure 6: Requests description view of the PREP web interface.

References

[1] The CMS Collaboration 2006 *CMS Physics Technical Design Report, Volume I: Detector Performance and Software* CERN/LHCC 2006-001, CMS TDR 8.1

[2] Evans D *et al* 2008 *Large Scale Job Management and Experience in Recent Data Challenges within the LHC CMS experiment* Proceedings of Science (ACAT08) 032

[3] Kuznetsov V, Evans D and Metson S 2010 *The CMS Data Aggregation System* Procedia Computer Science (ICCS 2010) vol 1, issue 1, p 1535