

21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015)



Contribution ID: 382

Type: poster presentation

HLT configuration management system

The CMS High Level Trigger (HLT) is implemented running a streamlined version of the CMS offline reconstruction software running on thousands of CPUs. The CMS software is written mostly in C++, using Python as its configuration language through an embedded CPython interpreter. The configuration of each process is made up of hundreds of “modules”, organized in “sequences” and “paths”. As an example, the HLT configurations used for 2011 data taking comprised over 2200 different modules, organized in more than 400 independent trigger paths. The complexity of the HLT configurations and the large number of configuration produced require the design of a suitable data management system. The present work describes the designed solution to manage the considerable number of configurations developed and to assist the editing of new configurations. The system is required to be remotely accessible and OS-independent as well as easily maintainable easy to use. To meet these requirements a three-layers architecture has been chosen. On top of the “ConfDB” database a business logic manager has been introduced to handle the database operations, to perform the read and write rights check and to send a configuration to the user in a suitable format for the user interface. The graphical user interface (GUI) provides the features to display, modify and manage the configurations. The GUI design was carried out first by exposing paper sketches to the end-users and based on the their feedbacks a software mockup was implemented. At the end of the development process usability test will be carried out in order to measure the impact that the new GUI has on the development of configurations for the CMS-HLT.

Primary author: DAPONTE, Vincenzo (Universite de Geneve (CH))

Presenter: DAPONTE, Vincenzo (Universite de Geneve (CH))

Track Classification: Track1: Online computing