A Tour Through the CMS Data Model

Roger Wolf (on behalf of the CMS Collaboration)

Institut für Experimentalphysik der Universiät Hamburg Luruper Chaussee 149 22761 Hamburg

Abstract. The data model of the CMS experiment is outlined and the role of dedicated analysis software tools and the *Physics Analysis Toolkit* (*PAT*) therein are described. They support the standardization of common analyses operations like the association, combination or isolation of reconstructed objects in a user configurable way. They facilitate event content management and data access for the end-user sustaining the full flexibility of the CMS data model at the same time.

Keywords: CMS Data Model High Energy Physics
PACS: 02, 10

INTRODUCTION

The *Large Hadron Collider* (LHC) at CERN restarted operation in November 2009, fist at lower center of mass energies, then with an intermediate center of mass energy of $\sqrt{s} = 10\,\text{TeV}$ and finally at the designed center of mass energy of $\sqrt{s} = 14\,\text{TeV}$. The instantaneous luminosity thereby is expected to vary between $10^{29}\,\text{cm}^{-2}\text{s}^{-1}$ at startup and $10^{32}\,\text{cm}^{-2}\text{s}^{-1}$ at design luminosity. CMS is one of the major experiments of the LHC [1]. It comprises a full silicon tracking system [2], a fine granular lead tungstenate electromagnetic crystal calorimeter, a hadronic calorimeter and a large system of muon chambers with several 10 millions of readout channels overall. Its trigger and data acquisition are designed to cope with the expected event rates of 25 MHz trigger input rate, which will be reduced to the order of 100 Hz output rate with a transfer band width of up to 150 MB/s to tape [3]. The operation of CMS will lead to large amounts of data in the range of several hundreds of Peta bytes per year [4]. To cope with the challenge of distributing such large amounts of data and to guarantee short transfer cycles from the reconstruction to the analysis a processing structure has been designed, which is organized in three layers, called Tiers [5]:

- A Tier0 center (T0) hosting large computing resources at CERN will serve for prompt reconstruction of the incoming data utilizing first calibration constants and jet energy corrections.
- A group of central Tier1 centers (T1) will serve for data replication, re-processing and worldwide distribution via the grid.
- Tier2 (T2) and Tier3 (T3) centers will serve for further distribution and common analyses of the data as well as for the production of adequate amounts of simulated events.

The large amount of expected data as well as the complexity of the detector require a flexible data model. Data will be provided in different formats resembling different amounts of detector information. The most important data formats for the user will be the RECO format with an event size of approximately 500 kB/event containing all relevant reconstructed object information and the *Analysis Object Data* format (AOD) with an event size of approximately 100 kB/event containing a reduced set of reconstructed object information, which should be relevant for most physics analyses. [1]

THE DATA MODEL OF CMS

The CMS *Event Data Model* (EDM) is based on independent C++ plug-in modules with the ability to read from and write to a flexible event content [4]. This event content is based on ROOT object storage, which is translated into independent Trees and Branches [7], for the storage of basic reconstruction objects like hits or energy deposits in

[1] All event sizes resemble a rough estimate from simulated events of top anti-top quark pair production at a center of mass energy of $\sqrt{s} = 10\,\text{TeV}$, which might vary by a few percent depending on the version of the used CMS reconstruction and analysis software.

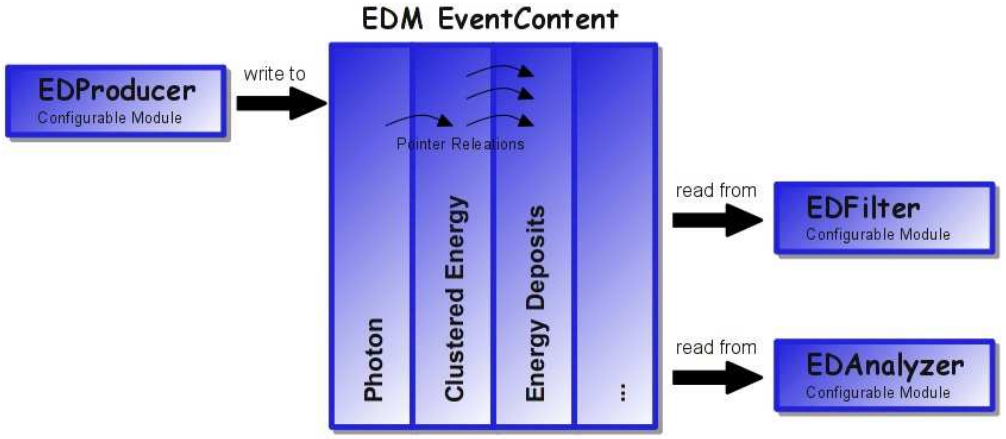

FIGURE 1. Graphical sketch of the EDM event content: basic reconstruction objects like energy deposits in the calorimeters are directly saved in the event content. They can be written to or read from the event content via independent C++ plug-in modules (indicated as EDProducer, EDFilter, EDAnalyzer in the figure). Clustered structures of these energy deposits can be combined to high level reconstruction objects like photons, via pointer relations. All collections are saved in parallel. Each collection may be kept or dropped at each reconstruction or analysis step.

the tracker, calorimeters or muon system of the detector. Other modules may combine this basic information to more complex analysis objects. E.g. hits in the tracker might be combined to tracks based on different algorithms or the same algorithms in different configurations. The basic objects are therein referred to via reference pointers in order minimize the redundancy of persistent information. In this way high-level analysis objects like electrons, muons or jets are reconstructed in several independent steps (represented by different modules) exploiting a hierarchical build-up procedure.

For instance a photon will be reconstructed energy deposits in the calorimeters, which will be combined to a clustered energy deposition [6]. The energy cluster will be referred to by a reference pointer. The energy cluster itself refer to the local energy depositions via own reference pointers. Each clustering or reconstruction step might be replaced by a different module exploiting an alternative algorithm or just a different configuration of the same algorithm. Collections of objects, which have been reconstructed with different configurations or different algorithms may be stored in parallel in the event content. Due to the reference pointer arithmetic this will be possible with minimal redundancy of persistent object information. A graphical sketch of the EDM event content is shown in figure 1 and 2.

Object collections may be kept or dropped at any state of the reconstruction or further analysis of the data. A data provenance instance provides a record of each reconstructed object summarizing all utilized modules and relevant configuration parameters that lead to the reconstruction of the object and thus of each single step of its reconstruction. In this way a clear identification of events and classification of objects can be guaranteed even after several skimming and replication steps. This model is optimized to the following requirements:

- flexibility of the reconstruction of high level analysis objects.
- guaranteed traceability and re-produceability of each reconstruction step for each high-level analysis object.
- minimal redundancy of persistent event information.

Both reconstruction and analysis should be possible based on the same data model and event content. To sustain the features of flexibility and especially of event provenance a decoupling step from the EDM event content (like the creation of private n-tuples of data) should be prevented or performed as late as possible during the analysis chain. The CMS data model provides tools to facilitate data analyses without the production of private n-tuples. They allow to perform analyses exploiting different C++ or python based analysis methods (ranging from plain ROOT to the full analysis data model).

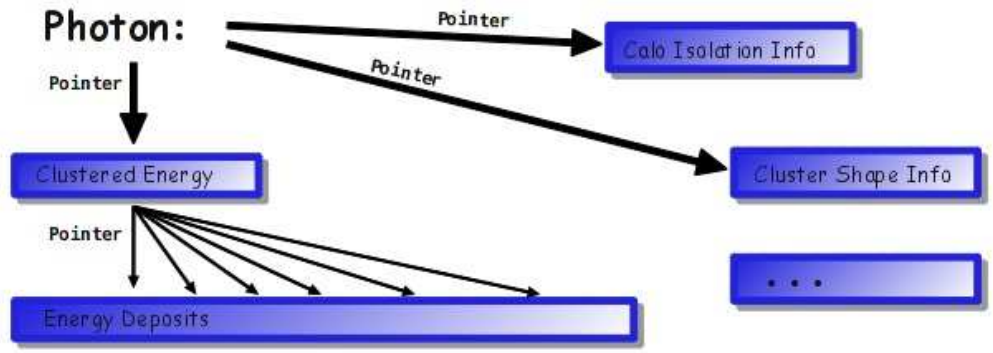

FIGURE 2. Sketch of the pointer relations for the example of an photon.

ANALYSIS TOOLS

The optimization of the event content with respect to flexibility and minimal redundancy of persistent object event information and the hereby implemented reference pointer arithmetics imply particular the danger of dead pointer relations in case of dropped object collections, which can complicate data access and event content management especially to beginners. While whole collections of energy deposits in the calorimeters or of hits in the tracker might be very space extensive the user might decide to drop these collections from the event content in later states of the analysis. This will turn all pointer relations to these objects from within higher reconstruction objects (like clustered energy deposits or photons) invalid at the same time. Dropping basic reconstruction information may thus have hidden influences on many higher-level reconstruction objects, which are difficult to foresee in all consequences by the user. It might thus very easily lead to unwanted loss of information. Among others this issue is addressed by the *Analysis Tools* packages of the CMS *Event Data Model*, which cover the following purposes:

- facilitate access to all analysis relevant data for the user, exploiting different C++ or python based analysis methods (ranging from plain ROOT to the complete set of features of the full CMS *Event Data Model*).
- provide standardized (python configurable) tools and algorithms for common analysis tasks, including tools for matching, object combination (to composed objects), object isolation, tag and probe methods and others.

The major component of the *Analysis Tools* packages in facilitating event content managements and access to all analysis relevant event information for the user is the *Physics Analysis Toolkit* (PAT) a toolkit to support event content configuration during later analysis steps after reconstruction. It circumvents the problem of dead pointer relations as described above by the introduction of an additional structure of PAT candidates (for high-level analysis objects like electrons, muons, jets, aso). These objects are produced via python configurable C++ plug-in modules during later analysis steps. This additional step allows directly to associate all analysis relevant information to each of the high-level reconstruction objects on the user's choice. Among others this information might include:

- object isolation in various definitions and detector components.
- object identification (like cluster shape information for photons or electrons).
- jet energy correction factors, object resolutions or reconstruction efficiencies.
- tracks associated to a jet, jet charge or jet flavor information.
- b-tagging information for jets.
- matched generator of trigger object information.

Apart from that it allows to store any kind of user defined information in C++ built-in or vector types via the definition of user functions. Depending on the user's configuration this information might internally be referred to via pointers (if the corresponding object collections are still kept in the event), or hard copied into the high-level analysis object, if the corresponding collection is dropped from the event. The different ways of accessing data are fully transparent when used during later analysis steps, as it will be provided by the same access functions at any time. The process of hard copying information into the high-level analysis objects allows to reduce the event content significantly in a python configurable and flexible way. An example is given for jets, which are reconstructed from calorimeter objects: in a typical event containing top anti-top quarks several hundred calorimeter objects might be present of which only a small fraction will be clustered into analysis relevant jets. The option to hard-copy the information of each calorimeter object into the reconstructed jet allows to drop the disc space extensive collection of all calorimeter objects preserving the information of each single calorimeter object that has been associated to the jet for later analysis purposes.
This fully transparent way to drop information within the CMS data model allows effectively to reduce the event size from typically 500 kB/event (100 kB/event) in simulated events containing a top anti-top quark pair in RECO (AOD) format to typically less than 16 kB/event at the same time decreasing the necessary time to access the reconstructed event information and preserving event provenance as introduced above. The PAT candidates are fully integrated within the CMS data model and as well as all other reconstructed event information may be accessed using plain ROOT taking advantage of all speed benefits in accessing the data. The reduced analysis layer is meant to replace common n-tuples used in former experiments. It is sometimes referred to as pat tuple.
The proposed user analysis cycle is to receive skims of reconstructed events from one of the associated T2 or T3 centers, to create and configure a pat tuple containing all information that is relevant for the given analysis on a mid term cycle and to perform fast interactive analyses on these pat tuples.

REFERENCES

1. R. Adolphi et al. **(CMS Collaboration)** *The CMS experiment at the CERN LHC*, JINST 0803:S08004,2008, 361pp.
2. **(CMS Collaboration)** *The Tracker Project Technical Design Report*, CERN-LHCC-1998-006, April 1998.
3. **(CMS Collaboration)** *CMS Physics TDR, Volume I*, CERN-LHCC-2006-001, 2 February 2006.
4. **(CMS Collaboration)** *CMS Computing TDR*, CERN-LHCC-2005-023, 20 June 2005.
   LCG Computing TDR, CERN-LHCC-2005-024, 20 June 2005
5. Francisco Matorras, contribution to these proceedings.
6. J. Nysten, *Photon reconstruction in CMS*, Nuclear Inst. Phsy. Res. **A**, 2004, pp. 194-198.
7. R. Brun and F. Rademakers, *ROOT-An Object Oriented Data Analysis Framework*, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl.Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.