The CMS Analysis Model

Benedikt Hegner
European Organization for Particle Physics (CERN), 11021 Geneva 23, Switzerland
E-mail: benedikt.hegner@cern.ch

Abstract. We review the options for performing user analyses at CMS. In this text we put the CMS physics analysis software framework and other tools in the context of a bigger picture of performing data analysis. The flow of physics data through layers of skimming is briefly described, and the user options for analyzing objects on local computing resources or laptop discussed in turn. We present the first experiences with this model obtained after the recent start of the LHC physics programme.

1. Introduction
The Large Hadron Collider (LHC) began operation in November 2009, first at a lower center of mass energy of $\sqrt{s} = 2.3$ TeV. During 2010 the energy was increased to $\sqrt{s} = 7$ TeV and the LHC now delivers an instantaneous luminosity of $10^{32}$ cm$^{-2}$ s$^{-1}$.

The Compact Muon Solenoid (CMS) is one of the two large experiments at the LHC [1]. It comprises a full silicon tracking system, a fine granular lead tungstenate electromagnetic crystal calorimeter, a hadronic calorimeter and a large system of muon chambers with 10’s of millions of readout channels.[2] Its trigger and data acquisition systems are designed to cope with expected trigger rates of 25 MHz, which will be reduced to the order of 100 Hz output rate with a transfer bandwidth of up to 150 MB/s to tape [3]. The operation of CMS will lead to large amounts of data in the order of petabytes per year [3]. The computing model is organized in four tiers in order to cope with the challenge of distributing such large amounts of data and to guarantee short transfer cycles from the reconstruction to the analysis [4]:

- A Tier-0 center 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 Tier-1 centres serves for data replication, re-processing, and world-wide distribution to additional centres via the grid.
- Tier-2 and Tier-3 centres will serve for further distribution and common analyses of the data as well as for the production of very large samples 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 reconstructed event information. One of these, the RECO format, contains all relevant reconstructed object information and has a typical event size of 500 kB/event. Another important one, the Analysis Object Data (AOD) format, contains a reduced set of reconstructed...
object information, that is required for performing most physics analyses. It has an event size of approximately 140 kB/event¹.

2. The data flow from the end user perspective
The described computing model can be mapped into a logical model of physics selection and analysis steps. The ideal scenario is shown in Figure 1. Data arriving at the Tier-0 are split according to trigger information into so-called Primary Datasets. Each Primary Dataset, which for example could contain all events passing a high $p_T$ muon trigger, then gets transferred to one or more Tier-1 centres. Basing the split on persistent information facilitates re-reconstruction, as it avoids migrations from one Primary Dataset into another.

In the next step, the Primary Dataset is split into a refined subset of events, so-called Central Skims. The selection can be based on trigger and reconstruction information and be already rather specific for a physics use case. The result then gets shipped to the Tier-2 centres. Central Skims are usually provided in the already mentioned AOD data tier.

Users have direct access to the data only at the Tier-2 centres. There they can run their analysis jobs, creating e.g. new datasets or plots. As a final step, users can copy the final selection of events to local resources and perform interactive analysis.

The technical experience with the CMS Event Data Model (EDM) was already presented during CHEP 2009 [5], and in this paper we focus on the software environment for the user-driven analysis jobs with analysis of first real data.

![Figure 1. Schematic view of the processing steps before a user can get direct access to data located at Tier-2 centres. Analysis jobs running on the Tier-2 produce a smaller output which is then copied over to local computing resources or laptop.](image)

3. The Physics Analysis Toolkit
The EDM has been optimized for flexibility of the reconstruction. As a consequence of the design the information in the AOD data tier is not optimized for usability in that information directly related to a physics object (e.g. a jet) might be stored amongst different collections (e.g. their b-tagging information). [6] This has motivated the development of a special software layer targeted at physics analysis, the Physics Analysis Toolkit (PAT). The PAT provides the physicist with a consistent interface to the event reconstruction information. Information such as generator or trigger matching, jet energy corrections, etc. can be computed with a rich set of tools, and then stored alongside the already existing object data. A more detailed description of the CMS Physics Analysis Toolkit can be found in these proceedings [7].

¹ All quoted event sizes represent a rough estimate based on simulation of $t\bar{t}$ events, including MC information.
One important feature set provided by the PAT is the possibility of trimming the event size to the level needed for a certain analysis. This can be done by thinning out object information or by simply removing objects from a collection. For example, in the following all electron candidates with a $p_T > 10 \text{GeV/c}$ are selected:

```cpp
selectedPatElectrons = cms.EDFilter(PATElectronSelector,
    src = cms.InputTag(patElectrons),
    cut = cms.string(pt > 10)
)
```

Selection cuts may be specified as strings in a configuration file, which is extremely convenient for the physicist. This functionality is provided by a custom expression parser, which is based on C++ reflection information and REFLEX [8].

CMS provides several tools to inspect and trim events to reduce event size. Choosing sensible settings the event size needed for an entire physics analysis can be as small as 5kB/event. This is small enough for data skims to be copied to a user’s laptop for further interactive analysis.

4. Interactive Analysis on local resources

Since its initial design the EDM was set-up such that it can be used in interactive environments on local resources (e.g. laptops). [5] Data storage is based on ROOT [9] and the EDM files can be opened and inspected with a standard-alone ROOT installation, e.g. via a TBrowser. This gives direct access to all stored data in the files, namely the members of all stored objects.

In addition to this so-called 'bare ROOT access', CMS provides a lightweight interactive environment (FWLite), which consists of ROOT, the data format libraries and dictionaries, and some basic helper functions. Using FWLite the EDM file looks like a real object storage and interactive object-oriented data analysis inside either a CINT or a Python shell are possible. Compiled code using the FWLite libraries can be set up in a way that the code is 100% interchangeable between FWLite and the full CMS software framework.

Together with the locally installable CMS event display Fireworks[10], FWLite and a skimmed down PAT file, a physicist should be able to perform analysis on his own laptop and to produce the final plots for publication. Due to the tight integration with ROOT, useful parallelization functionalities like PROOF-Lite can be used to further improve the turnaround times of individual analyses.

5. Integration of n-tuple analyses in a robust way

During the 2010 data taking period user feedback was systematically and continuously collected. Almost all of the analysis groups were using the PAT as part of their analysis workflow. [8] This greatly facilitated the exchange of knowledge amongst the analysis groups. Many users were taking advantage of the feature of running the CMS event display on their laptop, which helped them to quickly identify, understand and debug various problems inside the reconstruction algorithms.

There are several advantages of using a commonly designed and maintained software framework such as the PAT, namely

- the prevention of ‘re-inventing the wheel’,
- a more systematic scrutiny and validation,
- the possibility to share and exchange technical expertise,
- an easier comparison of results across groups, and
- the possibility to use the PAT as a ‘common language’,


However, experience has shown that there was a tendency to write out plain n-tuples in a format inherited from previous experiments. As a result many co-existing n-tuple infrastructures with very distinctive kinds of bugs and flaws were created. Due to this code duplication so-called synchronization exercises were carried out, in which these n-tuples were validated against each other. This significantly slowed down the analysis activities of some physics analysis groups.

While this procedure ensures a rigorous check of selection and n-tuple producer code, a single, common tool with a decent set of tests (and at most one independent test-implementation) provides a by far better maintenance solution for the long term. This situation as well hindered beginners to apply their expertise gained at local resources (using custom n-tuples) in the distributed environment of grid processing (EDM files). This led to a 'disconnect' of many people from the grid resources. Finally the inspection of special events passing the cuts with an event display is just not possible with custom data formats.

While it is very encouraging that the physics analysis tools are used for all data intensive production analysis, the tools provided for interactive analysis were mainly ignored. This was a major concern and demanded some rethinking on what tools CMS provides for this use-case.

To overcome the mentioned problems, CMS introduced a lightweight n-tuple making functionality as integral part of its software framework. Vectors of basic types are stored into an EDM file. This is entirely driven by Python job configuration files, and a typical configuration can be as follows:

```
<...>
  tag = cms.untracked.string("Dau1Pt"),
  quantity = cms.untracked.string("daughter(0).pt")
<...>
```

with `tag` as the name of the quantity in the produced n-tuple, and `quantity` as string expression for the number to be stored. It is evaluated using the already mentioned string expression parser, and provides a very convenient way of defining an n-tuple content.

Being an EDM file, the produced 'EDM-tuple' inherits its advantages:

- EDM files contain provenance information [11], which records the configuration parameters used during the creation of the data. This greatly facilitates reproducibility and introspection.
- All tools available for dealing with EDM files can be used, which is most important for running analysis jobs on the grid.

In particular, the combination of an entirely configuration-driven tool and the provenance tracking makes the created n-tuples almost self-descriptive. For each number stored in the EDM file the corresponding string expression can be looked up in the provenance information. Another advantage of this setup is the possibility to store n-tuple data alongside standard CMS data types. This enables the user to freely combine n-tuples with PAT objects in his final analysis dataset. This might facilitate the move from an n-tuple-centric approach to a PAT object oriented approach as both are supported within the same framework.

In the meantime many individuals, and at least two of the central physics analysis groups, replaced their custom n-tuple tools by the EDM tuple described above. They directly benefit from all the advantages related to maintenance and ease of validation mentioned above.

6. Further improvements resulting from practical experience

Significant feedback has been forthcoming from physicists using the analysis tools. In particular several concrete proposals were received for extending the physics analysis software. Following
the dynamic and adaptive model of the PAT, we were able to address most of these in a rather short time scale [7]. An instructive example of a feature that was asked for is a ROOT \texttt{TTree::Scan} like feature in FWLite, with configurable tables as a fast way of inspecting an object collection. The provided solution is again based on the string expression parser and the columns defined via simple strings. Together with a similar functionality for histogram creation, this has become a frequently shown example of how to do interactive analysis with official CMS data types, and was well received within the CMS community.

Adequate documentation is an essential piece of a physicist’s analysis environment. In CMS the main resource for end-user documentation is the \textit{WorkBook}. It is a wiki-based document that serves as the first entry point for a user on how to go about approaching physics analysis. In addition CMS provides regular analysis software tutorials [12]. In the beginning the incompleteness of the documentation was perceived to be the main reason for people not taking advantage of all of the centrally provided tools. The introductory analysis chapter of the WorkBook was therefore rewritten. It now covers the interactive tools of CMS more prominently and guides users towards the intended analysis workflows. The introductory tutorials were changed in a similar way. The feedback to the first two tutorials using the improved content is very promising, and the mentioned improvements seemed very worthwhile [13].

7. Summary

In 2009 and 2010, CMS gained first experience with this analysis software environment based on the Physics Analysis Toolkit, which provides a rich set of tools aiding the physicist in performing data analysis.

The PAT was used rather successfully for the data intensive processing carried out during the preparation of the first physics publications of CMS. The underlying technology is flexible and powerful, and the tools could quickly be adjusted to the evolving user needs.

On local resources, the approach of an n-tuple based interactive analysis is still predominant. CMS established a common EDM n-tuple tool serving this use case. This is slowly replacing the plentiful frameworks individuals and groups inherited from previous experiments.

The recently improved documentation and training on the CMS analysis software seems to cause a shift from using n-tuples to using the official data formats directly. It is too early though to draw any real conclusions whether this will be a lasting effect. The provided analysis environment greatly reduced the time physicists spend on maintaining individual software, and ensured the productivity of the physics analysis groups during the first LHC physics run.

References

[1] CMS Collaboration (2008): The CMS experiment at the CERN LHC, \textit{JINST} 0803:S08004, pp361
[2] CMS Collaboration (2006): CMS Physics TDR, Volume 1, CERN-LHCC-2006-001
[3] CMS Collaboration (2005): CMS Computing TDR, CERN-LHCC-2005-023
[4] CMS Collaboration (2005): LCG Computing TDR, CERN-LHCC-2005-024
[5] P. Elmer, B. Hegner, L. Sexton-Kennedy (2009): Experience with the CMS event data model, \textit{J. Phys.: Conf. Ser.} \textbf{219}, 032022
[6] Fabozzi F, Jones C D, Hegner B and Lista L 2008 Physics Analysis Tools for the CMS Experiment at LHC \textit{IEEE Trans. on Nuclear Science} 55 6 3539-3543
[7] A. Hinzmann (2010): Tools for Physics Analysis at CMS, these proceedings
[8] C.D.Jones, L.Lista and G.Petrucciani (2009), Expression and cut parser for CMS event data. Proceedings CHEP 09
[9] R. Brun, F. Rademakers (1996): ROOT - An Object Oriented Data Analysis Framework, Proceedings ALHENP’ 96 Workshop, see also root.cern.ch
[10] C.D. Jones et al. (2010): Event Displays for the Visualization of CMS Events, these proceedings
[11] C.D. Jones et al (2009): File Level Provenance Tracking in CMS, \textit{J. Phys.: Conf. Ser.} \textbf{219}, 032011
[12] S. Malik (2010): Perspective of User Support for the CMS Collaboration, these proceedings
[13] K. Lassila-Perini et al. (2010): Planning and organization of an e-learning training program on the analysis software in CMS, these proceedings