CMS Detector Description for Run II and Beyond

Gaëlle Boudoul (IPNL), Ianna Osborne (Fermilab)

On behalf of the CMS Collaboration

CMS Detector Challenges

- Run 2 is about to start with unprecedented beam conditions
  - 25 ns bunch spacing (instead of 50 ns)
  - Higher luminosity (1.3E34 $cm^{-2}s^{-1}$ in 2015, up to 1.7E34$cm^{-2}s^{-1}$ after the second long shutdown)
  - Higher energy (13TeV, compared to 8TeV)

- During the LHC Long Shutdown (LS1), several changes within the detector occurred to cover the CMS physics run 2 program
  - Completion of muon coverage
  - Photodetectors (Hadron Calorimeter) replacement
- Several hardware changes are also foreseen during Run 2
  - L1 Trigger upgrade
  - HCAL upgrade (photo-detectors, electronics)
  - Pixel upgrade

- Beyond Run2, diverse upgrades are proposed, where dramatic changes of the CMS detector are foreseen
  - 2023: High Luminosity at LHC project (HL-LHC)
  - Tracker , Calorimetry, Muon chambers upgrades…

CMS Detector Description

- The CMS detector description models the CMS Detector in great detail, including all changes introduced during LS1 or planned for the Phase 1 and Phase 2 upgrades.
  - This is a crucial part to ensure precise simulation of the physical process and detector response during collisions
  - A very high degree of accuracy is required in the description of active and passive materials

- CMS Detector Description (DD) is an integral part of the CMS software multithreaded framework.
  - CMS software (CMSSW) has evolved to be more flexible and to take advantage of new techniques
  - In this presentation, we will discuss the limitations of the Run I DD model and changes implemented for the restart of the LHC program in 2015.

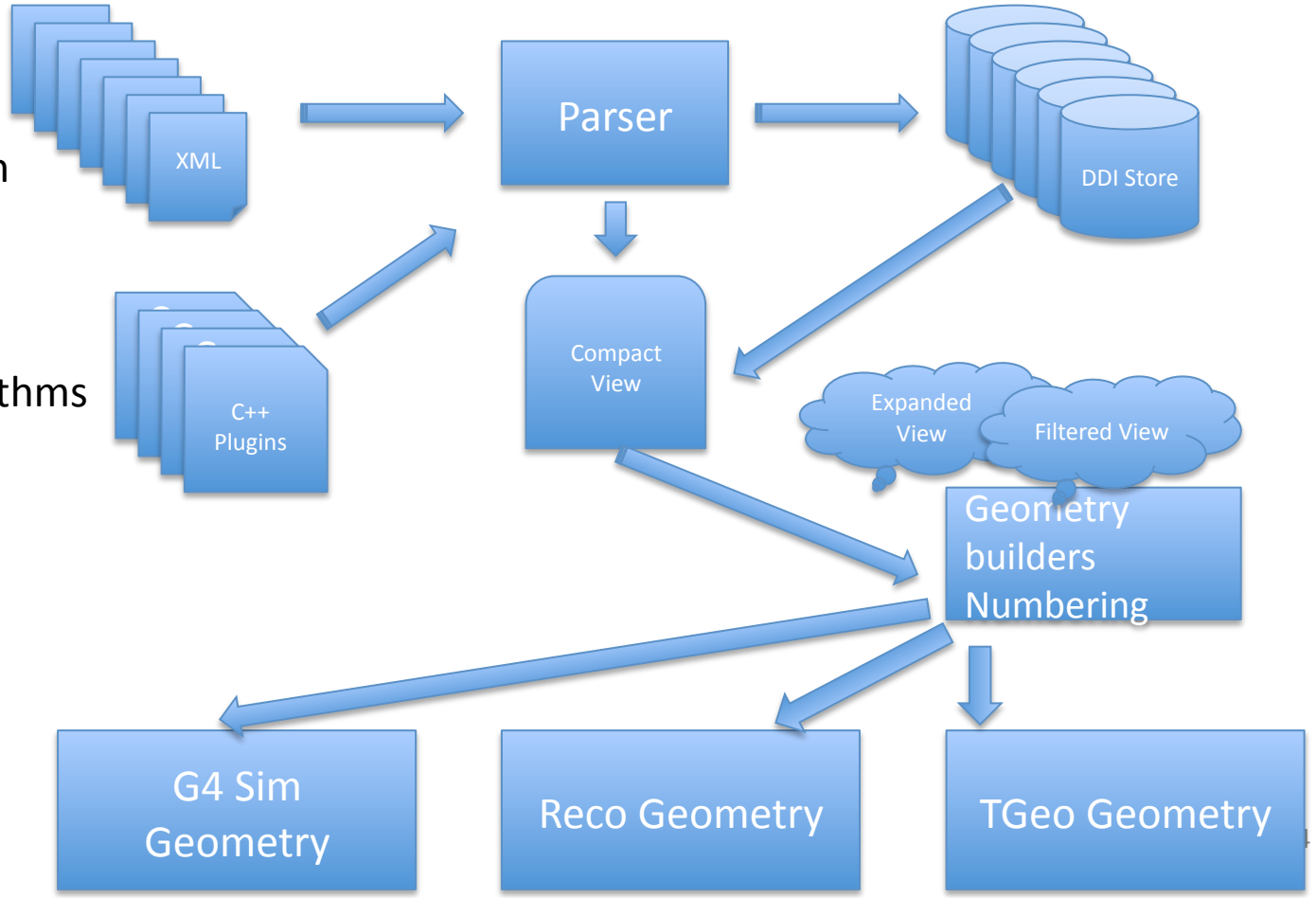
- The DD is a common source of information for Simulation, Reconstruction, Analysis, and Visualisation
  - Allowing different representations as well as specific information for each application.

Original Concepts

- Detector Description (DD) is the main source for other geometries
  - Describes Logical volumes associated with a Solid and a Material and their local or global positioning (similar to G4)
  - Constructs a Compact (DAG), an Expanded (Tree) and a Filtered views
  - Provides Stores for Materials, Solids, Logical Parts, Specifics, Rotation matrices
  - Modular sub-detector description in XML and C++

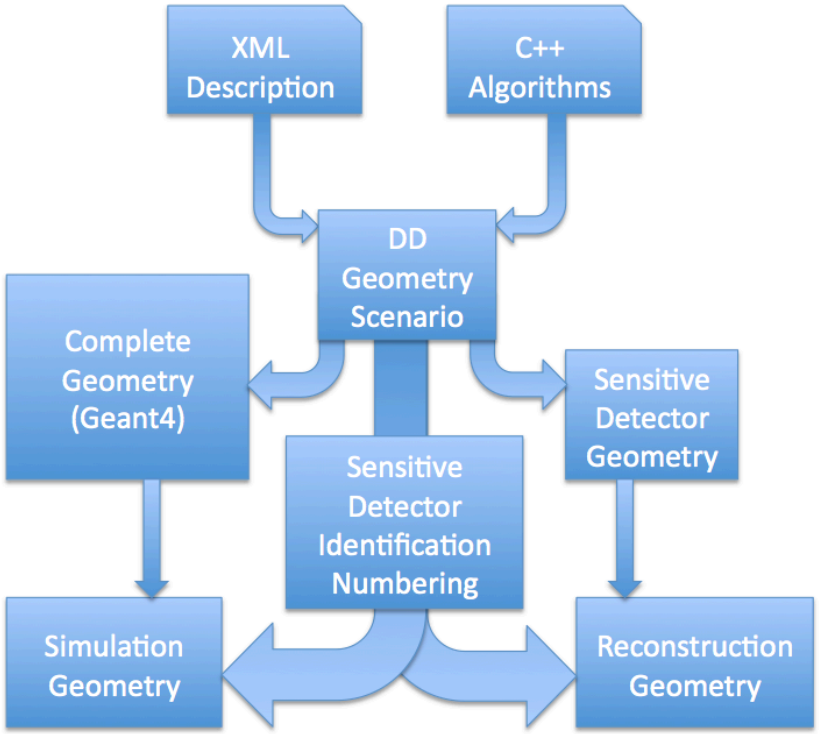
Modular description as a list of XML files

Algorithms



Geometry Incarnations

- Multiple geometries describing one detector
  - DD to G4 geometry for simulation
    - Simulation geometry is constructed from DD, sensitive volumes assigned unique IDs
  - DD to "skimmed" Reco geometry
    - Only sensitive volumes with their unique IDs
  - DD to ROOT (TGeo) for visualisation
    - Two geometries constructed – simulation and reconstruction
- Many to one mapping between simulation ar reco Geometry
  - Different material budget (= different simulation geometries) correspond to one reconstruction geometry
  - Both have identical unique IDs associated
  - with sensitive volumes
- ROOT geometries are either produced on demand (in full framework visualization) or read from a file distributed with a release



New Techniques

- More robust algorithmic positioning with DD Algorithm implemented as C++ plugins
  - Easier to use and debug
  - Moving away from volume positioning as a verbose XML descriptions
- Faster View searches based on different RegExp criteria
- Use of less error-prone Vector parameter description
- Deployment of C++11/14 standard
  - New language features for cleaner code and more efficient implementation
- Moving towards proper multi-threading (see the talk about CMS Threaded Framework by C. Jones – This session)
- Embraced GitHub for development
  - More eyes on code and its easier discussion allows better quality code
  - No restriction on who can modify what – the code is accepted or rejected based on the discussion followed by formal approval
  - Still many CMSSW development branches to follow

Example: Improved XML Description

- Old Vector description
  - No information on how many entries; defined by the end of parsing

```
<Parameter name="phioff" value="0.0*deg"/>
<Parameter name="phioff" value="0.0*deg"/>
<Parameter name="phioff" value="0.0*deg"/>
<Parameter name="phioff" value="10.0*deg"/>
<Parameter name="phioff" value="10.0*deg"/>
```

- New Vector description
  - Defined number of entries is checked on parsing

```
<Vector name="phioff" type="numeric" nEntries="5">
  0.0*deg, 0.0*deg, 0.0*deg, 10.0*deg, 10.0*deg
</Vector>
```

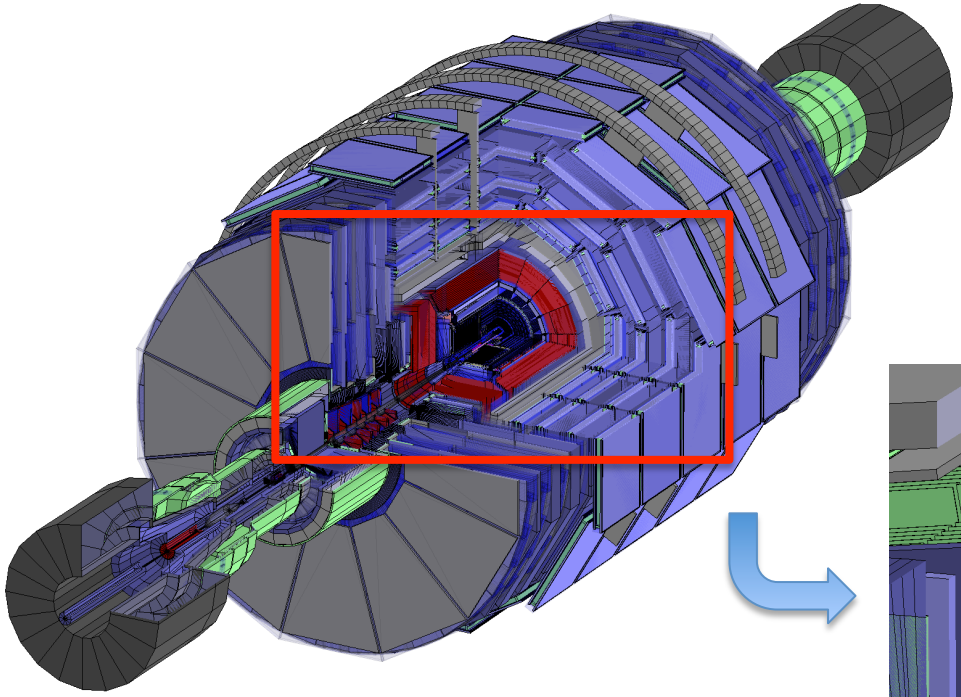
Increase of the robutness of the code:
parameter values given in a single place rather than spread over the xml files

Example: Improved XML Description

```
126   -   <rChild name = "AlignmentTubes:ATEV" />
127   -   <ParE name="n" value="1" />
128   -   <ParE name="radius" value="0*mm"/>
129   -   <ParE name="center" value="564*cos(157.5*deg)*mm"/>
130   -   <ParE name="center" value="564*sin(157.5*deg)*mm"/>
131   -   <ParE name="center" value="0*mm"/>
132   -
133   -    <!-- first rotation -->
134   -    <!-- first rotate the mirror about 0 degrees w.r.t. to the x-axis -->
135   -    <ParE name="rotateSolid" value="0*deg"/>
136   -    <!-- theta=0*deg -->
137   -    <ParE name="rotateSolid" value="0*deg"/>
138   -    <!-- phi = 0*deg -->
139   -    <ParE name="rotateSolid" value="0*deg"/>
140   -    <!-- rotate by 0*deg around (theta,phi) -->
141   -
142   -    <!-- second rotation -->
143   -    <!-- rotate the tube, so it has the appropriate angle w.r.t. the z-axis -->
144   -    <ParE name="rotateSolid" value="0*deg"/>
145   -    <!-- theta=0*deg -->
146   -    <ParE name="rotateSolid" value="0*deg"/>
147   -    <!-- phi=0*deg -->
148   -    <ParE name="rotateSolid" value="(157.5-90)*deg"/>
149   -    <!-- rotate around (theta,phi); the angle is the position angle in phi -->
150   -  </AlgoPosPart>
81    +   <String  name="ChildName"    value="AlignmentTubes:ATEV" />
82    +   <Numeric name="N"            value="1" />
83    +   <Numeric name="StartCopyNo"  value="1" />
84    +   <Numeric name="IncrCopyNo"   value="1" />
85    +   <Numeric name="StartAngle"   value="0*deg"/>
86    +   <Numeric name="RangeAngle"   value="360*deg"/>
87    +   <Numeric name="Radius"       value="0*mm"/>
88    +   <Vector  name="Center" type="numeric" nEntries="3"> 564*cos(157.5*deg)*mm, 564*sin(157.5*deg)*mm, 0*mm</Vector>
89    +   <Vector  name="RotateSolid" type="numeric" nEntries="6"> 0*deg, 0*deg, 0*deg, 0*deg, 0*deg, (157.5-90)*deg </Vector>
90    + </Algorithm>
```

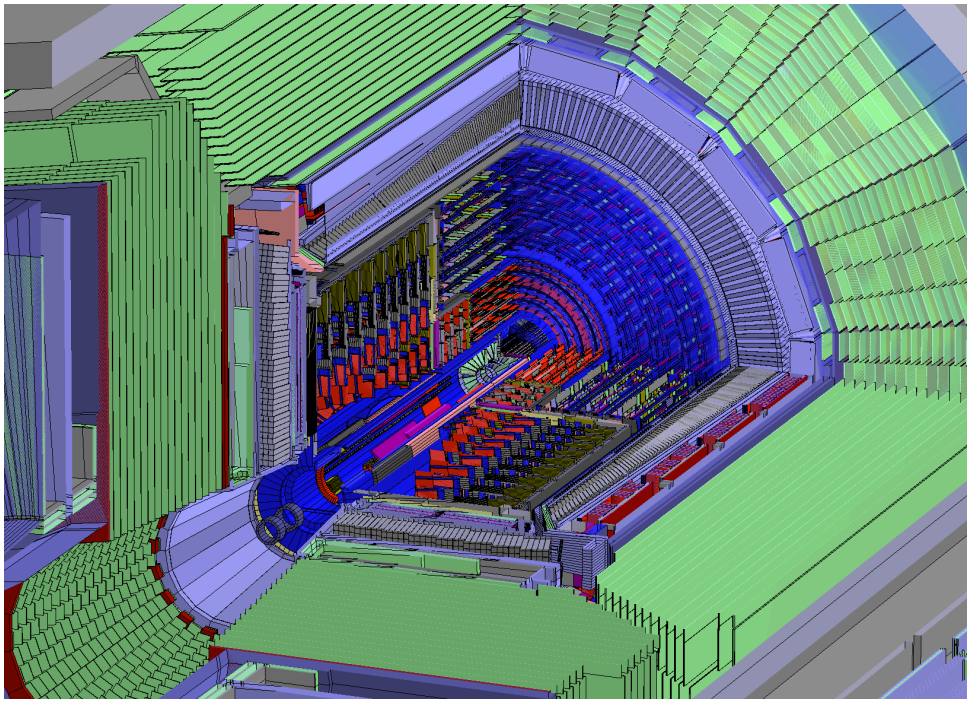
- New Algorithm description
  - Uses Vectors

Example of CMS representation



Three-dimensional view of the CMS detector

Passive and active material can be displayed



Next challenges to be adressed

- Fully multi threading of DD code is limited by its design due to internal choices to save time and memory which are less relevant today. We are working to remove limitations including:
  - Global state
  - XML double parsing
  - Writable stores
- Improving CMS geometry implementations (User code)
  - Hardcoded global constants and statics
  - Evaluating increase in 32 bit detector ID defintions
    - It is not sufficient to describe future subdetectors
  - Reducing duplication in XML by replacing single differences between object definitions with algorithms where possible

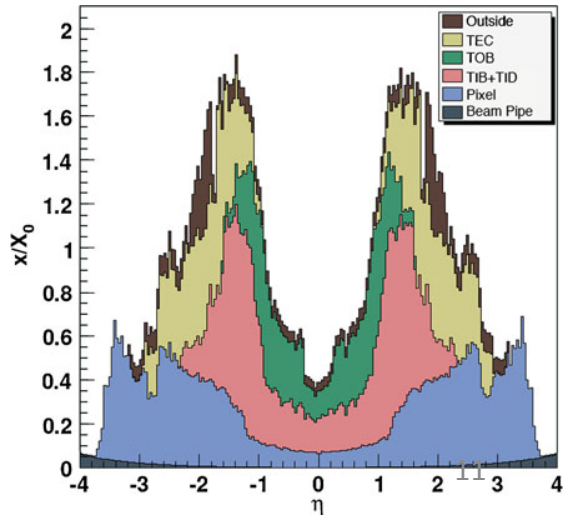
Integrating new geometries

- Important for CMSSW developers to be able to easily integrate new detector concept
  - CMS major changes are ahead of us and the design for CMS for HL-LHC is still to be finalized
  - Physics performance studies from realistic simulations are essential to design and develop hardware

- XML description are created from detector concepts or engineering drawings
  - Visualization tools developed within CMSSW are used to inspect the geometry.
  - Modular description of SubDetector (see next slide)
  - The geometry can be first tested within the full CMSSW framework using python configuration file before populating the CMS Conditions DataBase
    - Usage of Database has more versioning flexibility and better performance

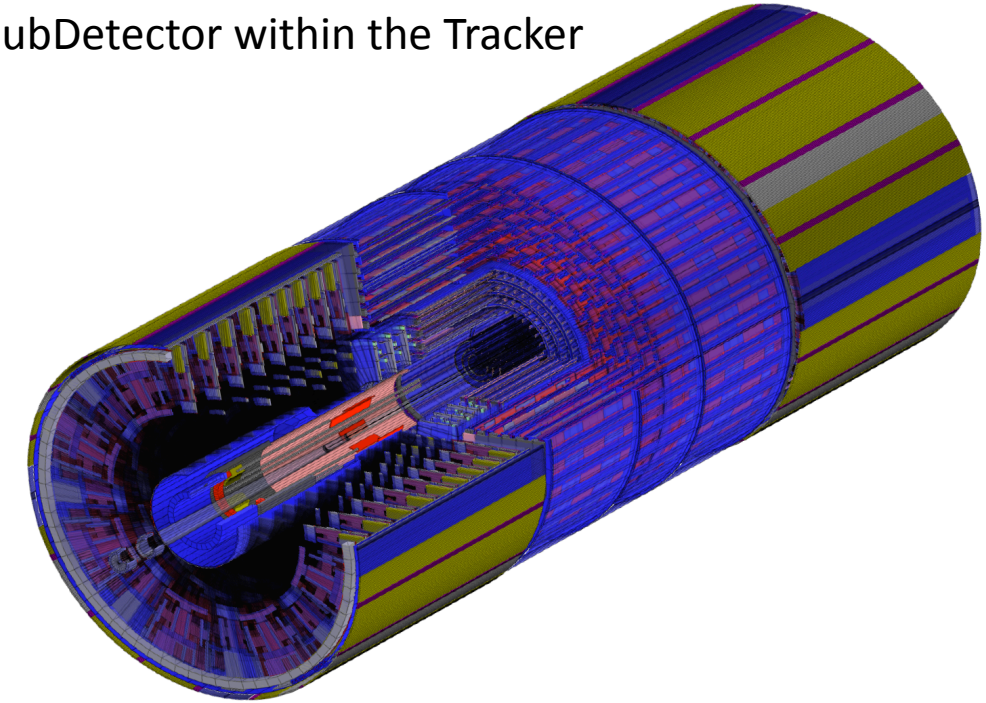
- Validation Procedure fully in place
  - Material Budget histograms check
  - Unit Test
  - Test samples routinely produced to keep track and identify any observed changes

Example of CMS Tracker Material Budget plot



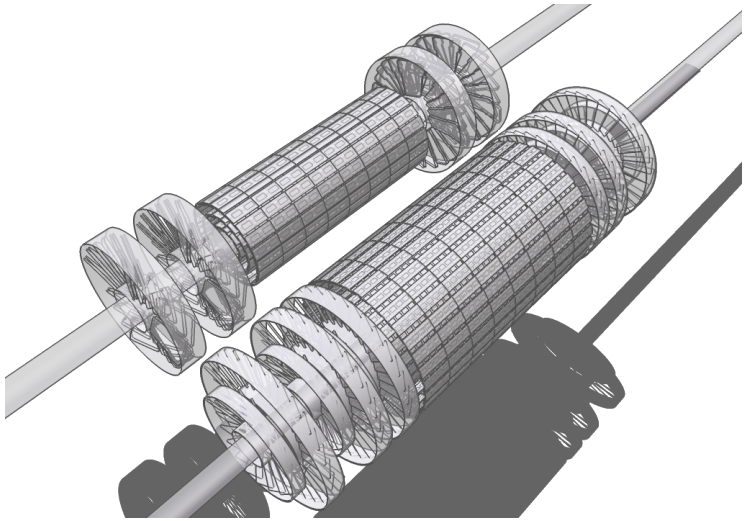
Modular CMS Detector Description

Example of the Pixel SubDetector within the Tracker



Present Pixel Geometry (3 layers- 2disks)

Present Pixel or the future Pixel description can be exchanged within the Tracker



Future Pixel Geometry (4 layers- 3disks)

Summary

- Many developments occurred during the long Shutdown
  - Cleaner and faster Detector Description
- Both Run 1 and Run 2 Detector Descriptions coexist in the same software release
  - Ongoing effort to support the CMS upgrade detector concepts within the same release as run1/run2
- Detector Description improvements continue
  - Isolated workarounds tackled individually
  - Legacy code cleanup
  - Enforcing better coding practices
- Geometry Description is crucial for Detector R&D developments requiring precise physics performance studies
  - The DD together with the Framework should address the need of flexibility in order to describe different detector concepts for future upgrades