

Geometry Description Markup Language (GDML)

Witek Pokorski

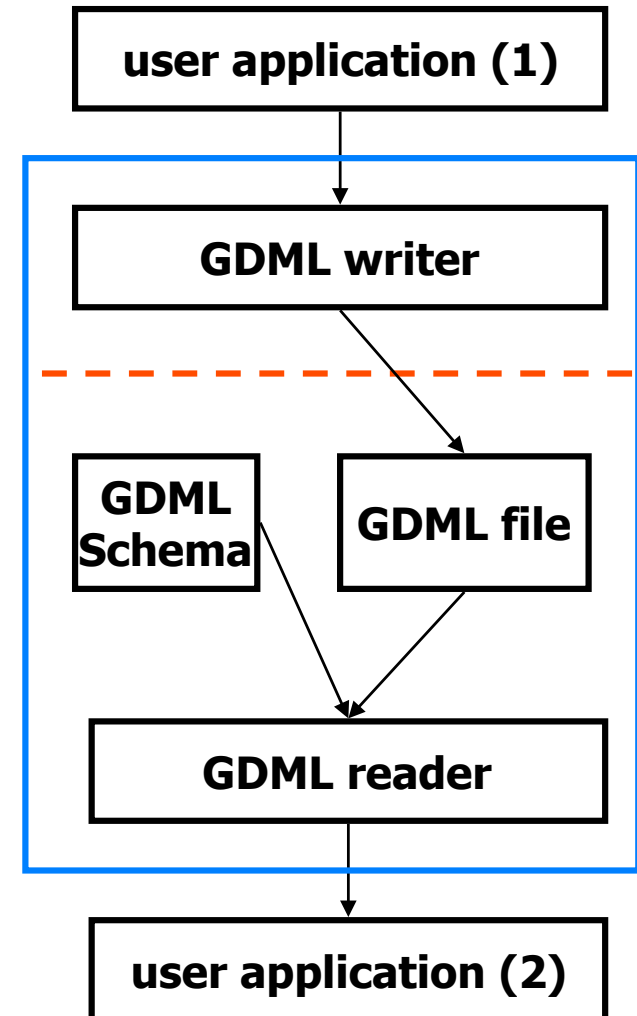
24.02.2010

GDML - Motivation

- simulation toolkits come with their native geometry description formats
 - many (most?) of the users do not implement geometry in those formats
- users use their own geometry description formats providing more flexibility, but:
 - they are integral parts of experiment software frameworks
 - cannot be easily exported in application independent way
- GDML has been developed
 - to have an application independent and flexible geometry format
 - to be able to interchange geometry between different applications for the purpose of
 - physics validation/comparison, visualization, debugging

GDML components

- GDML is defined through XML Schema (XSD)
 - XSD = XML based alternative to Document Type Definition (DTD)
 - defines document structure and the list of legal elements
 - XSD are in XML -> they are extensible
- GDML can be written by hand or generated automatically
 - 'GDML writer' allows writing-out GDML file
- GDML needs 'reader'
 - 'GDML reader' creates 'in-memory' representation of the geometry description



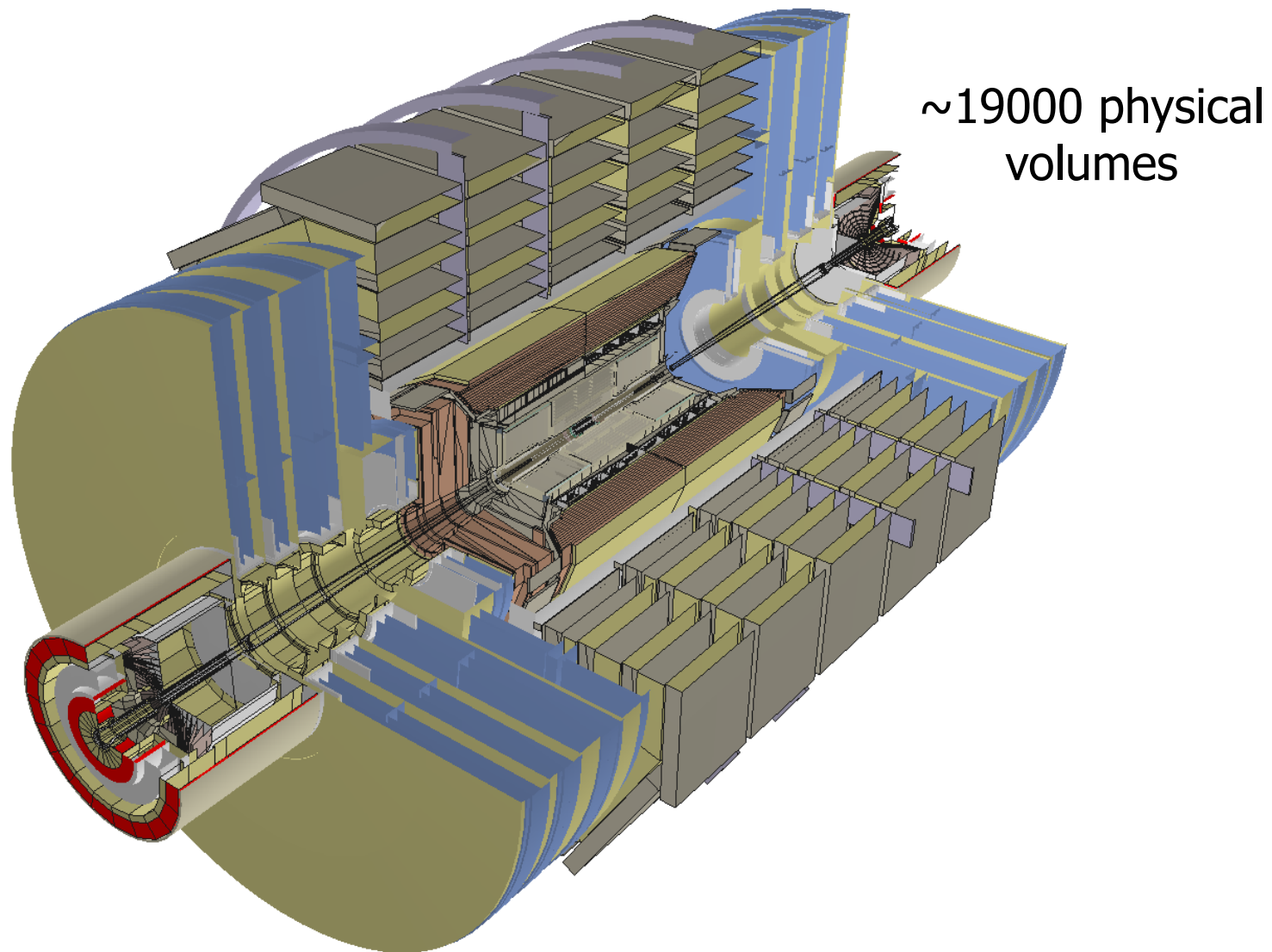
GDMML document



What exists

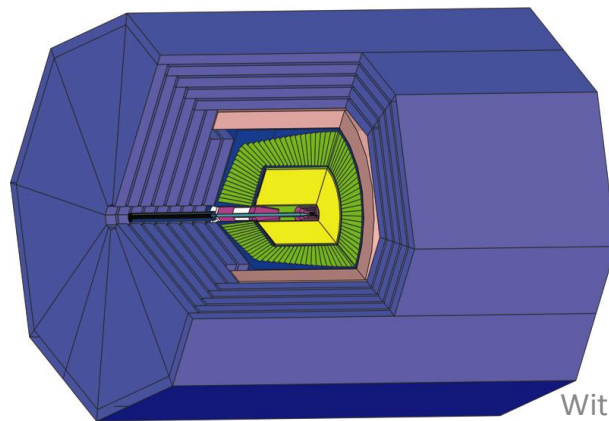
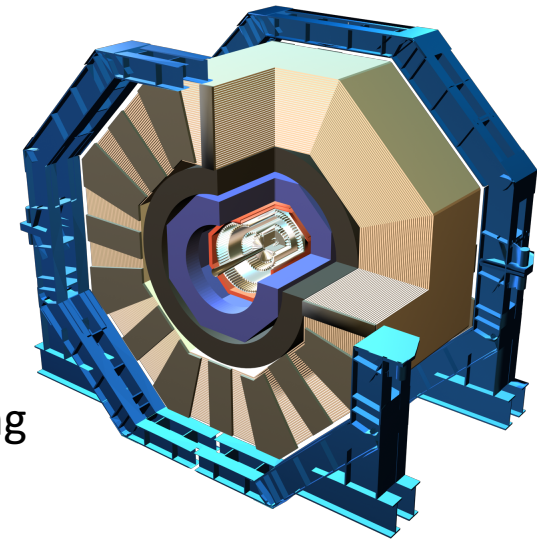
- GDML reader and writer for Geant4 is part of the Geant4 release
- GDML reader and writer for ROOT is part of the ROOT release
- the implementations are complete, but... some further development effort would always be welcome...

CMS detector: G4->GDML->ROOT

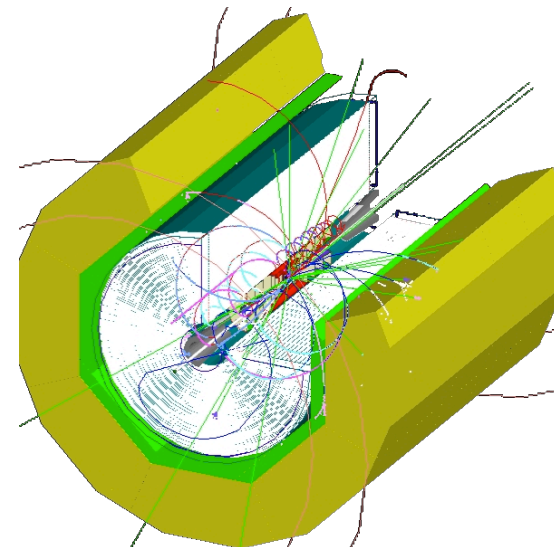


GDML as primary geometry source

- Linear Collider - Jeremy McCormick, SLAC
 - Linear Collider Detector Description (LCDD) extends GDML with Geant4-specific information (sensitive detectors, physics cuts, etc)
 - GDML/LCDD is generic and flexible
 - several different full detector design concepts, including SiD, GLD, and LDC, where simulated using the same application



GLD



LDC

Summary – what GDML can do

- GDML is an application independent geometry description language
 - basically any detector geometry can be described using it
- GDML does not provide application specific elements like for instance sensitive detectors
 - however, volumes can have 'auxiliary' field for storing any key-value pairs (ex: sensdet/tracker)
 - users' code can interpret those values
- GDML is XML so it can be extended and embedded in other schemas