DD4hep Status

HEP detector description supporting the full experiment life cycle

M.Frank, F.Gaede, N.Nikiforou, A.Sailer
• Motivation and Goals

=> Introduction / Reminders

• Concepts and Design

• Going to the 'real world'

• Summary
Motivation and Goal

- Develop a detector description
  - For the full experiment life cycle
    - detector concept development, optimization
    - detector construction and operation
    - 'Anticipate the unforeseen'
  - Consistent description, single source of information, which supports
    - simulation, reconstruction, analysis
  - Full description, including
    - Geometry, readout, alignment, calibration etc.
What is Detector Description?

- Description of a tree-like hierarchy of 'detector elements'
  - Subdetectors or parts of subdetectors

- Detector Element describes
  - Geometry
  - Environmental conditions
  - Properties required to process event data
  - Optionally: experiment, sub-detector or activity specific data

Diagram:
- ILD
- Ladder
- Module
- VXD
- TPC
- EndA
- Sector1
- Ecal
- Hcal
- Geometry
- Alignment
- Conditions
- Readout
- Visualization
- Subdet. data
- Reconstr. data
• Motivation and Goals

• Concepts and Design

=> Reminder

• Going to the 'real world'

• Summary
DD4Hep - The Big Picture

Compact description xml
Detector constructors c++ python
Geometry Display

Generic Detector Description Model
Based on ROOT TGeo c++

Conditions DB
Alignment / Calibration

Extensions where required
GDML Converter xml
TGeo => G4 converters
Reconstruction Extensions
Analysis Extensions

SLIC [SiD Simulation]
Geant4 Program
Reconstruction Program
Analysis Program
• Motivation and Goals

• Concepts and Design

• Status of Ongoing Work
  - Need to reshuffle work-list on demand of stake-holders:
    • Support for simulation
    • Support for reconstruction
    • Standard detector palette: DDDetectors
  - Alignment: DDAlign

• Future work – next steps

• Summary
Simulation: Generic Geant 4 Gateway

- Simulation = Geometry + Detector response + Physics
- Attempt for formalization of Geant4
  - Ideally: configuration without extra (C++) user code
- DDG4
  - Bootstrap Geant4 from DD4hep in memory geometry
  - Configure using XML, python or Cint (ROOT 5)
  - Configure Geant4 actions, physics-list, processes, particle constructors, sensitive actions, I/O etc using module palette
Simulation: DDG4

- **Concept**
  - Walk through the geometry and convert on the fly from ROOT to Geant4
  - Instantiate sensitive detectors from palette [similar to palette of detector constructors]
  - Instantiate physics list, -constructors and -processes
  - Start simulating

- **Palette of sensitive detectors**
  - Basic concepts present

- **Processing chain is implemented**
  - Validation in progress – time consuming process
DDG4 Design Illustration

Flexible configuration

Flexible definition of the physics list
- Define particles, processes, physics constructors or use/extend predefined physics lists

Flexible data input
- Programmable sequence. Input from particle gun, lcio, stdhep or HepMC (text) – easily extensible
- Modules to smear and boost primary vertices
- Modules to construct interaction overlays
- Further extensions may independently added

Provide user programmable sequences
- Either as explicit object type using ABC
- Or registering a member function as callback

Diagram:
- G4RunManager
  - G4VUserPhysicsList
  - G4VUserPrimaryGeneratorAction
  - G4UserRunAction
  - G4UserEventAction
  - G4UserTrackingAction
  - G4UserSteppingAction
  - G4UserStackingAction

- Geant4TrackingActionSequence
  - Geant4TrackingAction
  - Geant4TrackingAction
  - ...
Standard Detector Palette (DDDetectors)

- Mostly arose from the SiD model
  - Layer based detectors
  - Tracker barrel & endcap
  - Several calorimeter constructs
- Partially with measurement surfaces
  (see also talk by F. Gaede)
- Plugin mechanism to enhance detector elements
  - Neat mechanism to attach user defined optional data
    => Proof that 'anticipate the unforeseen' works
  - NOT intrusive to detector constructors
  - Flexible definition of the measurement surface
Extension Plugins for DDDetectors

- **Example**
  - Tracking surfaces
  - Apply plug-ins based on detector patterns

- **Same plug-in, different detectors, common Pattern:**
  'layered modules' build of one or several 'slices'
Alignment and Detector Conditions

- **Fundamental functionality to interpret event data from existing ('real') detectors**
  - Selling argument for existing (e.g. LHC) experiments
  - Necessity to handle imperfections
    - Geometry => (Mis)Alignment
    - Anomalous conditions
      - Pressures, temperatures => Gains, refractive indices
      - Contraction, expansions
Alignment and Detector Conditions (2)

- Functionality to handle the displacement of volumes is implemented
- Input / Output from xml is provided

But:
To fully use this functionality it must be combined with validity interval related data
Alignment and Detector Conditions (3)

- **Validity interval related information is not handled by the alignment support package**
  - Should not be handled for good reasons:
    - This is typically called the conditions database
  - Scope is much broader than only alignment
    - Environmental conditions etc.

- **Will extend the DD4hep toolkit to provide such functionality**
  - 'DDCond' – last extension of the DD4hep toolkit
  - Then apply alignment operations using data therein...
Alignment and Detector Conditions (3)

- Validity interval related information is not handled by the alignment support package
  - ...and should also not be handled
  - This is typically called the conditions database

- Will extend the DD4hep toolkit to provide such functionality
  - 'DDCond' – last extension of the DD4hep toolkit
  - Then apply alignment operations using data therein...
Documentation

- [http://aidasoft.web.cern.ch/DD4hep](http://aidasoft.web.cern.ch/DD4hep)
- [https://svnsrv.desy.de/basic/aidasoft/DD4hep/trunk](https://svnsrv.desy.de/basic/aidasoft/DD4hep/trunk)

In the svn doc area

- DD4hep manual
  - core API: 39 pages
- DDG4 manual
  - simulation: 40 pages
- DDAlign manual
  - alignment support 15 pages
- DDRec manual (F.Gaede)
- First issues, to be completed

Doxygen documentation
Toolkit Clients

Known client designs

- SiD: geometry conversion to DD4hep: M. Frank
  original design: J. McCormick
- ILD: F. Gaede et al., ported complete Mokka model ILD_o1_v05
- CLICdp: starting new design after CDR
- FCC-eh: P. Kostka et al.
- FCC-hh: starting
- LHCb: investigations by technical student

DD4hep core rather stable
DDG4 simulation under validation
Toolkit Clients (2)

- Design and implementation of simulation framework DDG4 finished
  - Validation ongoing: ILD, FCC-eh, CLICdp starting
- Possible new requirement (FCC)
  - Support for fast simulation
  - Parametrized simulation
  - Heterogenous simulation
    - Full, fast and parametrized simulation in parallel depending on subdetector or subdetector region
Summary and Outlook

- The DD4hep toolkit (+extensions) start to become accepted: Client validation has started
- Basic DD4hep API essentially stable
- Simulation kit DDG4 being validated
- Reconstruction extensions start to appear: DDRec
- Detector palette DDDetectors established
  - Hope of user contributions
- Alignment support implemented
  - Requires conditions support for full functionality
    => DDCond: extension to be developed
- Validate, verify, enhance and document
Questions and Answers
Backup slides
Implementation: Geometry

Subdetector Hierarchy (Tree)

- Detectors
  - DetectorElement
    - PlacedVolume
      - [TGeoNode]
        - [TGeoMatrix]
          - GDML content

Geometry

- Alignment
- Conditions
- Readout
- Visualization
- Segmentation

Subdetector status (conditions)

- Material
- Envelope
  - [TGeoShape]
    - [TGeoBox]
    - [TGeoCone]
    - [TGeoTube]
DDG4 Configuration Example (Incomplete)

<pre>
<sequences>
  <sequence name="Geant4EventActionSequence/EventAction">
    <properties Control="true"/>
    <action name="Geant4Output2ROOT/RootOutput">
      <properties Control="true" Output="simple.root"/>
    </action>
  </sequence>
  <sequence name="Geant4GeneratorActionSequence/GeneratorAction">
    <action name="Geant4ParticleGun/Gun">
      <properties .... />
    </action>
  </sequence>
  <sequence sd="SiVertexBarrel" type="Geant4SensDetActionSequence">
    <properties Control="true"/>
    <filter name="GeantinoRejector"/>
    <filter name="EnergyDepositMinimumCut"/>
    <action name="Geant4SimpleTrackerActionAction/SiVertexBarrelHandler">
      <properties Control="true"/>
    </action>
  </sequence>
  ....
</sequences>
</pre>

- **Geant4 event action setup**
- **Geant4 generator action setup**
- **Sensitive detector setup**

Instance type from palette  Instance name for reference
Complete Mokka model ILD_o1_v05 ported
(F.Gaede, L.Shaojun)

- VXD, FTD, SIT, TPC, SET, beam pipe
- Ecal, Hcal, Yoke, Beamcal, Lcal, LHcal
- services
- two generic SensitiveDetectors