DDG4

A Simulation Framework Based on DD4hep\(^{(a)}\) and Geant4

M.Frank\(^{(1)}\), F.Gaede\(^{(1,2)}\), N.Nikiforou\(^{(1)}\), M.Petric\(^{(1)}\), A.Sailer\(^{(1)}\)

\(^{(1)}\)CERN \(^{(2)}\)Desy

\(^{(a)}\) DD4hep: A General Purpose Detector Description Toolkit, CHEP2013, Amsterdam, NL
- Motivation, goals and the grand picture
  => Introduction / Reminders

- Concepts and Design

- Side remarks: Users and extensions

- Summary
Motivation and Goal

DD4hep: a detector description

– For the full experiment life cycle
  • concept development, optimization, construction and operation
  • 'Anticipate the unforeseen'
– Consistent description, single source of information
  • Support for simulation, reconstruction, analysis
– Full description, including
  • Geometry, readout, alignment, calibration etc.

DDG4\(^{(1)}\): Simulation support for DD4hep

– Require minimal user effort

\(^{(1)}\)AIDA-2020 project/Horizon-2020: WP3 Advanced Software
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
DD4Hep - The Big Picture

Extensions where required

Compact description
xml

Detector constructors
python
c++

Geometry Display

Generic Detector Description Model
Based on ROOT TGeo
c++

Conditions DB
Alignment / Calibration

Extensions

GDML Converter
xml

TGeo => G4 converters

Reconstruction Extensions

Analysis Extensions

Reconstruction Program

Analysis Program

Geant4 Program
• Motivation and Goals
• Concepts and Design
  – … life is a wish-list
  – The design and some illustration examples
• Side remarks: Users and extensions
• Summary
Simulation: The Wish

- **Simulation** = Geometry + Detector response + Physics
- **Minimalistic approach**
  - Ideally: configuration without extra (C++) user code
- **DDG4 Wish:**
  - Bootstrap Geant4 from DD4hep memory model
  - Configure the simulation application
  - Run...
Configuring Simulation Application: Concept

- Walk through the geometry and convert on the fly from ROOT to Geant4
- Hook into the Geant4 provided entry points
- Instantiate detector response (sensitive detectors) from plugin palette
- Instantiate physics list, -constructors and -processes from plugin palette
- Run...
Geant4 Provided Hooks
[and what we want to do inside]

Main issue: 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, Icio, 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
Example of an Action Sequence: Event Overlay with Features

- Combine simple and reusable modules
- Input module
  - Any data format
- Primary vertex smearing
- Primary vertex boost
- Common:
  - initialization, final merge
Another Example: MC Truth Handling

Registers itself as global MC truth handler

- Callback when hit is created
- Connect to stepping action by callback: Remember if track created secondaries

Geant4Sensitive

- Connect to begin/end event by callback:
- Store user track at end

Geant4ParticleHandler

- Automatically called as part of the event generation

Geant4GeneratorActionSequence

Geant4GeneratorAction

Geant4SteppingActionSequence

Geant4TrackingActionSequence

Geant4EventActionSequence
Design Considerations

- **External configuration**
  - Requires Plugin mechanism
  - Requires Property mechanism to configure plugin instances
  - Allows to naturally support configuration using XML, python or ROOT-AClick

- **Interactivity in Geant4**
  - For all plugins export properties and selected member functions

- **Formalization and external setup of physics**
  - Formalize construction of physics lists, physics constructors and particle type constructors in Geant4
Simple Basic Block for all Entities

- **Geant4Context**
  - Access Geant4 internals and Geometry
- **PropertyManager**
  - External configuration [similar to e.g. Gaudi]
- **Geant4UIMessenger**
  - Interactivity from the Geant4 prompt
Construct Specialized Components with Geant4Action as Base Class:

- Configurable physics actions (physics list)
- Configurable action sequences to handle sensitive detectors
- Configurable action sequences to handle
  - Events (e.g. event output) => Event action
  - Tracks => Tracking action
  - Simulation step => Stepping action
  - ...if the user application wants to interact with these
- And of course also provide the modules to populate these sequences!
Simulation: Plugin Palettes

- **Palette of sensitive detectors**
  - *Generic* sensitive detectors for trackers & calorimeters
  - Adopt *generic* designs for other detector 'classes'

- **Palette of IO handlers**
  - Input: Icio, StdHep(Icio), HepEvt (ascii), HepMC(ascii)
  - Output: Root, Icio

- **MC truth handling w/o record reduction**

- **Physics lists, Physics/particle constructors etc.**
  - Wrapped factory plugins directly taken from Geant4
  - Users extend physics list (e.g. QGSP)
• Motivation and Goals
• Concepts and Design
• Side remarks: Users and extensions
• Summary
Documentation

- http://aidasoft.web.cern.ch/DD4hep
- https://svnsrv.desy.de/basic/aidasoft/DD4hep/trunk
- Doxygen documentation
- In svn trunk/doc:
  - DD4hep manual, ~ 40 pages
  - DDG4 manual, ~40 pages
  - First issues, to be completed
  - Living documents
Known Toolkit Users

Users are mandatory for feedback to avoid developments in thin air (i.e. purely academic)

- 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, A. Salzburger et al.
- LHCb: investigations by technical student

<table>
<thead>
<tr>
<th>DD4hep</th>
<th>DDG4</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

(1) See A. Sailer's poster in Session B: “Integration of DD4hep in the Linear Collider Software Framework” Booth 6, Wednesday + Thursday  https://indico.cern.ch/event/304944/session/10/contribution/290
To Keep Us Off the Street ...

• **Possible improvements of DDG4**
  
  – **Support for fast and parametrized simulation**
    
    • Speed-up by avoiding full Geant4 machinery
  
  – **Heterogenous simulation**
    
    • Full, fast and parametrized simulation depending on sensitive region
  
  – Take action on demand, hope for external contributions

• DDG4 is the 2\textsuperscript{nd} episode of the DD saga
  
  – DD4hep, DDG4, DDRec, **DDAlign and DDCond** (to come)
Summary and Outlook

- The DD4hep toolkit (+extensions) is getting accepted: => Validation by users has started
- Simulation toolkit DDG4 being validated
  - Toolkit to simulate particle collisions in HEP detectors with minimal effort: simple, easy, flexible
  - DDG4 can host user plugins: extensible
- Basic DD4hep detector palette established
  - Hope for further user contributions
  - Hope to offer 'complete' sensitive detector palette
- Hope for contributions to complete the plugin suite to cover all simulation needs (I/O, MC truth, etc)
Questions and Answers
Backup slides
Implementation: Geometry

**Subdetector Hierarchy (Tree)**

- **Detectors**
  - **DetectorElement**
    - **PlacedVolume** [TGeoNode]
      - **Envelope** [TGeoShape]
      - **LogicalVolume**
        - **Material**
  - **Alignment**
  - **Conditions**
  - **Readout**
  - **Visualization**
  - **Segmentation**

**GDML content**

- **Geometry**
  - [TGeoBox]
  - [TGeoCone]
  - [TGeoTube]

**Subdetector status (conditions)**
Geant4 Interactivity

- Geant4 interactivity interfaced to every action object
  - Enabled on request
  - Actions have properties (similar to Gaudi)
    - Interrogate properties
    - Modify properties
Configure DDG4 Application with python

```python
kernel = DDG4.Kernel()
lcdd = kernel.lcdd()
kernel.loadGeometry("file:"+install_dir+"/DDDet/"
kernel.loadXML("file:"+example_dir+"/DDG4_field"
DDG4.importConstants(lcdd)
```

- Python configuration snippets
  - Loading geometry
  - Configuring actions
  - Steer Geant4 until it's prompt/batch
- C++ config ~ same
- Alternative: xml
  Load xml with lcdd

---

```
# First particle generator: pi+
gen = DDG4.GeneratorAction(kernel,
   "Geant4IsotropeGenerator/IsotropPi+")
gen.Particle = 'pi+'
gen.Energy = 100 * GeV
genMultiplicity = 2
gen.Mask = 1
kernel.generatorAction().adopt(gen)
# Install vertex smearing for this interaction
gen = DDG4.GeneratorAction(kernel,
   "Geant4InteractionVertexSmear/SmearPi-
-gen.Mask = 1
gen.Offset = (20*mm, 10*mm, 10*mm, 0*ns)
gen.Sigma = (4*mm, 1*mm, 1*mm, 0*ns)
kernel.generatorAction().adopt(gen)
```
Configure DDG4 Application from XML

```xml
<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="Geant4SimpleTrackerAction/SiVertexBarrelHandler">
      <properties Control="true"/>
    </action>
  </sequence>
  ....
</sequences>
```

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
- two generic SensitiveDetectors