EIC Software Infrastructure Review

Geometry Description and Detector Interface

Markus Diefenthaler

On behalf of the EPIC Collaboration

EPIC

EPIC Software

The design of the **modular simulation and reconstruction toolkit** for the development of the EPIC detector and the EPIC science program is based on the **EIC Software: Statement of Principles:**

2 We will have an unprecedented compute-detector integration:

- We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure.
- We aim for autonomous alignment and calibration.
- We aim for a rapid, near-real-time turnaround of the raw data to online and offline productions.

3 We will leverage heterogeneous computing:

- We will enable distributed workflows on the computing resources of the worldwide EIC community, leveraging not only HTC but also HPC systems.
- EIC software should be able to run on as many systems as possible, while supporting specific system characteristics, e.g., accelerators such as GPUs, where beneficial.
- We will have a modular software design with structures robust against changes in the computing environment so that changes in underlying code can be handled without an entire overhaul of the structure.

4 We will aim for user-centered design:

- We will enable scientists of all levels worldwide to actively participate in the science program of the EIC, keeping the barriers low for smaller teams.
- EIC software will run on the systems used by the community, easily.
- We aim for a modular development paradigm for algorithms and tools without the need for users to interface with the entire software environment.

8 We will provide a production-ready software stack throughout the development:

- We will not separate software development from software use and support.
- We are committed to providing a software stack for EIC science that continuously evolves and can be used to achieve all EIC milestones.
- We will deploy metrics to evaluate and improve the quality of our software.
- We aim to continuously evaluate, adapt/develop, validate, and integrate new software, workflow, and computing practices.

Integral parts of the modular simulation and reconstruction toolkit:

Geometry Description (this talk); **Data Model** and **Reconstruction Framework** (next talks).

Detector Simulations

Detector Simulations in Geant4

- De-facto standard for detector simulations for at least the next decade.
- Support for high concurrency heterogeneous architectures via multi-threading.
    - NP community supports next phase in concurrent Geant4: sub-event parallelism.

Simulation

Geometry Description

Role of Geometry in Detector Simulation

- Based on Geant4:
  - Hierarchical geometry with solids, logical volumes, and placements.
  - Concept of sensitive detectors and hits: Links geometry elements to a specific algorithm.
- Increasing level of complexity from detector concept to full detailed description of the running experiment.

Role of Geometry in Reconstruction

- Conceptual geometry description in terms of read-out elements, not all details necessary.
- Mapping between sensitive geometry elements and hits.

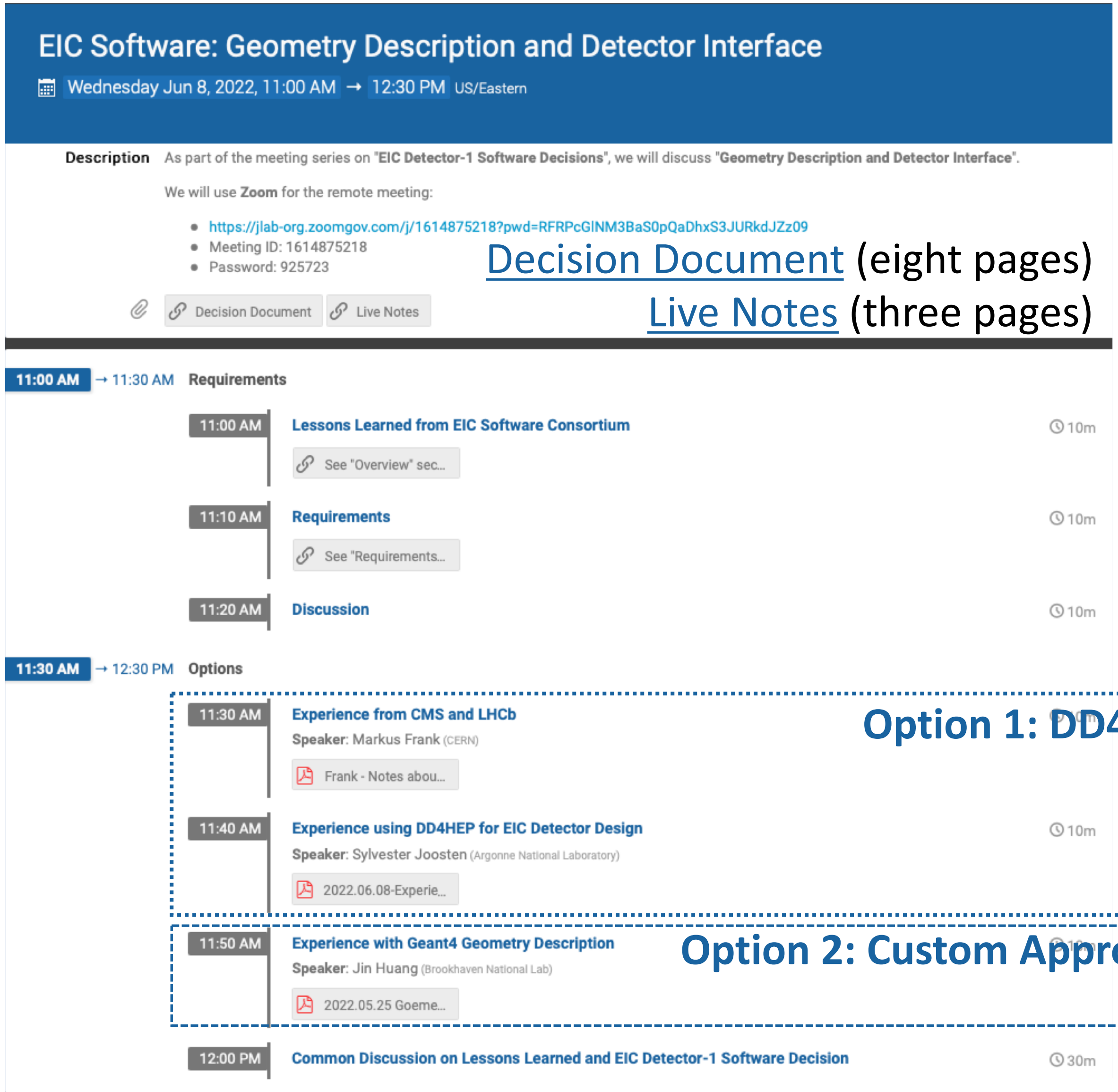
Requirements for Geometry Description and Detector Interface

1. **The geometry information should be the same in both simulation and reconstruction.**
2. Fast simulation systems should, as much as possible, **be able to use the common exchange format**.
3. The geometry system **should allow to include misalignment** and more general condition data.
4. Geometry description format should be **independent of a specific software technology**.
5. Geometry description **should be modular**. It should be possible to specify different geometry components in isolation with ideally zero dependency between different modules (detectors). Each detector component should have the ability to change the level of detail independent of other parts of the detector system.
6. Geometry description **should allow to specify logical information** (sensitivity, B-Fields) in addition to the solids, material and placements. In particular, sensitivity is recognized as a critical issue.
7. It **should be possible to make the geometry description persistent**. Different equivalent output formats should be supported (e.g. ROOT files, GDML files) and it should always be possible to translate one format into another in a simple manner.
8. Hits output files produced in a simulation job should be as much as possible self-describing, in particular **it should be possible to locate hits in space** without the need to run the simulation job. A *self-describing* format for the hits would be ideal, but in case this is not possible, the additional libraries to manipulate hits should not depend on the simulation stack used to produce the hits.
9. It should be possible to **change sensitivity attributes without changing other static aspects of the geometry**.
10. Geometry exchange format should **allow clients to use a subset of the features clearly stating which are the optional ones.** We should support existing interesting frameworks without discouraging other R&D activities. Since it is difficult to support all use-cases, the minimal set of mandatory elements to support should be clearly specified and what to do with non-supported ones should be stated (e.g. ignore visualization attributes if not needed).
11. **Support for export and import from CAD** should be included. Simplified CAD files will be provided via the Detector Menagerie.
12. Geometry information **should have support for versioning,** also including the Detector Menagerie.

Discussion

Prior Discussions

- Jason Webb (BNL), **Geometry Description and Geometry Frameworks in HEP/NP Experiments**, Presentation at EIC Software Consortium Meeting, May 1–2 , 2017, Jefferson Lab.
- Andrea Dotti (SLAC), **Geometry and Detector Interface: Implementation**, Presentation at EIC Software Meeting, July 6–7, 2017, SLAC National Accelerator Laboratory.
- Andrea Dotti (SLAC), **Geometry Interface**, Presentation at EIC Software Consortium Meeting, Oct. 16–17, 2017, Argonne National Laboratory.
- Markus Frank (CERN), **DD4hep for EIC**, Presentation at EIC Software Meeting, Jul. 10, 2019, Brookhaven National Laboratory.



Decision Document (eight pages)
Live Notes (three pages)

Option 1: DD4hep

Option 2: Custom Approach

Custom Approach vs. DD4hep

Custom Approach

Requirements

- Does not meet requirement 4 (independent of a specific software technology) and maybe requirement 5 (modular geometry description).

Concerns

- A custom approach will allow the use of all features of Geant4 but there might be a substantial amount of work needed for its implementation,
- including having to maintain a larger code base than in case of DD4hep.

DD4hep

← Has been chosen.

Requirements

- Meets all requirements.

Concerns

- DD4hep uses ROOT TGeo for the geometry description. This limits Geant4 simulations to the features being supported in TGeo.
- ROOT project regarding the support of ROOT TGeo:
  - *"The ROOT project is not going to support features such as parameterized volumes or parallel worlds. Replicas are supported, they are called divisions in ROOT. The ROOT project points out that missing features could be added on top of DD4hep and not TGeo."*
- There has been a concern raised about DD4hep support beyond the run time of CMS and LHCb at HL-LHC.
  - Support during HL-LHC era guaranteed.
  - In general cannot plan for software for more than one decade in advance and have to - as we are - plan for changes of our software stack.

DD4hep – Geometry Description and Detector Interface for EPIC

- A full implementation of the EPIC Detector in DD4hep is available:
  - Geometry description in DD4hep successfully used in EPIC Detector simulations.
  - Detector interface in DD4hep successfully implemented in EPIC Reconstructions.

- Training of the EPIC Collaboration in how to describe detector subsystems in DD4hep and has started. Members of the ATHENA proto-collaboration are already familiar with DD4hep.

- EIC Software connections to Geant4, ROOT, DD4hep, LHCb , and Key4Hep developers will allow to develop DD4hep for the needs of EPIC in specific and the EIC in general.

Example for feature EPIC requested:

> Allow to set MeanExcEnergy, MeanEnergyPerIonPair and BirksConstant in G4Materials ionisation parameters. Specify values in the compact description of the materials:

```
<material name="Ice">
  <D type="density" value="1.0" unit="g/cm3"/>
  <composite n="2" ref="H"/>
  <composite n="1" ref="O"/>
  <constant name="BirksConstant"         value="123.456*mm/MeV"/>
  <constant name="MeanExcitationEnergy" value="79.7*eV"/>
  <constant name="MeanEnergyPerIonPair" value="50*eV"/>
</material>
```