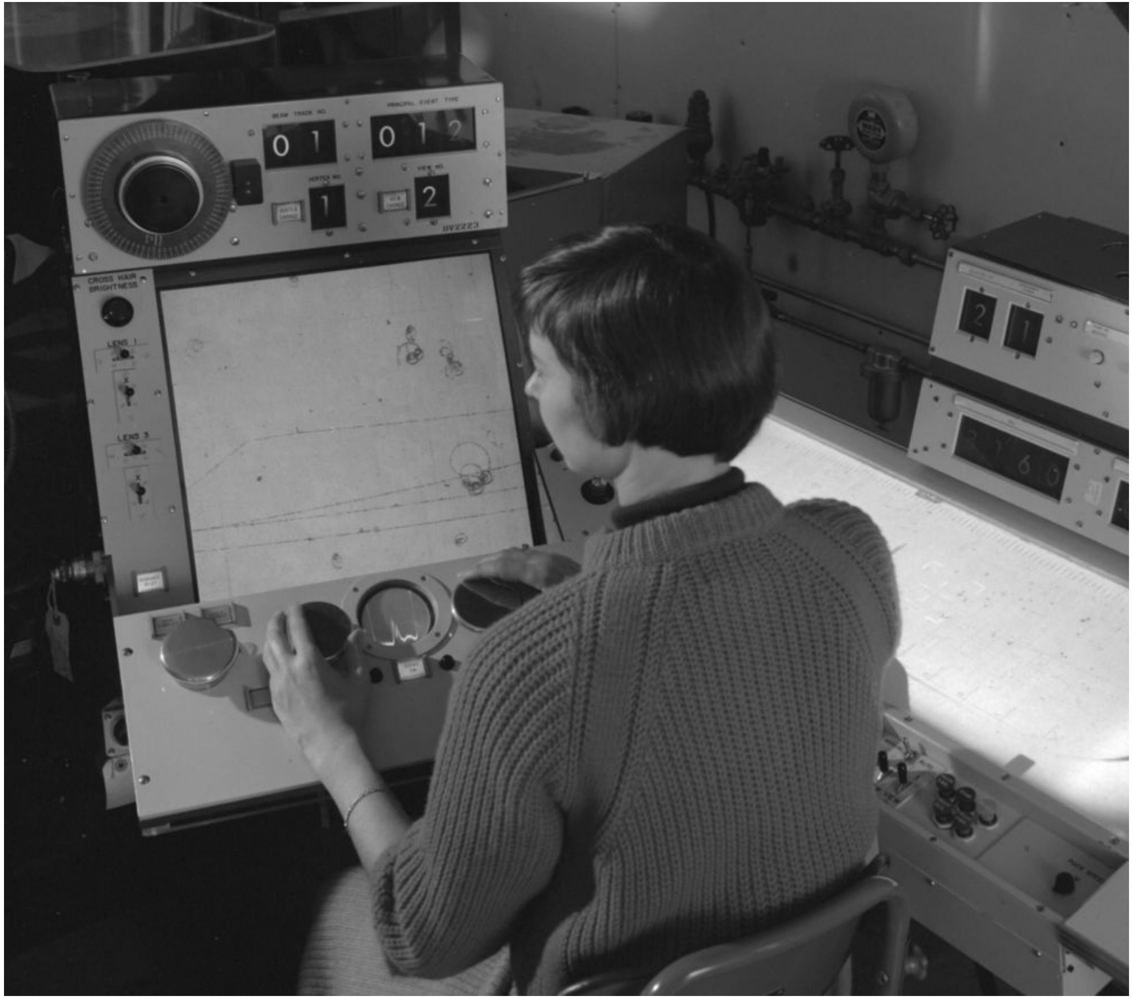


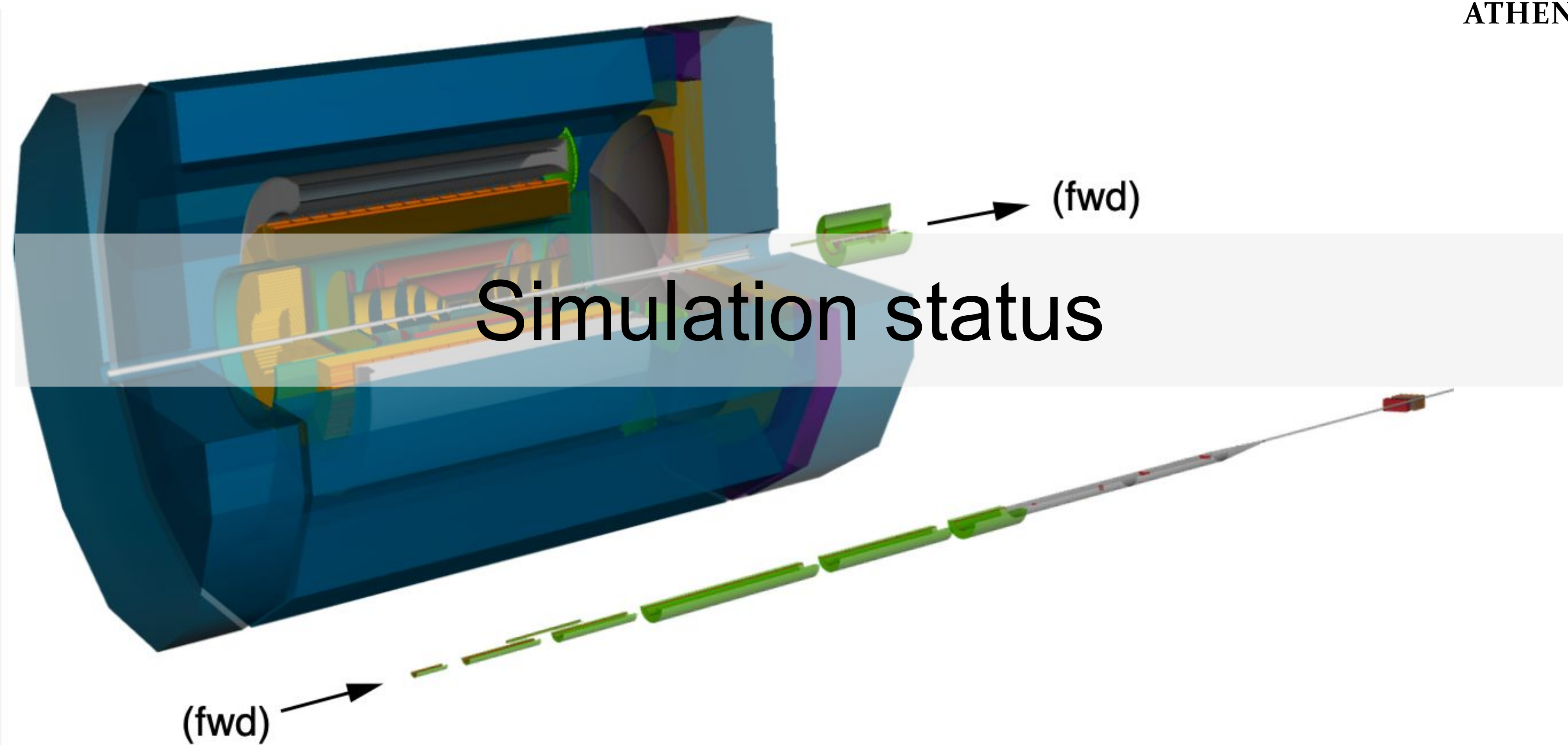
https://indico.jlab.org/event/505/contributions/9207/attachments/7478/10398/pivarski-jlab-history-programming-languages.pdf

ATHENA

Bi-Weekly Meeting

Thursday 2022-02-24

The Software and Computing WG Conveners:
Andrea Bressan (University of Trieste and INFN) ,
Dmitry Romanov (Jefferson lab) ,
Sylvester Joosten (Argonne National Laboratory) ,
Whitney Armstrong (Argonne National Laboratory) ,
Wouter Deconinck (The University of Manitoba)



Simulation status

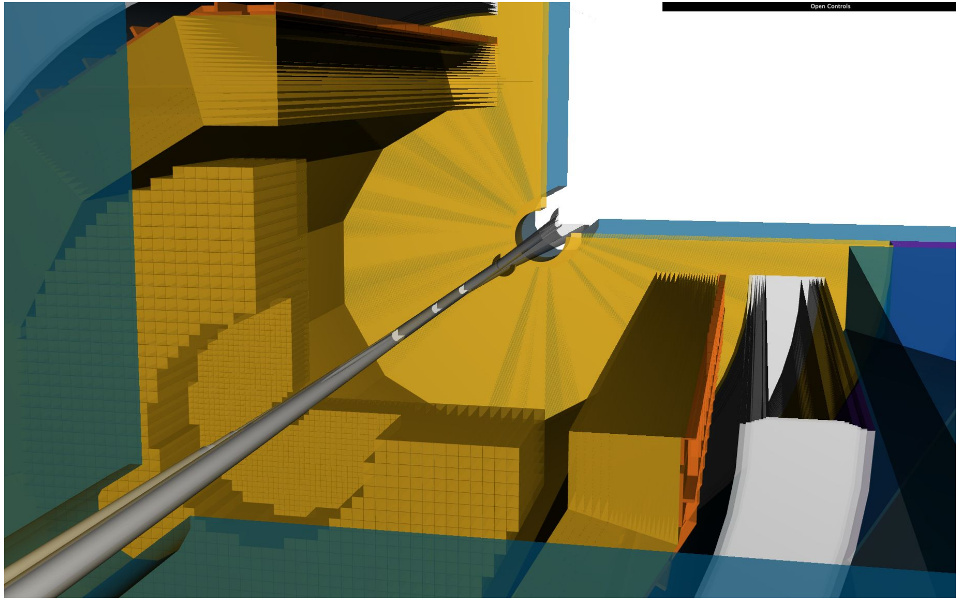
Simulation status: central calorimetry

✅🚧**Electromagnetic calorimetry**

- ✅ nECal ❌ nECal support
- ✅🚧 bECal minor tweaks for realism
- 🚧pECal realism and performance tweaks

🚧 **Hadronic calorimetry**

- 🚧 holistic pECal + pHCal validation
- 🚧 bHCal and nHCal validation and improved realism



Simulation status: central PID

✅ **RICH detectors**

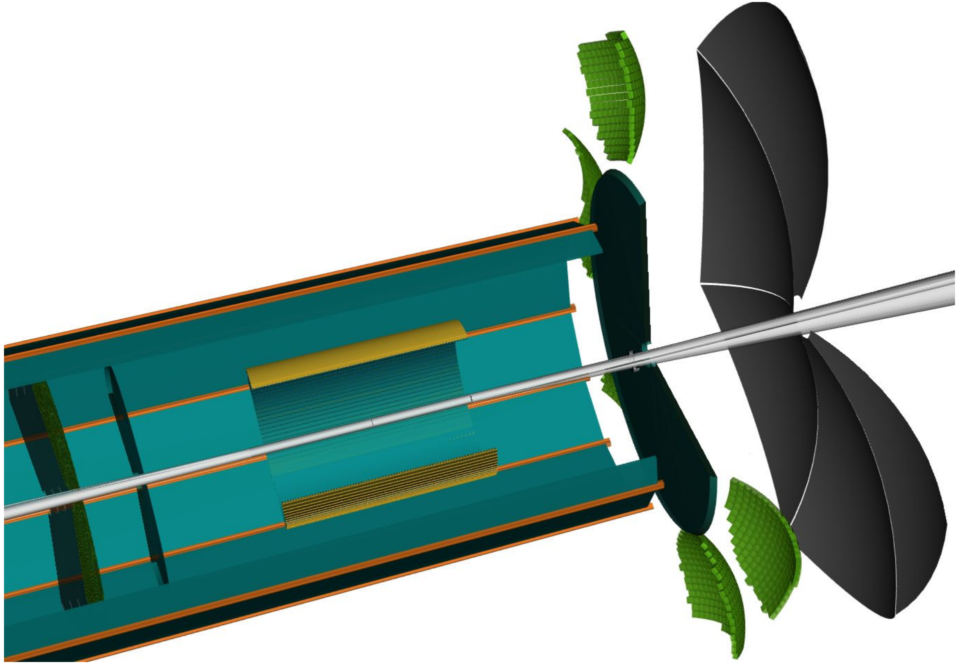
- ✅ dRICH and pfRICH fully functional

🚧 **DIRC**

- ✅ Still using material stand-in
- 🚧 Finalization of realistic DIRC plugin (fix overlaps and final validation with DIRC experts)

✅ ❌ **bTOF**

- ✅ Realistic detector
- ❌ bTOF support and services
- ❌ Holistic reoptimization of bTOF + tracking system



Simulation status: central tracking

✅🚧 **Vertex tracker**

- ✅ Approximate implementation due to (older) ACTS limitations
- 🚧 Proper cylindrical implementation
- ❌ Additional details: foam rings, connection to tracker services

✅ **Silicon trackers**

- ✅ Realistic implementation including services
- ❌ Additional details: connect detectors to services/support (service rings), mechanical support connected to rest of detector

✅🚧 **MPGD rings**

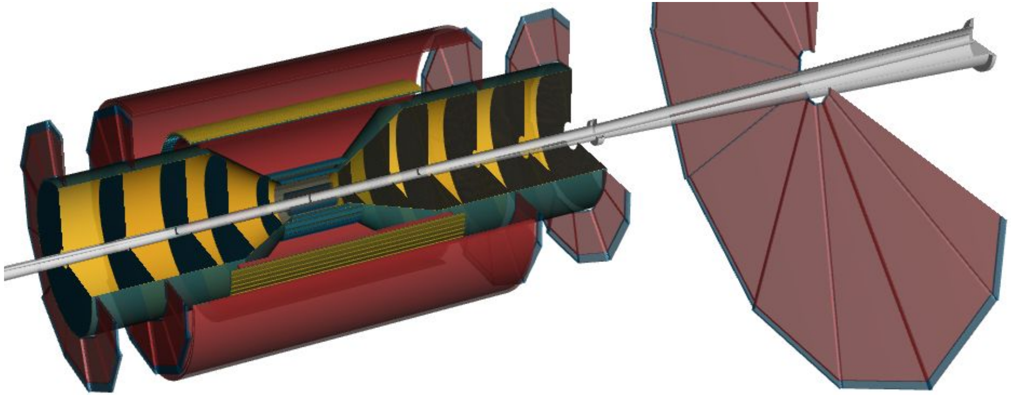
- ✅ Trapezoid implementation
- 🚧 Minor geometry optimization (rounded inner radius, better support)
- ❌ *Additional support details?*

❌ **bTOF (See PID)**

- ❌ Integrate bTOF into tracker services/support
- ❌ Holistic reoptimization of bTOF + tracking system

❌ **bECAL imaging layer 1**

- ❌ Integrate first imaging layer into ACTS tracking geometry



Simulation status: central beamline and far-forward/backward

✅ **Beamline**

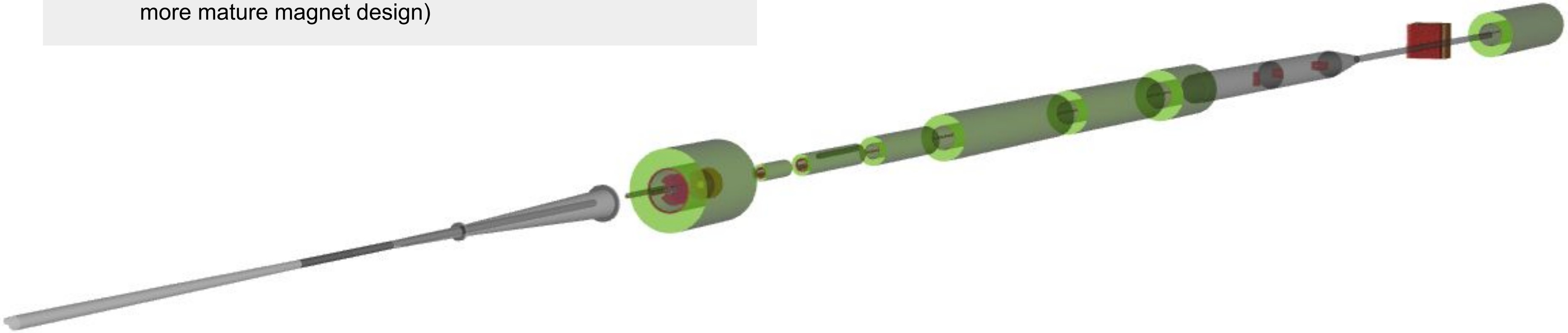
- ✅ Direct translation from project CAD (Be + Au)

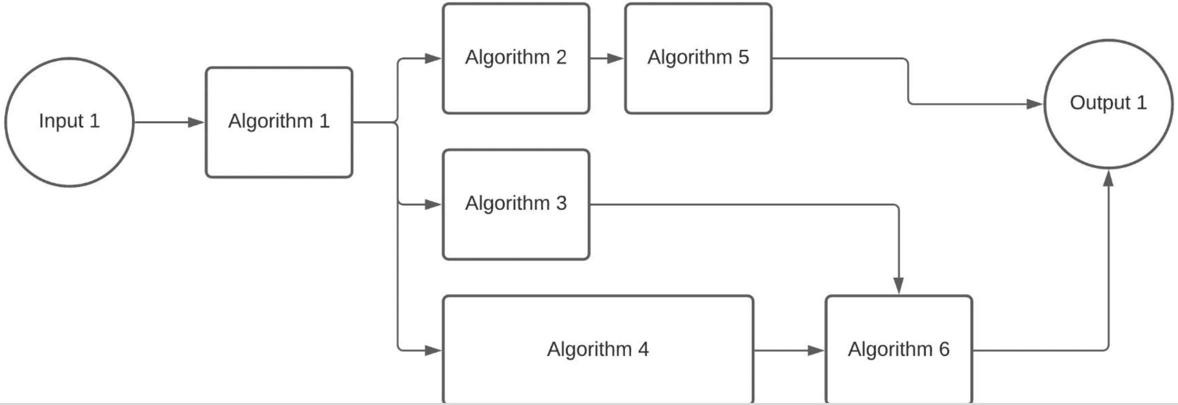
✅ **Far-forward instrumentation**

- ✅ Magnet system accurately reproduces machine lattice
- ✅ Beampipe and drift section between B1apf and B2apf consistent with preliminary design concept.
- ✅ RPs and OMDs with service/support (including RF shielding)
- ✅ Realistic ZDC ECal & HCal implementation
- ❌ Additional details: motors and detector frames (require more mature magnet design)

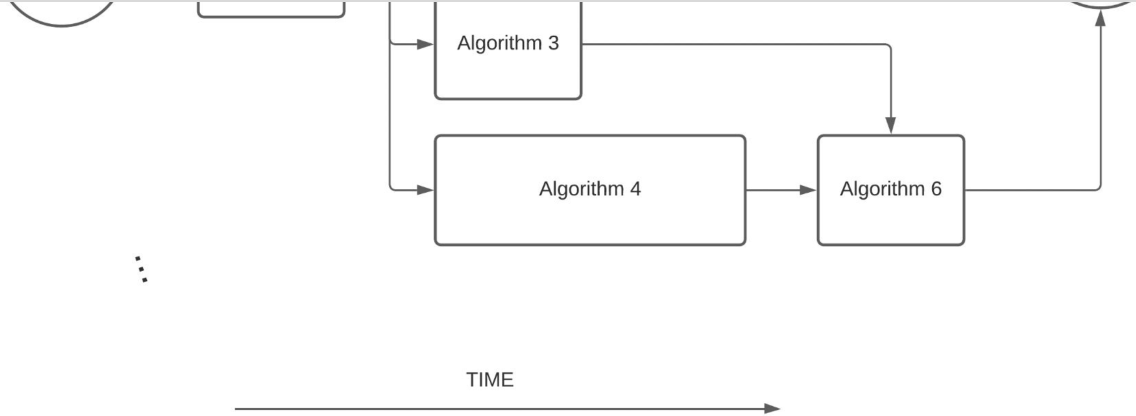
❌ **Far-backward instrumentation**

- ✅ Stand-alone DD4hep implementation in Far-Backward working group (**not** under umbrella of SWG)
- ❌ Urgent TODO: needs to be merged into the rest of the ATHENA geometry for completeness!





Reconstruction status



Track Reconstruction

Regular meetings with Acts developers

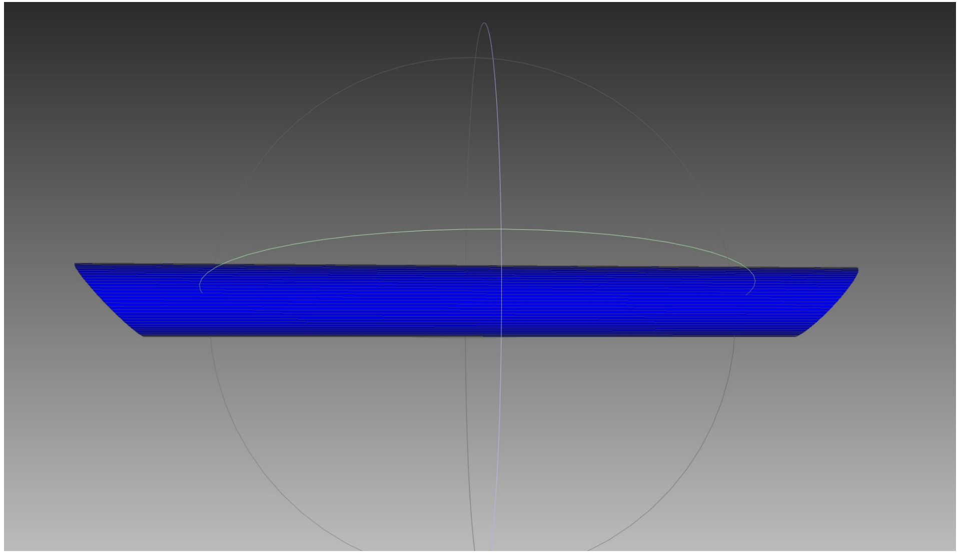
- Monday 10:30 Eastern (~weekly)
- Attended by team of 5 core Acts developers

Recent discussions:

- Vertex finder (Wenqing Fan, Yue Shi Lai)
- B0 tracker support (Sakib Rahman)
- Data model representation for Acts objects (Sylvester Joosten)

Main goals:

- Sandbox development of vertex finding must be integrated into main repository



JUGGLER GAUDI

https://eicweb.phy.anl.gov/EIC/eicd/-/blob/master/eic_data.yaml
https://eicweb.phy.anl.gov/EIC/juggler

Reconstruction status

LEGEND:
- ✅ : done
- 🚧 : ongoing
- ❌ : task not allocated to anyone or not fully defined

ATHENA

Status (incomplete)

- ✅🚧 Basic calorimeter reconstruction
  - 🚧 Validate/improve ECal calorimetry reconstruction
  - 🚧 HCal reconstruction and particle flow
  - 🚧 Integrate AI algorithms in main reconstruction
- ✅ Tracking with ACTS working well
  - 🚧 Provisions for B0 geometry
  - 🚧 Arbitrary track projections
  - 🚧 Realistic track seeding
  - 🚧 Realistic vertexing
- ✅ Basic neutral particle reconstruction
- 🚧 Realistic PID
  - ✅ Truth PID for all particles
  - 🚧 Integrate IRT for dRICH/pfRICH in main framework (major effort)
  - ❌ DIRC reconstruction
  - ❌ Realistic e/π separation
- 🚧 kinematic reconstruction
  - 🚧 Tune algorithms for inclusive kinematics
  - ❌ Integrate SIDIS reconstruction, particle flow, …
- 🚧 FF reconstruction
  - ✅ Fast FF reconstruction
  - ✅ RP/OMD proton reconstruction
  - ✅ ZDC reconstruction (NEW!)

- Will discuss reconstruction work prioritization and task distribution during next SWG meeting
- (SWG will resume bi-weekly meetings Thursday March 3 at 12:00pm EST)

ATHENA

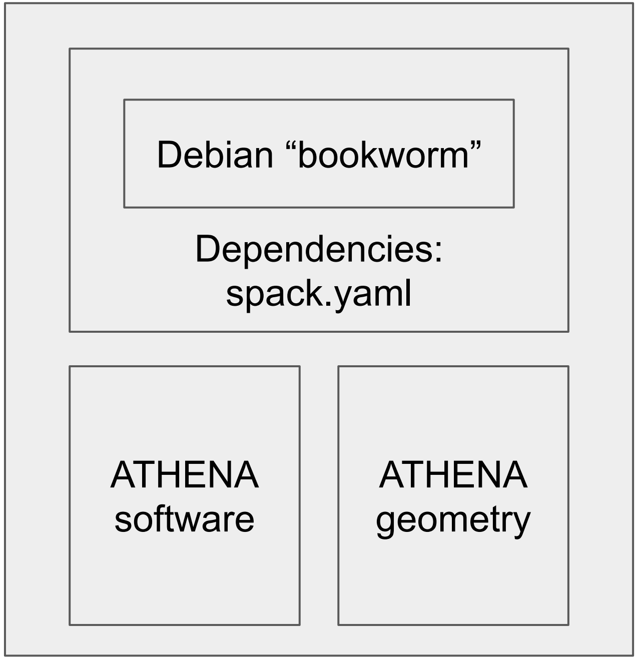
Infrastructure updates

(Internal notes)

- Software updates and what it buys us
  - Acts 15 -> 17: eta-binned chi2 limits, off-axis cylinders for B0, improved ITK seed finder
  - DD4hep 1.18 -> 1.20: diquark bugfixes, optical materials, geant4 11+ support
  - Podio 0.14 -> 0.15: mutable types
- Transition to EDM4hep data model
  - Simulations up to start of reconstruction: mcparticles -> MCParticles
  - Reconstruction starting with digitization
- Far forward updates:
  - ZDC clustering enabled now
- GPU acceleration:
  - CUDA on eicweb
  - Concurrent gaudi tests
- ML integration in reconstruction:
  - Tensorflow Lite
- Vertex seeding

Dependency Upgrades

- base system:
  - debian version December 2021
  - new gcc@11 compiler (from gcc@10)
- acts@15 (from @17)
  - eta-binned chi2 limits, off-axis cylinders for B0, improved ITK seed finder
- DD4hep@1.20 (from @1.18)
  - diquark bug fixes, optical materials, geant4 11+ support
- podio@0.15 (from @0.14)
  - separation of const and mutable codes
  - used by current EDM4hep data model
- Development of a multiple eic-shell containers, e.g. `eic-shell --ml`



Debian “bookworm”

Dependencies:
spack.yaml

ATHENA
software

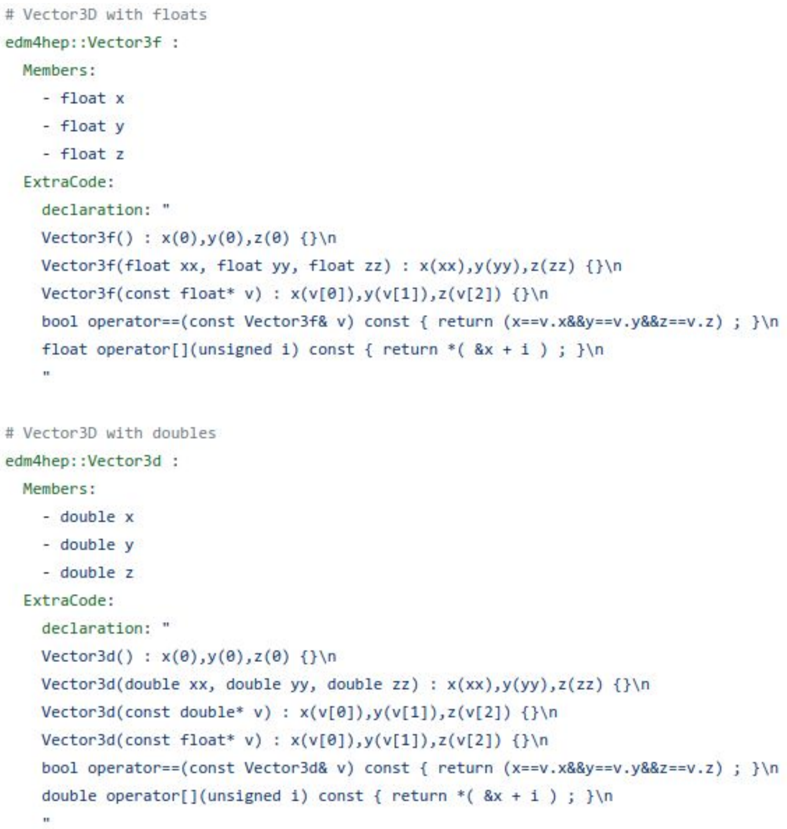
ATHENA
geometry

Transition to EDM4hep

Aspects involved in this transition:

- Different output from geant4 simulations:
  - dd4pod.yml to edm4hep.yml
  - mcparticles to MCParticles (more than just a change to uppercase, though)
  - *Hits collections are different, but primarily needs changed support in digitization algorithms
- Different output from reconstruction:
  - eicd.yml to edm4hep.yml
  - Most algorithms need modifications
  - Most written data structures will have minor to major changes

Status: writing .edm4hep.root files, algorithm conversion nearing completion, completion early next week



```
# Vector3D with floats
edm4hep::Vector3f :
  Members:
    - float x
    - float y
    - float z
  ExtraCode:
    declaration: "
    Vector3f() : x(0),y(0),z(0) {}\n
    Vector3f(float xx, float yy, float zz) : x(xx),y(yy),z(zz) {}\n
    Vector3f(const float* v) : x(v[0]),y(v[1]),z(v[2]) {}\n
    bool operator==(const Vector3f& v) const { return (x==v.x&&y==v.y&&z==v.z) ; }\n
    float operator[](unsigned i) const { return *( &x + i ) ; }\n
    "

# Vector3D with doubles
edm4hep::Vector3d :
  Members:
    - double x
    - double y
    - double z
  ExtraCode:
    declaration: "
    Vector3d() : x(0),y(0),z(0) {}\n
    Vector3d(double xx, double yy, double zz) : x(xx),y(yy),z(zz) {}\n
    Vector3d(const double* v) : x(v[0]),y(v[1]),z(v[2]) {}\n
    Vector3d(const float* v) : x(v[0]),y(v[1]),z(v[2]) {}\n
    bool operator==(const Vector3d& v) const { return (x==v.x&&y==v.y&&z==v.z) ; }\n
    double operator[](unsigned i) const { return *( &x + i ) ; }\n
    "
```

Heterogeneous Computing (GPUs)

Hardware: eicweb GPU acceleration

- dual Nvidia T4 (2 x 2560 cores)
- eicweb now has GPUs accessible to jobs (tag: gpu)
- use for tensorflow training (x50 faster training for imaging_shower_ML benchmarks

Check with `nvidia-smi` whether your jobs is GPU-enabled

Software: CUDA containerization

- proof of concept of running jug_xl with CUDA acceleration inside the container
  - large develop environment
  - small runtime environment
- software that currently supports CUDA ‘out of the box’
  - acts, veccore, vecgeom, vecmem
- benchmarking of run times
  - unlikely to speed up enormously with current scop
- operational issues to explore
  - OSG running on disparate hardware
- future directions where it does matter
  - calorimetry clustering
  - AI/ML training and inference

Heterogeneous Computing (Avalanche Scheduler)

Our main goal of using Gaudi:

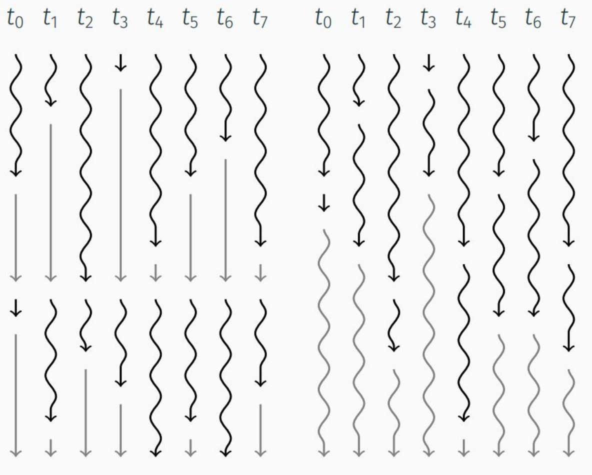
- claim all cores on a node
- let Gaudi schedule tasks to occupy all cores simultaneously
- optimization through small algorithms that allow flexibility in their scheduling
- requires thread-safe algorithms

Our actual use of Gaudi until now:

- single threaded running, linear operations

Near term goals:

- verify thread-safety of algorithms
- start running with avalanche scheduler by default
- stricter rejection of algorithms that are thread-unsafe


t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7

Artificial Intelligence and Machine Learning

Platform of choice (training): tensorflow

- well aligned with POD data model
- supported by data science community

Status of implementation:

- Currently `pip install tensorflow` inside CI
- Working on development of jug_ml container that has tensorflow preinstalled

Platform of choice (inference): tensorflow-lite

- only does inference, no training
- intended for microcontrollers, embedded systems, predictable running times
- GPU acceleration using OpenCL
- requires tflite model conversion, not full tensorflow model, some limitations on available operations (e.g. no SeLU)

Status of implementation:

- proof of concept gaudi algorithm for imaging calorimeter PID implemented

Software support and training

ATHENA

Office Hours and Spack Helpdesk

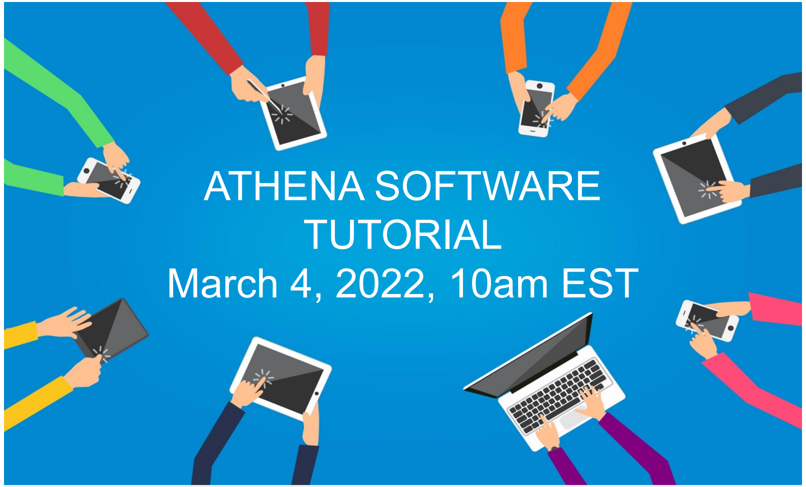
- **Regular office hours** (3x/week), 2pm Eastern on MWF
- Please ask questions in spack **#software-helpdesk** (preferably not the #general channel since that is sometimes perceived as noise by others)

Trainings and Tutorials

Planning new software tutorials after the conversion to EDM4hep is complete and functional.

Need input from all WGs **by this weekend**

- Contact person for each WG who can help with tutorial facilitation
- List of specific training use cases to aid in prioritization between the following:
  - DD4hep geometry coding in xml
  - DD4hep plugins in c++
  - Running small-scale simulations
  - Adding Gaudi algorithms
  - Tuning Acts track strategies
  - …
  - Many other topics are possible



ATHENA SOFTWARE TUTORIAL
March 4, 2022, 10am EST

Tutorial timeline

- **Friday March 4**
- **10am EST**, 9am CST, 7am PST, 4pm CET

Tentative agenda

- Running DD4hep with current geometry
  - No additional geometry coding
- Running Gaudi with existing algorithms

Useful dates and meetings

- (Almost) weekly ACTS meeting Monday’s at 10:30am EST
- NA-friendly key4HEP meeting on Tuesday March 1 at 10:00am EST
- **SWG will resume bi-weekly meetings Thursday March 3 at 12:00pm EST**
- SWG Tutorial session March 4 at 10:00 am EST
- *… and SWG office hours every Monday-Wednesday-Friday at 2:00pm EST*