Discussion points on Tools for Layout and Simulation studies

Valentina Cairo

September 23rd 2020

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Disclaimer

- **This presentation is far from being comprehensive**!
  - Our main goal is to find a framework that best suits our needs.
  - Ideally, we want to be able to:
    - create certain detector geometries which include 4D tracking sensors
      - 2 things to keep in mind: those sensors may have different passive material than 3D sensors and they imply being able to run 4D tracking
    - simulate and reconstruct objects based on these layouts
    - ultimately, we would like to see the impact of such varied layouts on some physics benchmarks

- These questions depend on a variety of possibilities., i.e. **which detector layout do we want to use as a benchmark**? One of the detectors at future colliders or shall we start from the HL-LHC scenario?
  - Most of what we learn for HL-LHC can be ported over future detectors.

Snowmass Efforts

- For what concerns future detector scenarios, **a set of tutorials is being put in place by the Snowmass MC task force** to help new users ramp up (most of these tutorial still have to take place, so we will learn more in the coming weeks)

https://snowmass21.org/montecarlo/energy

MC/Simulation Framework Tutorial Series

We are holding a series of tutorials for a variety of proposed future colliders. The goal of these tutorials is to prepare outside collaborators to conduct physics studies for these machines.

Machine	Date	Link
ILC	Aug 28	https://indico.fnal.gov/event/45031/
CEPC	Sept 8	https://indico.fnal.gov/event/45183/
FCC-ee/hh	Sept 22-23	https://indico.cern.ch/event/945608/
LHeC/FCC-eh	Sep 25	https://indico.fnal.gov/event/45185/
Whizard for e+e-	Sept 28	https://indico.fnal.gov/event/45413/
FCC-ee/hh	Sept 29	https://indico.cern.ch/event/949950/
Muon Collider	Sep 30	https://indico.fnal.gov/event/45187/

Snowmass Efforts

SLAC

- For what concerns future detector scenarios, a set of tutorials is being put in place by the Snowmass MC task force to help new users ramp up (most of these tutorial still have to take place, so we will learn more in the coming weeks)

Most of these frameworks support **fast simulation**

Snowmass EF MC Task Force - Framework/MC Info

File Modifica Visualizza Inserisci Formato Dati Strumenti Componenti aggiuntivi Guida

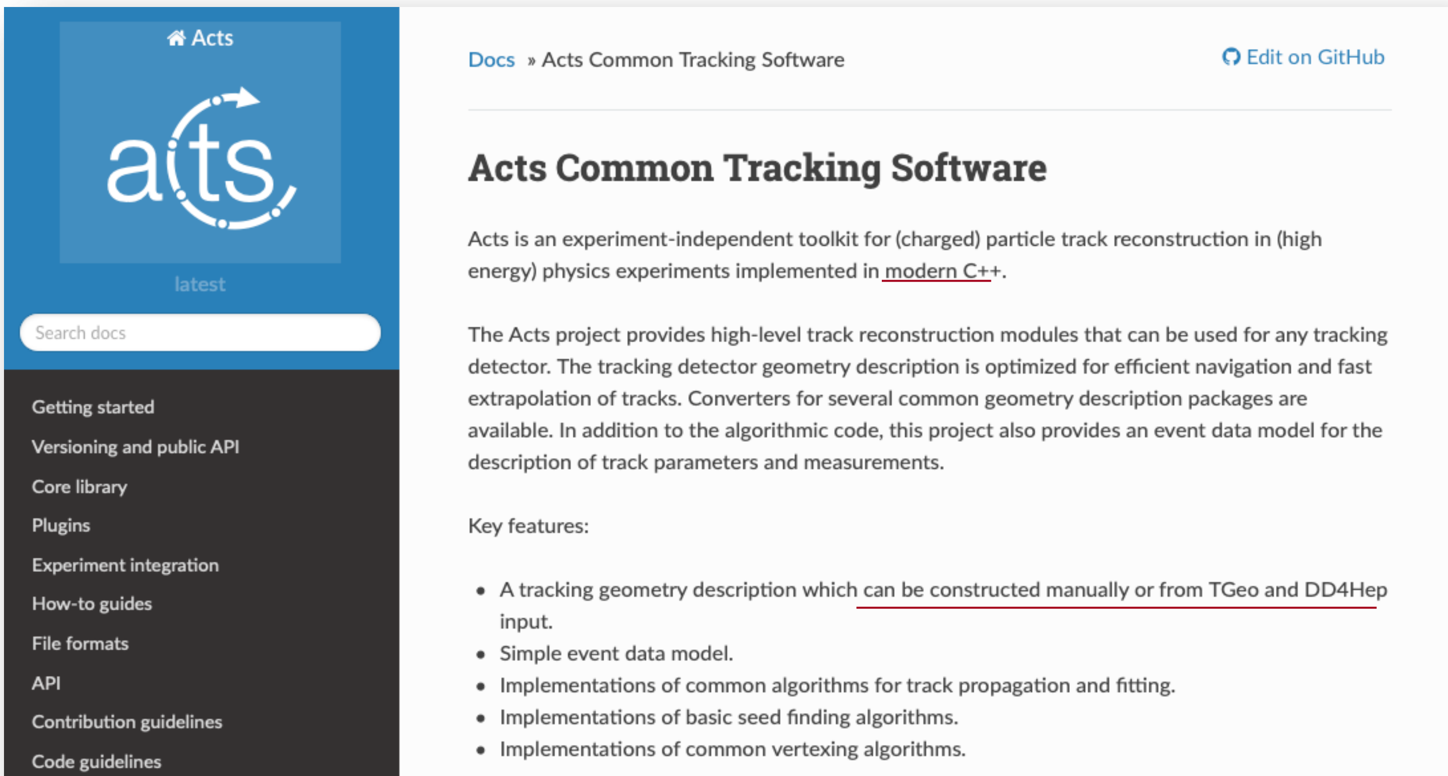
100% Solo visualizzazione

fx Which proposed collider are you responding on behalf of?

	A	C	D	E
1	Which proposed collider are you responding on behalf of?	Have you developed a simulation framework based on GEANT?	Is this framework publicly-available? If so, where?	Have you developed a DELPHES detector card (or other form of fast and/or parameterized simulation)?
2	ILC	Yes.	https://github.com/iLCSoft	Work is currently ongoing.
3	CLIC	Yes, DD4hep+DDG4+ddsim	e.g: /cvmfs/clicdp.cern.ch/iLCSoft/builds/2020-02-07/x86_64-slc6-gcc62-opt/init_ilcsoft.sh github.com/aidasoft/dd4hep	yes, CLICdet cards Stage1/2/3, documentation in https://twiki.cern.ch/twiki/bin/view/CLIC/DelphesMadgraphForBSMReport and https://arxiv.org/abs/1909.12728 (contains concrete instructions)
4	CEPC	Yes, CEPC has the Mokka-C, a Geant-4 Full simulation toolkit.	The CEPCsoft are capsuled into a package on cvmfs, the installation can be found at http://cepcsoft.ihep.ac.cn/guides/scratch/docs/cvmfs/	Yes, we have developed a Delphes card, which is also validated on the Full simulation tools. In addition, many small fast simulation tools are developed.
5	FCC-ee	Yes	Yes, the code is on github: https://github.com/HEP-FCC/ and it is installed centrally on cvmfs (same as FCC-ee). Also for the CDR, full simulation samples where produced with CMSSW.	Yes
6	FCC-hh	yes	Yes, the code is on github: https://github.com/HEP-FCC/ and it is installed centrally on cvmfs	yes
7	LHeC/FCC-eh	Yes: DD4HEP interface to GEANT	Yes: http://dd4hep.web.cern.ch/dd4hep/	Yes
8	Muon collider	Yes	yes, github	No DELPHES, no parametric simulation

What about hit time propagation?

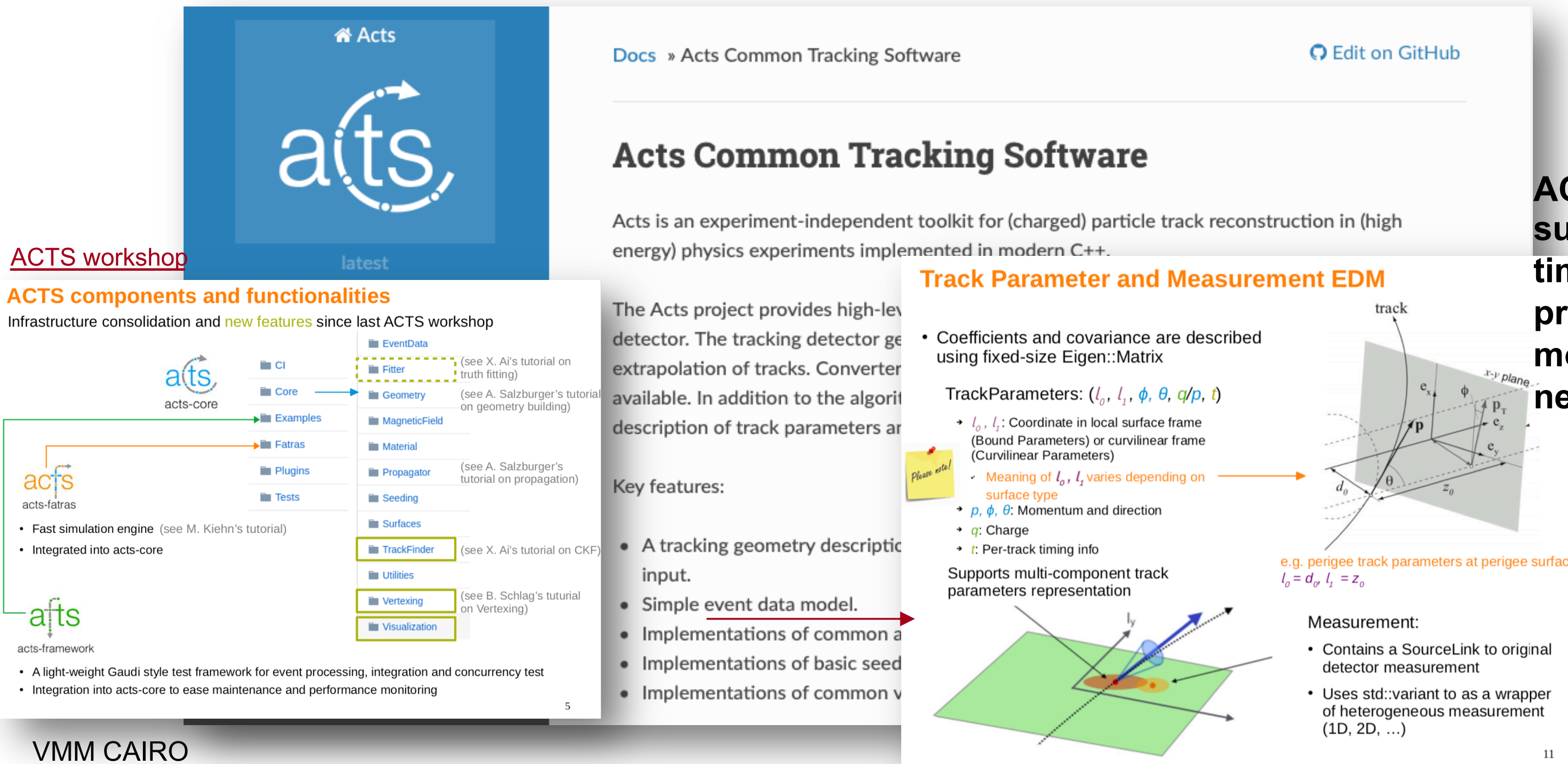
- What remains unclear is how those collaborations are dealing with time integration during track reconstruction
  - Basically impossible to know the details for all of them, so as a starting point I investigated what for me was the easiest thing to look at: **ACTS**



What about hit time propagation?

- What remains unclear is how those collaborations are dealing with time integration during track reconstruction
  - Basically impossible to know the details for all of them, so as a starting point I investigated what for me was the easiest thing to look at: ACTS

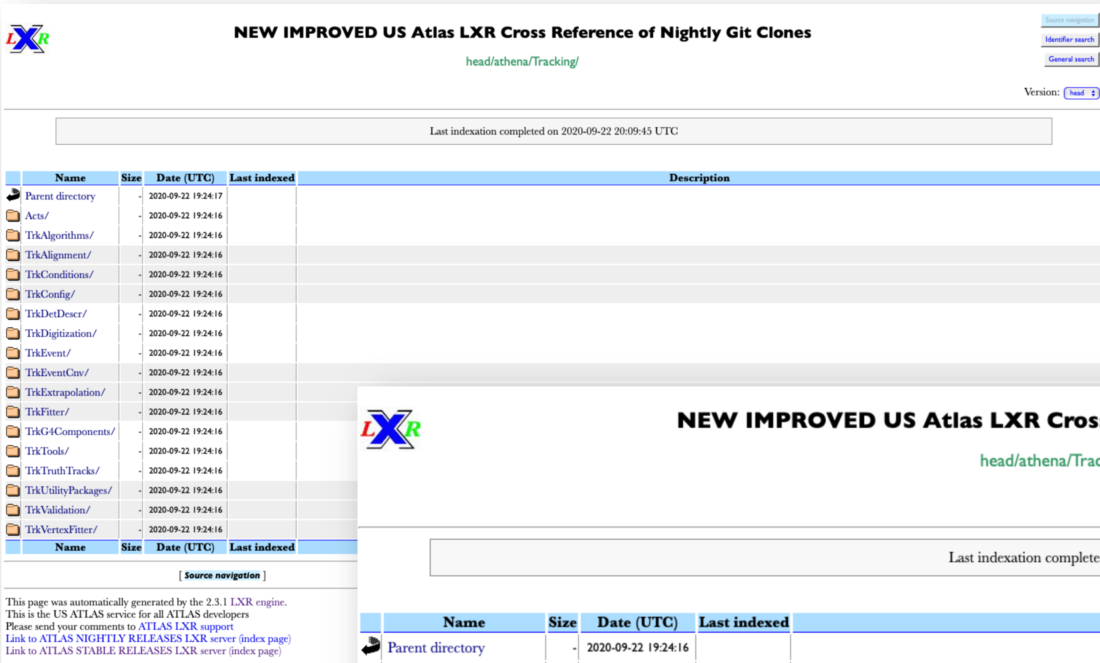
ACTS workshop



ACTS supports time propagation, more in the next slides

ACTS Integration

- Ongoing experiments like ATLAS are planning to switch to ACTS components for tracking in HL-LHC
  - Integration in Athena has already started (ACTS Vertexing for RUN3 already!)



NEW IMPROVED US Atlas LXR Cross Reference of Nightly Git Clones

head/athena/Tracking/

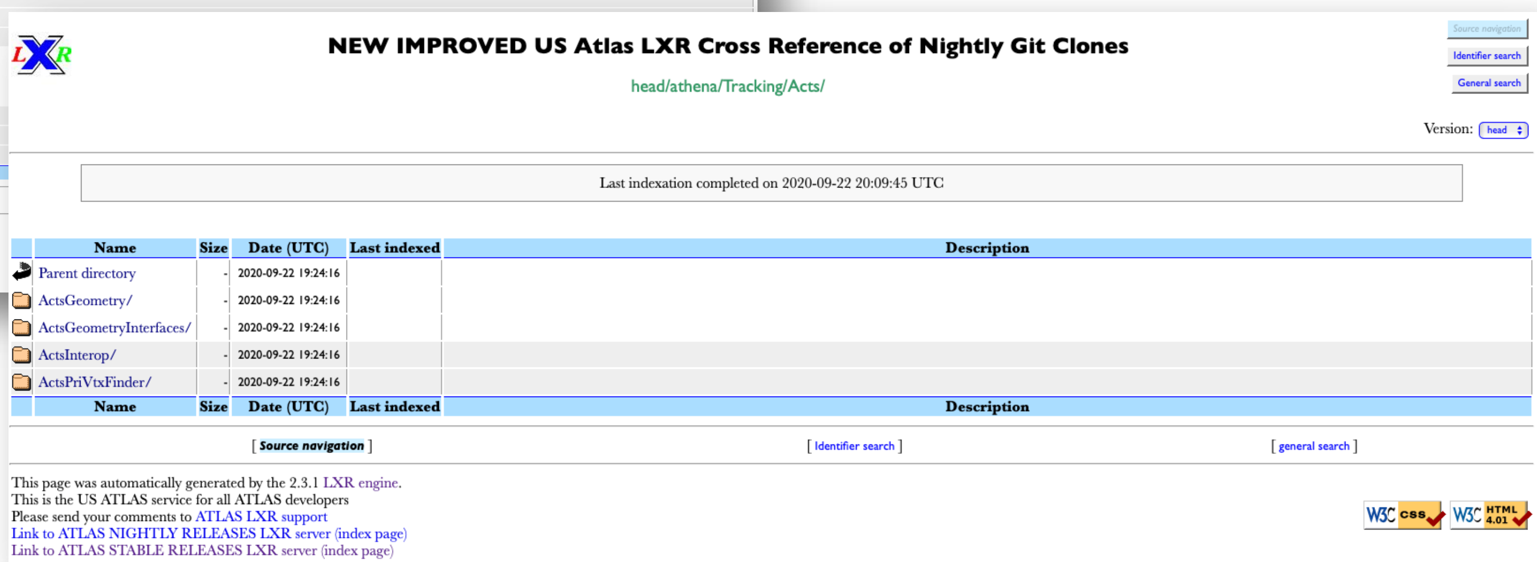
Version: head

Last indexation completed on 2020-09-22 20:09:45 UTC

Name	Size	Date (UTC)	Last indexed	Description
Parent directory	-	2020-09-22 19:24:17		
Acts/	-	2020-09-22 19:24:16		
TrkAlgorithms/	-	2020-09-22 19:24:16		
TrkAlignment/	-	2020-09-22 19:24:16		
TrkConditions/	-	2020-09-22 19:24:16		
TrkConfig/	-	2020-09-22 19:24:16		
TrkDetDescr/	-	2020-09-22 19:24:16		
TrkDigitization/	-	2020-09-22 19:24:16		
TrkEvent/	-	2020-09-22 19:24:16		
TrkEventCnv/	-	2020-09-22 19:24:16		
TrkExtrapolation/	-	2020-09-22 19:24:16		
TrkFitter/	-	2020-09-22 19:24:16		
TrkG4Components/	-	2020-09-22 19:24:16		
TrkTools/	-	2020-09-22 19:24:16		
TrkTruthTracks/	-	2020-09-22 19:24:16		
TrkUtilityPackages/	-	2020-09-22 19:24:16		
TrkValidation/	-	2020-09-22 19:24:16		
TrkVertexFitter/	-	2020-09-22 19:24:16		

[Source navigation]

This page was automatically generated by the 2.3.1 LXR engine.
This is the US ATLAS service for all ATLAS developers
Please send your comments to ATLAS LXR support
Link to ATLAS NIGHTLY RELEASES LXR server (index page)
Link to ATLAS STABLE RELEASES LXR server (index page)



NEW IMPROVED US Atlas LXR Cross Reference of Nightly Git Clones

head/athena/Tracking/Acts/

Version: head

Last indexation completed on 2020-09-22 20:09:45 UTC

Name	Size	Date (UTC)	Last indexed	Description
Parent directory	-	2020-09-22 19:24:16		
ActsGeometry/	-	2020-09-22 19:24:16		
ActsGeometryInterfaces/	-	2020-09-22 19:24:16		
ActsInterop/	-	2020-09-22 19:24:16		
ActsPriVtxFinder/	-	2020-09-22 19:24:16		

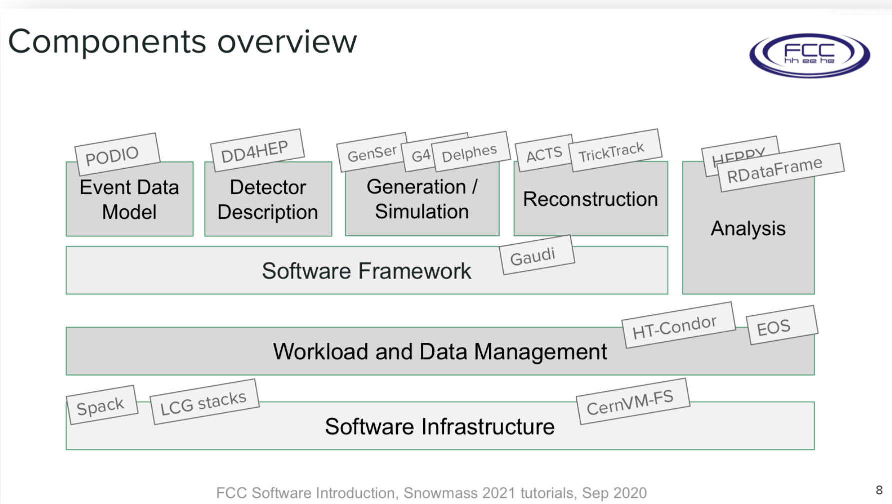
[Source navigation] [Identifier search] [general search]

This page was automatically generated by the 2.3.1 LXR engine.
This is the US ATLAS service for all ATLAS developers
Please send your comments to ATLAS LXR support
Link to ATLAS NIGHTLY RELEASES LXR server (index page)
Link to ATLAS STABLE RELEASES LXR server (index page)

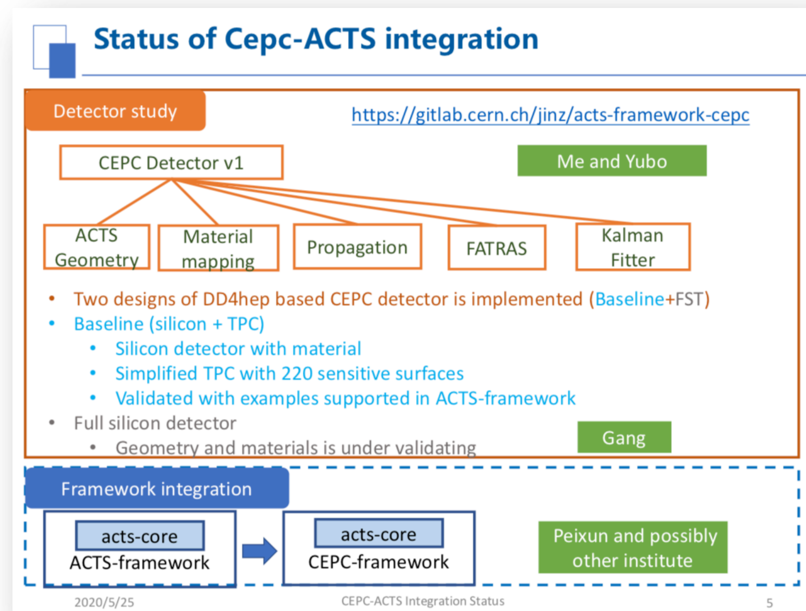
ACTS Integration

- Ongoing experiments like ATLAS are planning to switch to ACTS components for tracking in HL-LHC
- Future detectors are also integrating ACTS in their software infrastructure

FCC



CEPC



As a side note (it should not affect us, but worth mentioning) ACTS is very well supported for cylindrical geometries, but there is little support at the moment for test-beam geometries

Time in ACTS

- Directly from the main developers (Andreas Salzburger and Fabian Klimpel):

  - the time component is fully integrated in the ACTS propagation (and measurement structure), i.e. you can already now define the time as a measurement parameter (with error), and the propagation engine will use it as a standard parameter to be propagated (including error propagation).

  - also the ACTS Kalman Filter can already swallow time measurements and would simply perform update and filtering step on it

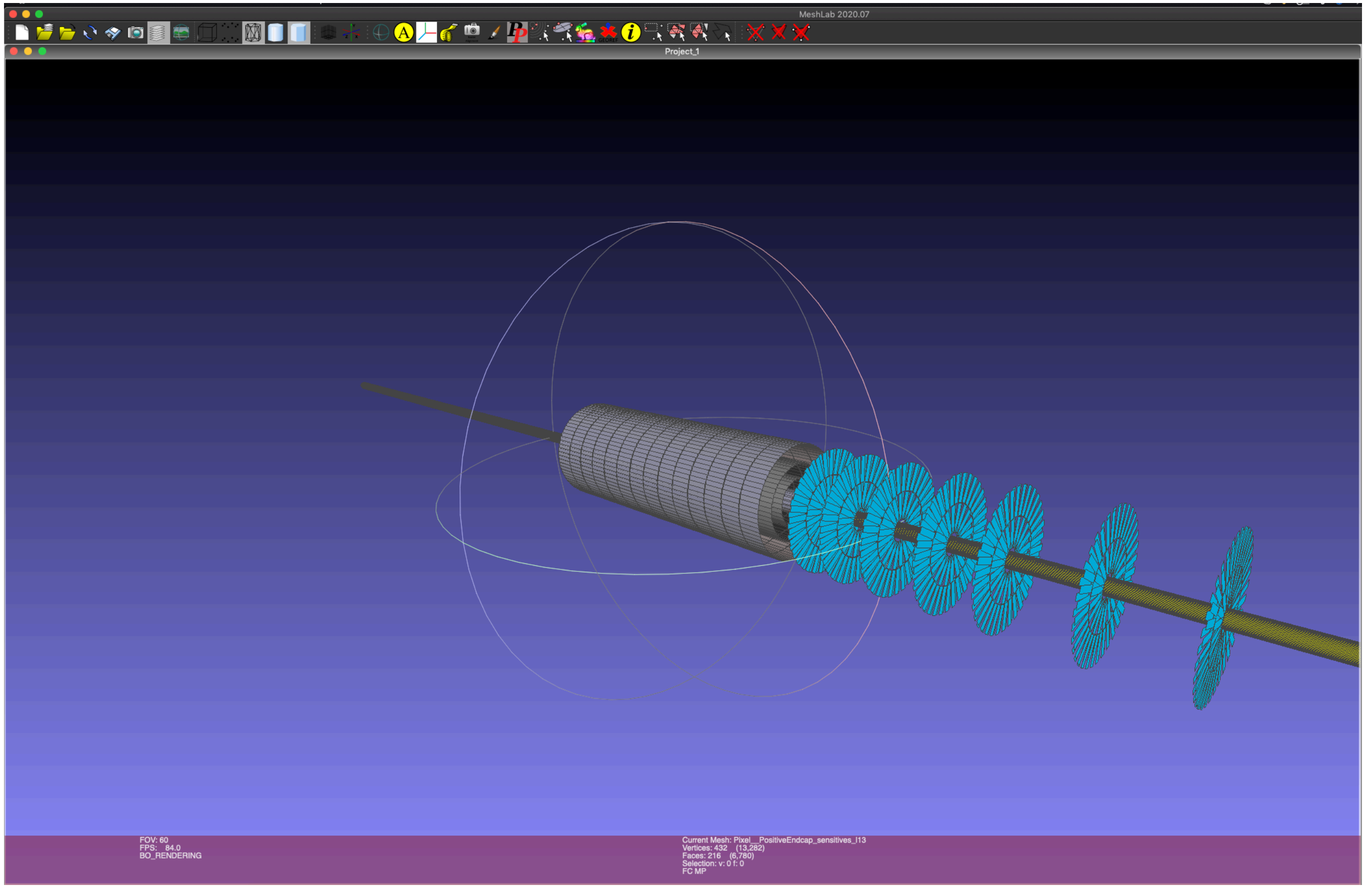
  - *Very Recent* PR (https://github.com/acts-project/acts/pull/461) for the new **ActsFatras**/Digitization where, as a start, at least time smearing will be supported for measurements

- The developers would also be happy to collaborate with us and outline **how to use (and thus validate) the code extending their current unit test validation**

Just a quick look to understand better…

- Many examples are currently provided. Tried to run a few on **geometry building** and Fatras (see back-up for instructions if you want to try it yourself)

MeshLab visualizer



Just a quick look to understand better…

SLAC

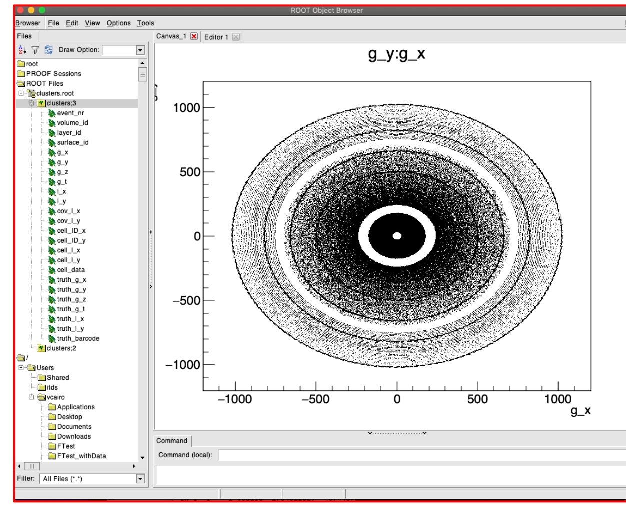
- Many examples are currently provided. Tried to run a few on geometry building and **Fatras** (see back-up for instructions if you want to try it yourself)

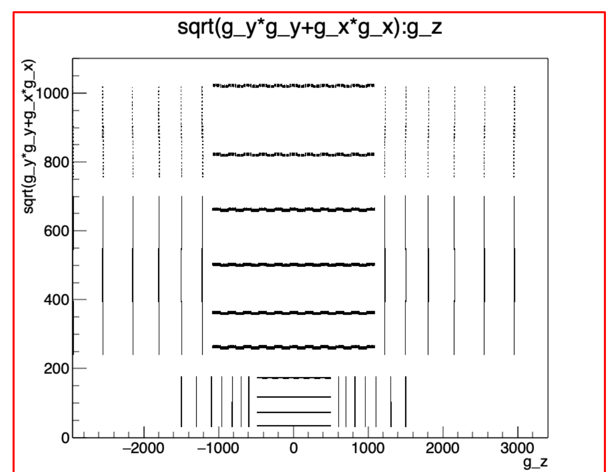
Generation and simulation are done in a single step for now
Some key options for running Fatras (./ActsExampleFatrasGeneric --help):

```
--evg-cms-energy arg (=14000)       Center-of-mass energy collision in GeV
--evg-beam0 arg (=2212)             PDG number of the first beam particle
--evg-beam1 arg (=2212)             PDG number of the second beam particle

--evg-input-type arg (=pythia8)     Type of evgen input 'gun', 'pythia8'
--evg-hard-process arg (=HardQCD:all = on)
                                    Pythia8 process string for the hard scatter
--evg-pileup-process arg (=SoftQCD:all = on)
                                    Pythi8 process string for the pile-up
--evg-pileup arg (=200)             Number of instantaneous pile-up events

--pg-pdg arg (=13)                  PDG number of the particle, will be
                                    adjusted for charge flip.
--bf-values arg (=0 0 0)            In case no magnetic field map is handed
                                    over. A constant magnetic field will be
                                    created automatically. The values can
                                    be set with this options. Please hand
                                    over the coordinates in cartesian
                                    coordinates: {Bx,By,Bz} in Tesla.
--bf-map arg                        Set this string to point to the bfield
                                    source file.That can either be a
                                    '.txt', a '.csv' or a '.root' file.
                                    Omit for a constant magnetic field.
--bf-name arg (=bField)             In case your field map file is given in
                                    root format, please specify the name of
                                    the TTree.
```

./ActsExampleFatrasGeneric -n10 --bf-values 0 0 2.5 --output-root true --output-csv true
(still uses the generic geometry from the previous page)

clusters.root hits.root particles_final.root particles_initial.root particles.root





Next possible steps and (big) questions

The following is all open for discussion

- A first goal could be to try and **start from the currently available geometries**, either the generic ML one or the ITk one and try to **modify them to add layers of detector with time info.** We could discuss this with the developers and try to build an Example that actually contains time information
  - If any of the future detectors has a geometry available in TGeo or DD4Hep, this can be plugged into ACTS to build a tracking geometry

- <u>From a pure track-reconstruction perspective, using ACTS to reconstruct tracks with a parametrization that accounts for time is most likely our best bet</u>
  - One might easily hack other existing platforms, but then those developments would only serve this project and will not be used in the future

- If the ultimate goal is to get to a full **physics prospect study** though (for instance, what is the impact on btagging and thus on certain physics topologies?) then the question becomes more complex… **We would need to use ACTS within a framework that allows to build other objects**
  - If we target ATLAS in HL-LHC, full ACTS integration in Athena may not arrive on time for a Snowmass paper due next Summer
  - For Future Colliders, like FCC and CEPS we would need to check with the current developers

Extra Slides

Instructions

As a reference, look at
https://acts.readthedocs.io/en/latest/getting_started.html
And https://indico.cern.ch/event/917970/overview (but some of the options used in the April workshop are already superseded, so you can use the instructions below as a reference)

On lxplus
git clone https://github.com/acts-project/acts source
cd source/
source CI/setup_lcg96.sh
cmake -B build -S . -DACTS_BUILD_EVERYTHING=on
cmake --build build
cd build/bin

Instructions - Geometry

./ActsExampleGeometryGeneric –help (for curiosity)
./ActsExampleGeometryGeneric -n1 --output-obj true --geo-detector-volume BeamPipe Pixel SStrip LStrip
(This uses a generic geometry created for the ML challenge and it will create .obj files that you can visualize in a 3D visualizer, like MeshLab)

Instructions - Fatras

./ActsExampleFatrasGeneric –help (for curiosity)
./ActsExampleFatrasGeneric -n10 --bf-values 0 0 2.5 --output-root true --output-csv true

```
[vcairo@lxplus768 bin]$ ./ActsExampleFatrasGeneric -n10 --bf-values 0 0 2.5 --output-root true --output-csv true
PYTHIA Warning in PhaseSpace2to2tauyz::trialKin: maximum for cross section violated
PYTHIA Warning in MultipartonInteractions::init: maximum increased by factor 4.172
03:38:36    Sequencer      INFO      Added reader 'EventGenerator'
03:38:36    Sequencer      INFO      Added algorithm 'FlattenEvent'
03:38:36    Sequencer      INFO      Added writer 'CsvParticleWriter'
03:38:38    Sequencer      INFO      Added writer 'RootParticleWriter'
- BField (scalor to/in) Tesla set to: 1
03:38:38    Sequencer      INFO      Added algorithm 'ParticleSelector'
03:38:38    Sequencer      INFO      Added algorithm 'FlattenEvent'
03:38:38    Sequencer      INFO      Added algorithm 'FatrasAlgorithm'
03:38:38    Sequencer      INFO      Added writer 'CsvParticleWriter'
03:38:38    Sequencer      INFO      Added writer 'CsvParticleWriter'
03:38:39    Sequencer      INFO      Added writer 'RootParticleWriter'
03:38:42    Sequencer      INFO      Added writer 'RootParticleWriter'
03:38:43    Sequencer      INFO      Added writer 'RootSimHitWriter'
03:38:43    Sequencer      INFO      Added algorithm 'DigitizationAlgorithm'
03:38:43    Sequencer      INFO      Added writer 'CsvPlanarClusterWriter'
03:38:45    Sequencer      INFO      Added writer 'RootPlanarClusterWriter'
03:38:45    Sequencer      INFO      Processing events [0, 10)
03:38:45    Sequencer      INFO      Starting event loop with 10 threads
03:38:45    Sequencer      INFO        0 services
03:38:45    Sequencer      INFO        0 context decorators
03:38:45    Sequencer      INFO        1 readers
03:38:45    Sequencer      INFO        5 algorithms
03:38:45    Sequencer      INFO        9 writers
 PYTHIA Warning in SimpleSpaceShower::pT2nextQCD: weight above unity
 PYTHIA Error in StringFragmentation::fragment: stuck in joining
 PYTHIA Error in Pythia::next: hadronLevel failed; try again
 PYTHIA Warning in StringFragmentation::fragmentToJunction: bad convergence junction rest frame
03:39:23    Sequencer      INFO      finished event 3
03:39:25    Sequencer      INFO      finished event 2
03:39:31    Sequencer      INFO      finished event 4
03:39:33    Sequencer      INFO      finished event 5
03:39:33    Sequencer      INFO      finished event 8
03:39:33    Sequencer      INFO      finished event 1
03:39:36    Sequencer      INFO      finished event 0
03:39:36    Sequencer      INFO      finished event 7
03:39:37    Sequencer      INFO      finished event 9
03:39:37    Sequencer      INFO      finished event 6
03:39:37    RootParticle   INFO      Wrote particles to tree 'particles' in 'particles.root'
03:39:37    RootParticle   INFO      Wrote particles to tree 'particles' in '/eos/home-v/vcairo/ACTS/source/build/bin/particles_initial.root'
03:39:37    RootParticle   INFO      Wrote particles to tree 'particles' in '/eos/home-v/vcairo/ACTS/source/build/bin/particles_final.root'
03:39:39    RootPlanarCl   INFO      Wrote particles to tree 'clusters' in 'clusters.root'
03:39:39    Sequencer      INFO      Processed 10 events in 54.838948 s (wall clock)
03:39:39    Sequencer      INFO      Average time per event: 48.100153 s/event
```

From the tutorial

https://indico.cern.ch/event/917970/contributions/3862003/attachments/2045143/3426537/zoom_0.mp4

Truth content

```
In [5]: h = pd.read_csv('/home/msmk/code/acts/sim-gun-test/event000000000-truth.csv')
        h.head()
```

Out[5]:

	hit_id	particle_id	geometry_id	tx	ty	tz	tt	tpx	tpy	tpz	te	deltapx	deltapy	
0	0	4503599644147712	504403295704449033	17.785538	82.659142	-1502.5	5.021653	0.045841	0.208956	-3.811316	3.818778	-0.000439	0.000588	2.
1	1	4503599644147712	504403433143402505	15.380033	71.694099	-1302.5	4.353219	0.045776	0.208962	-3.811317	3.818778	0.000064	-0.000005	4.
2	2	4503599644147712	504403570582355977	12.977911	60.728767	-1102.5	3.684784	0.045843	0.208788	-3.811326	3.818778	-0.000067	0.000173	8.
3	3	4503599644147712	504403708021309449	11.293963	53.059425	-962.5	3.216882	0.046107	0.209114	-3.811305	3.818778	-0.000263	-0.000326	-2.
4	4	4503599644147712	504403845460262921	9.600341	45.378079	-822.5	2.748976	0.045741	0.209426	-3.811292	3.818778	0.000366	-0.000312	-1.

FATRAS

- A few words about FATRAS

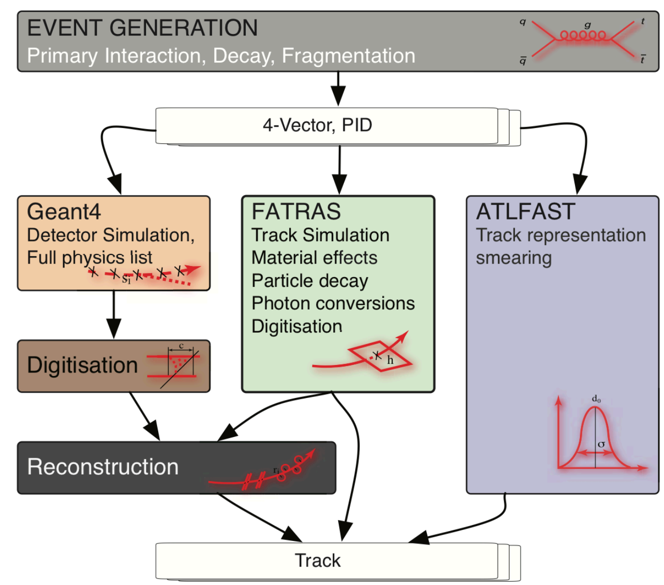
README.md

FAst TRAck Simulation package

This packages provides tools to run a fast track simulation on top of the core library. The fast track simulation (Fatras) uses the actor plug-in mechanism of the `Acts::Propagator` and its predictive navigation to simulate particle trajectories through the tracking detector.

As a fast track simulation, it uses the surface-based reconstruction geometry description, i.e. `Acts::TrackingGeometry`, as a simulation geometry instead of a detailed volumetric description. Interactions and material effects are simulated using parametrized models:

- Multiple Coulomb scattering is simulated by Gaussian (mixture) approximations.
- Ionisation loss is simulated using the Bethe-Bloch formalism.
- Radiation loss is simulated using the Bethe-Heitler formalism.



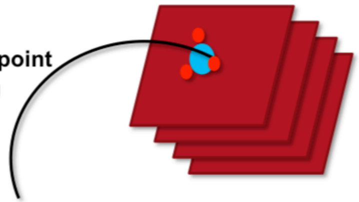
(a) Track simulation in ATLAS

HGTD (link)

Time Association Strategy

SLAC

Starting point: Extrapolate from **last hit on track** to **HGTD layer 0**



Then, for each layer:

1. Look for **surfaces** near the **extrapolated crossing point**
   - Pick up all surfaces in a ~5x5cm region around the crossing
   - Currently optimised for efficiency rather than speed
2. Look for the **cluster best matching the track**
   - Candidates: All clusters on surfaces selected in 1.)
   - For each cluster: Use KalmanUpdator to attempt to add measurement to track
   - Keep the one cluster with the best incremental chi²
3. Keep cluster if **satisfactory fit outcome**
   - Cutting at chi²/n.d.f < 5
4. In case of **successful** extension: Use **updated** track parameters from Kalman forward filter for extrapolation to next layer

Result: Set of 4 outcomes (either one cluster or failure) for the 4 layers.

VMM CAIRO 3

Information Collected for Later Analysis

SLAC

Result of this procedure:
Decorated to xAOD::TrackParticles for analysis in **InDetPhysValMonitoring**

Stored information:
Track quality based on presence/absence of hits on last Pixel/Strip layer
True production time from truth particle's production vertex
Reco track time based on chi² weighted average of cluster times on track
- Also available: Version with arithmetic mean
- Also storing RMS of times of associated clusters – sensitive to PU contamination!
- See below for def of cluster time!

Information **per layer:**
- **Presence/absence** of associated cluster
- **Cluster time –** Defined as time measured in HGTD corrected for time of flight from track origin (0,0,z_0)
- **Incremental chi²**
- **Location** in x,y,z,R
- **Truth classification of cluster –** see extra slides
- Information on **existence** of a cluster on the layer from the track regardless of association outcome
  - Using truth – see extra slides

VMM CAIRO 4