SOFTWARE R&D FUNDED BY ARGONNE LDRD (ULAB)

AUTOMATED WORKFLOWS FOR DETECTOR DESIGN

SYLVESTER JOOSTEN  
sjoosten@anl.gov  
On behalf of the Argonne EIC LDRD team

WHITNEY ARMSTRONG  
warmstrong@anl.gov

This work is supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under contract DE-AC02-06CH11357, and Argonne’s LDRD program.
A NEW DETECTOR FROM SCRATCH?

From the EIC Yellow Report to an optimized EIC detector

Detector & reconstruction requirements
Extensive list of key performance parameters inform detector choices. This table of requirements could be interpreted as a series of automated tests that a detector implementation needs to pass.

Physics requirements
Detector design has to enable many key physics measurements, while being flexible enough to accommodate new developments through the next 2 decades.
Enabled by modular, interoperable toolkit

DETECTOR OPTIMIZATION AS AN AUTOMIZED WORKFLOW

Event Sample
- Simple tracks
- Physics events

Detector geometry
- Detector A
  - Variation #1
  - …
  - Variation #N
  
- Detector Ω
  - Variation #1
  - …
  - Variation #M

Simulation + Digitization + Reconstruction

Benchmark analyses
- Detector benchmarks
- Reconstruction benchmarks
- Physics benchmarks
EVENT GENERATION
Standardized around HepMC3 data format

- Validate detector and benchmark reconstruction chain against YR requirements (using particle guns)

- Characterize and benchmark detector setup for desired physics observables (using physics event generators)

- Towards a fully integrated setup for on-demand event generation to cover all NAS/YR requirements for EIC
- Software chain standardized around the HepMC3 format
Parametrized detector implementations with DD4hep + NPDet

- Parametrized detector description defined using DD4hep
- NPDet: an extension for DD4hep that adds parametrized detectors for NP experiments (e.g. Cherenkov counters) and various DD4hep based tools (visualization, conversion, inspection, …)
- Single, unique source for geometry: dispatches full GEANT4 simulation and provides geometry for reconstruction
- Aim towards library of configurable detector options to feed into benchmark & optimization process.

https://github.com/AIDASoft/DD4hep
https://eicweb.phy.anl.gov/EIC/NPDet
https://eicweb.phy.anl.gov/EIC/detectors
MAIN TOOLKIT
Simulation/digitization/reconstruction

**DD4hep:** Geant4 geometry fully defined through detector plugin library, provides wrappers to run Geant4

**ACTS:** Experiment-independent tracking toolkit developed for HEP in modern C++, ACTS’ geometry constructed from DD4hep via plugin

**GAUDI:** Generic open project for building event processing frameworks. Enables modern task-based concurrent execution in a heterogeneous computing environment.

**Podio:** Robust data model definition to cross the boundaries between the tools

**Project Juggler:** Prototype event processing framework for EIC, tying together GAUDIO, ACTS, DD4hep and Podio. Used for the digitization and reconstruction step.

Note similarity with key4hep approach!
# BUILDING THE SOFTWARE

## Containerization and deployment

<table>
<thead>
<tr>
<th>Config</th>
<th>Build</th>
<th>Deploy</th>
<th>Downstream</th>
</tr>
</thead>
<tbody>
<tr>
<td>init</td>
<td>rundefault</td>
<td>singularity</td>
<td>singularity: #9868 (Child)</td>
</tr>
</tbody>
</table>

### Container Registry

- **Image Repository**: 5 Image repositories
- **Expiration Policy**: will run in 6 days

With the GitLab Container Registry, every project can have its own space to store images. More information

- **containers/eic_container/eic**: 8 Tags
- **containers/eic_container/eic_builder**: 8 Tags
- **containers/eic_container/node**: 1 Tag
- **containers/eic_container/rdt_builder**: 1 Tag
- **containers/eic_container/alpine**: 1 Tag

---

**Leverage own GitLab server instance** (eicweb), with continuous integration and dedicated build cluster, to automatically build container

**Reproducible container build with Spack**

**Docker container used in CI toolchain (and local development on MacOS)**

**Singularity container used for deployment to HPC environments (and local development)**
NEW PARADIGM IN SIMULATION WORKFLOW
From classic CI/CD to detector and physics benchmarks

- **CI/CD**: continuous integration & Deployment
- **Pipeline**: Collection of interdependent jobs that run in a certain sequence
- **Job**: A task (script) that runs in a CI pipeline
- **Artifact**: File(s) produced by a job uploaded to GitLab
- **Pipeline trigger**: Connecting pipelines

Many repos chained together, organized with GitLab groups
Automatic detector tests and documentation

Geometry overlap checks running as part of every merge request
Automatic visualizations for detector geometries, saved as job artifacts (browsable!)

https://twiki.cern.ch/twiki/bin/view/CLIC/DawnVisualization
PIPETLINE TRIGGERS

Connecting pipelines for immediate feedback

Change in detector geometry triggers reconstruction and physics benchmarks for immediate feedback

Also dispatch from Jugger CI: Due to merge request trigger, we can ensure proper operation of new/updated reconstruction algorithms
OPEN CHALLENGES

- Need full library of Monte Carlo generators that cover entire YR program
- Automatic publication/documentation of performance metrics to webpage for discoverability
- Compiling/combining metrics from benchmarks to guide optimization
- Distributed data persistency required, as GitLab artifact approach does not scale for large simulations
- Dispatch to HPC integrated in workflow (custom singularity gitlab runner? K8s? Manual pre-computations?)
EXAMPLE:
SIMPLE BROWSER-BASED WORKFLOW
BROWSER-BASED WORKFLOW EXAMPLE

Let’s change VertexTrackerRadius1 from 30mm to 29mm
BROWSER-BASED WORKFLOW EXAMPLE
Simplest workflow uses the browser-based GitLab IDE
BROWSER-BASED WORKFLOW EXAMPLE

Commit changes to a new branch, and…
BROWSER-BASED WORKFLOW EXAMPLE

...create a new Merge Request

**New Merge Request**

- **From** vertex_tracker_radius into master
- **Change branches**

**Title**

Tweaked the radius

Start the title with Draft: or WIP: to prevent a merge request that is a work in progress from being merged before it's ready.

Add description templates to help your contributors communicate effectively!

**Description**

- **Write**
- **Preview**

Describe the goal of the changes and what reviewers should be aware of.

*Markdown and quick actions are supported*

**Attach a file**
BROWSER-BASED WORKFLOW EXAMPLE

…the Merge Request then triggers the CI.
ALL WITHOUT LEAVING THE BROWSER OR INSTALLING ANY SOFTWARE LOCALLY!
Our software R&D in the context of the EICUG software EoI
A WORKFLOW-CENTRIC SOFTWARE APPROACH FOR EIC

- **Workflows** - leveraging the power of GitLab CI
  - ✓ Automized simulation-reconstruction-analysis pipelines
  - ✓ Low bar of entry, were able to onboard external collaborators (students!) with minimal training, already doing valuable work.
  - ✓ **next**: build out analysis portfolio, better dispatch to HPC, persistent distributed data storage, “publish” benchmark results through GitLab Pages

- **Detector simulations**
  - ✓ Geometry and simulation tooling around DD4hep
  - ✓ **next**: develop library of parametrized detector concepts, accelerate simulation with GANs

- **Reconstruction**
  - ✓ Functioning generic tracking with ACTS (and Gaudi)
  - ✓ **next**: reconstruction for other subsystems, improve task-based concurrency, accelerate part of workflow on GPUs, or with AI.

- **Data model**
  - ✓ Data model definition with podio
  - ✓ **next**: optimize and freeze exact data model, together with community

- **Data analysis preservation**:
  - ✓ Analysis as part of benchmark portfolio pipeline
  - ✓ **next**: website to aggregate analysis results and monitor quality of results (in the vein of benchmarks/tests).

- **Explore user-centered design**
  - ✓ Modular workflow with low bar of entry for analysis
  - ✓ **next**: automatic publication of pipeline results to website

- **Discoverable software**
  - ✓ Software pipeline fully containerized, software stack built and managed with Spack
  - ✓ **next**: integration of newer software tools with Spack
THE END