EICrecon Roadmap

Wouter Deconinck

EICrecon: Evolutions In Progress

- JMultifactory -> JOmniFactory: better podio integration, more uniform algorithms
- Separating algorithms from factories
- Exceptional fatalities
- Modular algorithm

JMultifactory -> JOmniFactory

Direction:

- Factory uniformity to allow easier maintainability.
- JOmniFactory should allow for easier external wiring (read: config files).

Status:

- Nearly complete: Two remaining algorithm/factories need conversion.

Separating Algorithms from Factories

Direction:

- Algorithms turn const collection inputs into non-const collection outputs.
- Algorithms should not need to do collection memory management.
- Algorithms define their input, output, services, properties.
- Algorithm authors should not need to know details about JANA2.
- JANA2 provides input, output, services, properties as required.
- Interface definition (`algorithms::`) should remain interface only.

Status:

- Many factories are already thin front-ends to algorithm.
- Some factories try to be too smart, take over algorithm functionality.
- Gray areas: programmatic algorithm configuration definition.

Modular and Generic Algorithms

Direction:

- Modularity in the sense of:
  - Accepts only podio input, produces only podio output
  - Declares its own services and properties
  - Avoids the need for algorithm developers to be familiar with everything
- Modularity allows (theoretical) use in other frameworks, collaboration with key4HEP project
  - Can be used as the basis for k4FWCore and k4ActsTracking

Status:

- Most calorimetry algorithms are now modular
- Tracking algorithms most challenging due to reliance on Acts data types

Exception Fatality

Direction:

- Historically: multiple geometry options leads to missing collections.
- Currently: exceptions cause some critical errors to be ignored.
- Exceptions indicate an unrecoverable error and should be fatal.
- Fall-back options in algorithms should not result in exceptions.

Status:

- Conversion underway, but some challenges remain

EICrecon: Evolutions To Come (For Discussion)

- Tracking: Transition to full podio data model support
- Timeslices: Fold/Unfold algorithms
- Allow for both timeslices and physics events as input
- External algorithm wiring: toml, ini, yaml, json, python
- Conditions databases and configuration management
- Multithreading
- AI/ML integration: support for inference, integration with learning workflows
- Input batching and buffered processing
- GPU and other accelerator integration
- Integration with object storage, remote storage elements

Other Broad Areas for Discussion

- Development experience
- Algorithm modularity
- Digitization modularity
- Configuration and orchestration
- Multi-architecture support