Proposal: A new data matching scheme in CBM

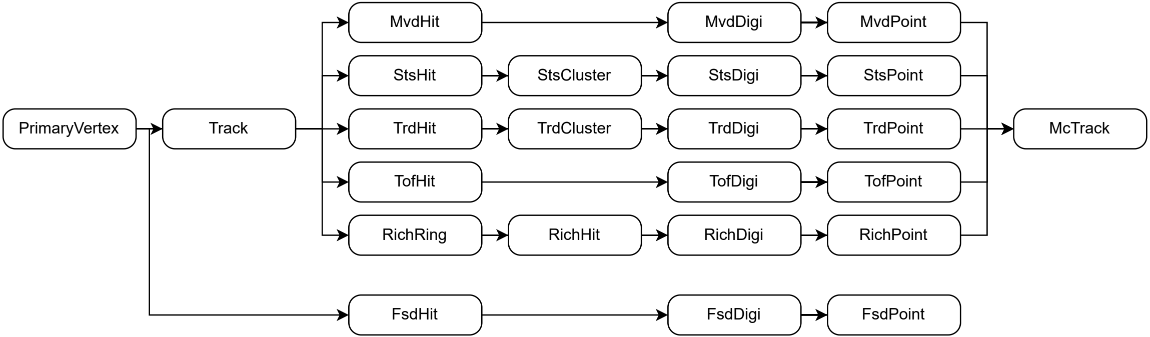
Sergei Zharko

CBM Department
GSI Helmholtzzentrum für Schwerionenforschung
Darmstadt, Germany

January 15, 2026

Introduction

- **Matching** – referencing of the data of different types:



- **Origin** – an object, which refers to object(s) of another data type(s)
- **Linked** – objects, to which the origin refers

Current matching/referencing

- References are stored in particular data types in different way
- MC matching relies on `CbmLink` and `CbmMatch` classes:
  - Unused data fields (e.g., weight = 1 for MC-track matches)
  - Frequent memory allocation: each `CbmMatch` object contains an std::vector of links
  - virtuality and dependency from ROOT (inheritance to `TObject`)

New Matching Approach

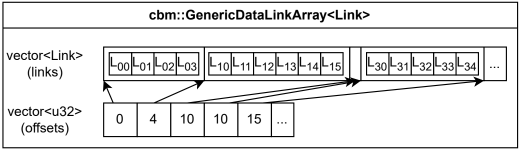
- A uniform way of linking data of different type
- Matching object levels:
  - **data link** – connector between two data objects of different types ($1 \to 1$ object relation, $1 \to 1$ type relation)
  - **data link array** – container of all data links for a sample of origins ($n \to m$ object relation, $1 \to 1$ type relation)
  - **data match** – a container of multiple data link arrays for different types of linked ($n \to m$ object relation, $1 \to n$ type relation)
- The types of linked data are defined in ECbmDataType enumeration
- Merge request !2250:
  - Classes for `DataLink`, `DataLinkArray`, `DataMatch`
  - Implementations of the data-link and data-link array (weighted link, simple link, reco-mc track link)

DataLink<ECbmDataType Origin, ECbmDataType Linked>

- A connector between an object of Origin and an object of Linked (1 → 1)
- Carries an index of the linked object (base class DataIndex<Linked>) in an array and extra information (e.g. weight)
- Multiple implementations of a concrete data link are supported:
  - SimpleDataLink<Index>: just an index of linked object (default)
  - WeightedDataLink<Index>: index + float weight (links to MC-points)
  - RecoMcTrackDataLink<Index>: a link between reco and MC tracks (different numbers of hits in a relation, purity)
- Class DataLink<ECbmDataType Origin, ECbmDataType Linked>:
  - assigne a particular implementation of a link to a pair of (origin, linked)

DataLinkArray<ECbmDataType Origin, ECbmDataType Linked>

- Container for links between data types A and B
- Uses contiguous memory layout for all origins
- Multiple implementations of a concrete data link array are supported:
  - `GenericDataLinkArray<Link>`: default
  - `FixedSizeLinkArray<Link, Size>`: optimization for origins with a fixed number of linked (e.g. StsHit→StsCluster)
  - other implementations can be provided for different (origin, linked) pairs
- Class `DataLinkArray<ECbmDataType Origin, ECbmDataType Linked>`:
  - assigns a particular implementation of a link array to a pair of (origin, linked)



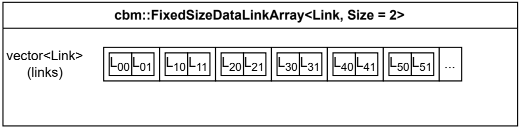
cbm::GenericDataLinkArray<Link>

vector<Link>
(links)

L_{00} L_{01} L_{02} L_{03} L_{10} L_{11} L_{12} L_{13} L_{14} L_{15} L_{30} L_{31} L_{32} L_{33} L_{34} ...

vector<u32>
(offsets)

0	4	10	10	15	...



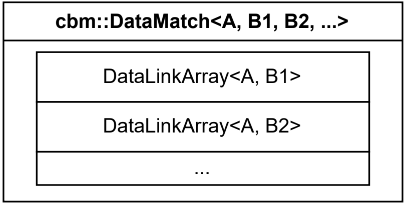
cbm::FixedSizeDataLinkArray<Link, Size = 2>

vector<Link>
(links)

L_{00} L_{01} L_{10} L_{11} L_{20} L_{21} L_{30} L_{31} L_{40} L_{41} L_{50} L_{51} ...

DataMatch<ECbmDataType Origin, ECbmDataType... Linked>

- Container for links between data types A and B1, B2, ...
- Underlying data format: tuple of `DataLinkArray<A, B1>, DataLinkArray<A, B2>, ...`
- Example: track→hit match: GlobalTrack → (StsHit, MvdHit, TrdHit, TofHit, RichRing)
- A single data match object is stored together with a reco-event, a digi-event, a digi-timeslice etc.



cbm::DataMatch<A, B1, B2, ...>

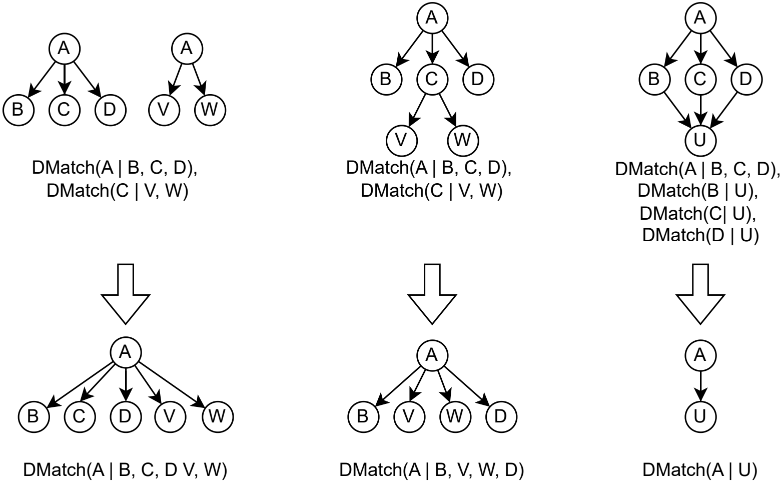
DataLinkArray<A, B1>

DataLinkArray<A, B2>

...

DataMatch<ECbmDataType Origin, ECbmDataType... Linked>

- Matches between different types can be combined automatically
- See _GTestCbmMatching



DMatch(A | B, C, D),
DMatch(C | V, W)

DMatch(A | B, C, D),
DMatch(C | V, W)

DMatch(A | B, C, D),
DMatch(B | U),
DMatch(C| U),
DMatch(D | U)

DMatch(A | B, C, D V, W)

DMatch(A | B, V, W, D)

DMatch(A | U)

Summary

Features

- Uniform: uses same API for all references
- Flexible: different types of links for different groups of data types
- Memory-optimized (contiguous memory storage for a data-sample, links do not allocate memory)
- Detached from actual data classes

TODO (next iteration)

- More use-cases for match combinations
- Testing
- Implementation with the new reco data classes

Your feedback is appreciated

- Use-case highlighting
- Code review
- Testing