Appendix C

The MuTag framework

The MuTagIMO tagging algorithm is implemented as a package within the exsisting MuTag framework[1]. Additional tools were developed to implement the techniques described in chapter 4 such as the MuTagMatchingTool and the MuTagAmbiSolverTool. In this way, no code was duplicated and the advantages of using a well developed framework were exploited.

The MuTag framework consists of an internal event data model (EDM), suitable to elegantly handle the data objects containing the information needed for muon tagging. Furthermore, the MuTag framework is designed in such a way that one *steering class* organizes the tagging procedure, data object conversion and configuration of Tools. Several tagging algorithms can be configured by configurating multiple instances of the steering class. This model allows, for example, the implementation of a muon tagger optimized for tagging cosmic muons. It also allows 'permutations' of software modules such as MuTagIMO tagging with Muonboy segments, or vice versa, MuTag tagging with MOORE segments.

In this section, the MuTag framework will be discussed. First, the MuTag EDM will be discussed. Secondly, the structure of the framework will be explained. Finally, a set of common tools provided by the framework is presented.

C.1 MuTag data objects

A *muon candidate*, i.e. an Inner Detector track with associated Muon Spectrometer segments can be fully described by the class MuTagObject (see figure C.1). The MuTagObject contains an element link to the Inner Detector Rec::TrackParticle and a pointer the matched Trk::Segment. This segment is stored in a MuTaggedSegment, described below. The MuTagObject owns a MuTagAuthor, telling the object by which MuTag tagging algorithm it has been tagged.

The MuTaggedSegment describes the association of the segment to the muon candidate track. The object holds an element link to the Muon Spectrometer Trk::Segment which was associated to the Inner Detector track. In order to access the matching of the

[1] developed by the Saclay group

segment to the track, the parameters of the track evaluated at the segments surface are written to the object as well as a `Trk::MeasuredAtaPlane` object. The set of matching variables (matchTheta, matchPhi, matchThetaAngle, matchPhiAngle) are stored as well. For the `MuTagIMO` algorithm, these matching variables correspond with

- matchTheta: The precision position matching variable $pull_{locY}$, as defined in Equation 4.9.
- matchPhi: The second position matching variable $pull_{locX}$ defined in the same way as $pull_{locY}$.
- matchThetaAngle: The precision direction matching variable $pull_{\alpha_{YZ}}$ as defined in Equation 4.10.
- matchPhiAngle: The second direction matching variable $pull_{\alpha_{XZ}}$, defined similar as $pull_{\alpha_{YZ}}$. This variable is cut on according expression 4.14.

The `MuTagContainer` is a `DataVector` of `MuTagObject`s, able to be stored on StoreGate when desired.

The `MuTag` EDM is only used in `MuTag` internally. For Athena to handle the muon candidates, dedicated converter Tools are implemented to convert a `MuTagObject` to the general EDM classes such as `Rec::TrackParticle`, `Trk::Track` and `Analysis::Muon`.

C.2 MuTag structure

The class organizing the different components of the `MuTag` framework is the `MuTagMain` class. This class defines the input collections and retrieves them from the transient data store, i.e. `StoreGate` as discussed in section 2.2. The collections are passed to a set of algorithm classe which are called `MuTagSubAlg`s, for processing. The main package also defines the output collections and writes them out to StoreGate.

The `MuTagSubAlg` is an abstract Algorithm defining common functionalities of the tagging algorithms such as the creation of `MuTagObject`s and the conversion of the internal `MuTag` EDM to the general EDM classes. Concrete implementations of `MuTagSubAlg`s are listed in table C.2, as well as the functionality of the algorithms.

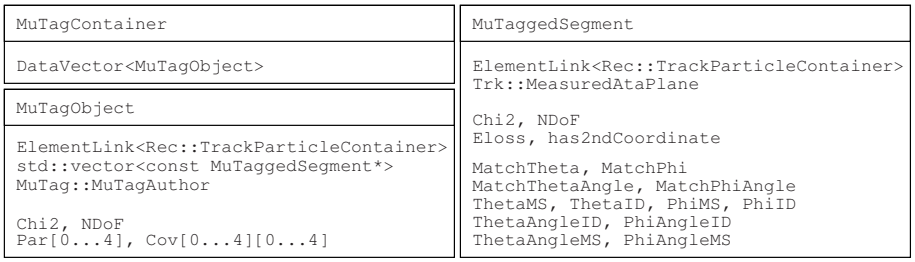

Figure C.1: *An UML diagram of the* `MuTag` *EDM classes.*

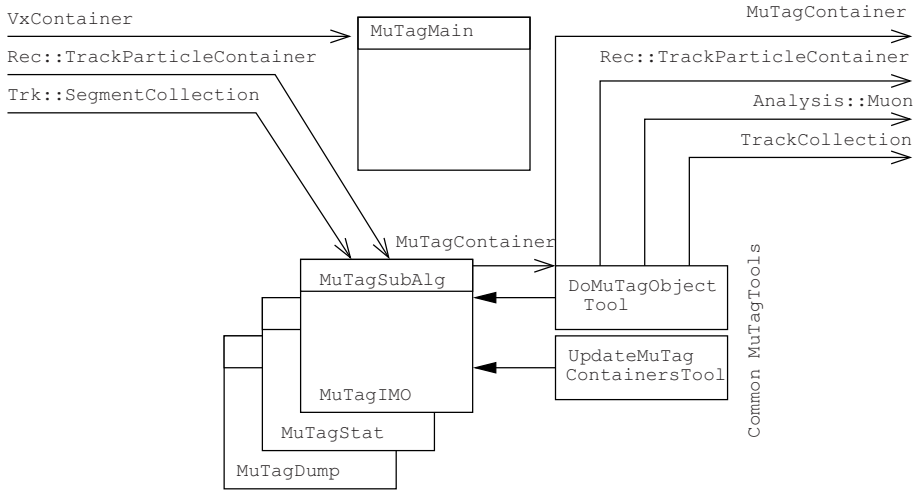

Figure C.2: *A schematic overview of the* `MuTag` *Framework.*

Algorithm name	Function
`MuTagDump`	Dumps content of the various output containers into logfiles
`MuTagStat`	Keeps track of the number of `MuTagObjects` created during the job
`MuTagInner`	Muon tagging using Inner Muon Spectrometer stations
`MuTagMedium`	Muon tagging using Middle Muon Spectrometer stations
`MuTagCosmicsInner`	Cosmic muon tagging using Inner Muon Spectrometer stations
`MuTagCosmicsMedium`	Cosmic muon tagging using Middle Muon Spectrometer stations
`MuTagIMO`	(Cosmic) muon tagging using the full Muon Spectrometer

Table C.1: *The list of the various MuTagSubAlgs with a brief description of its functionality.*

C.3 MuTag common tools

A set of common tools are available in the `MuTag` framework, designed to perform tasks common for the several concrete `MuTagSubAlg` implementations. The tools can be configured in such a way that within one MuTagMain application, different configurations of the same `MuTagTool` can be used. An example is the `SegmentFilterTool`, selecting a different set of Muon Spectrometer segments for the MuTagSubAlgs `MuTagInner` and `MuTagMedium`. The most important tools are listed below:

- `MuTagEDMHelperTool`, a tool providing several helper functions.
- `DoMuTagObjectTool` creates a `MuTagObject` object from the track and segment.
- `UpdateMuTagContainersTool` is called by the taggers at the end of each event, filling the output containers as defined by `MuTagMain` by converting the `MuTagObject`

to the appropriate Tracking EDM objects with help of the following converters:

  - MuTagToTrackTool converts the Inner Detector Rec::TrackParticles from the MuTagObject to a Trk::Track. Only the Inner Detector track is converted. The hits of the associated Muon Spectrometer segments are not stored on the resulting track.
  - MuTagToCombinedMuonTool converts the MuTagObject to an Analysis::Muon, which holds besides the Inner Detector Rec::TrackParticle, the set of associated Muon Spectrometer segments as well.
  - MuTagToParticleTool stores the Inner Detector Rec::TrackParticles from the MuTagObject.

- SegmentsFilterTool performs a preselection on the input Muon Spectrometer segment collection before passing them to the tagging algorithm. Typical selection criteria e.g. the segment station (for MuTagInner and MuTagMedium) and segment quality variables.
- TrackParticleFilterTool performs a preselection on the input Inner Detector Rec::TrackParticles before passing them to the tagging algorithm. Typical selection criteria are the track momentum, transverse momentum and the number of SCT and Pixel hits on the track.