Evaluation of Artemis and the Physics Analysis Event Data Model (PAEDM)

C. Collins-Tooth P. Sherwood

24 June 2004 Draft Version 0.1
30 June 2004 Draft Version 0.2
(added more PAEDM info and conclusions/recommendations)

1 Introduction

This document aims to provide an objective analysis of the Artemis framework and an overview of an alternative to Artemis, the PAEDM, for use within the context of a 'comparator' toolkit. It is envisaged that the comparator will provide a mechanism for the full reconstruction to be compared to Atlfast output.

2 Artemis Goals

Artemis provides a set of tools (small pieces of code performing often repeated actions which have been placed in a library) and classes that allow the writing of analyses programs which is well separated from previous processing stages, e.g. reconstruction.

A major effort has been made to keep the "C++ machinery" of the tools to a minimum, allowing the analysis programs to be written in the clearest possible manner.

We believe the separation between analysis and reconstruction, while highly desirable, is not always possible: analysis may push reconstruction to improved or new fucntionality. Artemis allows access to the underlying reconstruction objects, which in turn allows the mixing of a physics analysis (using Artemis interfaces) and investigation of the underlying reconstruction quantities.

3 Artemis Strengths

Extremely clear user code due to extensive use of tools and careful attention their construction.

Adaptors(1) These provide a decoupling of analysis from the data source. Example: Switching from DC-1 (Zebra) to DC-2 Analysis Object Data (AOD) would most likely require only that the user specify a different adaptor. The bulk of a user analysis would remain unchanged.

Adaptors(2) Adaptors provide a light weight, flexible way to extract information from reconstruction objects.

Example:Adaption of SimpleBTag to IJet

Adaptors can easily provide alternative definitions of concepts Example: "a track is a muon if there exists another track such that the combined mass is the mass of the ψ" is easily performed by adaptors, even though this is not the mainstream definition.

Even if new definitions become mainstream, adaptors would be able to adapt the current new, non mainstream concepts.

Simple Interface Heirarchy Provides a mechanism to allow tools to act on different subclasses. Provides a mechanism for future tools (e.g. Fastshower style "pingers")

Parallel Implementation Heirarchy Clear separation of common, hierarchical implementation, which is inherited privately.

Type safety Type recovery is performed through callback. No casting to go wrong.

Memory management for local objects and collections Provided by factory methods and shared pointers

4 Artemis Weaknesses

We believe that reconstruction and analysis are separable, and the that the interfaces manipulated by the programmer should be different in the two cases. Mixing between the two activities degrades the power of Artemis to enable the writing of concise analysis programs.

The underlying objects are not part of Artemis. This does not disturb greatly the flow of an Artemis program provided these underlying objects are used for simple activities such as plotting quantities (e.g. plotting the number of hits in the SCT for a given track).

However, they are not Artemis objects and cannot take part in relations among Artemis objects.

Today, the most important such relationship is associativity: any Artemis object can be associated with any number of other Artemis objects.

5 Work to make Artemis Viable

A number of items are fundamental to Artemis' viability. The tools currently available to Artemis developers are inadequate for the tasks at hand.

5.1 Segmentation Faults

Problems encountered typically arise from the rapidly changing environment external to Artemis. Issues include segmentation faults, perhaps due to overwrite errors, which are sometimes extremely difficult and time consuming to locate. Even when a fault is diagnosed (external to Artemis), it is not always possible to ensure that the faulty code is repaired. For this reason, possibly the most fundamental accessory required by Artemis is a debugger working under the python/athena framework. Detection of problems such as de-referencing null pointers, particularly in code used by, but external to, Artemis requires fluency with tools including Valgrind, Insure++ and GDB as well as possibly general techniques for detecting overwrites.

5.2 Artemis Work for DC-2 Analysis Object Data (AOD)

Artemis, by virtue of its adaptors, separates physics analysis from the data source. The changes needed to use a new data source are limited to using different adaptors.

To run on DC-2 data, Artemis–AOD adaptors need to be written. Reading DC-2 data would greatly reduce the quantity of non-Artemis code required to run an Artemis analysis, thereby diminishing the frequency of bugs in the external code.

The issue of recovering (from the AOD) the underlying reconstruction or fast simulation objects, together with how to exploit the idea of RTF navigation mechanisms (e.g. tracing particles to their constituents using the AOD *navigable* class) have not yet been fully explored in Artemis.

6 The Physics Analysis EDM (PAEDM) with AOD

An alternative framework for performing physics analysis, and which potentially could be used to compare fast and fully reconstructed information, is currently under development. The 'Physics Analysis EDM' (PAEDM) is becoming the mainstream method of doing an ATLAS analysis with the DC-2 data. The DC-2 data available for repeated analysis is the Analysis Object Data (AOD), which is a persistified representation of concrete PAEDM objects.

Using the AOD separates the reconstruction from the analysis step by providing persistified post-reconstruction files for repeated use. Currently, the results of the reconstruction are fed into the AOD via an 'AOD builder', which is semi-customisable (i.e. limited extra information not normally available in the AOD may be included by modifying the AOD builder, providing the data to be included has a 'Pool Converter' and is persistifiable). The amount of work needed to customise the AOD builder is unclear, and likely to be non-trivial.

The AOD contains a 'DataType' flag, which differentiates fast, full, and real data. This functionality is required, to allow the comparator toolkit to use the PAEDM.

7 Risks of Using the PAEDM and AOD in the Comparator

Typical data which might be compared for fast and full reconstruction might consist of:

Jet distributions, (e.g. P_t distributions, checking the result is flat in azimuthial ϕ space, η distributions etc.) using a range of jet algorithms.

Vertex distributions

Track multiplicities etc.

Quantities coming from physics analysis e.g. b-tagging efficiencies, purities etc.

These quantities can, in principle, be compared for fast and full reconstruction in the Artemis framework using the existing infrastructure and where necessary, writing or changing existing adaptors to gather information not currently gathered or used. Once the information is stored in ntuples or histograms, it can then be compared using (for example) the Kolmogorov test function HDIFF in HBOOK.

When the PAEDM is used, the process of comparison becomes somewhat different. The DataType flag should permit the filling of different ntuples or histograms (for later automated comparison), depending on whether one is processing fast or full reconstruction data.

It is assumed that the comparator would aim to compare b-tagging information from both reconstructed and Atlfast data. Several b-tagging algorithms are currently under development for the reconstructed data. Whilst switching from using the results of one b-tagging algorithm to another would be simple under the Artemis framework (only an adaptor to the new b-tagging objects would be needed), multiple b-tagging results would be handled slightly differently in the PAEDM. 'ParticleBuilders' would have to be written in the PAEDM, which would function in a similar manner to Artemis adaptors. The ParticleBuilders output results to memory (StoreGate), and allow use of the data they gather for histogramming etc.

A further risk is the AOD itself. Although the AOD will have to be used by whichever framework is chosen (Artemis or PAEDM), the transition to AOD presents special issues (in particular for a comparator) which need to be thought through.

Each AOD file may contain only one DataType (i.e. Fast, Full or FastShower), or it may contain multiple DataTypes. If multiple DataTypes are present, PAEDM developers must be careful to perform only actions which make sense for a particular DataType.

AOD files themselves may contain the summarised data of multiple 'partitions'.[1]

If multiple partitions are present in any AOD file, it may become difficult to trace the provenance of data used in an analysis, because, there is no guarantee *per-se* that the partitions in the AOD file represent part of a coherent dataset. For example, the AOD may be formed from partitions which came from different generators.

The AOD structure has not yet been adopted by other tools which may eventually be used by the comparator. Tools such as AMI (Atlas Metadata Interface) and Ganga (job submission and management) are currently unable to use AOD. It is as yet unclear whether these tools will be modified, and if so, when.

8 Conclusions

Some decisions must be made regarding the future of Artemis. To date, the project represents approximately 18 person-months of effort. Mothballing, retirement and, at the

[1] ATLAS 'partitions' are files which, together, form a dataset.

other end of the scale, increased development effort to improve the available tools for debugging and interface to the AOD are all options which could be pursued.

The next step in evaluating the PAEDM would be to invest some time producing a simple analysis job which builds a customised version of the AOD, and compares some rudimentary quantites for fast and full reconstruction. This would show that, in principle, the PAEDM is a viable alternative to Artemis for use in the comparator toolkit. It would also provide a clearer insight into the structure of the AOD and its limitations (which can not yet be obtained from use of Artemis, since it can not yet interface with the AOD.)

If such a test of the PAEDM were successful, a logical progression would be to recommend the use of the PAEDM (with its greater level of support and wider user base) for the comparator toolkit and other similar ATLAS analyses.

It is anticipated that the largest effort will be needed to customise the AOD, since at the time of writing it only contains collections of electrons, muons and photons. Additional items such as b-tag objects may present significant work to incorporate into the AOD.

It is hoped that this document has gone some way towards providing an understanding of what would be lost if the Artemis approach were abandoned, and the direction of effort required to further investigate the PAEDM approach.