Dual-use tools and systematics-aware analysis workflows in the ATLAS Run-2 analysis model

David Adams[1], Paolo Calafiura[2], Pierre-Antoine Delsart[3], Markus Elsing[4], Steven Farrell[2], Karsten Koeneke[5], Attila Krasznahorkay[4], Nils Krumnack[6], Eric Lancon[7], Wim Lavrijsen[2], Paul Laycock[8], Xiaowen Lei[9], Sara Strandberg[10], Wouter Verkerke[11], Iacopo Vivarelli[12], Martin Woudstra[13], on behalf of the ATLAS Collaboration

[1]Brookhaven National Laboratory, [2]Lawrence Berkeley National Laboratory, [3]Centre National de la Recherche Scientifique, [4]European Organization for Nuclear Research, [5]Albert-Ludwigs-Universitaet Freiburg, [6]Iowa State University, [7]Centre d'etude de Saclay Gif-sur-Yvette, [8]University of Liverpool, [9]University of Arizona, [10]Stockholm University, [11]Nikhef National Institute for Subatomic Physics, [12]University of Sussex, [13]University of Manchester

E-mail: `SFarrell@lbl.gov`

Abstract. The ATLAS analysis model has been overhauled for the upcoming run of data collection in 2015 at 13 TeV. One key component of this upgrade was the Event Data Model (EDM), which now allows for greater flexibility in the choice of analysis software framework and provides powerful new features that can be exploited by analysis software tools. A second key component of the upgrade is the introduction of a dual-use tool technology, which provides abstract interfaces for analysis software tools to run in either the Athena framework or a ROOT-based framework. The tool interfaces, including a new interface for handling systematic uncertainties, have been standardized for the development of improved analysis workflows and consolidation of high-level analysis tools. This paper will cover the details of the dual-use tool functionality, the systematics interface, and how these features fit into a centrally supported analysis environment.

1. Introduction

The second run of the Large Hadron Collider (LHC) [1] brings several computing and software challenges to the ATLAS experiment [2]. Higher luminosity will mean more data and hence more disk usage. Analyses will grow in complexity compared to Run-1. Any latencies in the analysis chain due to computing resource shortage or difficulties with software will slow down productivity. Thus, it will be highly beneficial to have a well-optimized analysis model which uses resources efficiently and does not hinder analysts' ability to publish results in a timely manner.

The analysis model in Run-1 had some issues which would not be sustainable for the future of ATLAS. The original intended model involved the use of the Athena framework [3] to analyze an Athena-based EDM (AODs) [4]. However, during Run-1 most analysts were predominantly analyzing simpler ROOT-readable n-tuples (D3PDs) with non-Athena code. The software environment consisted of numerous frameworks and standalone analysis tools (hereafter collectively referred to as the "ROOT environment"). Combined performance (CP) groups, which are responsible for providing recommendations on reconstructed object selections and corrections, were burdened with developing tool implementations for both environments. The widespread adoption of D3PDs was unexpected and initial production was handled by physics groups. The production was then centralized in 2012 but was managed by a small group of people. The existence of multiple D3PD formats consumed a large amount of computing and disk resources.

The goal for Run-2 was to learn from the lessons of Run-1 and upgrade the analysis model. One clear requirement was that analysis would need to be supported in both the Athena environment and the ROOT environment. This meant that the EDM and the analysis software tools provided by the CP and physics groups would have to be dual-use. Additionally, ATLAS would need some kind of robust, flexible data-reduction framework to reduce disk consumption and effort of end-users.

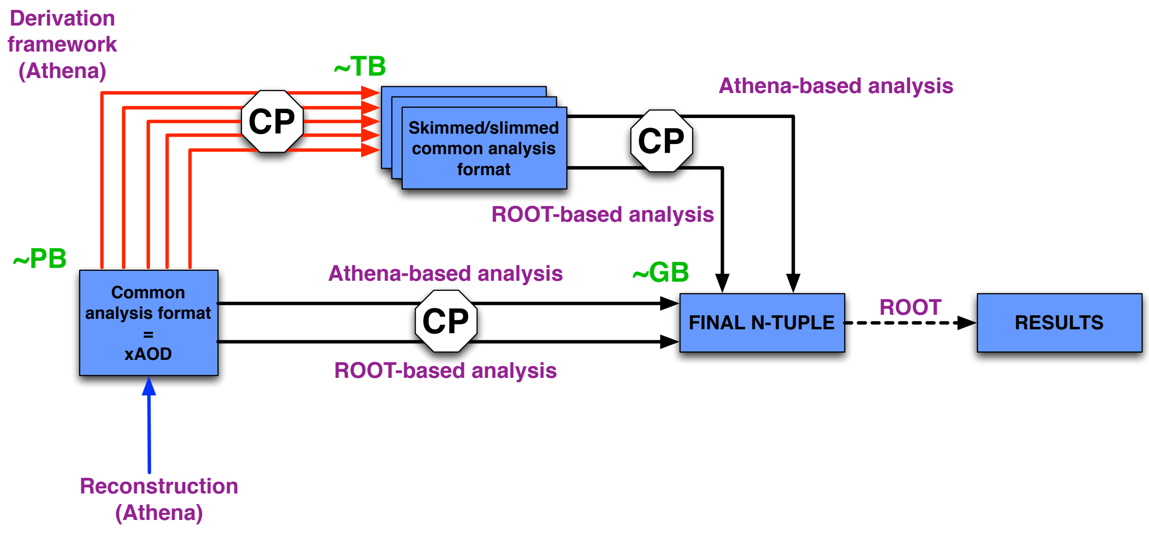

Figure 1. The ATLAS Run-2 analysis model consists of a new EDM (xAOD), a centralized data-reduction framework (Derivation Framework), and a dual-use tool infrastructure for applying combined performance group recommendations.

2. Overview of the Run-2 analysis model

The ATLAS Run-2 analysis model, shown in Figure 1 consists of the following components:

- The **xAOD EDM** is a dual-use format; i.e., it can be read in Athena and in ROOT. Physics objects are split into interface classes and variable payload (auxiliary store) classes, both of which are organized into containers. [5].
- The **Derivation Framework** is a centralized data-reduction framework. It reads in xAODs and then applies fixes, corrections, and preliminary selections, writing the results to analysis-specific output files (*derivations*). The derivation framework is designed to reduce the O(PB) size of the input xAODs down to O(TB) size for input into analysis jobs.
- **Dual-use combined performance (CP) tools** apply all the recommended prescriptions of the combined performance groups. They generally can run in the Derivation Framework or in a user's analysis.

The new xAOD EDM brings some powerful features to the analyzer which can be exploited when writing analysis software. The separation between the object interface and the variable payload makes the EDM flexible and dynamic. Variables can be added and removed as desired by the user. A specialized type of auxiliary store allows for *shallow copies* of object containers. When an object container is shallow-copied, auxiliary variables are only duplicated when modified. Otherwise, when requested, they are taken from the original container. This feature is illustrated in Figure 2. Shallow copies are very useful for applying object corrections and systematic variations in an analysis workflow, which is described further in Section 5.

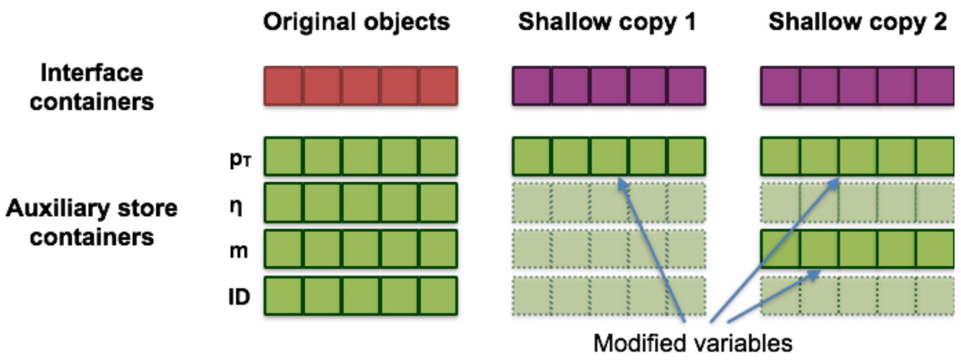

Figure 2. Shallow copies in the xAOD EDM. Interface containers are duplicated for each shallow copy, but the auxiliary store of the copy efficiently references unmodified variables from the original object's store, reducing memory consumption and copy-time.

3. Dual-use tools

As with the dual-use EDM, dual-use tools are those that can be used in either Athena or the ROOT environment. In Athena, tools depend on various framework components in order to perform functions such as interacting with the event store and logging messages. Additionally, Athena provides functionality to retrieve tools (via a tool service) and a build system (CMT) for compiling and integrating tools into the application. Analogous components were developed (or adopted) for use in the ROOT environment to give a similar look and feel. These include alternate implementations of the event store, tool store, and message streams. A build system which had already achieve widespread popularity for the ROOT environment in Run-1, RootCore, was chosen as the analog to CMT.

Combined performance tools in the Run-2 analysis model are dual-use tools that are used to implement the recommendations of the ATLAS combined performance groups. These recommendations include

- **Object corrections** - energy calibrations, resolution smearing, etc.
- **Object selections** - cleaning bad objects, official identification criteria, etc.
- **Object weights** - reconstruction and identification efficiencies.

To make tools dual-use, the underlying machinery which is specific to each environment must have a similar interface. The actual implementations of the components are then controlled via compiler switches. The class layout is shown in Figure 3. Dual-use tools must inherit from the AsgTool base class and should usually implement some abstract interface which itself inherits from IAsgTool. This pattern supports the component design model of Athena. The AsgTool base class provides event store access through a common interface, as well as messaging functionality.

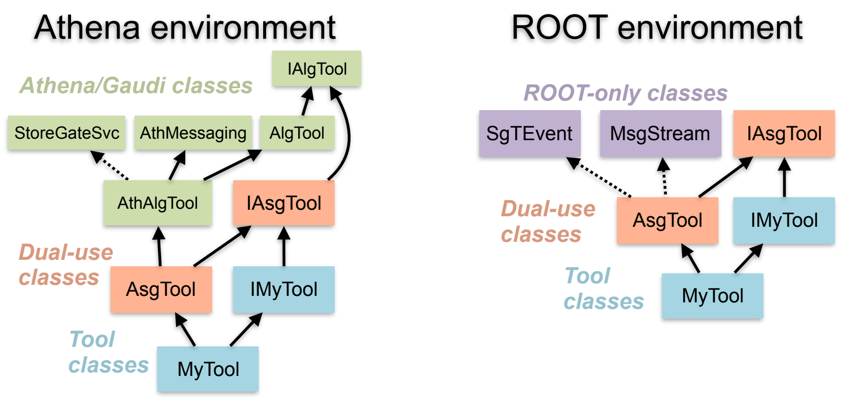

Figure 3. Class diagram for dual-use tools in the Athena environment (left) and the ROOT environment (right). The implementation of the tool classes is unchanged when switching environments. Compiler directives control which underlying base classes and framework services get used.

4. Systematics

The evaluation of systematic uncertainties in Run-1 ATLAS analyses was cumbersome. Tools that provided systematic variations varied greatly in interface and implementation, which made usage difficult and error-prone. For Run-2 the goals were to make things easier for analyzers and add support for more complicated use-cases: evaluating correlations of systematics and evaluating systematics at magnitudes other than $\pm 1\sigma$.

In the new analysis model, the management and interaction with systematic variations has been standardized. New classes were introduced to represent and organize systematic variations, and a new abstract interface was provided to ensure that tool developers provide a consistent behavior to users.

4.1. Systematics representation

An individual systematic variation is represented by a *SystematicVariation* object. It consists of a name and an optional variation magnitude (i.e., number of sigmas). Internally, the object holds a string built from these two pieces which can be parsed to extract the magnitude.

In the statistical interpretation of physics data, it is typical to treat systematic variations as nuisance parameters and evaluate the model of the data at a variety of arbitrary points in nuisance parameter space. Thus, a *SystematicSet* object was introduced to represent a collection of systematic variations that are to be applied simultaneously. Internally, the SystematicSet holds a set of SystematicVariation objects and a cached combined string name and hash. The hash allows for efficient lookup in hash containers where the SystematicSet is used as the key.

Upon initialization, systematics-aware tools register their affecting and recommended systematics in a *SystematicRegistry* object. This registry is a singleton object which helps to catch compatibility errors in the systematics and can be used to inform the user about which systematic variations are available based on their tool configuration. The interaction between the user, tools, and the registry can be seen in Figure 4.

4.2. Systematics interface
All systematics tools implement the ISystematicsTool abstract interface, which harmonizes the handling of systematics across all tools for the user. This interface allows to configure a tool for a particular SystematicSet, to query which systematics affect a given tool, and to query which subset of those systematics are recommended to be evaluated in a given analysis.

The application of the systematics in the analysis job is done in the event loop. Usually the user has a list of SystematicSet objects and some per-event code which needs to be run for each set. The user passes a requested SystematicSet to their tools via the ISystematicsTool interface. Each tool filters the requested systematics with its own affecting systematics, and then updates its internal configuration according to the filtered result. The tools return a status code signifying success or failure. Next, the user applies the tools to xAOD objects via the dedicated tool interfaces, and the tools operate on the data according to their current systematic settings. The filtering and configuration lookups in the tools are very efficient thanks to the SystematicSet's cached hash value and use of hash maps. This process is illustrated in Figure 4.

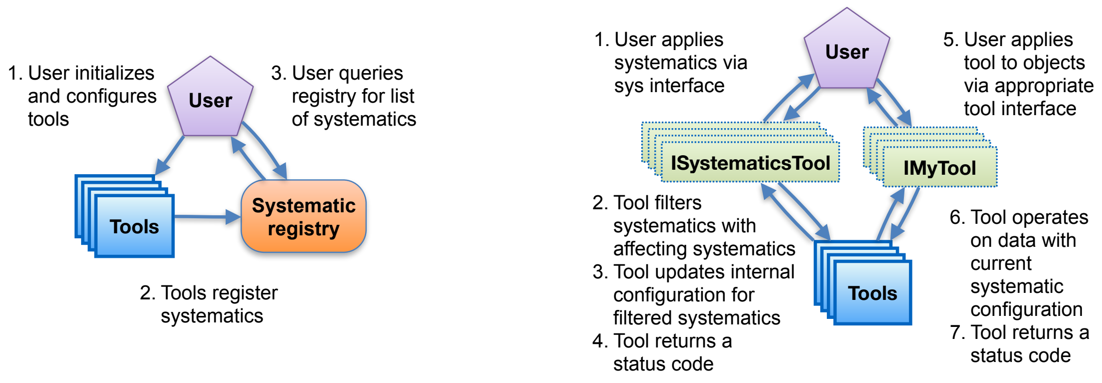
1. User initializes and configures tools
User
3. User queries registry for list of systematics
Tools
Systematic registry
2. Tools register systematics
1. User applies systematics via sys interface
User
5. User applies tool to objects via appropriate tool interface
ISystematicsTool
IMyTool
2. Tool filters systematics with affecting systematics
3. Tool updates internal configuration for filtered systematics
4. Tool returns a status code
Tools
6. Tool operates on data with current systematic configuration
7. Tool returns a status code

Figure 4. User initialization (left) and application (right) of tools and their associated systematics. The initialization phase occurs once at the beginning of a job, while the application procedure is performed for every event and every systematic variation.

5. Analysis workflows
The flexibility of the dual-use tool model means that analyzers can design their software workflows in a variety of ways. However, as illustrated in Figure 1, it is expected that most analysis workflows will have the same general features. An application reads an input xAOD (typically a derivation), applies CP tools, and writes out some "final" analysis n-tuple which can be subsequently used for making plots and other results. As mentioned in Section 2, shallow copies can be used to represent variations on objects in an intuitive way but with minimal memory cost. Thus, a popular model for analysis code is to generate shallow copies of each object type for each systematic variation, and then apply the various correction, selection, and efficiency tools to the objects with the appropriate systematics settings for each shallow copy. Multi-object tools are then also applied, which perform operations like object overlap removal and calculation of physics quantities like the missing transverse energy. Finally, the shallow copies are written to the "final" analysis n-tuple.

Despite the greater simplicity in configuring and using CP tools in the new analysis model, high level analysis tools are frequently used to simplify things even further. These "super-tools" manage and apply all of the CP tools for the user, combining the full chain of operations into a compact interface. Physics groups often provide super-tools in order to ensure consistent object and event definitions for all analyzers. Others attempt to work across various physics group conventions. QuickAna is one such tool that aims to provide harmonized object definitions while also supporting the conventions of the various physics groups.

The example workflow model described above is shown with descriptions in Figure 5.

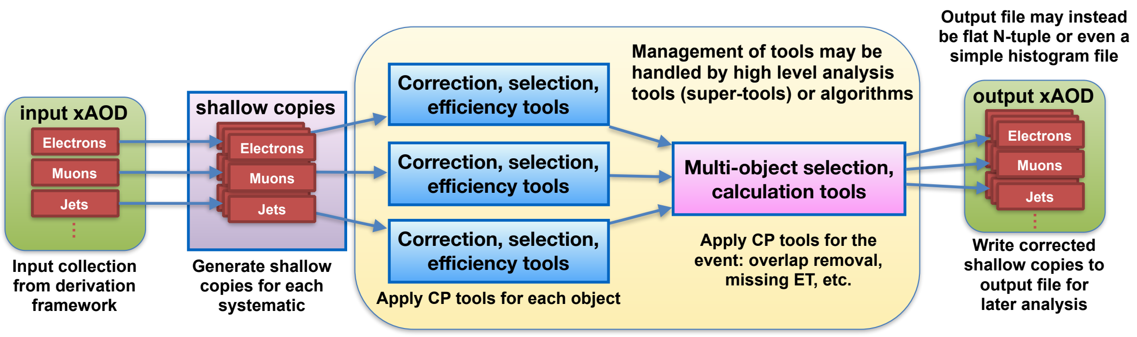
input xAOD
Electrons
Muons
Jets
Input collection from derivation framework
shallow copies
Electrons
Muons
Jets
Generate shallow copies for each systematic
Correction, selection, efficiency tools
Correction, selection, efficiency tools
Correction, selection, efficiency tools
Apply CP tools for each object
Management of tools may be handled by high level analysis tools (super-tools) or algorithms
Multi-object selection, calculation tools
Apply CP tools for the event: overlap removal, missing ET, etc.
Output file may instead be flat N-tuple or even a simple histogram file
output xAOD
Electrons
Muons
Jets
Write corrected shallow copies to output file for later analysis

Figure 5. An example analysis workflow in the new analysis model which uses shallow copies of object containers to represent systematic variations, applies CP tools to the objects, and writes the results to an output file.

6. Future work

Future work is foreseen on the analysis model to further improve the software experience of analyzers. Some functionality which doesn't yet work as well outside of Athena will be improved (e.g., meta-data handling). In general, the trend towards greater framework flexibility with dual-use components is expected to continue. The concept may be extended to include dual-use algorithms and other services. Finally, more dual-use tools will be written to share advanced analysis techniques across ATLAS such as background estimations, reconstruction of complex events, and analysis optimization.

7. Conclusion

The new analysis model, and particularly the dual-use tool design, has transformed the way ATLAS analysis code is written. It provides harmonized interfaces and behaviors for tools, improving their general quality and usability. It allows tool developers to support multiple frameworks with very little effort. It also gives users greater freedom in choosing an analysis framework. As a result, it should be effective at improving the overall productivity of ATLAS in Run-2.

References

[1] Evans L and Bryant P 2008 LHC Machine *JINST* **3** S08001
[2] ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider *JINST* **3** S08003
[3] Calafiura P, Lavrijsen W, Leggett C, Marino M, Quarrie D 2004 The athena control framework in production, new developments and lessons learned *Interlaken, Computing in high energy physics and nuclear physics* 456-458
[4] A Farbin 2008 ATLAS analysis model *J. Phys.: Conf. Ser.* **119** 042012
[5] Eifert T, Gillberg D, Koeneke K, Krasznahorkey A, Moyse E, Nowak M, Snyder S, Van Gemmeren P 2015 Implementation of the ATLAS Run 2 event data model *Proceedings of the CHEP 2015 conference J. Phys.: Conf. Ser.*