ATLAS datasets

David Adams
July 31, 2002

Introduction

ATLAS users carrying out analysis will access collection of event data for which raw data were collected under known conditions, additional data were constructed with specified reconstruction algorithms and event selections were made with specified selection algorithms. Here we call these collections datasets, propose an interface and describe the operations that are used to create them.

Definitions

We begin with brief definitions of some of the terms used in the following.

Event data

ATLAS data is acquired for a series of beam crossings or the simulation thereof. Event data is that associated with one such crossing. This includes both the raw data read off the detector and the reconstructed data derived from it.

Event ID

Each beam crossing is labeled with a unique index called the event ID. These identifiers can be ordered. These might be implemented as 64-bit integers.

EDO

The unit of event data is the event data object or EDO. Each EDO is labeled with a content ID that specifies the kind of data (the content) that resides within.

Event view and content ID

An event view is a subset of the event data corresponding to a single event ID. The view is a collection of EDO's indexed by content ID. We restrict our consideration to collections in which there is exactly one EDO for each included content ID. The collection of content ID's in an event view is called the content of the view. Event view will sometimes be abbreviated as event.

In the ATLAS transient world defined by StoreGate, the content ID is specified by the EDO type-key, *i.e*. the transient type (class ID) and a string key.

Different views of an event may hold different versions of an EDO with the same ID. They might be different because they were produced with different versions of an algorithm or because their algorithms had different input.

Except for the first, each EDO is created by an algorithm taking other EDO's from the same event as input. The latter are the parents of the former. An event view is said to be complete if the parents of all EDO's are included in the view. An event view is consistent if the contained EDO's constitute a subset of some complete view. If a parent EDO in a consistent view were replaced with a different version, the resulting view would be inconsistent.

Dataset definition

Here we define the dataset interface. To simplify the language, we speak of what the dataset contains or provides when we really mean that the dataset has an interface that provides access to this information. Implementations are discussed in later sections.

Event list

A dataset contains data for a specified collection of event ID's. Any event ID can appear no more than once in this collection. The dataset provides means to return this collection of event ID's and is able to return the smallest and largest of these identifiers. We will use the phrase "a dataset includes the events in an event ID list" to mean the dataset holds data for the ID's in that list.

Content

A dataset has a content expressed as a list of content ID's. The view provided for each event in the dataset is restricted to this content or a subset thereof.

Two datasets have the same content if they have the same content ID list and have nonoverlapping content if none of the content ID's in one appears in the other.

Data

The event and content ID lists specify the scope of the dataset. Any EDO contained in a dataset may be uniquely identified within that scope by its event and content ID's. The dataset provides means to access any of its EDO's identified in this manner. We do not specify the means and expect it will depend on the nature of the dataset. One dataset might return a transient address while another return a persistent identifier that requires a data store for resolution.

The actual scope of a dataset may differ from the nominal scope defined by the event and content ID lists. An EDO specified by an event-content ID pair in the nominal scope may be absent. However no EDO's may be included that are outside the nominal scope. The actual scope is a subset of the nominal scope.

A dataset provides the means to report an absent EDO as such and to distinguish between this and an EDO that is out of nominal scope or is inaccessible for some other reason (e.g. file not available).

Persistence

There is at least one mechanism by which datasets persist, i.e. they can be created in on process, written out and then later read back and used in another process.

Identifier

Each dataset has a unique identifier that can be used to reference the dataset. A means is provided to dereference these identifiers, i.e. to locate the dataset associated with an identifier. The scope of the uniqueness and dereferencing depend on the scope of the dataset. A dataset that only exists with one process may be identified by its address in memory. Many datasets will shared throughout the ATLAS collaboration and must be identified and located in a global manner.

Locking

A dataset must not be modified after it is written to persistent store or used to create another dataset. This is accomplished via a locking mechanism. A dataset may not be modified once it is locked and may not be written out or used to create other datasets until it is locked. A dataset cannot be unlocked.

History

Most datasets are created from one or more other datasets called the parents. A dataset is able to report its parent or parents, e.g. by identifier. A dataset should also provide a means to identify the algorithm that was used to construct it from its parent or parents.

Event and content selection

Below we discuss the transformation of datasets. The simplest of these simply restrict the view of a dataset by restricting the scope to a subset of the events or content of the original dataset. Rather than introducing specialized datasets to describe these transformations, we require that all dataset provide mechanisms to select events and to select content. Of course, these operations may only be performed on a dataset that has not been locked.

Summary

In summary, a dataset is required to provide an event ID list, a content ID list and a means to access the EDO associated with any pair formed by taking one entry from each of these lists. It also provides persistence, reference, locking and history.

Transformations

The preceding section described the interface for reading a dataset. Here we describe the operations that can be used to transform datasets. We assume a dataset cannot be modified after is locked. Instead one or more datasets are transformed into one or more new datasets.

There must be a means to create a dataset in the absence of existing datasets. We do not place any restrictions on this procedure except that the result satisfies the conditions defining a dataset.

Below we identify some fundamental operations that can be used to transform datasets. The transformations are illustrated in figure 1.

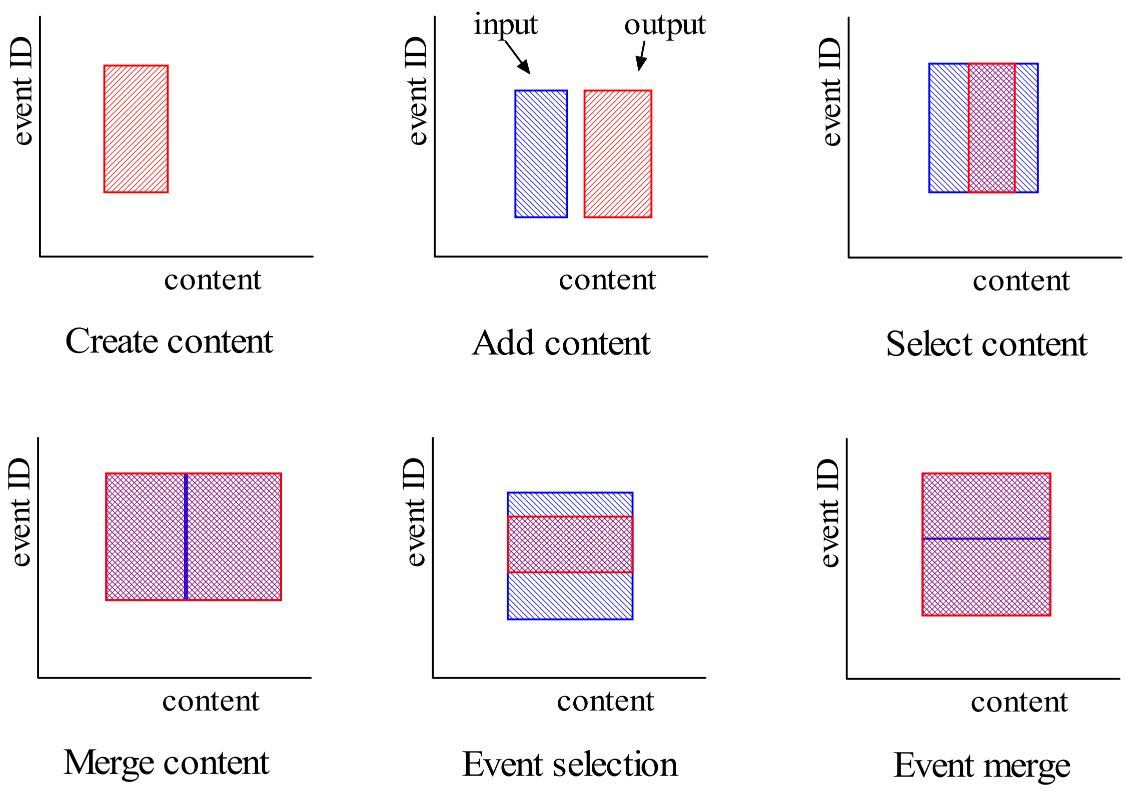


Figure 1. Fundamental dataset transformations. In each figure, the input dataset or datasets are shown in blue and the output in red.

Add content

Adding content is a one-to-one transformation in which each event is processed independently by a sequence of algorithms that use the event view from the input dataset as input. The output dataset includes the same set of event ID's and the content for each event is the collection of newly created EDO's for that event.

Select content

Selecting content is another one-to-one transformation in which the output dataset includes the same set of event ID's as the input. The transformation specifies content for the output dataset that is a subset of the content of the input.

Merge content

Merging content is a many-to-one transformation in which all of the input datasets and the output dataset cover the same event range. The view for each output event is obtained by merging the input views for the event. If the same content ID appears in two input views, we require that both refer to the same EDO. If not, the merge is invalid. The content of the output dataset is the union of the contents of the input datasets.

Arbitrary merging of content can result in a dataset with inconsistent views, i.e. with event views containing EDO's that are not derived from a common history. We may want to restrict merging to forbid this.

Event selection

Event selection is a one-to-one transformation in which a subset of the events from the input dataset are used to construct the output dataset. The input and output datasets have the same content and the view presented by any event in the output dataset is exactly the same as the view for the corresponding event in the input dataset. Typically event selection is performed by running a series of selection algorithms to determine which events are accepted.

Event merge

Merging events is a many-to-one transformation that creates an output dataset whose event views are the union of those in the input datasets. All input datasets and the output dataset have the same content. If an event ID appears in two input datasets, then the corresponding event views must be identical, i.e. reference the same set of EDO's.

Composite datasets

The specification of the data (EDO's) in a dataset is proportional to the number of events and will be very large for the event samples acquired in the ATLAS experiment. We expect many different datasets based on different event and content selections but we also expect there will be considerable overlap between these datasets.

Of the transformations described above, only one (add content) creates new EDO's. The datasets produced by the other four transformations can only include EDO's that appear in their input dataset(s). Two of the remaining (select content and event selection) select a subset of the data in an existing dataset and are included as part of the dataset interface. The remaining two are composite datasets that reference multiple constituent datasets and include instructions for constructing the composite from these. Here we describe these compositions.

Content merged dataset

A content merged dataset includes two or more constituent datasets with the same event range. The view presented for each event is the union of those for each of the views of that event in the constituent datasets. This describes the result of a merge content transformation. The constituent datasets are subject to the same restrictions as the input datasets for that transformation.

Event merged dataset

An event merged dataset includes two or more constituent datasets with the same content. The event list is the union of those from the constituent datasets. This describes the result of an event merge transformation. The constituent datasets are subject to the same restrictions as the input datasets for that transformation.

Virtual Data

Here we briefly discuss the application of a virtual data system as envisioned by GriPhyN (see Technical Report GriPhyN-2002-02 at www.griphyn.org). Unlike the GriPhyN studies to date, we identify the dataset rather than the file as unit of data. The fundamental transformations are described above and an actual transformation will likely be a compound transformation expressed as a DAG combining these.

ADB event collection

The architecture for ATLAS event database is spelled out in the ADB document (copy available at http://www.usatlas.bnl.gov/~dladams/hybrid). The ADB envisions an event collection whose purpose and interface is very similar to the dataset described here. An event collection is composed of event headers that provide access to event data. This sounds very much like a possible implementation for the dataset described here. However it is not clear that other implementations such as the composite datasets described above or the HES file dataset described below would be supported.

HES dataset

The BNL HES (hybrid event store) design (also available at http://www.usatlas.bnl.gov/~dladams/hybrid) assumes the existence of datasets similar to those described here. It assumes that event data is stored in files and defines an interface for these files that includes that described for datasets here. We can define a dataset implementation called HES file dataset that holds a single HES file (by logical name). If this file holds data by reference instead of value, then we have something that looks very similar to the ADB model.

The HES document envisions multiple input streams combined with user configuration to define the input event view. This functionality is included in the dataset described here and we propose leaving it there and using a single unadorned dataset to define the input event view.

One advantage of the HES file dataset is that it can provide direct access to the data if the included file holds EDO's by value instead of reference.

Example

Figure 2 provides an example using most of the transformations described above. Each box represents a dataset transformation and the lines indicate how the dataset produced by one transformation is used an input for the next. The flow is from top down. The input to this DAG (directed acyclic graph) is a collection of raw data files.

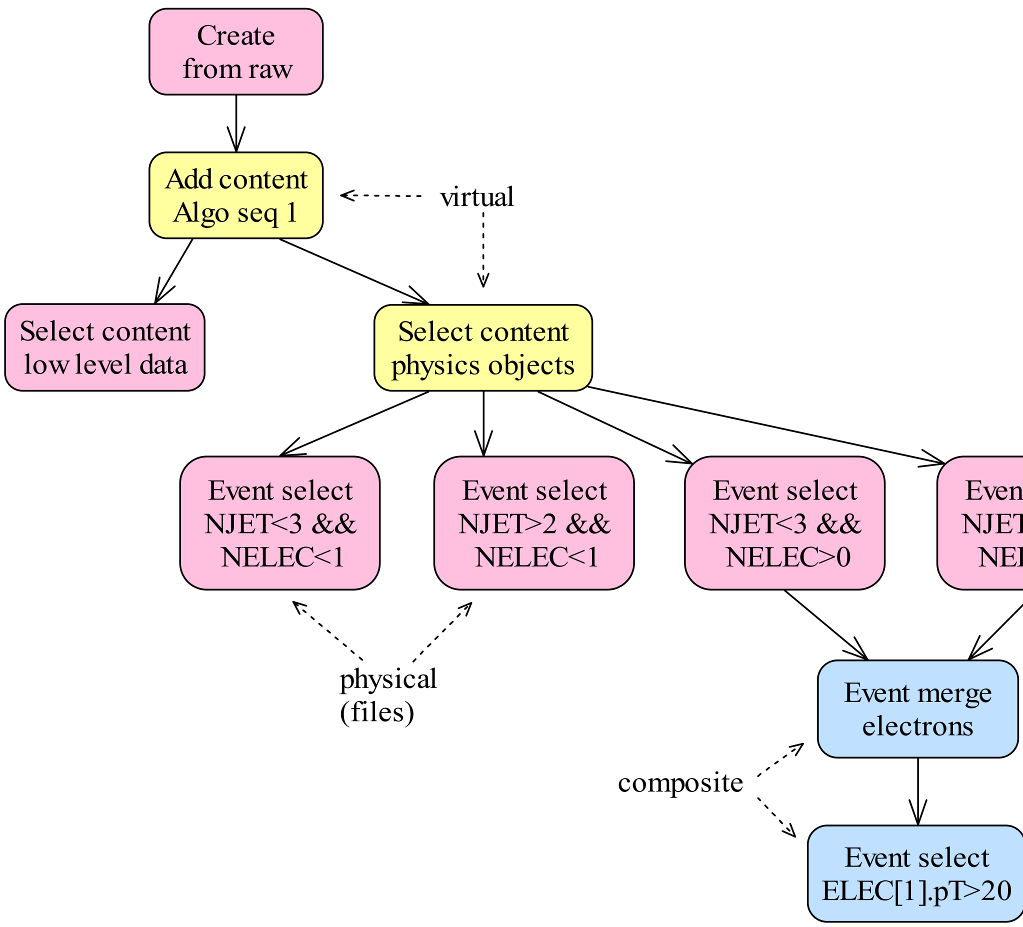
The colors denote the nature of the dataset produced by each transformation: yellow for virtual, pink for physical (the dataset directly specifies the files holding the data) and blue for composite (the dataset points back to other datasets).

The first box represents construction of a raw dataset from the raw data files. An algorithm sequence is the applied to add content including clusters, tracks, jets and high level physics objects (electrons, taus, …). Content filters are then used to separate the new data so that the low-level data is written to one physical dataset and the high-level physics objects are directed to a separate virtual dataset. Event filters are then used to divide the latter into four exclusive physical datasets. Each of these references a different collection of files enabling users to access the data from one without having to access the files in the other datasets.

We imagine that actual production would define a larger number of exclusive streams retaining the notion that the full dataset can be recovered by merging these streams. It might also replicate a small fraction of the data in smaller streams. This can also be done in a later step taking the exclusive streams as input.

The next transformation merges the two datasets containing electrons to provide a dataset holding all events with electrons. The new dataset is a composite: it points pack to the input datasets rather than directly referencing the files.

The final transformation uses the previous as input and selects events with an electron above a transverse momentum threshold. Again it points back to other datasets rather than directly referencing files. It must also carry an event ID list to indicate which events in those files are included. We might have chosen instead to replicate those events and make this a physical dataset.



Example of dataset transformation DAG.