

Physics Analysis Tools - Overview

Kétévi A. Assamagan

Physics Analysis Tools (PAT)

- Mailing list:
atlas-phys-analysis-tools@cern.ch
- Web page and documentation:
<https://uimon.cern.ch/twiki/bin/view/Atlas/PhysicsAnalysisTools>
- Coordination: **Ketevi A. Assamagan**
 - Analysis Model: **Amir Farbin**
 - Connection to Distributed Analysis: **Tadashi Maeno**
 - Representation on the Event Management Board (EMB): **Kyle Cranmer**
 - The PAT coordinator reports to the **Physics Coordinator and the ESRAT Coordinator**
- Meetings
 - Working group meetings a every 2 weeks
 - Meetings on Analysis Model discussion/Meetings about the rest of PAT activities
 - Working meetings at software weeks
 - Physics Analysis Tools workshops - see the workshop summary reports on the PAT page
 - The UCL workshop - 2004
 - The Tucson workshop - 2005

PAT Mandate

➤ Objectives:

- ✓ Propose a unified, baseline framework for Analysis - extend the RTF work to the analysis domain
 - Common classes in the analysis domain
 - Tools to build these objects
 - General/common tools for analysis
 - Navigations and Associations
 - Tag based event selection
 - Interactive tools, Event Display & Visualization
 - Concrete, thorough analysis examples
 - Common look and feel to the user
- ✓ Analysis Model
- ✓ Interface to distributed analysis efforts
- ✓ Interactions with combined performance groups & physics groups
- ✓ Interaction with the user

Analysis Data objects

➤ ESD - Event Summary Data

- ✓ As an output of the standard reconstruction on the raw data
- ✓ Contains reconstruction output data, at the level of hits on track and calo cells
- ✓ Calibration, alignment studies, refitting tracks, track extrapolation, analyses
- ✓ Available only at Tier0 and Tier 1 centers
- ✓ Projected size: 500 kb/events, although ~700 kb/event today
- ✓ Production of AOD, event Tags, customized NTuples
- ✓ [Contents of the ESD as of 11.0.3](#)

Analysis Data objects

➤ AOD - Analysis Object Data

- ✓ As an output of the reconstruction on the ESD
- ✓ Contains reconstruction output data, at the level TrackParticles and CaloClusters, very loosely identified particles
- ✓ Analysis data objects - contents can be customized by users/groups
- ✓ Available at Tier0, Tier 1 and Tier 2 centers
- ✓ Back Navigation to ESD expected to work in practice
- ✓ Projected size: 100 kb/event, although ~170 kb /event today
- ✓ Production of event Tags, customized NTuples, Event Views
- ✓ Data access speed today not adequate for analysis directly on AOD - some improvement expected
- ✓ **From the release 11.1.0, Fast and Full simulation AOD saved in the same file according the recommendations of the MC Truth Task Force**
- ✓ [Contents of the AOD as of 11.0.3](#)

Event Tag & Tag Databases

- ATLAS tag database is intended to be an event-level metadata system
- Tags are intended to support efficient identification and selection of events of interest for a given analysis
 - Expectation is that, in practice, first-level cuts will be possible (and could greatly reduce candidate sample sizes) via queries that are supportable in relational databases
- Nominal space budget in Computing TDR is 1 KB/event
- Tags must be producible from AOD, though tag databases contain or are linked to sufficient navigational information to allow retrieval of event data at all production processing stages

Event Tag Production

- Current Tier 0 model is that tags will be written into local ROOT files as AOD are written (or as AOD files are merged), and later loaded into a master database at CERN
- Tag databases are exported to all Tier 1 centers, and eventually to all Tier 2s
- Tags and tag infrastructure are strongly related to the streaming model: when events are written exactly once, for example, the several groups that may have an interest in the event either add a reference to the event to their group-specific event lists, along with associated metadata (tags), or flag the event as one of theirs in a master event tag
 - Note: Model allows for the *possibility* that different physics working groups may define their own custom tags in addition to an experiment-wide common tag
 - This option has not been explored to date
- The content of the event tag is still being defined for the initial running and for the steady state operation - [see the event tag working group page](#)

Example of Tag Based Event Selection

1. **To count the number of events in a given category:** For example, in some trigger studies, the **SQL query: `select count(*) from rome_003038_merge_J5_Pt_280_560_AOD_tags where (PJetPt1 > 400000 and PJetPt2 > 350000)`**; gives the number of events in the J5 sample that overlap both the 400 GeV single jet and 350 GeV dijet thresholds. This takes about a second and a half to run directly with the database
2. **To select events for Athena analysis:** For example, adding the following to the JobOptions file

EventSelector.Connection

= '<mysql://reader@lxfs6021.cern.ch/CollectionsRome>'

EventSelector.CollectionType = 'ExplicitMySQLIt'

EventSelector.Query = "(EventNumber BETWEEN 21801 AND 21850)

AND (EventNumber <> 21814) AND (PJetPt1 > 60000)" selects only events that have a 60 GeV jet in them, in a particular event range. The **EventNumber <> 21814** bypasses a "killer event" that crashes the code, and this provides a simple way to bypass it.

3. Other queries

- "highest ET jet in the sample",
- "most missing ET in the sample",
- "poorest balance between events with exactly two jets" (or the five or ten events with the worst balance) and so on.

4. The time it takes to run these queries:

- Less than the Athena initialization times.
- The time to run over the sample is of course much faster when one is picking just a few events out of 10000 than when one reads all 10000.

T. LeCompte

Athena-Aware NTuple (AAN)

- Athena-Aware in the sense that you can use this NTuple as an input to Athena jobs (note that you cannot do this with an ordinary NTuple)
 - To pre-select events
 - To navigate back to the AOD and ESD to access the full event
 - The AAN acts as the user's customized NTuple and also as the user's own tag for event selection
 - The AAN is the recommended way today to create customized NTuples
 - See the [UserAnalysis](#) package for an example of how to produce AAN

AAN - Pre-selection

- ROOT can read AttributeList and branches, but cannot reach POOL
- Athena knows AttributeList
- Could be

```
query = ""
```

```
f = ROOT.open('AANTuple.root')
```

```
for evt in f:
```

```
    if evt.branchX > 0:
```

```
        query += " EventNumber=%d OR" %  
                evt.EventNumber
```

```
EventSelector.InputCollection=['AANTuple']
```

```
EventSelector.CollectionType="ExplicitROOT"
```

```
EventSelector.Query=query
```

Event View

- Overlaps: for example, a given TrackParticle may have been used in:
 - The electron/photon reconstruction, cluster-Track association
 - Combined muon reconstruction, muon spectrometer track
 - Inner Detector Track association/combo
 - Hadronic tauJet reconstruction: one / 3-prong
- Some overlaps removed during AOD making but not all
- Event View
 - Overlap removal - analysis dependent / configurable
 - UserData, book-keeping of analysis flow/results and more. See the talk by K. Cranmer

User Interface to Distributed Analysis

- PAT interests in Distributed Analysis
 - Understand the requirements of Distributed Analysis Tools on user application
 - Provide required Physics Analysis Tools toward a common user interface to Distributed Analysis. Perhaps a unique interface cannot be achieved but not interested in a proliferation either.
- What is out there
 - DIAL (Distributed Interactive Analysis on large Datasets) as common interface to back-end systems (local batch, Panda, DAPS, etc)
 - GANGA (Gaudi-Athena And Grid Alliance) as a common framework with wrappers around back-end systems
 - Back-end systems with there own user interfaces, i.e., tools to make Panda usable for analysis in Athena
- For the long term: DIAL as the interface to Panda (Production and Distributed Analysis), integration of DIAL into Panda

User Interface to Distributed Analysis

- DIAL examples and tutorials by D. Adams, see <http://agenda.cern.ch/fullAgenda.php?ida=a056739>
- Tutorial on GANGA by D. Liko, see <http://agenda.cern.ch/fullAgenda.php?ida=a055708>
- Distributed Analysis on Panda - how to submit user analysis job on the OSG production system - T. Maeno:
<https://uimon.cern.ch/twiki/bin/view/Atlas/DaonPanda>
 - User are invited to start experimenting with this and provide feedback

Analysis Model-PAT view point ...

- (1) Histogram manipulation (operations and fitting) at the ROOT/PAW/ATHENA prompt - there is not much for PAT to do here.
- (2) LCG Minuit Example. PAT needs to provide the whole AIDA interface as long as fitting is concerned and this needs to be integrated into ATHENA. - S. Binet
- (3) Interface to external packages which calculate things like exclusion plots for a given model provided a set of measurements - this may be done (in order of preference) through python bindings, C++ wrappers for ROOT, or standardize input/output formats.
- (4) Multi-variable analyses - Neural Networks, Fisher discriminant, etc. A general Multi-Variate Analysis (MVA) package, which provides a ROOT-based tool for discrimination analysis. K. Voss & A. Hoecker
- (5) More advanced Maximum Likelihood fitting. A generalized framework for defining the statistics problems. So a user has objects which represent their data, the observables, parameters, hypothesis, model, PDFs, etc... **RooFit provides much of the engine:** RooFit provides excellent implementation some of the core objects for these statistics concepts. Commonalities w/ (4) may lead to integrated tools.

In PAT, some discussion and work have started on (4) and (5)

Multi-Variate Analysis Tools (K. Voss & A. Hoecker)

- Work in progress, details presented at the PAT session of the Orsay software week : <http://sourceforge.net/projects/mva>
- There are 2 phases to these tools
 - The training phase: optimize the separation of signal and backgrounds using user-defined discriminant variable on well defined data samples
 - Apply the pattern recognition obtained in training phase to some unknown data
- The discriminant method includes cut optimization, likelihood analysis, Fisher discriminants and 2 neural network methods
- The input data is ROOT Tuples or ascii. The training produces output in user defined formats to be fed into the analysis in ATHENA or otherwise
- Some demonstrations already exist:
 - Calorimeter cluster classification for local hadronic calibration
 - Gamma-jet separation in $H \rightarrow \gamma\gamma$ analysis
 - Electron-jet separation of HLT studies
 - Details of how to package this for public consumption to be carried out during PAT meetings - to be installed as an external package.

Multi-Variate Analysis Tools (K. Voss & A. Hoecker)

- MVA is a framework package that can host any MVA tool a user wants to use. If he/she doesn't like the ones we already provide, he can just send us another one, and we'll add it to the factory. What we do is primarily setting a factory framework that allows the user to apply all methods in parallel, in a technically very easy way.
- The MVA factory allows us to make detailed comparison plots (and numerical estimates) to assess the performance of each method in exactly the same environment. It is hence easy for the user to choose the best - and - simplest one.

More Advanced ML Fitting

- HEP experiments perform complicated fits of 10's of observables for 100K's events extracting 100's of parameters/measurements.
- This ultimately necessitates
 - highly optimized core tools- fits can take days
 - scripting environment- building, configuring, bookkeeping fits requires good organization
 - Validation tools such as toy Monte Carlos, plots, etc.
- RooFit Features
 - provides implementation of variables, PDFs, datasets, etc which handle complexities such as "caching"
 - Library of PDFs
 - some plotting and toy Monte Carlo features
 - access through C++ (compiled or interpreted) or python
- We get all of this for free because RooFit is included in ROOT5

ML Fitting: Generalized Approach

Amir Farbin

- Writing a fit in RooFit requires writing C++ which instantiates RooFit objects, puts them together to form a “model”, reads configuration/data files, runs the fit, extracts results, makes plots, etc...
 - ➔ User must write complicated and tedious code
 - ➔ Ex: BaBar created multiple analysis specific fit packages using RooFit
- More generalized approach (Ex: MLFit from BaBar):
 - User defines input data format, the observables, hypothesis (ie signal/bkg), and model (ie what PDF for each observable/hypothesis).
 - MLFit builds fit by dynamically instantiating RooFit objects using a PDF factory and writes template configuration file for parameters/measurements.
 - Standardized representation of concepts such as observable or hypothesis allows writing generalized tools which calculated upper limits, do likelihood scans, do hypothesis testing, calculate goodness of fit, and make projection plots, sPlots, or exclusion plots rather than requiring each fit to provide its own implementation

I4Momentum Interface

- Section 3.2 of the RTF recommendations for details and look in the CVS repository under Event/FourMom for the implementation
- An interface to provide 4-momentum accessors, enforcing a uniform access to kinematic variables
- Uniform naming, common « look and feel »
 - ✓ Less code to write, less documentation to read
 - ✓ Avoid bugs:
 - o $P_x = \cos(\text{phiVert}) / \text{PtInvVert}$
 - o → wrong !
 - o $E_T = E * \sin(2 * \text{atan}(\exp(-\text{eta})))$
 - o → correct but three times slower than $E / \cosh(\text{eta})$;

I4Momentum Interface

➤ I4Momentum already implements some tools: `pt()`, `eta()`, `phi()`, `e()`, `m()`, etc

✓ The computation of the Z 4 momentum is now (done by Rousseau):

```
m_hlvZ=m_egamma[0]->hlv()+m_egamma[1]->hlv();
```

✓ whereas it used to be:

```
HepLorentzVector hlv[2];  
const LArCluster* clus=0;  
double eta,theta,px,py,pz,en;    // ..... 2 electrons  
for ( int i=0 ; i<2 ; ++i )  
{  
    clus = m_egamma[i]->get_Cluster();  
    eta = (clus)->eta();  
    theta = 2. * atan(exp(-eta));  
    px = (clus)->et() * sin((clus)->phi());  
    py = (clus)->et() * cos((clus)->phi());  
    pz = (clus)->et() / tan(theta);  
    en = (clus)->et() / sin(theta);  
    hlv[i].setPx(px);  
    hlv[i].setPy(py);  
    hlv[i].setPz(pz);  
    hlv[i].setE(en);  
}  
m_hlvZ=hlv[0]+hlv[1];
```

Concise & cleaner code with I4Momentum

Object Navigation

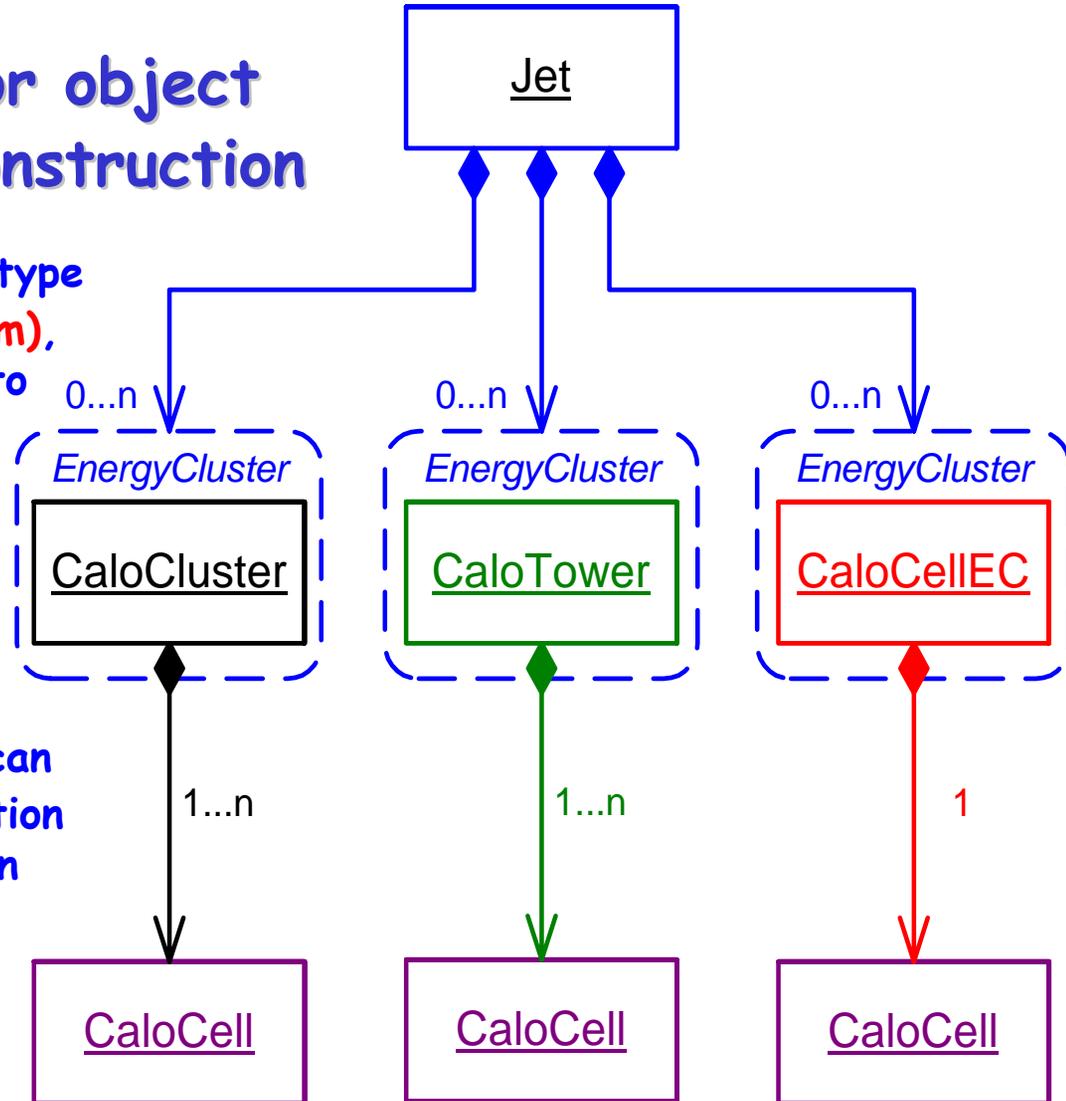
 Original motivation for object navigation from jet reconstruction

 **Jet** constituents are of generic type **EnergyCluster (INavigable4Momentum)**, their concrete type is not exposed to the **Jet** itself;

 clients need to retrieve objects of specific concrete type at any node of the relational tree behind a **Jet** -> need a navigation system;

 constituent objects in the tree can be composites themselves -> navigation must be possible to any given level in the tree;

 constituent objects can contribute their kinematics with a weight to the composite object -> weights must be retrievable and propagated correctly;



Object Associations

- Non constituent associations, for example
 - ✓ Jet \leftrightarrow TrackParticle Association: a muon is associated to a jet but the muon does not belong to the jet
- Constituent associations are handled by Navigation
- User associations of **anything to anything** in the analysis domain - implemented as:
 - ✓ AssociationLink : 1-to-1
 - ✓ AssociationVector : 1-to-many
 - ✓ AssociationMap : many-to-many
- Examples
 - ✓ Overlap associations at the cell and hit on track levels can only be one at the ESD. The results need to be propagated to the AOD to be able to check for overlaps down to cell and hit level even on the AOD.

Conclusions

- Please try the Athena-Aware NTuple and give us feedback
- Please try the distributed analysis tools and give us feedback
- Please try the Event View and give us feedback
- There is still a lot of work ahead for PAT
 - Proposal for a model for analysis
 - Interface to distributed analysis
 - AOD data access speed issues
 - Overlap checking tools
 - Etc
- The number of PAT developers is very thin. We need you to get involved
 - To contribute to the development of tools
 - To test existing tools and give feedback