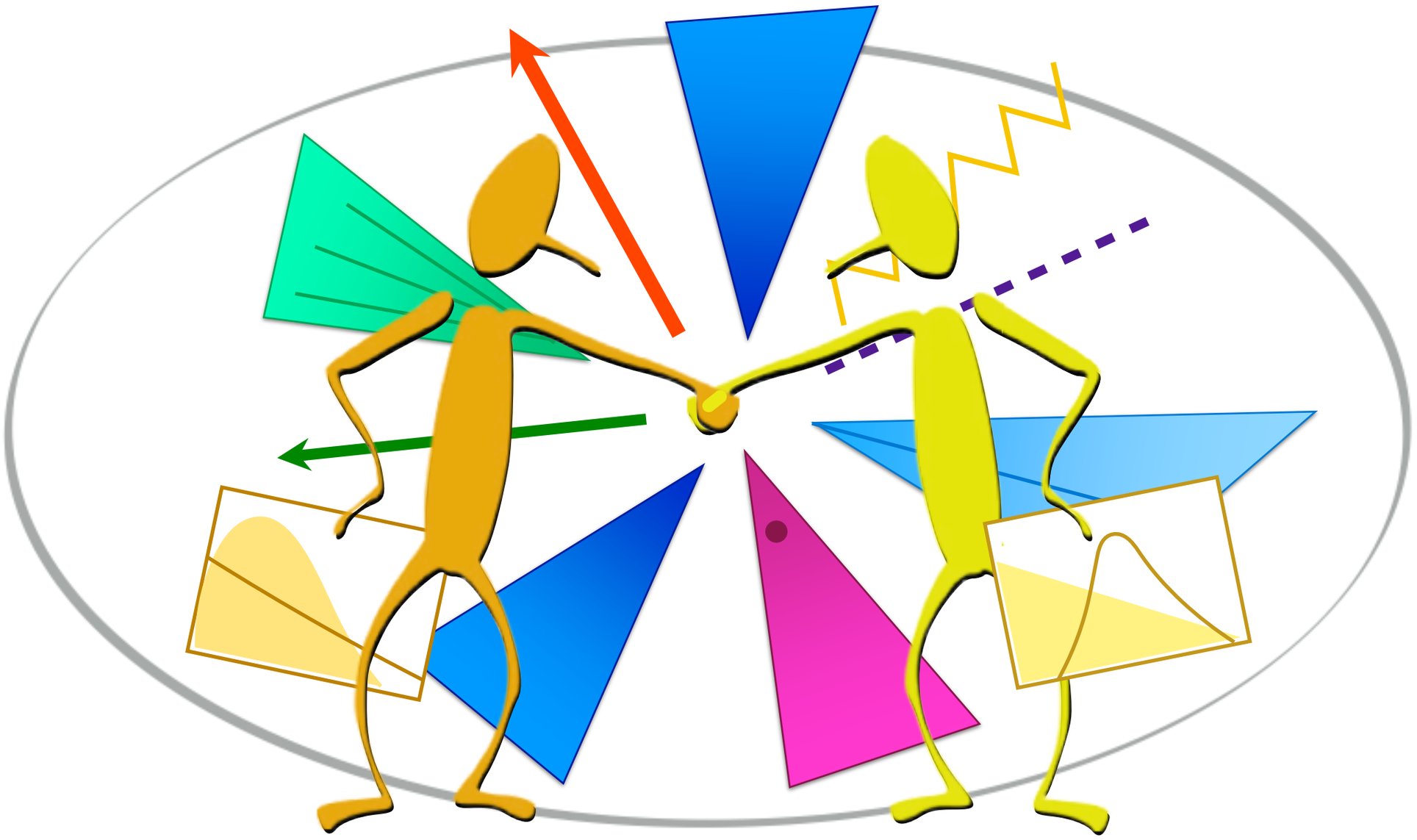


An **A**nalysis **D**escription **L**anguage and **adl2tnm**

Harrison B. Prosper (FSU)

ADL4LHC Workshop, 6 May 2019

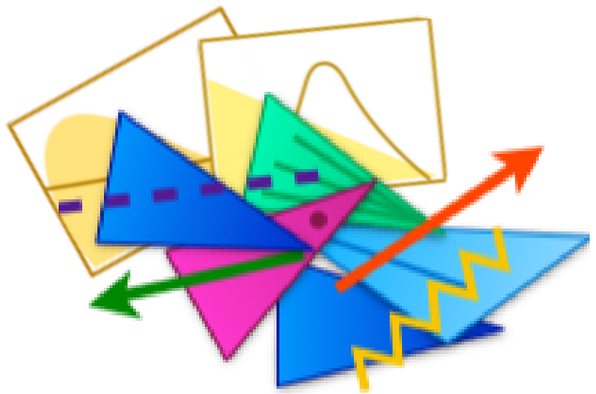
Properties of an ADL for the LHC

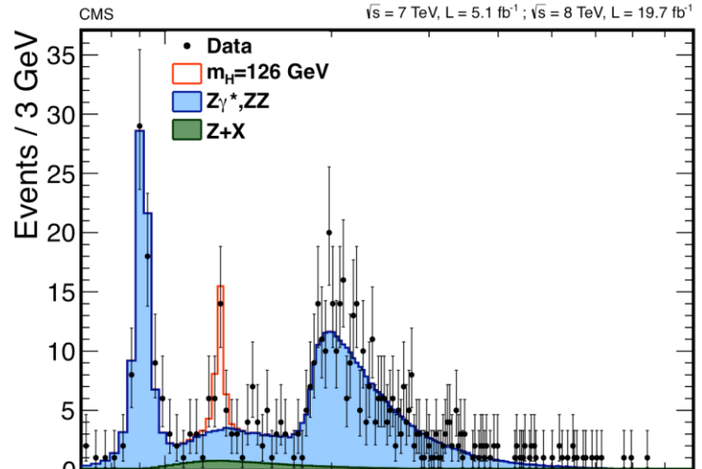
Basic requirements:

- Public
- Complete
- Easily learned
- Demonstrably correct

Desirable features:

- Self-contained
- General programing language-independent
- Analysis framework-independent





A Proposal for a **L**es **H**ouches **A**nalysis **D**escription **A**ccord

D. Barducci, G. Chalons, N. Desai, N. de Filippis, P. Gras, S. Kraml, S. Kulkarni, U. Laa, M. Papucci, H. B. Prosper, K. Sakurai, D. Schmeier, S. Sekmen, D. Sengupta, J. Sonneveld, J. Tattersall, G. Unel, W. Waltenberger, A. Weiler.

Abstract: We present the first draft of a proposal for “a Les Houches Analysis Description Accord" for LHC analyses, a formalism that is capable of describing the contents of an analysis in a standard and unambiguous way independent of any computing framework. This proposal serves as a starting point for discussions among LHC physicists towards an actual analysis description accord for use by the LHC community.

LH 2015 New Phys WG report (arXiv:1605.02684)

Subsequent Developments - I

- A preliminary ADL proposal developed during & after LH2015.
- CERN workshop on Nov 2016 to refine ADL syntax. https://indico.cern.ch/event/572170/overview.

- Several *proofs of principle* developed
  - Approach: **transpiler** (convert ADL to executable C++)
    - lhada2rivet (Philippe Gras)
    - adl2tnm (Harrison Prosper)
    - lhada2checkmate (Daniel Dercks)

- Approach: **interpreter** (runtime interpretation)
  - CutLang (Gökhan Ünel) based on ADL principles, but with different syntax.

Subsequent Developments - II

- Several ATLAS & CMS published analyses implemented and tested and results compared using lhada2rivet, adl2tnm and CutLang.
  - Analysis examples taken from a LH17 (re)interpretation tools comparison summary. Used Delphes samples / data formats.
  - Short note in LH2017 proceedings. https://arxiv.org/abs/1803.10379
  - Results presented at the 4th (Re)interpretation workshop at CERN. https://indico.cern.ch/event/702612/

Subsequent Developments - II

- lda2tnm and CutLang development continuing.
  - CutLang publication https://arxiv.org/abs/1801.05727
  - CERN summer student for CutLang. lex/yacc-based parsing.

  - adl2tnm adapted to handle CMS nanoAOD, but needs a major re-write to render more robust.

Introducing the ADL - I

The first ADL proposal adhered to the following principles:

- human readability
- computer language independence
- analysis framework independence, **but note…**

the ADL is analysis framework independent precisely so that it offers a standard input to analysis frameworks, rather like an SLHA text file offers a standard input to SUSY calculators.

We should accept the healthy reality that a physicist will work with whatever analysis framework she or he judges to be the easiest to work with. And note that by the 2040s, today's favored frameworks may be considered obsolete; but that should not be the fate of the ADL.

Introducing the ADL - II

The proposed ADL consists of

- a plain text file describing the analysis using a HEP specific language with a syntax rules that include standard mathematical and boolean operations and 4-vector algebra.

- a library of self-contained functions encapsulating variables that are non-trivial to express within the ADL.

Introducing the ADL - III

The LHADA ADL comprises blocks with a key value structure.

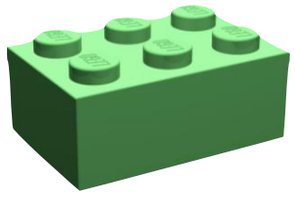
```
blocktype blockname
   # comment
   key1 value1
   key2 value2
   key3 value3 # comment
```

- There is a clear separation of analysis components.
- The same key can appear multiple times.
- The current version includes:
  - Main block types: **info** **object** **define** **region** **table**
  - Main keywords: **select reject take**

Blocks: `info`

Provides general information about the analysis, e.g. publication information, benchmark scenarios, event generators used, validation material, etc.

```
info analysis
  experiment     ATLAS
  id             EXOT-2016-32
  publication    Eur.Phys.J. C77 (2017) no.6, 393
  sqrtS          13.0 # TeV
  lumi           36.1 # 1/fb
  arXiv          1704.03848
  hepdata        https://www.hepdata.net/record/ins1591328
  doi            10.1140/epjc/s10052-017-4965-8
```



Blocks: `object`

```
object muonsVeto
  take Muon
  select pt > 5
  select |eta| < 2.4
  select softId == 1
  select miniPFRelIso_all < 0.2
  select |dxy| < 0.2
  select |dz| < 0.5

# jets - no photon
object AK4jetsNopho
  take AK4jets j
  reject dR(j, photons) < 0.4 and
         photons.pt/j.pt [] 0.5 2.0
```

Blocks: `define and region`

Implements selection criteria that define analysis regions, e.g. signal, control regions.

- A region can be considered a boolean condition in another block

```
define MR      = fMR(megajets)
define METl    = met + leptonsVeto[0]
define Rsql    = sqrt(fMTR(megajets, METl) / MR)
define Rsq     = sqrt(fMTR(megajets, met) / MR)

# Boost pre-selection cuts
region preselection
   select AK4jets.size >= 3
   select AK8jets.size >= 1
   select MR > 800
   select Rsq > 0.08
```

Blocks: region

```
region WcategoryCRW
    select preselection
    select leptonsVeto.size == 1
    select WjetsMasstag.size >= 1
    select dphimegajets < 2.8
    select MT [] 30 100
    select bjetsLoose.size == 0
```

```
region WcategoryCRL
    select AK4jets.size >= 3
    select AK8jets.size >= 1
    select MR > 800
    select leptonsVeto.size == 1
    select Rsql > 0.08
    select WjetsMasstag.size >= 1
    select dphimegajets < 2.8
    select MT [] 30 100
    select bjetsLoose.size == 0
```

Current Projects

Now will give some details of current projects

- **transpilers** (ADL to executable C++)
  - lhada2rivet (Philippe Gras)
  - adl2tnm (Harrison Prosper)
  - lhada2checkmate (Daniel Dercks)

- **interpreters** (runtime interpretation)
  - CutLang (Gökhan Ünel) based on ADL principles, but with different syntax.

lhada2rivet

Philippe Gras

- Rivet BSM analysis support added with version 2.5.2
- A prototype transpiler **lhada2rivet.py** for Rivet. https://github.com/lhada-hep/lhada/tree/master/lhada2rivet.d
- Particles and jets are implemented using Rivet specific truth level objects. Smearing added in RIVET.
- Can be used for phenomenological analyses.
- Associated C++ code validated before the code generation
  - ease debugging by isolating problem in user's code from the possible ones in the generated code
- Needs further testing.

adl2tnm

Harrison Prosper

- A prototype transpiler **adl2tnm.py** converts ADL to C++.

- C++ code can be executed within the TNM (TheNtupleMaker) generic ntuple analysis framework. Only depends on ROOT.

- Assumes that a standard extensible type is available to model all analysis objects. Uses adapters to translate types.

- Can be used for any experimental or phenomenological analyses https://github.com/hbprosper/adl2tnm.

- **Warning**: **adl2tnm.py** does not yet use a formal grammar for the ADL. We have plans to rewrite **adl2tnm.py** using tools such as PLY. The current version of **adl2tnm** is fragile!

CheckMATE

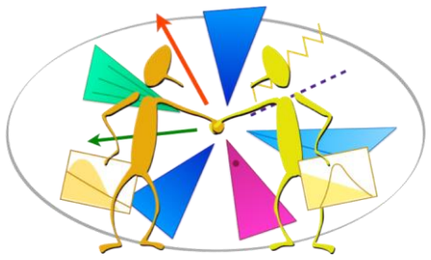
Daniel Dercks, nee Schmeier

- A prototype transpiler from ADL to CheckMATE C++ code.
- Used the ATLAS monojet analysis ATLAS-1604-07773 as an example.
- Works with Delphes objects.
- Tested a simple version of automatic function download, and confirms that function database would be very useful if structured carefully.
- Useful feedback on the block and keyword content.

CutLang

Gökhan Ünel

- CutLang is both an ADL and a runtime interpreter. CutLang adheres to the ADL principles, but since its development preceded the preliminary ADL accord developed at Les Houches, the syntax of CutLang differs in detail from that of the accord.
- Special features added to the CutLang syntax to enhance its runtime interpretation capabilities.
- Works with multiple data formats:
  - LVL0, ATLAS & CMS open data, Delphes, LHCO, FCC. More can be easily added.
- Uses lexx/yacc-based parsing, and is, therefore, quite robust.
- Used in a real ATLAS exotic analysis.
- Documentation: arXiv:1801.05727, https://cutlang.hepforge.org



CERN Analysis Preservation Support Group

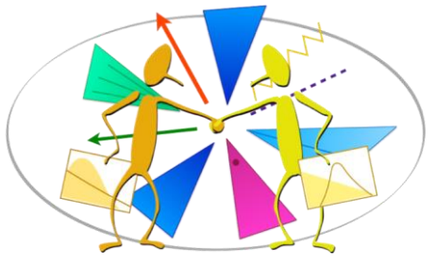
- There is a major ongoing effort to preserve the content of LHC analyses: http://analysis-preservation.cern.ch
- A database to store analysis information is designed and working.
- The CERN Analysis Preservation Support Group is very supportive of the idea of an ADL and state that an ADL can be readily incorporated into the current system, which can store:
  - ADL files
  - Self-contained functions
  - Object definitions

Summary and Plans

- Human readable ADLs have a role in HEP.
- ADLs exist and have been tested on several representative analyses.
  - An ADL based on LHADA and the CutLang ADL are being refined.
  - 3 transpilers and one interpreter targeting different analysis frameworks.
  - **adl2tnm** is to be rewritten using formal parsing tools

- This is an **open** project that very much welcomes contributions.

- We hope the discussions in this workshop will encourage further developments.



Thanks!

Many thanks to Sezen Sekmen whose initiative this is and

to Gökhan Ünel for being the first to show that a workable ADL is feasible.

Also thanks to:

D. Barducci, A. Buckley, G. Chalons, E. Conte, N. Desai, N. de Filippis, B. Fuks, P. Gras, S. Kraml, S. Kulkarni, U. Laa, M. Papucci, C. Pollard, K. Sakurai, D. Schmeier, D. Sengupta, J. Sonneveld, J. Tattersall, W. Waltenberger, A. Weiler.