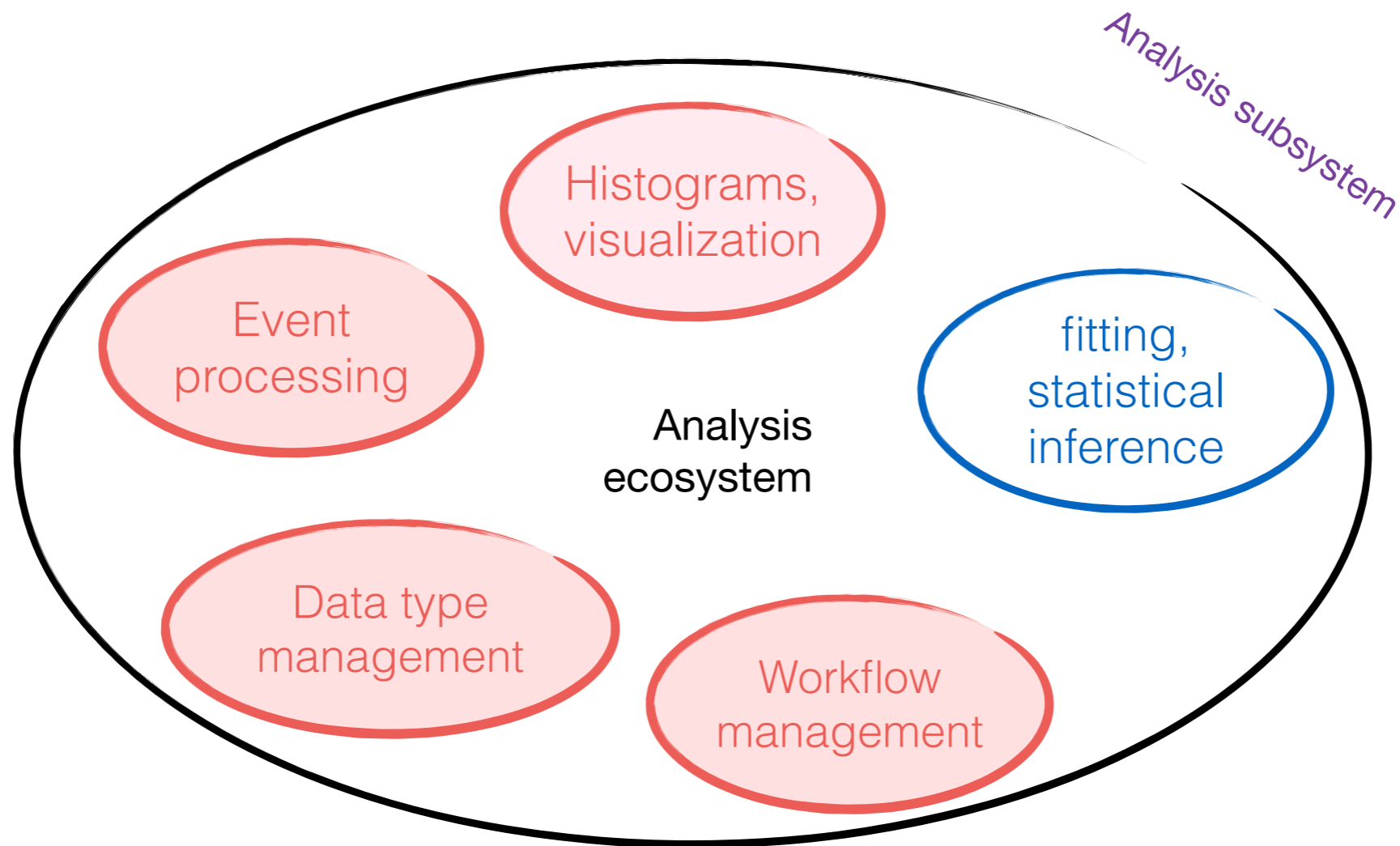# Analysis Description Language for LHC-type analyses

G. Unel (UCI/ATLAS) S.Sekmen (KNU/CMS). H. Prosper (FSU/CMS)

HSF WLCG Virtual Workshop

19-24 November 2020

Analyses in HEP

- Analysis frameworks based on general purpose languages (GPL): A computer language that is broadly applicable across many application domains (C++, Python, …).

What is next?

- **A**nalysis **D**escription **L**anguage for HEP analyses:
  - A domain specific declarative language capable of describing the physics content of an LHC analysis in a standard and unambiguous way.
    - Domain-specific language (DSL): a computer language specialized to a particular application domain (regular expressions, Makefile, SQL, …).
      - Designed to model how domain experts think about specific problems they wish to solve.
    - Declarative language: A language that expresses the logic of a computation without describing its control flow. Describes what needs to be done, but not how to do it.
  - Human readable with the aim of shifting the focus from programming to physics.

- ADL consists of
  - a plain text file describing the analysis using an easy-to-read DSL with clear syntax rules.
  - a library of self-contained functions encapsulating variables that are non-trivial to express with the ADL syntax (e.g. MT2, ML algorithms). Internal or external (user) functions.

LHADA (Les Houches Analysis Description Accord): Les Houches 2015 new physics WG report (arXiv:1605.02684, sec 17)

CutLang: Comput.Phys.Commun. 233 (2018) 215-236 (arXiv:1801.05727), ACAT 2019 proceedings (arXiv:1909.10621)

WHY ADL?

Motivation / use case	Exp	TH/ Pheno	Public
Analysis preservation	✔	✔	✔
Analysis design, implementation	✔	✔	✔
Analysis communication, clarification, synchronization, visualization	✔	✔	✔
Analysis review by referees	✔	✔	
Interpretation studies, analysis reimplementation	✔	✔	✔
Easier comparison/combination of analyses	✔	✔	✔

Currently   cern.ch/adl

- LHADA U CutLang = ADL
  - syntax rules: https://twiki.cern.ch/twiki/bin/view/LHCPhysics/ADL

- Execution is based on ROOT

```
object goodEle : ELE
  select    Pt(ELE)        >    10
  select abs({ELE}Eta )  <    2.4
  select      {ELE}AbsEta ][  1.442 1.556

define goodZreco : goodEle[0] goodEle[1]

algo          preselection
  select        ALL                # to count all events
  select        Size(ELE)    >= 2  # events with 2 or more electrons

algo          testg
  preselection
  select        Size(goodEle) >= 2  # events with 2 or more electrons
  histo         h1mgoodReco, "Z candidate mass (GeV)", 100, 0, 200, {goodZreco}m
  select        {goodZreco}q == 0    # Z is neutral
  histo         h2mgoodReco, "Z candidate mass (GeV)", 100, 0, 200, m(goodZreco)
```

  - Math., Logical, HEP specific functions
  - New variables, objects & user functions
  - Histograms, Tables, efficiencies
  - Reducers, optimizers, etc

- Interpreter & Framework
  - CutLang v2 : https://github.com/unelg/CutLang
  - Multiple input data types: OpenData, NanoAOD, Delphes,...
  - Recently used in ATLAS exotics + FCC pheno analyses

- Transpiler
  - ADL2TNM : https://github.com/hbprosper/adl2tnm

- database for LHC analyses:
  - https://github.com/ADL4HEP/ADLLHCanalyses

Outlook

- ADL concept and tools are actively being developed
  - US: UCI, FSU
  - KR: KNU
  - TR: BOUN, IU
  - CERN via summer students

- The available analysis database is growing

- The interpreter is ready for public testing

- Planning for
  - Systematic error evaluation
  - Statistical tool integration

Thank you for your attention