Contribution ID: **559** Type: **Oral**

Progress in Analysis Preservation with a Focus on Reinterpretation

Thursday, October 13, 2016 3:00 PM (15 minutes)

LHC data analyses consist of workflows that utilize a diverse set of software tools to produce physics results. The different set of tools range from large software frameworks like Gaudi/Athena to single-purpose scripts written by the analysis teams. The analysis steps that lead to a particular physics result are often not reproducible without significant assistance from the original authors. This severely limits the ability to re-execute the original analysis or to re-use its analysis procedures in new contexts: for instance, reinterpreting the results of a search in the context of a new physics model. We will describe two related packages that have been developed to archive analysis code and the corresponding analysis workflow, which enables both re-execution and re-use.

Following the data model of the W3C PROV standard, we express analysis workflows as a collection of *activities* (individual data processing steps) that generate *entities* (data products, such as collections of selected events). An activity is captured as a parametrized executable program in conjunction with its required execution environment. Among various technologies, Docker has been explored most extensively due to its versatility and support both in academic and industry environments. Input parameters are provided in form of JSON objects, while output data is written to storage that is separate from the execution environment and shared among all activities of the workflow. Further, each activity publishes JSON data, in order to allow for semantic access to its outputs.

The workflow itself is modeled as a directed acyclic graph (DAG) with nodes representing activities and directed edges denoting dependencies between two activities. Frequently, the complete graph structure and activity parameters are not known until execution time. Therefore, the workflow graph is iteratively built during run-time by a sequence of graph extensions, that collectively represent a *workflow template*. These extensions, also referred to as stages, schedule new nodes and edges as soon as the required information is available. As the dependency structure of the activities is fully captured in the DAG, mutually independent activities can be computed in parallel, distributed across multiple computing resources, e.g. using container orchestration tools such as Docker Swarm or Kubernetes.

Both activities and workflow stages descriptions are defined using an extensible JSON schema that allows for composition and re-use of both individual activities as well as partial analysis workflows across separate analyses. Finally, it enables us to store and richly query, inspect and present workflow information in the context of analysis archives such as the CERN Analysis Preservation Portal.

Tertiary Keyword (Optional)

Secondary Keyword (Optional)

Data processing workflows and frameworks/pipelines

Primary Keyword (Mandatory)

Preservation of analysis and data

Primary authors: CRANMER, Kyle Stuart (New York University (US)); HEINRICH, Lukas Alexander (New York University (US))

Presenter: HEINRICH, Lukas Alexander (New York University (US))

Session Classification: Track 8: Security, Policy and Outreach

Track Classification: Track 8: Security, Policy and Outreach