https://twiki.cern.ch/twiki/bin/view/Geant4/StatTest

Statistical Testing Tool

Status and Plans

Andrea Dotti (adotti@slac.stanford.edu) ; SD/EPP/Computing

Geant4 21st Collaboration Meeting – Ferrara, 12-16 September 2016

U.S. DEPARTMENT OF ENERGY
Office of Science

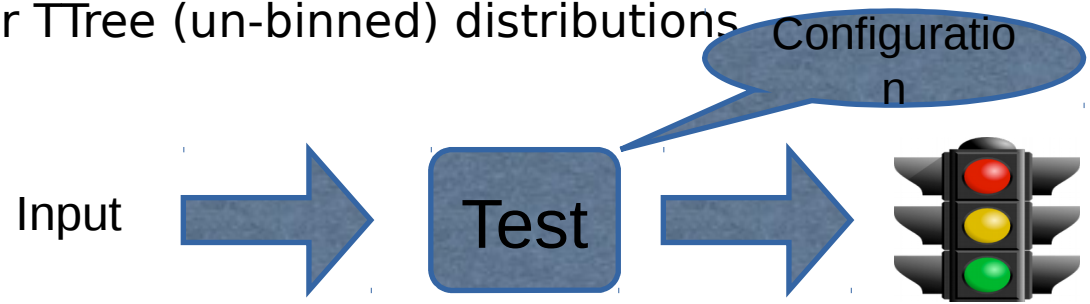
SLAC NATIONAL ACCELERATOR LABORATORY

Requirements from CDash/CTest integration (2012)

- Current Nightly and CDash system:
  - Code stability oriented
  - Success defined as: Run to completion, stderr empty
- Investigate automatic extension to include some physics regression testing
  - Few physics oriented test exists, not unified, interpretation of results and automatization missing
- Geant4 has “unified” analysis approach: g4analysis
- Requirements:
  - NFR: **Ease to work with**: developers should do minimal work. Ideally in simple cases none
  - NFR: **Ease to interpret results**: a “shift” system could be put in place. Results should be interpreted by non-experts
  - FR: Statistical tests to support: **compare to reference, compare to theory**
  - FR: Both **binned** and **un-binned** distributions
  - FR: System should **provide summary or verbose output**

Basic Idea

- Derived from my experience in ATLAS' DQ
- Basic idea: an "DQ Algorithm" is a combination of:
  - **An input** (e.g. an Histogram) with simulation results
  - **A Test** (e.g. implementation of Kolmogorov test)
  - **An Output**: a simple flag based on Test output
  - (ATLAS "Region" concept, i.e. group of tests belonging to a detector with "algebra" on output not implemented for G4)
- A simple way to configure define an set of Algorithms is provided
- Pre-requisite: G4 application should provide ROOT file with TH1 (binned) or TTree (un-binned) distributions



Define Result

SLAC

- An Output is defined as a three-state objects:
  - Two **thresholds** (low, high) are specified and compared to the output of the test R
  - If **R < low** then FAILED
  - If **low < R < high** then NOTPASSED
  - If **R > high** then SUCCESS

Recent Work from P. Arce (2015/2016)

SLAC

Extension of the system to support parsing of TXT log file and perform NormalCDF test on values extracted from log file

List of available tests

BinnedAndersonDarlingTest	Histograms
AndersonDarlingTest	Ntuple column
Binned1DChi2Test	Histograms
BinnedWeighted1DChi2Test	Histograms with w/ != 1.
KolmogorovSmirnovTest	Ntuple column
BinnedKolmogorovSmirnovTest	Histograms
NormalCDF	Text log files

What's Next

The utility was developed specifically for ctest/cdash integration, but to be used also in standalone mode

- Actually some CMS guy asked me to put it on github for public use
- I would like to keep it as-it-is with minimal improvements tailored to this use-case

Available to anybody that wants to perform statistical testing between two versions of Geant4 (see twiki and included examples for instructions)

But: we have started to discuss the possibility of using it in the context of DoSSiER for automatic statistical testing for web-presentation

Issues and Challenges

SLAC

DoSSiER integration:

- the application is a python script relying on ROOT files => not suitable for DoSSiER as it is
- TGraphs are missing (Julia has a ROOT script, I'll take a look at it) => TGraphs are very common in DoSSiER
- We probably do not need all the different algorithms, for web-presentation one is probably enough (let's say Chi2) => suggest to re-write the one we need/want in JAVA for DoSSiER

A final remark form my personal experience (from G4 Validation and ATLAS-Tile DQ Leader): these automatic tools never really "just work" (statistical regression testing is tricky), we should never use an automatic tool to reject/approve a tuning or a validation, but instead use it to "guide the eye" of the expert to where something is smelling funny...

- For example, DoSSiER web-page could put a flag when comparing two set of data, but no emphasis or automatic action should be taken