Contribution ID: **14** Type: **not specified**

ROOT and C++11

Wednesday 13 March 2013 17:20 (25 minutes)

ROOT as a platform is positioned at a demanding intersection of requirements: Data acquisition and analysis in modern HENP asks for often highly optimized implementations, while at the same time many typical users of ROOT are non-experts and are interested in getting correct answers to their scientific questions easily.

The C++11 standard updated the core C++ language with a number of paradigms which allow maintaining a user-friendly, high level of abstraction without overly compromising efficiency, e.g. through shifting work to check type constraints or dependencies from run time to compile time (some of which were already proposed in TR1). New C++11 features like anonymous function objects, automatic type deduction and very easy to use abstractions for multithreaded and asynchronous programming put formerly advanced techniques into the reach of non-experts.

This represents both a challenge and an opportunity for ROOT: to reduce the potential for misuse of its libraries by baffled users, and to build on a common language of modern C++ to communicate design decisions both internally and externally.

Starting from typical use cases problems in representative usage patterns will be shown and possible solutions will be outlined with a focus on usability without compromised efficiency.

Author: BANNIER, Benjamin (Department of Physics and Astronomy, Stony Brook University, SUNY, Stony Brook, New York 11794-3400, USA)

Presenter: BANNIER, Benjamin (Department of Physics and Astronomy, Stony Brook University, SUNY, Stony Brook, New York 11794-3400, USA)