Contribution ID: **291** Type: **oral presentation**

Automatic Procedures as Generated Analysis Tool

Thursday 30 September 2004 16:30 (20 minutes)

The photo injector test facility at DESY Zeuthen (PITZ) was built to develop, operate and optimize photo injectors for future free electron lasers and linear colliders. In PITZ we use a DAQ system that stores data as a collection of ROOT files, forming our database for offline analysis. Consequently, the offline analysis will be performed by a ROOT application, written at least partly by the user (a physicist). To help the user to develop safe filters and data visualisation (graphs, histograms) with minimal effort in an existing ROOT framework application, we provide a GUI that generates C++ source files, compiles and links them to the rest of the application. We call these C++ routines “Automatic Procedures” (AP). Standard filter conditions and data visualisation can be generated by click or drag- and-drop, while more complex tasks may be expressed as small pieces of C++ code. Once compiled by ACLiC (ROOTs Automatic Compiler Linker), an Automatic Procedure may be reused without repeated compilation. E. g. the injector shift crew will run a number of ROOT applications, controlled by APs in regular intervals. Alternatively every AP can be read in and loaded to the GUI for further improvement. A number of APs can run in a logical sequence, parameters can be transferred from one AP to an other. They can be selected by picking a point from a graph.
The GUI was constructed with Qt, because that offers a comprehensive GUI programming toolkit.
The photo injector test facility at DESY Zeuthen (PITZ) was built to develop, operate and optimize photo injectors for future free electron lasers and linear colliders. In PITZ we use a DAQ system that stores data as a collection of ROOT files, forming our database for offline analysis. Consequently, the offline analysis will be performed by a ROOT application, written at least partly by the user (a physicist). To help the user to develop safe filters and data visualisation (graphs, histograms) with minimal effort in an existing ROOT framework application, we provide a GUI that generates C++ source files, compiles and links them to the rest of the application. We call these C++ routines “Automatic Procedures” (AP). Standard filter conditions and data visualisation can be generated by click or drag- and-drop, while more complex tasks may be expressed as small pieces of C++ code. Once compiled by ACLiC (ROOTs Automatic Compiler Linker), an Automatic Procedure may be reused without repeated compilation. E. g. the injector shift crew will run a number of ROOT applications, controlled by APs in regular intervals. Alternatively every AP can be read in and loaded to the GUI for further improvement. A number of APs can run in a logical sequence, parameters can be transferred from one AP to an other. They can be selected by picking a point from a graph.

The GUI was constructed with Qt, because that offers a comprehensive GUI programming toolkit.

Keywords: Automatic Procedure, ROOT, ACLiC, Data Analysis, Data Visualisation, GUI, Qt

Authors: ASOVA, G. (DESY ZEUTHEN); ABRAHAMYAN, K. (DESY ZEUTHEN); WINDE, M. (DESY ZEUTHEN); CASTRO-GARCIA, P. (DESY); WEISSE, S. (DESY ZEUTHEN)

Presenter: ASOVA, G. (DESY ZEUTHEN)

Session Classification: Core Software

Track Classification: Track 3 - Core Software