The New Data Acquisition System

C. Broude, R. Lutter, O. Schaile, K. Schöffel, K. Steinberger, and P. G. Thirolf

1. Overview

A development project has been started aiming at a new data acquisition and analysis system to overcome the limitations in performance and reliability of the existing MLLE system at the BL for multi-parameter and/or high rate experiments. The MLLE system will be replaced by a modern hardware and software architecture (POWERPC CPUs instead of Eltec E6, C++ instead of Fortran). While a significantly improved data acquisition performance is mandatory for upcoming complex detector setups (e.g. the MINIBALL Germanium and Silicon detector array with VME and/or VXI based electronics) also "conventional" experiments will benefit from improved performance of CAMAC readout. Along with the implementation at the BL, the new system will also replace the existing data acquisition system of the ISOLDE collaboration at CERN and several member groups of the MINIBALL and ISOLDE collaboration have expressed their interest in adopting the new system.

The new data acquisition system consists of two main parts as shown in fig. 1: a front-end system for data readout, event building, and data transport based on GSI's Multi Branch System (MBS) [1] and a back-end system responsible for setup, run control, histogramming, and data analysis written within the ROOT framework [2].

2. Front-end system

We are implementing two different MBS hardware configurations: a single-branch, pure CAMAC configuration based on a CVC processor (Struck, Motorola 68040) running under LYNXOS V2.2 for small experiments and a VME-based dual-processor system using POWERPCs (CES RIO2-8061) under LYNXOS V2.5 for more complex setups (CAMAC, VME, VXI). The POWERPCs are interfaced to CAMAC via the VME Subsystem Bus (VSB) with a CBV controller (CBV-1000, Struck) and use a dual-ported memory for internal communication: processor #1 is responsible for data readout and reads raw data from CAMAC to memory, processor #2 is responsible for event building and data transport and formats data from memory to final MBS event structures for data storage and monitoring.

The CVC setup has been successfully tested offline using data from a pulser and a γ source as well as in an in-beam experiment. Implementation and testing of the POWERPC has been started. In small experiments readout may simply be defined by lists of CAMAC commands (CNAFs), but in more complex cases the user has to supply his own readout function written in C. The readout function has then to be compiled and linked to the MBS kernel. To ease the generation of readout functions a set of ROOT classes has been developed describing the structure of the experiment: triggers, subevents, modules, and parameters. The user may write a set of C++ statements to a configuration file which later will be read by ROOT's powerful C++ interpreter ROOTCINT. This generates readout code tailored to the needs of the specific experiment. A graphical interface has been written as a prototype to allow hardware configuration via mouse clicks.

3. Back-end system

The online data analysis system is compiled C++ code using classes from the CERN ROOT system. A modified client-server model has been implemented; that is, there is a central server which handles requests from local and remote clients via socket message passing. However, due to some restrictions of the memory mapped file system in a C++ environment, it is necessary to give some server features to other running processes.

In the prototype used in the first experiment, there is a server which handles one- and two-dimensional histogram space allocation, directory listings, and storage to and retrieval from disk. It maps as a consumer to the memory mapped file described below which is created by the online analysis process.

There are two clients: one simply passes commands entered on the keyboard or executed in shell scripts to the server for execution. The other is a display client which is run in the same way, but whose function is to request histograms from the server and display them in all of the modes supported by ROOT. These client functions can also be controlled by a graphic user interface written in Tcl/Tk. Features of the display server are mouse control of expansion and scaling, linear-log mode and synchronous display update.

The actual online analysis is done by a process which links to the MBS data stream using a method due to J. Holeczek [3]. The process receives data one event at a time either from the online readout or from previously stored events in a file. It builds histograms using ROOT histogram classes via C++ code tailored to the specific experiment. These histograms are stored in a memory mapped file which is accessed for read only by the server described above. The online analysis process makes use of PTHREADS; there is an analysis thread, and a thread each for timed updates of the memory mapped file and asynchronous listening on a socket for messages from the server.

The system was first used with an existing experiment at the Q3D spectrograph. Previously this experiment used the MLLE system with monitoring code written in Fortran. This has been converted to C++ and implemented as a user function in the analysis process described above. About 90 one-dimensional and 90 two-dimensional spectra are filled using a total of 10 Mbyte histogram space in memory.

It was tested offline by applying it to data taken earlier with the MLLE system. These data had been converted by the programs MLLE2HBOOK and H2ROOT to a ROOT file in a straightforward manner.

The ROOT system includes a C++ implementation of the function minimization package MINUIT which can be used as described in [4]. ROOT provides a higher level access to MINUIT for fitting of functions to the contents of one- or two-dimensional histograms without the need for knowledge of internal details of MINUIT. These functions can be given as formulae (ROOT contains a formula class with most of the frequently used functions) or defined by the user as C++ functions.

A simple graphical interface has been implemented to ease the access to these functionalities. The user can define limits and approximate peak positions in a histogram with the help of mouse inputs. Depending on the number of positions given, gaussians with linear background may then be fitted. Optionally start parameters may be given as numerical input, similarly fixed values may be assigned to the parameters. Other fitting functions can easily be implemented.

In two dimensional histograms, a region enclosed by a polygon may be selected. The selected contents may then be projected on the x-axis taking the spread of the values along the y-axis as errors. Fits of polynomials of any degree or other functions may then be applied to the resulting one-dimensional histogram.

4. Outlook

After completion of the offline tests of the POWERPC configuration an in-beam experiment is scheduled for spring '98 to test the new system under realistic conditions. Further developments will focus on the integration of the existing software functionality in a graphical user interface and on the implementation of additional functionality (data storage in ROOT trees, remote run control, and hardware control, see fig.1).

References

[1] H. Essel, N. Kurz *et al.*: The GSI Multi Branch System. See also http://www.gsi.de

[2] Rene Brun, Nenad Buncic, Valery Fine, and Fons Rademakers: The ROOT System. See also http://root.cern.ch

[3] J. M. Holeczek, TGoosyLMD - A GOOCLI and APECLI interface to ROOT, GSI Darmstadt

[4] MINUIT , Function Minimization and Error Analysis, CERN Program Library Long Writeup D506

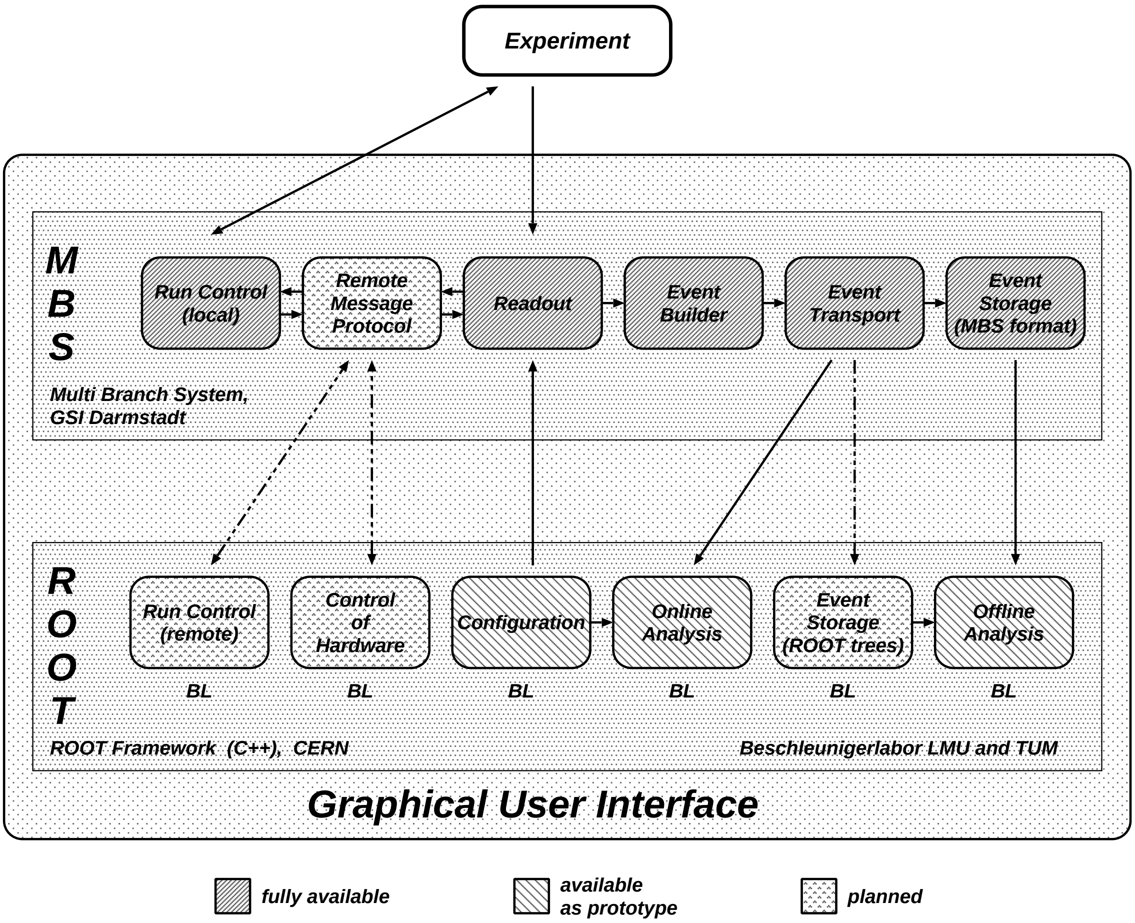

Fig. 1: The new data acquisition system