Data Acquisition and Analysis for the 3-Spectrometer-Setup at MAMI

Michael O. Distler, Harald Merkel, Markus Weis

Abstract— **The revised data acquisition system for the 3-spectrometer-setup at the Mainz Microtron (MAMI) is based on VME front-end computers running Linux to read out the detector electronics (CAMAC, Fastbus, etc.). Event building and online analysis is done on PCs running Linux as well. Three major software packages AQUA, COLA, MEZZO handle the data acquisition, analysis/simulation, and slow control, respectively. The software is written in C++ and Java. One major design goal has been a good communication and interaction between data acquisition/slow control and analysis/simulation.**

I. Introduction

The 855 MeV continuous wave electron accelerator MAMI (Mainz Microtron) is located right on campus of the university of Mainz. One of the experimental groups is the A1 collaboration which has build and operates the 3-spectrometer-setup[1]. In order to fulfill the requirements of a new phase of double and triple coincidence experiments the old Motorola 68020 based VME front-ends have been phased out. At the same time the data acquisition software has been rewritten from scratch.

The new data acquisition system has been especially designed to fit into the daq/analysis concept that ranges from the detector readout and slow control to the online and offline analysis and the final cross section. One major design goal has been to achieve a good communication and interaction between data acquisition/slow control and analysis/simulation. This involves a message system to handle the communication and a SQL database to make all the parameters of the slow control accessible for online and offline analysis.

In contrast to most of the "old" data acquisition and analysis systems used in nuclear physics the user interface is set up in a way that the user always deals with meaningful names instead of channel or histogram numbers. The names are user configurable in most of the cases. That way it is possible to have students doing useful data analysis in very little time.

This work was supported by the Deutsche Forschungsgemeinschaft (SFB 443).

Michael O. Distler is with the Institut für Kernphysik, Gutenberg-Universität Mainz, 55099 Mainz, Germany (telephone: +49 6131 39 22956, e-mail: distler@kph.uni-mainz.de).

Harald Merkel is with the Institut für Kernphysik, Gutenberg-Universität Mainz, 55099 Mainz, Germany (telephone: +49 6131 39 25812, e-mail: merkel@kph.uni-mainz.de).

Markus Weis is a student at the Institut für Kernphysik, Gutenberg-Universität Mainz, 55099 Mainz, Germany (telephone: +49 6131 39 25740, e-mail: weism@kph.uni-mainz.de).

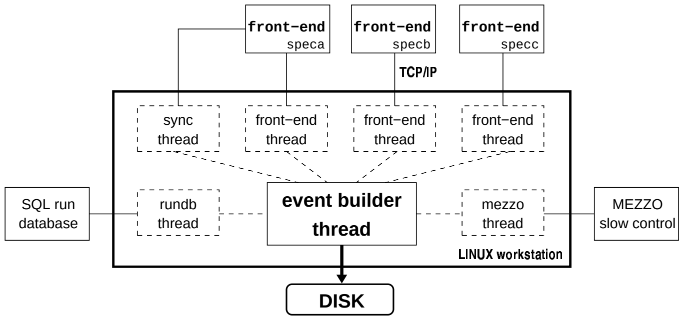

Fig. 1. Schematic view of the AQUA event builder threads.

II. AQUA - Data AcQUistion for A1 experiments

The detector systems for each of the three spectrometers are very similar and consist of scintillator paddles, wire chambers and a gas Cherenkov detector. Each spectrometer is equipped with CAMAC and Fastbus electronics that is read out by one VME front-end which is located on the back side of the spectrometer. The front-ends send the data via fast ethernet to the event builder in the counting room.

A. Event building

For most experiments it is desirable not only to record coincidence data but also (pre scaled) single arm data for calibration purposes. Therefor a special counter module is part of the experiment trigger, which is incremented for every event, either single or coincidence event. This counter is read out by each individual front-end. According to this information the data streams from the front-ends are merged by the event builder.

Since the data acquisition system has been designed from scratch object-oriented methods could be used from the very beginning. The message system uses the same transport mechanism for status and control messages as for messages that contain detector data. Consequently there is only one TCP/IP connection open between the individual front-ends and the event builder. A watchdog thread on the event builder is monitoring the heartbeat signals send by the front-ends.

The event builder also fills a run database and collects status information from the slow control which is written to output stream as well. That way the slow control data is available for offline analysis.

B. Front-ends

The front-end system for the 3-spectrometer-setup at MAMI is based on VME computers (VMIC VMI7589 with

TABLE I
The AQUA data format.

byte	16bit word	16bit word
0	message length	
4	message type/number	
8	timestamp	
12	unit number	unit length
...	...	...
...	unit number	unit length
12+4U	data	

a Pentium 200 MHz CPU) running Linux to read out the detector electronics (CAMAC, Fastbus, etc.). The interface to the VME bus is implemented as a kernel driver[2] which takes care of the interrupt handling and the programming of the Universe chip. With these front-ends interrupt rates up to 30 kHz can be achieved. For the actual setup where the maximum rate is limited by the conversion time of the TDCs and the readout via the slow CAMAC bus the maximum rate is 2.5 kHz.

C. Data format

A new data format has been introduced that is optimized for high readout rates. All messages that are passed between event builder and front-ends share the same header definition (table I). The length of the message, the type and a timestamp are stored in three longwords. Possible message types are control messages (e.g. run start and stop), status and debug messages, or slow control messages. For an event data message the event number is part of the type field. The timestamp is contains the unix time in seconds except for the data messages where the timestamp gives the milliseconds since run start. This time resolution is possible since all daq computers are synchronized using NTP.

Every front-end has several readout units which can be physical units (e.g. a Fastbus crate) or logical units (scalers that are read out every 100 events). This organization is visible in the data format, too.

D. User interface

The graphical user interface is implemented in Java. It is used to configure the event builder. Any combination of the three spectrometer and various non-standard detectors can be selected for single, double or triple coincidence experiments. Another task of the user interface is the run control and the monitoring of the daq system. Visual and audio signal are generated in case of a system failure.

III. MEZZO - Slow control

A. Front-ends

The front-end computers for the slow control are also based on PCs running Linux. They provide the interface

TABLE II
Examples of Hardware Modules supported by MEZZO

ADAM Module 4011, 4012, 4014D, 4017, 4050, 4060
A&D Instruments FG 150 (scale)
BMC-PC20 (ADC, DAC, DI, DO)
Bruker MPS-Controller (KPh Mainz)
Collimator-Controller (KPh Mainz)
Danfysik MPS 8000 (magnet power supply)
Dataforth D1000 (pressure)
ILM200 family cryogen level meter
IPS 120-10 (magnet power supply)
LakeShore DRC91CA, LS208 (temperature)
LeCroy 4032a, 1440 (HV power supply)
NI MicroGPIB RS-232/IEEE-488
NRVD Powermeter
Phoenix Contact IBS PC ISA SC/I-T (Interbus S)
VDC/HDC-Controller (KPh Mainz)
uDAM Module 6017

to the control hardware (table II). Each channel (ADC, DAC, etc.) can be identified via port number, TCP/IP address, module name, and channel number. The readout values are sent directly to to the central process via fast ethernet without being filtered or converted. The typical readout frequency for each channel is 1 Hz.

B. Central process

On a Linux Workstation in the control room runs the central process. This main process of the control system collects the raw values from all front-end computers, converts them into physical values and applies a filter when the value is unchanged. This reduced data stream is sent to the data acquisition, the graphical user interfaces, and a monitor process to log all data.

The main task of the central process is to compare the nominal and actual values of each channel every time the value changes and to report mismatches.

The central process also manages logical names (aliases) of each channel. The user only deals with meaningful names instead of channel numbers. The aliases can be created, modified and deleted without having to restart the process. The 3-spectrometer-setup is represented as a hierarchical tree of these alias names and no hardware information is hard-coded in the software.

The nominal values for each channel and the "alias" information (limits, alarm thresholds, polling frequency, etc.) are stored in a SQL database (postgreSQL).

The front-end computers may eventually break down due to high radiation levels in the experimental hall. Therefor the central process has to check the communication to the front-ends periodically. The connection is cut in case of a timeout and a message is sent to the user interface.

C. Data format

Internally, the slow control uses a binary protocol that has the same packet header format as the data acquisition.

Packets from AQUA and MEZZO can be mixed without confusing both systems. Additionally, an ASCII protocol is available. The user can use any text terminal and "telnet" to do simple operations.

D. User interface

The graphical user interface is a JAVA application. It shows the status of the setup, displays actual and nominal values of the channels and allows the user to edit the "alias" information.

For simple operations or batch processing a command line interface is provided.

IV. COLA - Online/offline analysis and simulation

While the data acquisition and the slow control is localized at the site of the experiment, the data analysis usually takes place at all participating institutions of an international collaboration. So this part of the A1 software family is kept as portable as possible and implementations on nearly all common unix systems exist.

In former experiments, for each analysis of a new experiment a specially tailored program was developed based on a collection of libraries e.g. for event decoding and basic detector analysis. On the other hand, the task of a generalized analysis software is to provide a tool for the analysis of an experiment, even without knowing at the moment of the software development which reaction has to be analyzed. This concept can only be achieved by separating the detector dependent program source code from the description of the physical quantities in a high level language.

A. The Cola language

The Cola language is designed to describe nuclear reactions without the need to know the actual detector. So the basic data objects are relativistic four-vectors as reconstructed by the detector packages. From these four-vectors other four-vectors or scalar quantities can be derived and filled into histograms. Further, the derived scalar quantities and the raw data as well can be used to define cut conditions for the histogramming of the events.

An analysis of a neutral pion production experiment takes for example the following steps: Defining the missing four-momentum of the reaction, which is likely to be a pion, calculating the mass of the four-vector, and filling histograms in the center of mass coordinates of this four-vector with cut conditions on the missing mass around the expected pion mass.

B. Simulation

The integration of the accepted phase space via a Monte Carlo simulation is the second part of an analysis package to extract cross sections from count rates. In the Cola concept, this part of the analysis can now be performed very easy. The physics description in the Cola language for the histogramming of the count rates is sufficient as input to perform the completed phase space integration. Even high level corrections e.g. the radiative corrections can be performed automatically. By this concept, we are able to extract normalized cross sections already during the experimental data taking.

C. The histogram package

Several high quality histogram packages are already on the market. Nevertheless, we decided to use an own development of such a package to integrate the source code in the framework of the analysis. By this we were able to overcome some weaknesses of the existing packages, e.g. the missing automatic error calculation and propagation in weighted Monte-Carlo histograms. At the same time we could include the online histogram display in our user interface. The graphical user interface is based on the Motif version of the toolkit library wxWindows[3], which provides a portable GUI on most Unix Operating Systems. Via a simple macro language a high quality Postscript output can be created to be included in publications.

References

[1] I. Blomqvist et al., *The three-spectrometer facility at the Mainz microtron MAMI*, Nucl. Instr. and Meth. A403 (1998) 263

[2] J. Hannappel, `http://lisa2.physik.uni-bonn.de/~hannappel`

[3] Julian Smart, `http://www.wxwindows.org`