Trends and Challenges in Data Acquisition and Control Systems Summary of CHEP97 Session B

A.J. Lankford

Department of Physics and Astronomy
University of California
Irvine, CA 92697-4575
lankford@lankford.ps.uci.edu

Current topics in computing for data acquisition, control, and trigger systems for high energy physics experiments are discussed with emphasis on those topics presented in papers submitted to the International Conference on Computing in High Energy Physics, Berlin, April 7-11, 1997.

Fifty-one high quality presentations and posters on the topics of data acquisition, control, and trigger systems were presented at the International Conference on Computing in High Energy Physics, held in Berlin in April 1997. These presentations covered architectures, algorithms, implementations, and tools, as well as both hardware and software topics. They covered a wide variety of subjects ranging from highly technical topics such as custom gallium arsenide adder circuits and how to select an implementation of CORBA to broad topics such as high-performance architectures and software for distributed processing.

1 Challenges for New Trigger and Data Acquisition Systems

Whether to search for and study rare particle processes or to perform precision studies of other interesting processes, new high energy physics experiments demand very high interaction rates, and hence unprecedented trigger and data rates. Managing these data rates requires new trigger and data acquisition systems with increases in trigger complexity, data bandwidth, processing power, and software sophistication. Indeed, the increased resolving power of present and planned high energy physics experiments arises principally from our ability

Preprint submitted to Elsevier Preprint *1 July 1997*

to instrument and read out large numbers of channels with custom electronics and our ability to harness the power of large amounts of affordable compute power.

2 Data Acquisition Architecture

2.1 Characteristics of New Data Acquisition Architectures

Many of the architectural features of trigger/data acquisition systems proposed for the LHC detectors are characteristic of most new data acquisition systems. The LHC experiments [1,2] employ architectures which exploit multilevel trigger and data acquisition systems. Level 1 triggers are based on custom, pipelined logic in order to run in a nearly deadtimeless fashion. Level 2 buffers are digital, enabling long level 2 trigger latencies and hence the use of general-purpose processing elements. Higher level triggers are based on network switches and processor farms in order to achieve high performance data transport and processing. They also attempt to make effective use of commercial technology, products, and protocols. Moreover, the architecture and components are designed so as to be partitionable, scaleable, upgradeable, reliable, and maintainable.

2.2 Data Acquisition Architectures at CHEP97

At this meeting, in addition to revisiting the proposed architectures for the major LHC experiments [1,2], the data acquisition architectures of some experiments which will operate in the near future were discussed. Cornell's CLEO III [3] data acquisition system, designed to study B-physics at high luminosity, does not use pipelined, deadtimeless front-end electronics or switch-based event assembly, but does use a very fast, heavily buffered data collection and hierarchical event assembly system to keep deadtime quite small at expected trigger rates. For comparison, KEK's B-detector BELLE [4] and Frascati's CP-violation detector KLOE [5] do use switch-based event assembly, and SLAC's B-detector BABAR [6] uses both pipelined front-end electronics and switch-based event assembly. Fermilab's D0 detector [7], which pioneered multiport-memory based event assembly and online farms based on commercial processors, is upgrading the performance of its data acquisition system by adding more parallelism to its event assembly and PCI-based processors and WindowsNT to its farm. The new event assembly system uses a novel recirculation scheme in order to regulate data flow. D0 has also adopted what is now formally a three-level architecture, in which event assembly and the final stage of trigger selection occurs at Level 3.

The data acquisition systems of some recently commissioned experiments KTeV [8], Euroball [9], CAT [10], KASCADE [11], SND [12], AGS E896 [13], and of Fermilab's Development and Test Department [14] were also discussed. KTeV [8] has commissioned a very high-performance implementation of the general-purpose Fermilab DART data acquisition system [15]. In KTeV's implementation, which is VME based, events are assembled via dual-ported memories. The Euroball data acquisition system [9], by contrast, uses VXI for data collection and a Fiber Channel switch for event assembly. In addition, new general purpose data acquisition systems at the U.S. labs and at IAEA [16] were discussed, along with some of their implementation experience. Both Fermilab's DART system [15] and Thomas Jefferson Lab's (formerly CEBAF) CODA system [17,18] support a wide variety of fixed target experiments, ranging from small to quite large.

These experiments whose data acquisition systems were presented are pioneering the application of new hardware and software technologies to data acquisition. Their experiences, particularly experience with high rates and low deadtime, with network-based event building, and with software technologies for distributed computing, can provide valuable insight into the selection of architecture and of hardware and software components for future experiments.

3 Trigger Architectures

3.1 First Level Triggers

First level triggers are still the well-established realm of special-purpose, custom processors. High luminosity operation, whether at a collider or with a fixed target, demands complex first level triggers with two relatively new characteristics: pipelined architecture, in order to avoid deadtime during the latency of the first level trigger, and very sophisticated, selective algorithms, in order to suppress high rate backgrounds. The calorimeter trigger electronics under development for the CMS detector [19] exemplifies these characteristics, first in the use of detailed Monte Carlo studies to establish physics requirements and to study trigger selection algorithms, and then in its use of state-of-the-art electronics, such as custom GaAs integrated circuits, gigabit per second fiber optics, and 160-MHz point-to-point backplanes. Development of first level triggers for HERA-B [20] and STAR [21] were also reported.

3.2 Second Level Triggers

Whereas, first level triggers are the well-established realm of high-speed custom logic, and third level triggers are the well-established realm of general-purpose processor farms, second level triggers are a fertile field for exploring new techniques. Techniques under consideration for new experiments range from special-purpose to general-purpose processors, arranged in a variety of pipelined and parallel architectures. In general, there seems to be a clear trend away from complex hardware processors towards exploiting the cost effectiveness and flexibility of off-the-shelf computing for future second level triggers. This approach is epitomized in the architecture adopted by the CMS detector [1], where the second and third level trigger are incorporated into a single higher-level trigger. The CMS Letter of Intent [22] states, "The resources which would have been required for a hardware second level trigger are invested in the readout network and in the event filter processing power, both of which are more suitable for the integration of technological upgrades." This philosophy is also characteristic of the principal level 2 trigger options under consideration by the ATLAS [23,24] and HERA-B [25] experiments, although these two experiments maintain a distinct level 2 trigger, separate from higher-level event filters.

A challenge remains at the second level of triggering to demonstrate, in the face of high (approximately 100kHz) level 1 trigger rates, that general-purpose CPUs can provide the required amount of processing power at affordable costs and that data flow can be managed through the extensive networks foreseen for connecting front-end buffers to the trigger processors. Interesting results from studies by the level 2 trigger group of ATLAS [23] show that, in the architectures based on commercial CPUs and networks which they are considering, approximately one half of the requirement for processing cycles arises from moving and assembling data, matching the requirement for execution of trigger selection algorithms.

Development of custom logic for second level triggers continues for some experiments, as do potentially interesting custom computing architectures such as μENABLE [26] which is designed as a programmable (*i.e.* configurable) coprocessor composed of field programmable gate arrays.

3.3 Third Level Triggers

Third level trigger processing is the well-established realm of commercial, general-purpose CPUs. Level 3 trigger processors in new and future experiments generally sit on a commercial network, which more and more often is also the event assembly network. This configuration is the choice of nearly all new data acquisition architectures, for instance CMS [1], ATLAS [2], KLOE [5], BABAR [6], D0 [7], Euroball [9], SND [12], CLAS [17], ZEUS [27], and HERA-B [28]. The concept of an online processor farm as a trigger, however, seems to be gradually giving way to the online farm conceived as an "event filter". In this conception, the ability to run offline code in the online farm is generally elevated to the level of a requirement, and plans to run only offline code to perform full reconstruction is now often a goal. In some experiments, such as HERA-B [28] and BABAR, an additional trigger/data acquisition level is defined for full reconstruction. For instance, for HERA-B both level 3 and level 4 processors operate on fully assembled events. In BABAR, the "online event processing" stage, which performs the level 3 trigger algorithms, is followed by a "prompt reconstruction" stage which runs the full offline code.

4 Data Acquisition Hardware

4.1 Switch-based Event Assembly

Event assembly using network switches is the most topical solution to the need for a "parallel" event builder which, in order to avoid data bottlenecks, provides parallel data paths between front-end data sources and processor farm data destinations. Commercial network switches are also the solution which capitalizes on developments in the communications and computer industries. Candidate network technologies under consideration include ATM, FiberChannel, SCI, FDDI, and 100baseT or Gigabit Ethernet. Important experience in event assembly at high rates with network switches will be gained by experiments of the current generation working with this technique. For instance, last year ZEUS [27] began operation of an FDDI-based system. This year, CLAS [17] will start operation with ATM, and Euroball [9] will start operation with Fiber Channel. Next year, KLOE [5] will put an FDDI system into operation, and the following year CDF will start operation with ATM. Nonetheless, the performance of these systems is still at least an order of magnitude less than that proposed for the LHC. Meanwhile, laboratory benchmarking of candidate technologies continues. Components with adequate bandwidth performance exist today. The challenge in switch-based event assembly arises from the desire to choose the technology which will provide the longest lasting solution in terms of upgrades of performance and ease of maintenance. In using this technique, issues of control of data flow are of paramount importance, particularly with respect to error identification and recovery and to fault-tolerant operation.

4.2 Event Recording

New and future experiments generally define their maximum rate for writing events to archival storage according to the size of the data sample which they feel can be managed and processed offline. This leads to recording at high rate, typically between ten and thirty Mbytes per second. Recording at these rates seems to be a solved problem. For instance, KTeV [8] records at 18 Mbytes/s with DLT drives and BELLE [4] will record at 15 Mbytes/s with SONY DIR-1000 drives.

4.3 Modeling of Data Acquisition Architectures

Functional modeling is a vital tool for design of large data acquisition and trigger systems. It provides a means for extrapolating from small-scale prototypes to full-scale systems. A number of interesting modeling studies were presented at this conference. ATLAS [23] demonstrated that much can be learned about its level 2 architectural options using only "paper models", without full simulation. ALICE [29] and D0 [7,30] are using an object-oriented simulation language MODSIM [31] to guide their system design. A particularly interesting study of the ZEUS second level tracking trigger has been completed [32]. By modeling an existing configuration of this trigger, it was possible to validate the system model before using it to provide guidance to the design of upgrades. Similarly, studies of laboratory scale prototype and demonstrator systems of data acquisition architectures proposed for future systems can be used to validate models used for design. Then these models can be used to confidently extrapolate from prototype scales to full scale.

4.4 New Hardware Developments

For some applications, commercial solutions do not provide performance that is fully satisfactory. Consequently, the high energy physics community continues to develop some new hardware. Need for higher performance occurs particularly during the stage of data collection from front-end electronics which precedes event assembly into a processor farm. Consequently, new hardware developments tend to appear at the level of front-end electronics crates. For instance, CLEO has developed a Fastbus to VME interface [3,33] to allow control of commercial Fastbus digitizer modules by off-the-shelf VME single-board computers, at the same time providing a uniform processing environment at the crate level for both VME and Fastbus. In addition, CLEO has adopted several of the properties of the draft VME-P specification [34,35], including broadcast (MCST) and chained block transfers (CBLT) to speed data collection. KLOE has developed a bus called AUXbus [36], also for faster data collection. ALICE has developed a dedicated detector data link [37]. CES has developed a high-performance interconnect [38] to link PCI-based processor platforms, and a CERN/Liverpool group has implemented a T9000 transputer as a communication controller for a DEC Alpha microprocessor [39]. Custom hardware is typically required for timing and trigger distribution systems, such as CTTS [40], for performance reasons.

5 Techniques for Distributed Processing

One of the principal differences between today's data acquisition systems and those of a few years ago is the degree to which processing is distributed throughout the systems. For many years, embedded processors have been providing distributed computing power within the hierarchy of data collection, for instance, through the use of crate-level readout controllers. Microprocessor farms for high level triggers have more recently appeared, but have been in use for some time. In the last few years, workstations have proliferated for data acquisition control, monitoring, and online event reconstruction, providing a more diverse, that is less uniform and hierarchical, use of distributed processing. Control and coordination of distributed processors in the data acquisition environment has in the past been awkward. Techniques for providing control and coordination of distributed online processing was one of the strong themes of this conference. Among the tools discussed were finite state machines, object request brokers, and Java.

5.1 Finite state machines

Finite state machines (FSMs) have been in use for run control of some experiments for years. They have now been adapted in several independent fashions, for instance, by CODA [17], DELPHI [41], and ATLAS [42], to provide run control for distributed processors. The usual technique is to create an FSM proxy for each distributed processor and to establish a hierarchy of finite state machines, in which the state of each FSM depends upon the state of FSMs at lower levels within its domain. Finite state machines are now moving beyond run control to describe states of other software components. For instance, ZEUS [43] has used a Harel diagram and rule-based implementation to monitor the components of its trigger rates. In addition, various software tools [41,42] are being used to construct the code which implements the finite state machines.

5.2 Object request brokers

The distributed processing environment of data acquisition systems can be quite dynamic. The distribution of tasks can change frequently, particularly in the high-level, control end of the system, and particularly during the evolutionary development of the system and during periods of partitioned debugging of subsystems. Object request brokers (ORBs) are being used in data acquisition systems by several experiments including CLEO [3,44], ATLAS [42], and PHENIX [45] to dynamically establish inter-component communication in order to manage this problem. Various implementations of CORBA-compliant ORBs have been chosen by different experiments, often guided by the availability for the particular workstation or real-time operating systems chosen by the experiment. The NILE [46] project uses CORBA in a processor farm, along with replication of object groups on multiple nodes, to achieve robust distributed processing despite failures of processes, entire nodes, or network links.

5.3 Java

Intrigued by the fashion of Java, several investigations of its use in data acquisition systems have been initiated [47–49]. These studies report the merits of Java's platform independence and its ability to act as a mechanism to dynamically handle any system configuration. The most convincing applications to date, for instance Ref. [47], are at the back end of data acquisition systems, where Java can provide flexible client/server relationships for monitoring and control, as well as provide some balancing of processing load between server and clients.

5.4 Object Oriented Programming

Three years ago at CHEP94, one or two object oriented software systems for data acquisition were reported. Since then, object oriented software for data acquisition has become commonplace, not unusual. Most new data acquisition software reported at this conference is designed using object oriented techniques [1,3,17,41–43,45,48,50,51]. Object orientation has now become the natural choice for any new system.

6 Software Technologies and Operating Systems

Commercial and public-domain software components offer the opportunity to enhance the quality, robustness, and maintainability of software systems for data acquisition [42,44,45,48,52,53]. Several interesting products were identified for services such as databases, communication, operating system services, and code generation. Availability of source code greatly enhances the desirability of software packages from outside HEP, for reasons of long-term maintainability and adaptability.

Operating systems are now commonly employed on embedded processors in data acquisition systems. The need for real-time performance for this application frequently leads to the use of VxWorks [54] in the United States and LynxOS [55] in Europe. Microsoft Windows NT is now beginning to appear in the data acquisition environment. Detailed studies [53] show that it has not yet reached the real-time performance levels of LynxOS; however, it may be viable for applications which do not require extreme performance.

7 Summary

By virtue of the power which they endow to data acquisition and trigger systems, the capabilities of modern computing hardware and software will enable current and future experiments to reach further towards solving the mysteries of particle physics. Several challenges to harnessing the full capabilities of modern computing for trigger and data acquisition exist, particularly mastering the use of commercial switching networks for high-speed event assembly and mastering distributed computing in the real-time environment. Valuable experience is being accumulated towards these goals. Over recent years the focus of data acquisition sessions at CHEP has shifted noticeably from hardware to software, reflecting the ample performance available from commercial hardware products and yet the challenges of harnessing that performance. As we consider these software challenges, we can muse that the ideal software for a data acquisition system would be independent of the technology of its hardware platform and independent of the language in which it is implemented. Computing for data acquisition in high energy physics seems to be moving constructively in that direction. The reader is invited to review the many excellent papers on this subject presented at this conference.

8 Acknowledgements

The author would like to acknowledge all the excellent contributions on the subjects of trigger, data acquisition, and control systems made to this conference. He would also like to thank the organizers of the conference and his co-conveners L. Mapelli and B. Haynes of the sessions on data acquisition and control systems. The author's work is supported in part by U.S. Department of Energy Grant DE FG03 91ER40679.

References

[1] A. Kruse, CMS Online Event Filter Software, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[2] G. Ambrosini, *et al.*, The ATLAS DAQ and Event Filter Prototype "-1" Project, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[3] A. Wolf, *et al.*, The DAQ System for CLEO III, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[4] H. Fujii, *et al.*, High-Speed Data Transfer and Recording System for BELLE (KEK-B) Experiment, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[5] A. Doria, *et al.*, Message System and Data Transmission in the KLOE Data Acquisition System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[6] The BABAR Collaboration, Technical Design Report, SLAC-R-95-457, March 1995.

[7] D. Cutts, *et al.*, The D0 Level-3 Trigger / Data Acquisition System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[8] E. Blucher, *et al.*, KTeV Data Acquisition System and Online Software, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[9] G. Barazza, *et al.*, The Euroball Data Acquisition System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[10] H. Delchini, *et al.*, GAS - A General Application Server, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[11] H. Schieler, *et al.*, Data Transmission and Online Processing in the KASCADE Data Acquisition System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[12] D.A. Bukin, *et al.*, Data Acquisition System of SND Experiment, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[13] J. Schambach, *et al.*, Data Acquisition Systems for E896 at the AGS, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[14] J.M. Nogiec, *et al.*, An Open Distributed Monitoring and Control System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[15] G. Oleynik, *et al.*, The Fermilab DART Data Acquisition System at Running Experiments, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[16] A. Bazhenov, *et al.*, Using VXIbus-Based DAQs for Basic Research and Unattended Monitoring Applications, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[17] W.G. Heyes, *et al.*, A Distributed Object-Oriented Control System for the CODA Data Acquisition System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[18] D.J. Abbott, *et al.*, The Front-End Readout Control Software for the CODA Data Acquisition Toolkit, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[19] S. Dasu, *et al.*, Calorimeter Trigger Electronics for CMS Detector at LHC, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[20] T. Fuljahn, *et al.*, Implementation of the HERA-B First Level Trigger, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[21] V. Lindenstruth, *et al.*, Dead Time Free STAR Trigger Architecture, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[22] CMS Collaboration, Letter of Intent by the CMS Collaboration for a General Purpose Detector at the LHC, CERN/LHCC 92-3, October 1, 1992.

[23] D. Calvet, *et al.*, Performance Requirements of Proposed ATLAS Second Level Trigger Architectures from Simple Models, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[24] R. Blair, *et al.*, Options for the ATLAS Level-2 Trigger, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[25] J.D. Hansen, *et al.*, Demonstrator for a High-Performance Second Level Trigger, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[26] Y. Gal, *et al.*, μEnable: A FPGA-Based Coprocessor Proposed for Pre-processing in ATLAS, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[27] S. Polenz, *et al.*, A FDDI Based Processor Farm for the ZEUS Third Level Trigger, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[28] A. Gellrich, *et al.*, A Prototype System for the Processor Farm of the HERA-B Experiment at HERA, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[29] J. Yuan, *et al.*, Simulation of the ALICE DAQ System with GALSIM, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[30] D. Cutts, *et al.*, Simulation of the D0 Level-3 Trigger System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[31] The CACI Products Company (http://www.cacias1.com/).

[32] A. Quadt, *et al.*, Simulation of Parallel Processing Systems: The ZEUS Central Tracking Detector - A Case Study, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[33] R. Wanke, *et al.*, FRITZ: A Universal Fastbus-VME Interface, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[34] VME for Physics "Implementation Recommendations & Guidelines, Draft Document" (http://www.vipa_fnal.gov/vipa/doc2.html).

[35] VME64 Extension Draft (http://www.forcecomputers.com/technology/papers/v64ext.html).

[36] A. Aloisio, *et al.*, AUXbus: An Event Driven Bus-Based DAQ System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[37] P. Csato, *et al.*, Report on the ALICE Detector Data Link: method used, protocol and design status, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[38] M. Weymann, Linking PCI Based Processor Platforms, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[39] R.W. Dobinson, *et al.*, High Performance Computing Nodes Using a DEC Alpha Processor and a T9000 Transputer, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[40] C. Terella and I. Young, CTTS: The Composite Timing and Trigger System of the *JET* Machine, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[41] C. Gaspar and B. Franek, SMI++ - Object Oriented Framework for Designing Control Systems for HEP Experiments, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[42] D. Burckhart, *et al.*, Software Technologies for a Prototype ATLAS DAQ, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[43] U. Behrens, *et al.*, Rule-Based Monitoring of Trigger-Rates at ZEUS, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[44] A. Wolf, *et al.*, CORBA and the CLEO III Data Acquisition System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[45] E. Desmond, *et al.*, Overview of the PHENIX On-Line Computing System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[46] F. Previato, *et al.*, Demonstration of NILE, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[47] H. Beker and R. Divia, Evaluation of Data Acquisition Tools Based on Java, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[48] Y. Yasu, *et al.*, A Study of Network-Based Data Acquisition System, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[49] I.C. Legrand, A JAVA Based Environment to Control and Monitor Distributed Processing Systems, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[50] C. Arnault, *et al.*, A Component Oriented Framework for the Control-Command Environment of the Virgo experiment, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[51] J. Tutas, A C++ Based DAQ System for Small and Medium Sized Experiments with LabView as Graphical User Interface, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[52] R. Kaiser, *et al.*, Online Monitoring at HERMES, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[53] G. Ambrosini, *et al.*, Operating System Studies for DAQ Applications, submitted to the International Conference on Computing in High Energy Physics, Berlin, April 9-11, 1997.

[54] Wind River Systems (http://www.wrs.com).

[55] DOC-0080: Lynx RTS, LynxOS 2.3.1 for PowerPC, Development Operating System.