Online Data for Experiments

R. Pordes, M. Votava, R. Forster on behalf of the Computing Division

Computing Division, Fermilab, USA

Abstract

We report on a new project just getting underway at Fermilab to provide data acquisition software for the next generation of experiments at the lab*. These experiments are in the early stages of detector and data acquisition architecture design. ODE (Online Data for Experimenters) will incorporate components from DA systems already in use at Fermilab, specifically DART and the Run II collider program. DART, the existing Fermilab standard system, has proven robust and versatile over the last five years and has been used by many experiments and test beams beyond those which provided the original requirements for its development. Additional components from Run II of CDF and D0 are also being used. ODE includes components to provide a ”paperless counting room”, ”remote and distributed access and control”, ”large scale system administration, instrumentation and monitoring”, ”remote archiving of the data”, and ”integration of data bases for bookkeeping and tracking”. We provide a scripting language in python to define test sequences and acquisition steps. NT and Linux are supported. Java is the implementation language of choice. We plan to reuse the message logging, vme readout and resource management components from the CDF Online system, and the data archiving and staging components from the D0 Run II data handling system. We are integrating LabView for test stands and small scale data acquisition environments. We describe our first integrated test system within the ODE framework, targeted for use in early 2000, for a BTeV data acquisition pilot project.

Keywords: data acquisition,online,test stand

1 Introduction

With the end of data taking of the current generation of Fixed Target Experiments, the maturing of the upgrades for the Collider Run II data acquisition systems, and new proposals for experiments in the Fermilab Main Injector era, we are beginning to evaluate and develop the first components of a new data acquisition system in support of future experiments' data taking. Table 1 gives a summary of our current understanding of the data acquisition requirements of the Run II experiments and most of the proposed experiments at the Main Injector.

As reported in previous CHEP conferences, parts of the DART[1][2] system have been used by more than eight experiments over the past five years. DART has proven to be a scalable, flexible and robust hardware and software system for these experiments, and one of the collider experiments - DZero - has been using it extensively in their test stand systems during the development of the electronics and detectors for Run II. As DART hardware and software nears the end of its life cycle, it is appropriate and timely to move to the next generation of technology choices from those of DART which relied on: VxWorks, IRIX host, tcl/tk GUI, gdbm database system.

* This work is sponsored by DOE contract No. DE-AC02-76CH03000

	CDF/D0	Minos Near/Far	Kami	CKM	BTeV
No. of Channels	1M	50k	10k	20k	50M
Frequency, MHz	7.6	2.5/5.0	8	40	7.6
Level I Output, Mbytes/sec	256		5		25,000
Level II Output, Mbytes/sec	256	40	120	200	5,000
Event Size, kbytes	256	5	8	2 (pre L3)	150
Logging Rate, Mbytes/sec	15	0.5/0.1	12	10	200
Experiment Start	2001	2003	2006	2005	2006 ?
Experiment Life	5	10	5		8

Table I: System Requirements for HEP Fermilab Experiments

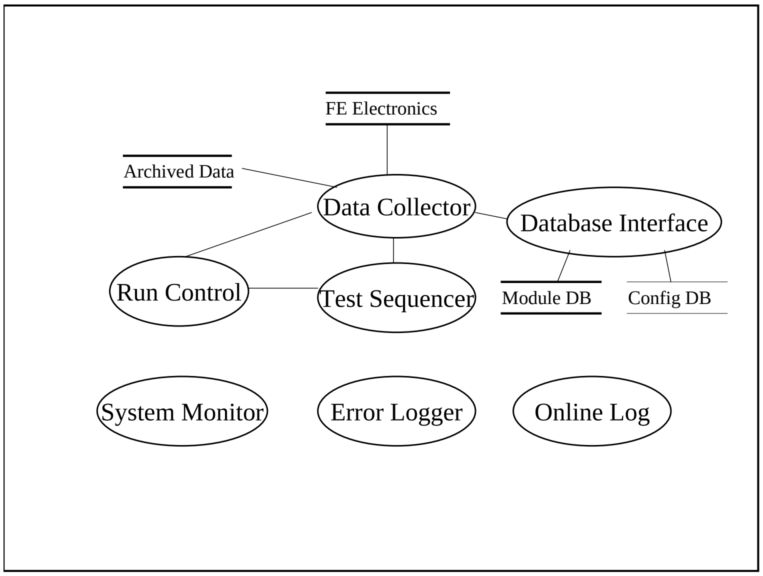

Figure 1: ODE Test Stand Software Architecture

In keeping with our tradition of close cooperation with end users, we are currently collaborating with CDF[3] and D0[4] on components of their data acquisition and online systems - especially in the area of diagnostics for the silicon detector electronics and run control with CDF, the data logging and database subsystems for D0, and online database design for both experiments.

We report here on the first steps towards a new Fermilab DA system - ODE[5] - to be fully supported on PC hardware platforms running either NT or Linux. Programming languages will include Java and scripting in JPython, and CORBA will be the programming standard for communicating with remote objects. It is premature to start designing the final DA architecture and framework for BTeV, CKM, etc. We are evaluating and developing components of a new system which can be used now for testing, extended for use in sub-system tests and commissioning, and then be used as a basis for the design and implementation of the full data acquisition systems.

Thus, the approach will be one of incremental development and release of Versions of ODE as needed throughout the lifecycle of the experiments' DA needs. The system is being designed and developed in collaboration with the engineers who are building the data acquisition modules and with the experimenters running test stand and test beams.

The initial components of ODE are shown in Figure 1.

A typical hardware architecture for ODE test stands is shown in Figure 2.

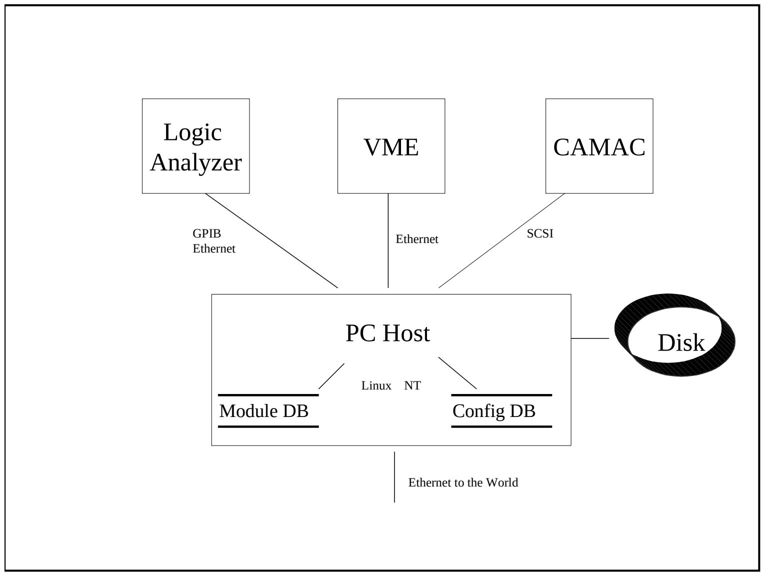

Figure 2: Typical ODE Test Stand

2 Components of ODE V001

The first real world application of ODE will be a teststand for the photomultiplier tubes for the BTeV RICH detector, shown in Figure 3, to be delivered this spring.

2.1 Data Acquisition

The data acquistion path for ODE is concentrating on an environment that is geared for test stands and test beams - small and easy to construct. The acquisition of CAMAC and GPIB data is supported by using LabView. We have interfaced the Jorway SCSI branch driver and crate controller using LabView. This allows us to continue using the existing installed hardware base. Data acquired through LabView can be viewed directly or sent through shared memory to the rest of the ODE system. This work has been done in an NT environment and we will be evaluating Labview under Linux shortly. Data from VME is acquired in ODE through an upgrade of the CDFVME test stand framework[8] reported on in RT99. We are replacing the CDF specific “real time corba” implementation with a public domain Corba package[9]. This framework can also be modified to support ”backplaneless” standalone boards if the boards have ethernet and a processer that can support a corba server. We are also evaluating the use of Linux or Real Time Linux as a replacement for VxWorks in the module testing environment. We will clearly provide a framework in which test sequences can be defined and executed and the results of long running sequences archived for later analysis. Board developers need to readily define a series of actions to debug/diagnose hardware and perform error rate tests over an extended period of time. The suite of tests needs to handle flow control based on returned values as well as support a batch mode operation.

2.2 Archiving of Administrative and Control Information

The initial version of ODE will incorporate MySQL databases for the storage of and access to detector and module information. The database interface will be standardized and allow either Mysql or Oracle to be used. We are currently working on interfacing the CDF slow controls and online DA databases to transfer data from the detector monitoring database (in a commercial

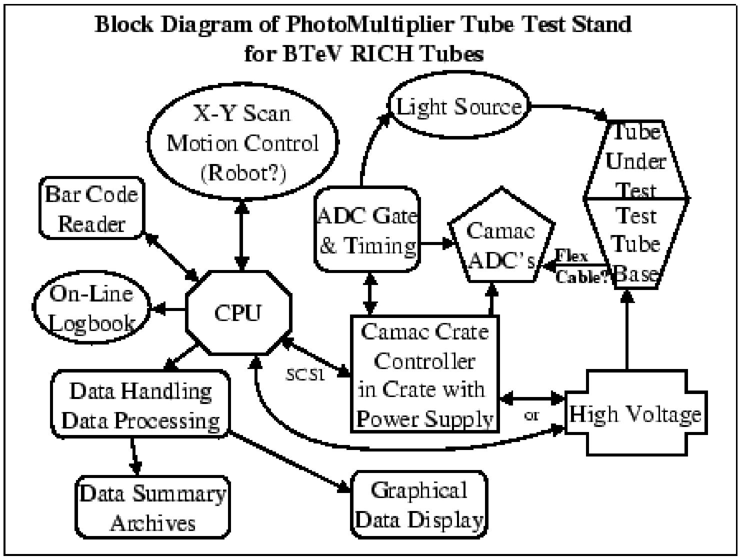

Figure 3: Photomultiplier Tube Test Stand

database from Intellution) and the online run conditions database (in Oracle). Additionally we plan to increase the logistics support available to the engineers and technicians by supporting bar-code reading of module serial numbers into the administrative databases.

2.3 Run Control/Message Logging

The initial version of run control (the ability to control the data taking configuration and duration) will be inherent in the DA framework - either LavView or CDFVME. This can only meet the needs of small scale integration tests. Subsequent releases of ODE will provide the "glue" run control and monitoring layer using the flexible, extensible GUI described in the poster paper "Dynamic graphical user Interfaces using XML and JPython". Data and the display of data can be defined in XML while test sequences and the GUI itself are coded in jpython.

As a message logging system, ODE will use merlin[6], the message logging facility delivered to CDF for Run II. Merlin currently uses commercial software, SmartSockets, for the underlying message passing protocol. We will extend Merlin to support other publish/subscribe packages, e.g., Voyager[7]

2.4 "Paperless Test Stand"

One clear decision that has been made for ODE is the idea of a paperless test stand. Test results, configurations, and system changes can all be stored in an electronic logbook and accessed by remote collaborators. Electronic logbooks can readily catalog screen dumps (via hotkeys), output from digital cameras, scanners, text files, etc. Several Electronic Logbook packages, including the new Fermilab HepNRC logbook[10], are under evaluation, and we will select one shortly.

2.5 Future Components of ODE

As work begins to converge on final DA architectures, ODE will expand to meet the more advanced needs of data analysis and long term storage. Collider experiments have adopted Root[11] as both their offline and online physics analysis software package. For ODE, we will evaluate root as well as JAS and IDL.

Continuing to build on expertise and tools gained from Run II experiments, long term data storage will be implemented using the ENSTORE interface to the Fermilab mass storage systems[12].

3 Conclusions

New experiments at Fermilab are just now defining requirements and architectures. We are in the process of evaluating tools and technologies which can be used in the infrastructure for test stand and test beam development. ODE developers include software and hardware engineers who are working closely with the physics community in defining requirements. Although ODE is in its infancy of development at Fermilab, it is a natural progression of collaborative work with the collider experiments.

References

1 G.Oleynik et al, DART data acquisition system, CHEP'98, Chicago, Autumn 1998.
2 G.Oleynik et al, CHEP'97, Berlin, Spring 1997.
3 `http://www-b0:8000.fnal.gov`
4 `http://d0server1.fnal.gov/www/online_computing.html`
5 `http://www-ese.fnal.gov/ods/ode`
6 `http://www-b0:8000.fnal.gov/merlin`
7 `http://www.obectspace.com`
8 `http://www-b0:8000.fnal.gov/core/cdfmve/index.html`
9 `ftp://ftp.parc.xerox.com/pub/ilu/ilu.html`
10 `http://www.hep.net`
11 `http://root.cern.ch`
12 `http://www-hppc.fnal.gov/enstore/index.html`