REFERENCES

1. DELPHI Collaboration, P.Aarnio et al., The DELPHI Detector at LEP. Nucl. Instr. and Meth. A303 (1991) 233.
2. B.Franek, Ph.Charpentier, C.Gaspar, Ph.Gavillet, F.Harris, M.Jonker, Architecture of the DELPHI on-line Data Acquisition Control System, IEEE 1991 (Santa Fé, New Mexico-USA - 2-9 Nov. 1991).
3. T.Adye, A.Augustinus, M.Donszelmann, T.Rovelli, R.Sekulin, G.Smith, The Design and Operation of the Slow Controls for the DELPHI Experiment at LEP. - Nucl. Instr. and Meth. A349 (1994) 160 and RAL-94-029, CERN-ECP/94-3 reports.
4. C.Gaspar, M.Donszelmann, DIM - A distributed information management system for the DELPHI experiment at CERN. The IEEE Eight Conference REAL TIME ‘93 on Computer Applications in Nuclear, Particle and Plasma Physics (Vancouver, June 8-11 1993).
5. M.Donszelmann, C.Gaspar, J.Valls, A Configurable Motif Interface for the DELPHI Experiment at LEP MOTIF’92 The 2nd Annual International Conference On Motif Application Development and Use (Washington, D.C., 30 Nov.- 4 Dec., 1992).
6. D.E.Comer, Internetworking with TCP/IP. Volume I, ISBN 0-13-474321-0, Prentice-Hall International Editions.
7. S.Waldbusser, RFC1271 - Remote network monitoring management information base.
8. J.N.Albert, OPS5 pour la surveillance de VAX. Experience utilisateur en OPS5. Symposium DECUS France Session 3IA-14 (Avril 1993).
9. J.C.Mogul, Efficient Use of Workstations for Passive Monitoring of Local Area Networks ACM SIGCOMM’90 Symposium on Communications Architectures and Protocols WRL Research Report 90/5.
10. B.Tangney, D.O’Mahony, Local Area Networks and their Applications, ISBN 0-13-539560-7, 1988 Prentice-Hall International Editions.
11. R.L.Kruse, B.P.Leung, C.L.Tondo, Data Structures and Program Design in C. ISBN 0-13-725649-3, 1991 Prentice-Hall International Editions.
12. Y.Langsam, M.J. Augenstein, A.M. Tanenbaum, Data Structures Using C. ISBN 0-13-036997-7, 1991 Prentice-Hall International Editions.
13. D.E. Comer, D.L. Stevens, Internetworking with TCP/IP. Volume II, ISBN 0-13-465378-5, 1991 Prentice-Hall International Editions
14. J. Case, A. Rijsinghani, RFC1512 - FDDI Management Information Base.
15. D. McMaster, K. McCloghrie, RFC1516 - Definitions of Managed Objects for IEEE 802.3 Repeater Devices.
16. F. Kastenholz, RFC1643 - Definitions of Managed Objects for Ethernet-like Interfaces Types.
17. Pirelli MIB - Specification for 7000 Series Hubs - Issue N 1.07, Pirelli Data Networks.
18. R.Brun et al, Physics Analysis Workstation - CERN Program Library entry Q121.
19. -Simulation Workbench for Client-Server Architecture and Performance Modelling SWAP Eurêka Project: EU 1341.

5.3 The Central Analyzer

The Central Analyzer module (Figure 6) is where the complete monitoring information is analyzed when a problem has been observed in at least one of the five layers. Its first task is to deliver a rapid diagnostic as to whether or not the reported problem could endanger the DCCS running and, if necessary, to give the location of the faulty entity (Multiport Repeater, Disk,…) as retrieved from the DCCS Configuration Database. The second function aims at better identifying the source of the abnormality, on the basis that a problem caused by a defect or a process misbehavior frequently manifests itself in different DCCS layers. This analysis currently consists of searching for coincidences in the various Layer Analyzers logs, following receipt of a trigger. Such a correlation study is a typical type of application for an Expert System which would evaluate rules based on accumulated experience and thus help to pinpoint the cause of the trouble.

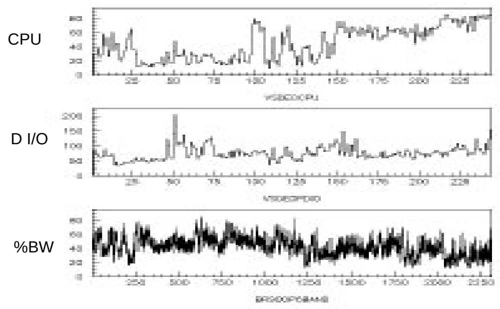


Figure 8: Example of trace plots of OSL and UCL variables

User interfaces similar to those of the Layer Analyzers are adapted to provide specific diagnostic information:

- The operator receives a Status (Warning, Alarm) indicating the severity of the fault.
- A complete error log is available for further investigation.

The GIN System is being commissioned and benchmarked on the DELPHI DCCS, in normal data taking conditions. Thanks to the facility to obtain all the DCCS diagnostic information from a single system, one can already follow precisely the activity of the various online domains.

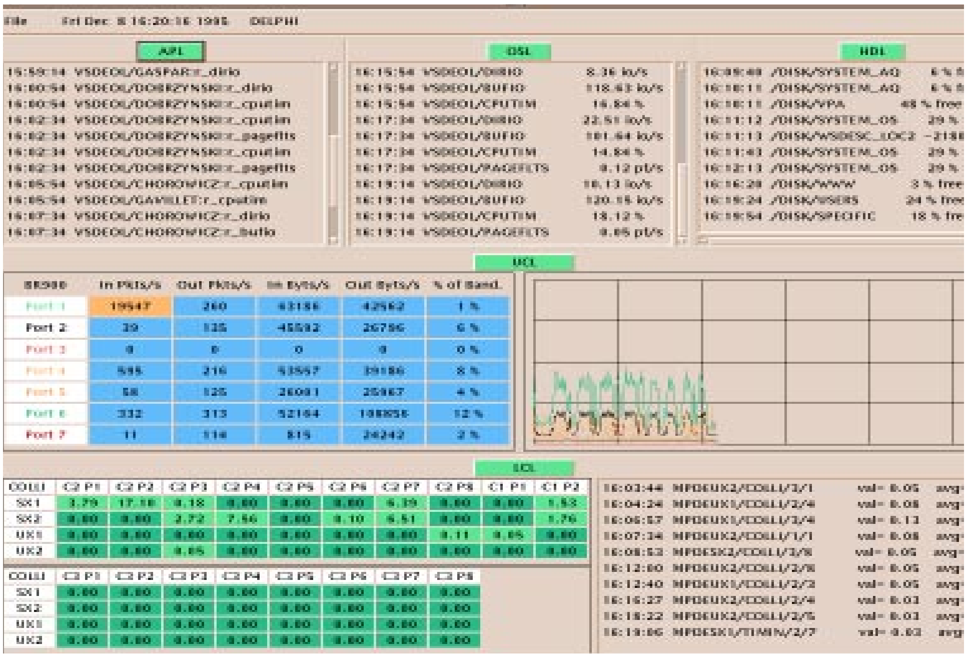
6 CONCLUSION

The complexity of HEP DCCSs and the variety of their distributed control tasks requires a coherent integrated approach to their monitoring. This problem has been recognized in other similar fields and Computer Software companies are already developing appropriate Monitoring Systems. Other initiatives include the recent Eurêka SWAP proposal [19] to promote a Software Engineering Workshop whose role would be to study the methods and tools required to master the performance of Large Distributed Information Systems. However, these projects should not prevent HEP online specialists from focusing their attention on the specific problems of the control of HEP experiments in order to guarantee that future systems will be well adapted to their requirements.

The information collected from the UCL and LCL network layers is structured *à la* RMON. For this purpose a standard SNMP library based on the Carnegie-Mellon University SNMPLIB has been developed [13]. The MIB part includes the necessary extensions [14, 15, 16, 17] for the network hardware used.

5.2 The Layer Analyzer

The data accumulated from each layer are available both for interactive display and more automated monitoring. The interactive mode enables the behavior of each layer to be followed in detail by means of MOTIF displays (Figure 7). A facility aimed at the System Expert allows the capture, via DIM, of data from any layer over a given time interval and the production of ntuples (Figure 8) for subsequent analysis with PAW [18]. In addition an Average Analyzer (AVGANA) provides an automatic detection of problems. Each new datum is compared with its moving average value (computed over the last N samplings) and may activate a trigger which is passed to a Central Analyzer.



```
15:59:14 VSDEOL/GASPAR::r_dirio
16:00:54 VSDEOL/DOBRZYNSKI::r_dirio
16:00:54 VSDEOL/DOBRZYNSKI::r_cputim
16:02:34 VSDEOL/DOBRZYNSKI::r_cputim
16:02:34 VSDEOL/DOBRZYNSKI::r_pageflts
16:02:34 VSDEOL/DOBRZYNSKI::r_cputim
16:02:34 VSDEOL/DOBRZYNSKI::r_pageflts
16:05:54 VSDEOL/CHOROWICZ::r_cputim
16:05:54 VSDEOL/GAVILLET::r_cputim
16:07:34 VSDEOL/CHOROWICZ::r_dirio
16:07:34 VSDEOL/CHOROWICZ::r_bufio
```

```
16:15:54 VSDEOL/DIRIO         8.36 io/s
16:15:54 VSDEOL/BUFIO       118.43 io/s
16:15:54 VSDEOL/CPUTIM       16.84 %
16:17:34 VSDEOL/DIRIO        22.51 io/s
16:17:34 VSDEOL/BUFIO       101.64 io/s
16:17:34 VSDEOL/CPUTIM       14.84 %
16:17:34 VSDEOL/PAGEFLTS      0.12 pf/s
16:19:14 VSDEOL/DIRIO        10.13 io/s
16:19:14 VSDEOL/BUFIO       120.15 io/s
16:19:14 VSDEOL/CPUTIM       18.12 %
16:19:14 VSDEOL/PAGEFLTS      0.05 pf/s
```

```
16:09:40 /DISK/SYSTEM_AQ        6 % f
16:10:11 /DISK/SYSTEM_AQ        6 % f
16:10:11 /DISK/VPA            48 % free
16:11:12 /DISK/SYSTEM_OS       29 %
16:11:13 /DISK/WSDESC_LOC2   -218
16:11:43 /DISK/SYSTEM_OS       29 %
16:12:13 /DISK/SYSTEM_OS       29 %
16:16:20 /DISK/WWW           3 % free
16:19:24 /DISK/USERS        24 % free
16:19:54 /DISK/SPECIFIC      18 % fre
```

SR900	In Pkts/s	Out Pkts/s	In Byts/s	Out Byts/s	% of Band.
Port 1	19547	260	63156	42562	1 %
Port 2	39	135	45592	26796	6 %
Port 3	0	0	0	0	0 %
Port 4	595	216	53557	39186	8 %
Port 5	58	125	26081	25967	4 %
Port 6	332	313	52164	108858	12 %
Port 7	11	114	815	24242	2 %

COLLI	C2 P1	C2 P2	C2 P3	C2 P4	C2 P5	C2 P6	C2 P7	C2 P8	C1 P1	C1 P2
SX1	3.79	17.18	0.18	0.00	0.00	0.00	6.39	0.00	0.00	1.53
SX2	0.00	0.00	2.72	7.56	0.00	0.10	6.51	0.00	0.00	1.76
UX1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.05	0.00
UX2	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00

COLLI	C2 P1	C2 P2	C2 P3	C2 P4	C2 P5	C2 P6	C2 P7	C2 P8
SX1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SX2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
UX1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
UX2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

```
16:03:44 MPDEUX2/COLLI/3/1      val= 0.05   avg=
16:04:24 MPDEUX1/COLLI/2/4      val= 0.08   avg=
16:06:57 MPDEUX1/COLLI/3/4      val= 0.13   avg=
16:07:34 MPDEUX2/COLLI/1/1      val= 0.08   avg=
16:08:53 MPDESX2/COLLI/3/8     val= 0.05   avg=
16:12:00 MPDEUX2/COLLI/2/8      val= 0.05   avg=
16:12:40 MPDEUX1/COLLI/2/3      val= 0.05   avg=
16:16:27 MPDEUX2/COLLI/2/4      val= 0.03   avg=
16:18:22 MPDEUX2/COLLI/2/5      val= 0.03   avg=
16:19:06 MPDESX1/TIMIN/2/7      val= 0.03   avg
```

Figure 7: MOTIF Display

Others parameters include the resources used, such as data transfer rates and disk space utilization.

• Upper Communication Layer (UCL). This is the highest layer of the various network segments, originating from Bridges, Switches,… e.g the Ethernet Collision Domains or FDDI rings. The variables of each segment are: Availability (Yes/No), Bandwidth use, I/O rates in frames/s & bytes/s. (This could be extended to the full RMON standard).

• Lower Communication Layer (LCL). This is the layer of individual network sub-segments, originating from Multiport Repeaters, Concentrators, Hubs, etc. For each such device the variables are, according to the media specification: Collisions, Noise, Short and misaligned frames, framing errors, CRC errors, ring status, ring reconfiguration times, etc.

5 IMPLEMENTION: THE GENERAL INFORMATION MONITOR (GIN) SYSTEM

Based on the above proposal we are developing a General Information moNitor (GIN) System for the DELPHI DCCS. Figure 6 gives a schematic view of GIN. Monitoring of each of the five layers is performed in two parts. The Information Collector is responsible for gathering the monitoring data and the Layer Analyzer treats the collected data. The data from each the five layers are also passed to a Central Analyzer. GIN is configured by means of a Configuration Database that describes the various system components.

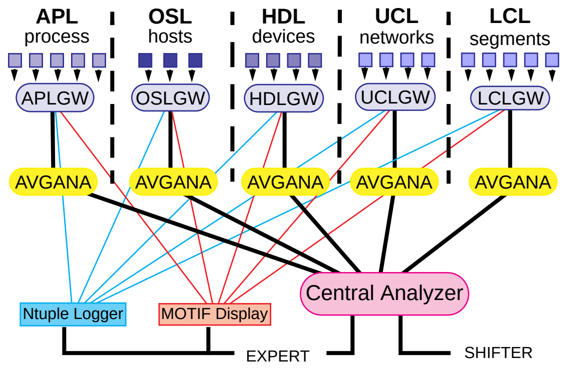


Figure 6: GIN System

5.1 *The Information Collector*

The data are acquired by means of distributed collectors, at a sampling frequency appropriate to each layer [9] and chosen to minimize the load induced on the DCCS. This data are published using DIM as the Information Manager.

The overall monitoring information represents about 10,000 data items. Each item is associated a Name constructed from well defined Naming Conventions [10]. Internally the various data structures are handled by means of a dedicated MEMLIB Memory Manager [11, 12].

For the APL, OSL and HDL layers the information about the various VMS entities (processes, devices, etc.) is gathered via calls to the standard VMS System Service routines embedded in a special interface library (VMSLIB).

resources used by applications. For example, in DELPHI Online we have used the following Digital tools: MONITOR, AMDS, DECPS. The main problem arises from the tools being specific to the manufacturers Operating System and thus forcing the use of as many sets of packages as there are Operating Systems present in the DCCS.

The commercial packages for the monitoring of network equipment are more universal. Low level standards have been defined such as the Simple Network Management Protocol (SNMP Version 1 and 2) [6] and the Remote Monitoring (RMON) [7] of the Management Information Base (MIB) [6]. A variety of products exist, from simple Segment Manager e.g LTM: LAN Traffic Monitor from Digital, up to tools incorporating the administration, configuration and monitoring of networks such as NetView[a], Openview[b], PolyCenter[c], etc.

Home made tools have often been produced to provide functionality not offered by commercial products. For example, the Central Cluster Panel (CCP) [8], developed jointly by Digital and DELPHI is a package that monitors the activity of the VMSCluster part of the DELPHI DCCS. Its scope is to assist operators in spotting abnormal computer behavior which could endanger the safe running of the DCCS. CCP has been found very useful in that it displays the monitoring of a complete VMSCluster on a single screen and can draw attention to the state of a critical online process. However, CCP has been found insufficient in that the activity of the network and devices (disks, tapes, etc.) is not monitored.

The current situation can be summarized by the existence of a wide range of products although none of them with the complete functionality required to monitor a DCCS. The problem is particularly acute for Client-Server applications where performance can only be guaranteed by a continuous monitoring and control of the entire DCCS.

4.2 Integrated Monitoring Approach

In order to achieve complete monitoring we first propose a model of the DCCS as a homogenous system composed of five layers. The aim is to provide a consistent view of the DCCS integrating computer nodes, application processes and the network. At each level characteristic parameters are defined that describe the System’s behavior.

• APplication Layer (APL). This first layer is composed of all application processes at work during normal operation. Each application is characterized by means of variables such as: State (of the associated SMI Object), process CPU, I/O and Page Fault rates.

• Operating System Layer (OSL). This deals with the resources provided by the Operating System. Typical parameters are: Global CPU rate, Memory requests per unit of time, I/O rate.

• Hardware and Device Layer (HDL). This includes the Computer Hardware (CPU, Memory, Network and Device Interfaces) and the Devices themselves (Disks, Tape units,…). Their physical state is evaluated in terms of Availability (yes/no) and Error rate.

[a] NetView is a trademark of the International Business Machines company.

[b] Openview is a trademark of the Hewlett Packard company.

[c] PolyCenter is a trademark of the Digital Equipment Corporation company.

the detector electronics.
- Real time performance (seconds to minutes).
- High I/O bandwidth and specific I/O time profile.
- Safety constraint in the Slow Control area.
- High efficiency and reliability imposed by the high operating costs of the Detector and Accelerator.
- Need of regular performance optimization.
- Adaptability to the:

Local evolving environment driven by the:

- Luminosity increase (e.g LEP).
- Rapid evolution of the computer and network market stimulating cost effective upgrades through the integration of new Workstations and Servers along with the adoption of new network standards (e.g. FDDI,ATM) and modern equipment (Routers, Switches, etc.)

For comparison one should keep in mind that the DCCSs at LHC era, will scale from current ones by more than one order of magnitude in size and probably in complexity!

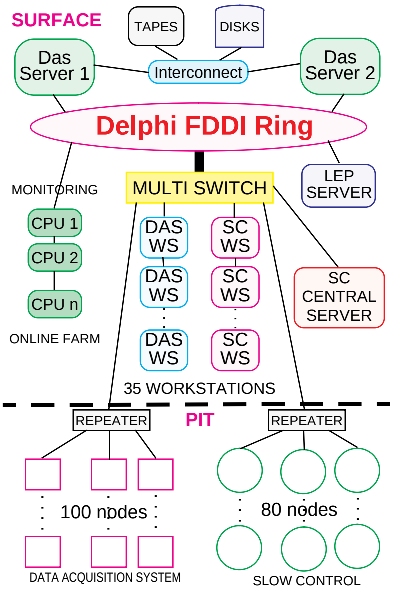


Figure 5: DELPHI DCCS

After more than five years of DELPHI running a variety of problems have been encountered ranging from equipment failure, Operating System and Network protocol errors up to hang situations caused by the applications. In these conditions one can easily infer the importance of the DCCS monitoring from both the point of view of stable running and performance optimization.

4 DCCS MONITORING

Here we briefly discuss the way today DCCSs are monitored in order to better appreciate the need for a new approach.

4.1 Current Monitoring Methods

The monitoring of a DCCS is normally based on combination of commercial and home made packages with the computer nodes and LAN often treated separately.

Commercial products are usually provided by the computer manufacturers. For the computer nodes, these tools address the monitoring of the Operating System and survey resources used by the applications. Resource analyzers enable the optimization of

- Across the domains:

  Ex.: Stop data taking if the Slow Control State of Detector j is Faulty.

For what concerns us here, one should note the distributed nature of the State Manager Interface which typically has one associated process per domain and partition.

2.4 *Information Management and Distribution*

All the online domains are providing and requesting information (most of it in real time) for several purposes:

- Establishing and maintaining interprocess communication across the DCCS.
- Coordinating the execution of distributed tasks.
- Providing the end user with an up-to-date status of the system operation.

The implementation of such a communication system is a typical example of a Client-Server application. In DELPHI, a Distributed Information Management System (DIM) has been developed [4], in which any process can become a DIM Server if it has to publish some information and/or a DIM Client if it needs information from one of the available services.

DIM is responsible for most of the communications inside the DELPHI Online System, offering information on about 15,000 Services provided by about 300 Servers.

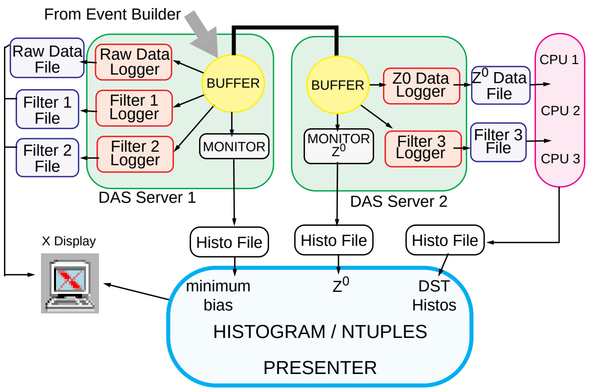


Figure 4: DELPHI Central Quality Checking

2.5 *User Interface*

Amongst others requirements the User Interface used in the DCCS domains should be able to access any information from any process of any domain and should allow the User freedom to chose which display to use and what information to see. The obvious choice is a MOTIF based User Interface using the Information Manager (DIM in our case) for the communication. The DELPHI User Interface (DUI) [5] developed for the DELPHI Online is another highly distributed facility.

3 THE DISTRIBUTED COMPUTER CONTROL SYSTEM (DCCS)

Figure 5 shows the DELPHI DCCS Architecture. This illustrates both the distributed architecture and the heterogeneity of today HEP DCCSs. From the above review of the DCCS tasks, one can more precisely identify its characteristics and required properties:

- Complexity: e.g. Several (more than 500) intercommunicating processes at work in 40 Workstations and Servers. About 200 Embedded Processors ensuring the control of

The Communication between the Detector and the Accelerator is simply implemented as a Client-Server application through which:

- The Experiment site gets updates of the beam and machine parameters from the Accelerator Running Data Base.

- The Experiment site sends updates of the luminosity, the background intensities as collected in the Experiment Online Database.

2.2 *Monitoring and Quality checking*

The task of verifying the data naturally follows. This aims to monitor the integrity and quality of the data acquired by the DAS, Slow Control and Accelerator Systems and, in the event of a problem being detected, calls for the necessary corrective actions.

This verification is performed in successive steps, first at the detector local level, in the partition workstation and then more centrally, when the detector data can be viewed as a whole e.g. at the physics event level. Figure 4 shows the organization of the DELPHI Central Quality Checking. One should note its distributed nature. The events, gathered from the detector, are passed through selective data loggers running in the DAS Central Servers which dispatch filtered streams where needed for specific analysis. An X11 based Presenter allows all results (statistics, histograms) to be merged.

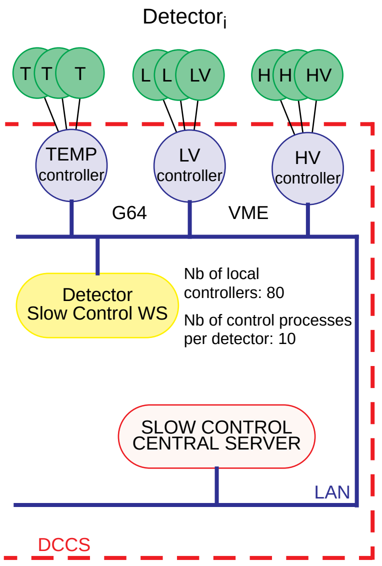


Figure 3: Slow Control System

Another characteristic of monitoring tasks is their continual change and growth as the experiments strives to improve the detector calibrations and the accuracy of the data used for physics analysis.

2.3 *Process Control and Coordination*

In the DELPHI experiment, the DCCS domains (DAS, Slow Control and LEP Communication) are coordinated through a so-called State Manager Interface (SMI) [2]. The functions of the various Control Processes (Readout, Run Control,...) are associated an Object characterized by its State (Ready, Running,...). The possible interaction between the various Objects is expressed using a formal language (State Manager Language) which allows the definition of AI-like rules by which each Object can specify logical conditions based on the States of other Objects. This system allows to coordinate the actions and commands:

• Within a domain:

Ex.: Do not start Data Acquisition if Detector j is Not Ready.

2 THE ONLINE OF HEP EXPERIMENTS

The principal domains of activity of an HEP DCCS are the Control and Acquisition of the detector data, the detector technical parameters (often called Slow Control) and Communication with the Accelerator Control System.

Although the volume of data and the acquisition rate are very different from one domain to another, the overall organization of the tasks remains essentially the same. Here we will describe the organization of the DELPHI experiment [1], but this can be taken as typical for any current large HEP Detector. Figure 1 gives a schematic view of the tasks and their relationships within DELPHI.

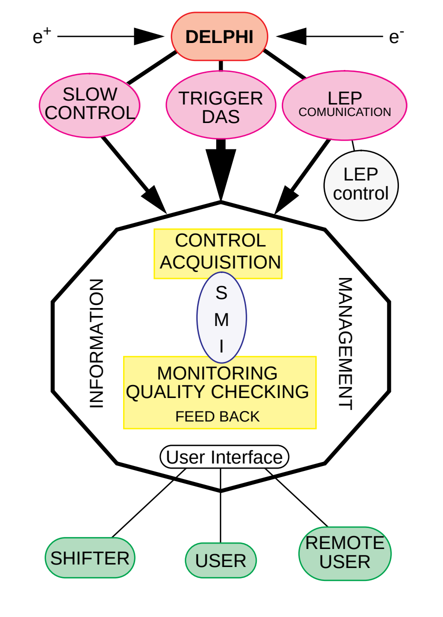


Figure 1: DELPHI Online Organization

2.1 *Control and Acquisition*

The real time control and data acquisition is the basic task, of highest priority. Let briefly review its characteristics from the point of view of the DCCS.

Figure 2 shows the typical tree like architecture of current Data Acquisition Systems [2]. From the values quoted one can note that these systems are typically composed of 20 detector partitions, each one comprising about five Embedded Controllers which are supervised locally by some five Control Processes in one workstation.

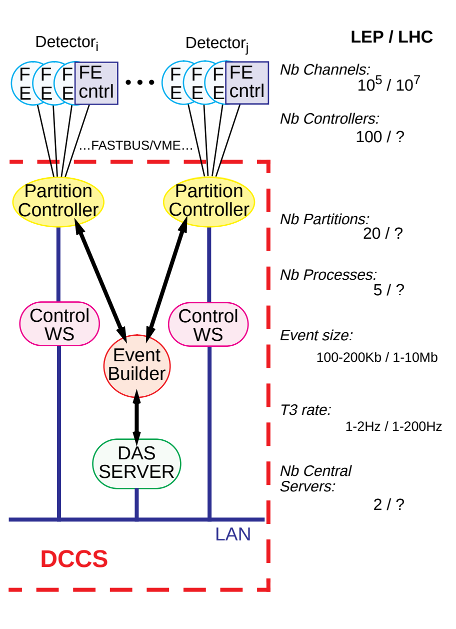


Figure 2: Data Acquisition System

Figure 3 gives the Slow Control architecture [3]. Every detector partition controls about four embedded controllers, dedicated to one or more surveys (High Tensions, Temperature,...). The necessary bandwidth is low so that the data accumulated in the embedded controllers can be transferred to the local workstation and the Central Server via the LAN. On the other hand one should notice that there are often more than ten autonomous processes controlling the various families of technical parameters.

ON THE RELEVANCE OF EFFICIENT, INTEGRATED COMPUTER AND NETWORK MONITORING IN HEP DISTRIBUTED ONLINE ENVIRONMENT

D.CARVALHO[1, 2], Ph. GAVILLET[1], V.DELGADO[1, 5], J.N.ALBERT[4], N.BELLAS[2],
J.JAVELLO[1],Y.MIERE[1], D.RUFFINONI[1], G.SMITH[3]

[1]CERN - CH-1211 Genève 23, Suisse; [2]Universidade Federal do Rio de Janeiro - Ilha do Fundão 21945-970 Brasil; [3]Rutherford Appleton Laboratory, Chilton GB; [4]LAL - Université de Paris-Sud Bâtiment 200, Paris, Orsay France; [5]ETSIT - Ciudad Universitaria 28040 Madrid, España

Large Scientific Equipments are controlled by Computer Systems whose complexity is growing driven, on the one hand by the volume and variety of the information, its distributed nature, the sophistication of its treatment and, on the other hand by the fast evolution of the computer and network market. Some people call them generically Large-Scale Distributed Data Intensive Information Systems or Distributed Computer Control Systems (DCCS) for those systems dealing more with real time control. Taking advantage of (or forced by) the distributed architecture, the tasks are more and more often implemented as Client-Server applications. In this framework the monitoring of the computer nodes, the communications network and the applications becomes of primary importance for ensuring the safe running and guaranteed performance of the system. With the future generation of HEP experiments, such as those at the LHC in view, it is proposed to integrate the various functions of DCCS monitoring into one general purpose Multi-layer System.

1 INTRODUCTION

Distributed Computer Control Systems (DCCS) are nowadays common in several sectors such as Communication, Aerospace, Defense, Transportation and over the last few years, in the Online Systems of High Energy Physics Experiments. They are normally organized around a few Central Servers and many Workstations and/or PCs in charge of local controls. In the HEP environment the architecture includes a layer of so-called Embedded Systems responsible for the control of the detector electronics under the supervision of one of the workstations. All these computer nodes are connected to a backbone LAN.

Stringent requirements are usually put on DCCSs, such as the availability of large CPU power, high I/O bandwidth, real time performance, reliability, fault tolerance and adaptability. Furthermore the DCCS has to accomplish a complex set of tasks with demands that can often conflict. The continuous monitoring of the entire system is required for tuning and ensuring reliability.

The presentation, after reviewing the characteristics of the HEP Online environment and its associated DCCS, describes how current monitoring methods should be evolving towards an integrated monitoring approach in which all the aspects of the DCCS running are considered.

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN/ECP 96-01
17 January 1996

ON THE RELEVANCE OF EFFICIENT, INTEGRATED COMPUTER AND NETWORK MONITORING IN HEP DISTRIBUTED ONLINE ENVIRONMENT

D.CARVALHO[1, 2], Ph. GAVILLET[1], V.DELGADO[1, 5],
J.N.ALBERT[4], N.BELLAS[2], J.JAVELLO[1], Y.MIERE[1], D.RUFFINONI[1],
G.SMITH[3]

[1]CERN - CH-1211 Genève 23, Suisse;
[2]Universidade Federal do Rio de Janeiro - Ilha do Fundão 21945-970
Brasil;
[3]Rutherford Appleton Laboratory, Chilton GB;
[4]LAL - Université de Paris-Sud Bâtiment 200, Paris, Orsay France;
[5]ETSIT - Ciudad Universitaria 28040 Madrid, España

Abstract

Large Scientific Equipments are controlled by Computer Systems whose complexity is growing driven, on the one hand by the volume and variety of the information, its distributed nature, the sophistication of its treatment and, on the other hand by the fast evolution of the computer and network market. Some people call them generically Large-Scale Distributed Data Intensive Information Systems or Distributed Computer Control Systems (DCCS) for those systems dealing more with real time control. Taking advantage of (or forced by) the distributed architecture, the tasks are more and more often implemented as Client-Server applications. In this framework the monitoring of the computer nodes, the communications network and the applications becomes of primary importance for ensuring the safe running and guaranteed performance of the system. With the future generation of HEP experiments, such as those at the LHC in view, it is proposed to integrate the various functions of DCCS monitoring into one general purpose Multi-layer System.

Presented at *Computing in High Energy Physycs ‘95* (September 18-22, 1995; Rio de Janeiro, Brazil.)