D0 Note #2935

DØ Run 2 Online Computing

Tasks and Schedules

4 May 1996
Stuart Fuess

Table of Contents

Introduction

The DØ detector is being upgraded in preparation for Run 2 of the Fermilab Tevatron Collider. In anticipation of higher data acquisition bandwidth requirements, the online computing system is being revised. This note deals explicitly with the *host* portion of the computing system, which is the section which deals with the logging of event data after it is transferred from the Level 3 software filter layer, the monitoring of that event data, the control and monitoring of the detector elements, and the general control of the experiment.

DØ had ample experience in the operation of a data acquisition system during Run 1. The online computing system was particularly reliable in this period, with an integrated downtime resulting from online host system problems being less than 1% of the total. The concepts and philosophy of the Run 1 system which so successfully operated are being carried over to the Run 2 system design.

The principal task of the online computing system is to control and record the event data stream. The parameters of the Run 2 trigger system define the data rate to be 20 Hz with an event size of 250 Kbytes. The steady state event bandwidth is thus 5 Mbytes/sec. To commission and calibrate the detector, it is also desired to have a burst capability of 1 kHz with a much reduced event size, possibly giving a short-term rate of 25 Mbytes/sec. In Run 1 the event data was recorded at approximately 2.2 Mbytes/sec, so the Run 2 upgrade requires a modest factor of 2 to 3 in increased steady state performance.

Hardware Architecture

In choosing the hardware components of the online computing system, the philosophy is to use commodity components wherever possible. The system is thus constructed of common processors connected by a standard network. Performance and reliability are achieved by redundancy, giving parallel paths through the system.

The event data arrives at the host system on the data cable from the Level 3 system. There is a VME standard interface to the data cable. The connection to the host node(s) is then by an adapter between the VME bus and the processor bus. We have targeted PCI as the most common processor bus, and hence expect this connection to be achieved by a VME-PCI bus adapter.

Figure 1 illustrates the architecture of the Run 2 online computing system. There are four sections tied together by a standard network:

- *Trigger and detector control and monitoring processors*

  These components, external to the *host* system, are either part of the trigger system or are front ends closely associated with downloading, control, and monitoring of the detector. The event data path through the trigger system and into the DAQ nodes is on a custom data cable bus, as used in Run 1 of the experiment. Control and monitoring communication with the processors in the trigger system will be via their standard network interfaces.

  Existing front end processors which will be reused in Run 2 have a Token Ring network connection. Access from the host system to these processors will be via a Token Ring bridge or gateway, as was done in Run 1. New front end processors will have Ethernet interfaces, and hence will be directly accessible.

  There are no planned acquisitions in this regime as part of the online computing upgrade. Changes to the processors in the trigger system or to front ends are considered in the appropriate relevant section.

- *Data acquisition processors*

  The principal task of these DAQ nodes will be to receive the event data from the Level 3 software trigger layer and to ship the events to a disk buffer in the logging nodes. Several processors may occupy either the same or distinct data cables from the Level 3 system. Software running on these nodes will guide the event data to its proper location in preparation for logging.

  As the principal operation in the DAQ nodes is event data transfer, they must be capable of handling the expected 5 Mbytes/sec data rate. Since they will also interface to the VME crates with the expected PCI to VME bus adapter, they clearly must also support the PCI bus. Additionally, a driver for the selected adapter must be available on the selected DAQ node.

  The output path for the event data will be via a standard network interface. Each node will be capable of handling the full event data bandwidth of 5 Mbytes/sec. Currently, FDDI network adapters would be capable of such rates.

- *Logging nodes*

  Event data will be written to disk on the logging nodes, then collected and written to serial media. The logging nodes will also provide access to some fraction of the data for event monitoring. The incoming path for the event data is again by a standard network interface. As with the DAQ nodes, each node should be capable of handling the full 5 Mbytes/sec network bandwidth. The logging nodes will need to spool the event data to disk, read it back, and deposit it on serial media. To achieve the desired I/O bandwidth each node will have multiple I/O busses. Some fraction of the event data will also be passed back over the network to the monitoring nodes.

The current plan is to use standard SCSI disk and tape devices. The SCSI-2 Fast and Wide bandwidth should be adequate. Higher bandwidth can be achieved by adding more controllers, so the logging nodes must be capable of supporting multiple I/O controllers.

The separation of DAQ and Logging nodes into distinct network-connected processors is not required. A single machine, possibly with multiple processors, could conceivably perform the DAQ and logging functions, possibly with higher bandwidths. However, the multiple-node solution is more robust, allowing for failure of individual components without compromising data acquisition. Additionally, component upgrades may be financially and technically easier with a distributed architecture. Finally, with multiple nodes the software tasks are distributed, possibly allowing each processor to be configured optimally for a task.

- *Control and monitoring nodes*

  User interaction with the experiment will be via a set of nodes which will host the various control and monitoring applications. The most resource-intensive activity on these nodes will be the event data monitoring processes. These processes are likely to be limited by CPU power, but also require a substantial network bandwidth for input of the event data information.

- *Network*

  The sections of the online system are tied together by a standard network. The main requirement of the network is that it be capable of handling the net bandwidth for event data transfer in addition to various communication and monitoring activities. A switched network may be best to isolate the various online functions in addition to providing the greatest aggregate bandwidth. A switched FDDI network is currently capable of providing adequate capacity. Lower bandwidth portions of the online system (the trigger systems, front ends, and monitoring nodes) need not be on a switched or high-bandwidth section of the network, but instead may be serviced by bridges and hub ports.

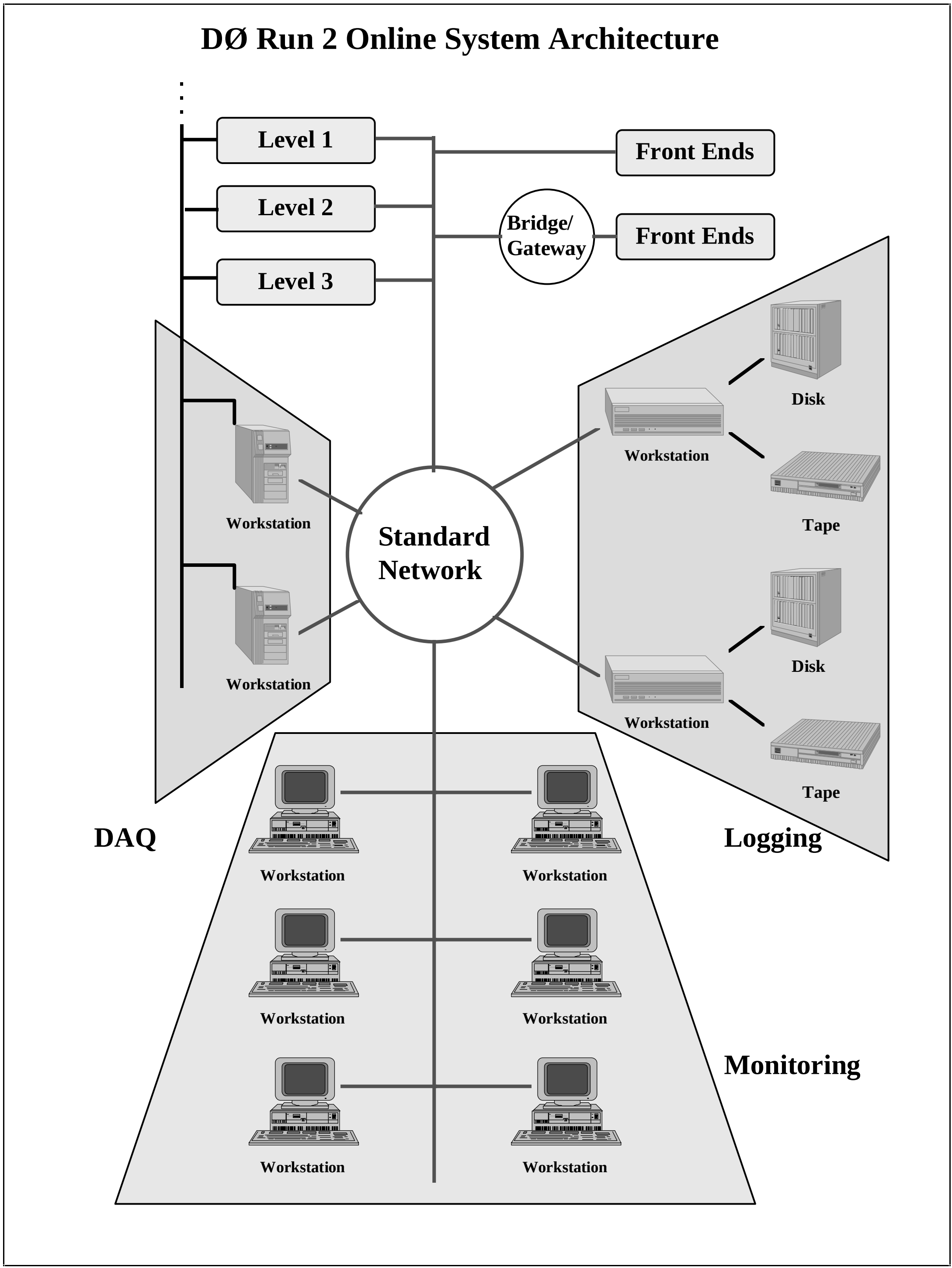

Figure 1. Online System Hardware Architecture

Hardware Acquisition

The WBS worksheet for the online computing system is shown in Figure 2. It includes a breakdown of acquisitions into 3 years, FY97 through FY99. Computing purchases are planned as late as possible in order to optimize the performance to cost ratio.

WBS	**1.5.1**	**ONLINE COMPUTING**										
Revision:	1 (Fuess)	September 30, 1994										
WBS	**ITEM**	**MATERIALS & SERVICES (M&S)**							**CONTINGENCY**		
1.5.1	**ONLINE COMPUTING**	**Unit**	**#**	**Unit Cost**	**M&S TOTAL**	**FY97**	**FY98**	**FY99**	**%**	**Cost**	**TOTAL Cost**
1.5.1.1	***Workstations***				260000	55000	55000	150000	17.69	46,000	306000
1.5.1.1.1	DAQ Workstations (CPU,mem,netwk,IO,soft)	ea	4	35000	140000	35000	35000	70000	20	28,000	168000
1.5.1.1.2	Monitoring Workstations	ea	6	20000	120000	20000	20000	80000	15	18,000	138000
1.5.1.2	***Network (switch,bridge,router)***	lot	1	80000	80000	0	80000	0	25	20,000	100000
1.5.1.3	***Disk/tape Peripherals***				90000	15000	15000	60000	25	22,500	112500
1.5.1.3.1	Disk	Gb	60	500	30000	5000	5000	20000	15	4,500	34500
1.5.1.3.2	Tape	ea	12	5000	60000	10000	10000	40000	30	18,000	78000
1.5.1.4	***Printers/Monitors/Xterms/PCs***	lot	1	75000	75000	10000	15000	50000	5	3,750	78750
1.5.1.5	***Software***	lot	1	100000	100000	10000	40000	50000	25	25,000	125000
	TOTAL ONLINE COMPUTING				605,000	90,000	205,000	310,000	19.38	117,250	722,250

Figure 2. WBS Worksheet for Online Computing System

Software Architecture

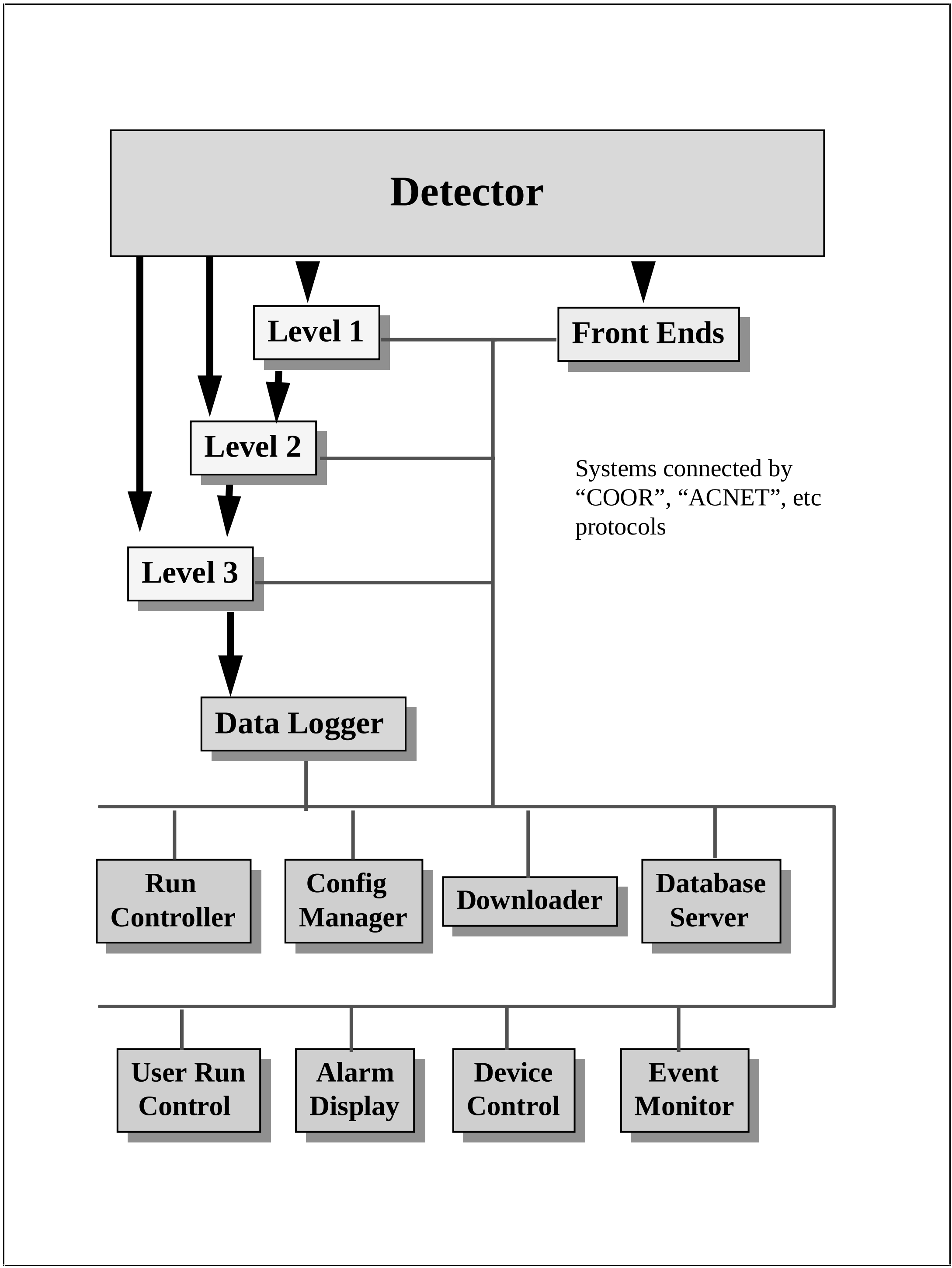

Figure 3. Online System Software Architecture

Software Task Definitions

0. General tools and frames

0.0 *Architecture*

Definition of hardware architecture, operating system, languages, and compilers.

0.1 *Generic interprocess communication*

A set of IP-based task-to-task communication routines. The **`ITC`** and **`ELNCON`** packages are the Run 1 prototypes. The **`D0IP`** package is the planned Run 2 version.

0.2 *Generic Client/Server framework*

A framework main program which is based upon the generic interprocess communication package. This is principally for use with host based applications. The framework should be supported on any platform where multiple similar applications are to exist. The program should be multi-threaded to ease programming of asynchronous events. The **`CLSPKG`** package is the Run 1 prototype for this framework.

0.3 *User interface tools*

In order to make client GUI interfaces as uniform as possible, a set of tools should be developed and shared.

0.4 *Database management tools*

Numerous applications will need associated databases. A single database choice should satisfy all needs. Tools to manipulate the database should be provided. The Run 1 prototype databases used ***RDB*** or ***DBL3***.

1. Level 1

1.1 *Level 1 framework*

Determine the OS interface, task list, and task structure. Initial implementation of exerciser and diagnostic software.

1.2 *Level 1 configuration management and monitoring*

Manage the communication with the host to set configuration parameters.
Provide local and host level monitoring functions.

- **1.2.1** Initial release to operate Level 1 Framework via host control
- **1.2.2** Release with auto rate control, information serving, and database logging
- **1.2.3** Release to jointly operate Level 1 FW and Level 1 CAL with Level 2

2. Level 2

2.1 *Level 2 framework*

Determine the OS interface, task list, and task structure. Initial local and host tasks. Initial implementation of exerciser and diagnostic software.

2.2 *Level 2 configuration management and monitoring*

Manage the communication with the host to set configuration parameters. Provide local and host level monitoring functions.

- **2.2.1** Release to operate Level 1 and Level 2 Framework
- **2.2.2** Release with auto rate control, information serving, and database logging
- **2.2.3** Release to jointly operatoe Level 2 FW and Level 2 Global processor with Level 1
- **2.2.4** Muon: CNAPS-128 parallel processor

2.2.5 Calorimeter: Alpha trigger processor boards
2.2.6 Tracking: DSP processor boards
2.2.7 Global: Alpha trigger processor board

2.3 *Level 2 data transport*

Manage the flow of data into and out of the Level 2 processors.

2.4 *Level 2 filter code*

Manage the loading of Level 2 filter code.

3. Level 3

3.1 *Level 3 framework*

Determine the OS interface, task list, and task structure. Initial local and host tasks.

3.2 *Level 3 configuration management*

Manage the communication with the host to set configuration parameters.

3.3 *Level 3 data transport*

Manage the flow of data into and out of the Level 3 processors.

3.4 *Level 3 filter code*

Manage the loading of Level 3 filter code.

3.5 *Level 3 monitoring*

Provide local and host level monitoring functions. This includes many of the functions of the Run 1 prototypes ***Supervisor*** and ***Surveyor***.

4. Configuration management

4.1 *Configuration management*

Using a well prescribed configuration description, manage the detector configuration state and control configuration changes. The Run 1 prototype for this task is the **COOR** task.

4.2 *Configuration downloading*

Implement requested configuration changes. The Run 1 prototype is a combination of the **COOR** and **COMM_TKR** tasks, and the specialized muon download tasks.

4.3 *Configuration recording*

Log configuration changes. The Run 1 prototype is **COOR** with the ***Tape Log Database***, plus **COPYCFG**.

4.4 *User configuration control*

The user interface to configuration management. The Run 1 prototype is **TAKER** communicating with the **COOR** task.

5. Run control

5.1 *Run state management*

Maintain the current state of runs and manage run state changes. The Run 1 prototype is **COOR** communicating with the Level 1 ***trigger control*** process and the Level 2 ***Supervisor***.

5.2 *User run control*

The user interface to run state management. The Run 1 prototype is **TAKER** communicating with the **COOR** task.

5.3 *Run information recording*

Log run activity. The Run 1 prototype is **COOR** with the ***Tape Log Database***.

6. Data logging

6.1 *Host data transfer*

Synchronize data transfer from Level 3 to Host. The Run 1 prototype is the **Data Logger** process.

6.2 *Data spooling*

Transfer data from Host receiver to final repository. In Run 1 this activity was performed by the **Data Logger** in concert with the **TWSERVER**, the **TWSLAVE**, and the **COPIER** tasks.

6.3 *Data information recording*

Log data file information. In Run 1 the **Data Logger** recorded information in the ***Tape Log Database***.

7. Calibration

7.1 *Front End calibration tasks*

Complete portion of calibration procedure resident in detector front end systems.

7.2 *Calibration databases*

Determine form of databases. Provide utilities for viewing and manipulating databases. The Run 1 duties were performed by ***DBL3***.

7.3 *Calibration routines*

Provide the routines to be executed in front ends, Level 2 or Level 3, or the Host. Provide the communication and downloading paths. In Run 1 this was the complete **CALIB** suite of programs.

8. DAQ monitoring

8.1 *Global monitoring*

Consolidate the monitoring functions for the individual DAQ components. Provide tools and user interface. This function was performed by the **GM_SERVER** and **GM_CLIENT** in Run 1.

9. Event monitoring

9.1 *Event data pool*

Provide the mechanism whereby events from the data stream are made accessible to monitoring tasks. The Run 1 prototype was **DAQFIL** and the global shared common. The monitoring of the data pool was performed by the **POOLBAR** task.

9.2 *Event monitoring processes*

Construct the various processes which take information from the event data pool and perform monitoring at various levels, from hardware performance to reconstructed physics quantities. In Run 1 this was performed by the various **EXAMINE** processes.

10. Accelerator interface

10.1 *Accelerator communication*

Generalize the tools to receive and send information from/to the accelerator, and to make this information available to external tasks. In Run 1 this was performed with a global shared common with the **IMPORT_DAQ** and **RDAQ$SERVER** processes.

10.2 *Accelerator data monitoring and logging*

Acquire accelerator data, monitor and display such, and log it to a database as necessary. In Run 1 this task was performed by **D0LUM_SERVER** and a ***DBL3*** database.

11. Control and monitoring

11.1 *Front ends*

Provide the code to execute in the control path front end processors. The Run 1 equivalents were the Shea/Goodwin 68K systems, the HV systems, and aspects of the muon front end processors.

11.1.1 Front end framework, EPICS or otherwise.

11.1.2 Front end VME-path data acquisition, principally for Test Beam needs.

11.1.3 Front end detector-specific tasks, principally Silicon detector monitoring tasks.

11.1.4 Front end with 1394-based monitoring

11.1.5 Front end with 1553-based monitoring

11.2 *Control path communication*

Provide the tools for communicating with and controlling control and monitoring path elements. In Run 1 these tasks were performed by CDAQ, the HDB hardware database and servers, and the GATEWAY processes.

11.3 *Alarm system*

Provide processes to generate, collect, and distribute significant events, including alarms. Provide access to the database containing alarm information. These functions were performed in Run 1 by the **ALARM_SERVER, ALARM_DISPLAY, ALARM_WATCHER, ALARM_LOGGER**, and several private alarm logging tasks. Also included in Run 1 is the **FRONTLAND** task to monitor front ends.

11.4 *Host control process*

Allow for host control and monitoring of control path devices. In Run 1 this was provided by the **PAMPAS** parameter page application.

11.5 *High Voltage control*

Manage the high voltage front end systems and the host level communication processes. The Run 1 tasks included **HV2** and **HVMON**.

11.6 *Specialized control processes*

11.6.1 Control of the D0 Clock. In Run 1 this activity was controlled by the **CLK_SERVER**.

11.6.2 Liquid Argon monitoring. In Run 1 this was performed by **LAREXAM**.

11.6.3 Muon system controls

11.6.4 Tracking system controls

11.6.5 Calorimeter controls

11.7 *Data logging*

Allow for periodic recording of control path information. In Run 1 this was performed by **DBMON** writing ***DBL3*** database information.

12. Other

12.1 *Electronic log*

Log writer and viewer. Run 1 prototypes are the TPU-based **INFO** and tcl-based **D0_LOG** tasks.

12.2 *System performance monitors*

The Run 1 prototype is the VMS **`Perf_Meter`**.

12.3 *DAQ process monitor*

In Run 1 performed by **`MONITOR_DAQ`**.

12.4 *Store monitor*

In Run 1 performed by **`STORE_KEEPER`**.

Software Task Milestones

Oct 1996 ***Basic tools available***

- Architecture determined
  - Hardware configuration
  - Operating systems
  - Languages and compilers
- Task-to-task IP communication
- Client/Server package
- Database determined
- Control mechanism (CDAQ) available
- Level 1, Level 2, Level 3 base frameworks

Oct 1997 ***Control path operational***

- FY97 hardware acquisition complete
- Host communication with Level 1, Level 2, Level 3
- Initial release of host control tasks
- Front end tasks
- Alarm system
- User interface tools

Apr 1998 ***Data path operational***

- FY98 hardware acquisition complete
- Detector data into host via data path

Jan 1999 ***Coordinated operation***

- Configuration and Run control active

Jul 1999 ***Online system ready***

- FY99 hardware acquisition complete
- Filter code loaded
- User interfaces ready

Software Organization Chart

The organization chart for the Run 2 Computing management is shown in Figure 4. Under the direction of the Run 2 Software leaders are four sub-groups, of which Online Software is one sub-group. The Online Software sub-group is divided into several projects, several of which currently have no assigned leadership. Several of the projects, for example Level 1, overlap with projects that are part of the upgraded detector development. In these cases the leadership represents the heads of the software portions of those broader efforts, and is expected to oversee both the software closely tied to that effort as well as the associated software on the online host system.

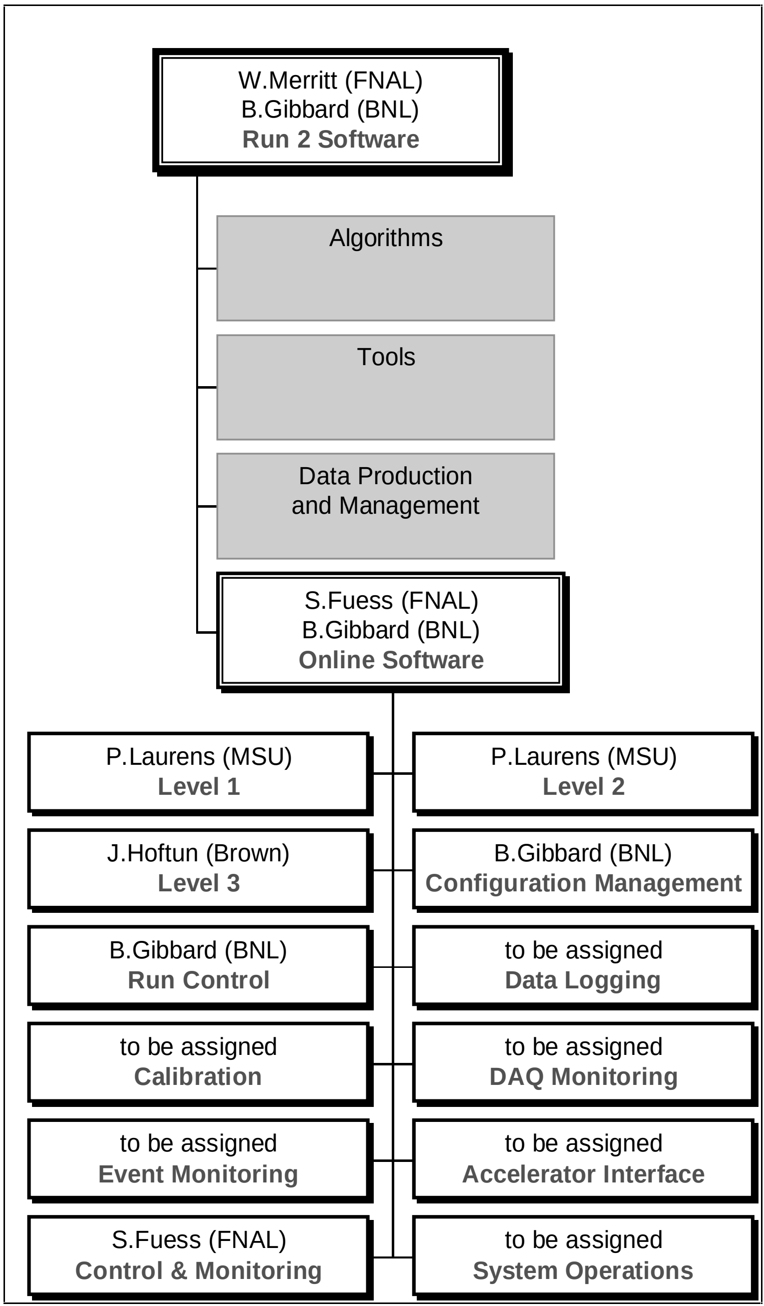

Figure 4. Run 2 Online Software Organization Chart

Software Task Assignments and Schedule

ID	Task Name	Duration	Start	Finish	Predecessors	Constraint	Resources
1	**Online Computing System**	**169w**	**1-Jan-96**	**1-Apr-99**		Must Finish On	
2	**Management**	**169w**	**1-Jan-96**	**1-Apr-99**		Must Finish On	
3	*Mangement*						
4	Management	169w	1-Jan-96	1-Apr-99		Must Finish On	PhysF[.5],PhysU[.5]
5							
6	**Online tools**	**39w**	**1-Jan-96**	**1-Oct-96**		Must Finish On	
7	*General tools*						
8	Hardware architecture plan	13w	1-Apr-96	1-Jul-96		As Soon As Possible	PhysF[.125],PhysU[.125]
9	Interprocess communication package	13w	1-Apr-96	1-Jul-96	8	As Soon As Possible	PhysU[.125]
10	Client/Server package	26w	1-Apr-96	1-Oct-96	9	Must Finish On	PhysF[.25]
11	Database decision	39w	1-Jan-96	1-Oct-96	8	Must Finish On	PhysF[.125],PhysU[.125]
12	*Level 1*						
13	Level 1 framework	39w	1-Jan-96	1-Oct-96	8	As Soon As Possible	CProfU[.25]
14	*Level 2*						
15	Level 2 framework	39w	1-Jan-96	1-Oct-96	8	As Soon As Possible	CProfU[.25]
16	*Level 3*						
17	Level 3 framework	39w	1-Jan-96	1-Jul-96	8	As Soon As Possible	PhysU[.25]
18	*Control and monitoring*						
19	Front end framework	39w	1-Jan-96	1-Oct-96	8	As Soon As Possible	CProfF[1]
20	Prototype control path communication	39w	1-Jan-96	1-Oct-96	11,19	As Soon As Possible	CProfF[.25]
21	VME-based DAQ	39w	1-Jan-96	1-Oct-96	19,20	As Soon As Possible	PhysU[.25]
22	Detector-specific tasks	39w	1-Jan-96	1-Oct-96	19	As Soon As Possible	PhysU[.25]
23							
24	**Communication paths**	**52w**	**1-Oct-96**	**1-Oct-97**	**6**	Must Finish On	
25	*General tools*						
26	Database tools	52w	1-Oct-96	1-Oct-97		As Soon As Possible	CProfF[.5]
27	Debugging and maintenance	26w	1-Oct-96	1-Oct-97		As Soon As Possible	PhysF[.25],CProfF[.5],PhysU[.25]
28	*Level 1*						
29	L1 operation by host control	13w	1-Oct-96	1-Jan-97	10	Must Finish On	CProfU[1]
30	*Level 2*						
31	L1 and L2 operation by host control	39w	1-Jan-97	1-Oct-97	10	Must Finish On	CProfU[1]
32	*Level 3*						
33	L3 configuration management	52w	1-Oct-96	1-Oct-97	10	As Soon As Possible	PhysU[.5]
34	*Calibration*						
35	Front end calibration tasks	52w	1-Oct-96	1-Oct-97	19	As Soon As Possible	PhysU[.25]
36	*Control and monitoring*						
37	Control path communication	52w	1-Oct-96	1-Oct-97	20	As Soon As Possible	CProfF[.5],PhysF[.5]
38	Alarm system	52w	1-Oct-96	1-Oct-97	10,37	As Soon As Possible	CProfF[.25],PhysF[.25]
39	Front end with 1394 support	52w	1-Oct-96	1-Oct-97	19	As Soon As Possible	CProfF[.25]
40	Front end with 1553 support	52w	1-Oct-96	1-Oct-97	19	As Soon As Possible	CProfF[.25]
41							
42	**Data path and monitoring**	**91w**	**1-Jul-96**	**1-Apr-98**	**24**	Must Finish On	
43	*General tools*						
44	Debugging and maintenance	26w	1-Oct-97	1-Apr-98		As Soon As Possible	PhysF[.25],CProfF[.5],PhysU[.25]
45	*Level 1*						
46	Auto-rate, serving, logging features	13w	1-Oct-97	1-Jan-98	29	As Soon As Possible	CProfU[.5]
47	*Level 2*						
48	Auto-rate, serving, logging features	13w	1-Oct-97	1-Jan-98	31	As Soon As Possible	CProfU[.5]
49	Data path operational	26w	1-Oct-97	1-Apr-98	29,31	As Soon As Possible	PhysU[.5]
50	*Level 3*						
51	Data path operational	78w	1-Jul-96	1-Jan-98	29,31,35,49	As Soon As Possible	PhysU[.5]
52	Monitoring tasks operational	39w	1-Jul-97	1-Apr-98	33	As Soon As Possible	PhysU[.25]
53	*Configuration management*						
54	Configuration framework	78w	1-Jul-96	1-Jan-98		As Soon As Possible	PhysU[.5]
55	Configuration downloading operational	52w	1-Jan-97	1-Jan-98	10,37,54	As Soon As Possible	PhysF[.25]
56	User configuration control	52w	1-Jan-97	1-Jan-98	10,54,55	As Soon As Possible	PhysU[.25]
57	*Run control*						
58	Run state management framework	78w	1-Jul-96	1-Jan-98	11	As Soon As Possible	PhysU[.25]
59	User run control	52w	1-Jan-97	1-Jan-98	10,51,58	As Soon As Possible	PhysU[.125]
60	*Data Logging*						
61	Host data path operational	78w	1-Jan-97	1-Apr-98	51	As Soon As Possible	PhysU[.5]
62	Data spooling operational	52w	1-Apr-97	1-Apr-98	61	As Soon As Possible	PhysU[.125]
63	*Calibration*						
64	Calibration databases operational	65w	1-Jan-97	1-Apr-98	11	As Soon As Possible	PhysU[.125]
65	*Event monitoring*						
66	Event data pool operational	13w	1-Jan-98	1-Apr-98		As Soon As Possible	PhysU[.25]
67	*Control and monitoring*						
68	Host control process	52w	1-Jan-97	1-Jan-98	37	As Soon As Possible	PhysU[.5]
69	High voltage control process	52w	1-Jan-97	1-Jan-98	37	As Soon As Possible	PhysU[1]
70	Clock control process	26w	1-Jul-97	1-Jan-98	37	As Soon As Possible	CProfF[.25]
71	Liquid Argon monitoring	26w	1-Jul-97	1-Jan-98	37	As Soon As Possible	PhysU[.25]
72	Specialized muon system controls	26w	1-Jul-97	1-Jan-98	37	As Soon As Possible	PhysU[.25]
73	Specialized tracking system controls	26w	1-Jul-97	1-Jan-98	37	As Soon As Possible	PhysU[.25]
74	Specialized calorimeter controls	26w	1-Jul-97	1-Jan-98	37	As Soon As Possible	PhysU[.25]
75	*System Operations*						
76	System performance monitors	13w	1-Jan-98	1-Apr-98		As Soon As Possible	PhysU[.125]
77							

Table 1a. Run 2 Online Software Task Schedule

ID	Task Name	Duration	Start	Finish	Predecessors	Constraint	Resources
78	**Coordinated operation**	**104w**	**1-Jan-97**	**1-Jan-99**	**42**	Must Finish On	
79	*General tools*						
80	User interface tools	52w	1-Jan-97	1-Jul-98		As Soon As Possible	PhysU[.125]
81	Debugging and maintenance	39w	1-Apr-98	1-Jan-99		As Soon As Possible	PhysF[.25],CProfF[.5],PhysU[.25]
82	*Level 1*						
83	Level 1 and Level 2 coordinated operation	52w	1-Jan-98	1-Jan-99	31	Must Finish On	CProfU[.5]
84	*Level 2*						
85	Level 2 and Level 1 coordinated operation	52w	1-Jan-98	1-Jan-99	31	Must Finish On	CProfU[.5]
86	Level 2 muon pre-processor	52w	1-Jan-98	1-Jan-99	85	As Soon As Possible	PhysU[.5]
87	Level 2 calorimeter pre-processor	52w	1-Jan-98	1-Jan-99	85	As Soon As Possible	PhysU[.5]
88	Level 2 tracking pre-processor	52w	1-Jan-98	1-Jan-99	85	As Soon As Possible	PhysU[.5]
89	Level 2 global processor	52w	1-Jan-98	1-Jan-99	85	As Soon As Possible	PhysU[.5]
90	*Configuration management*						
91	Configuration recording	52w	1-Jul-97	1-Jul-98	56	As Soon As Possible	PhysU[.125]
92	*Run control*						
93	Run information recording	78w	1-Jul-97	1-Jan-99	59	As Soon As Possible	PhysU[.125]
94	*Data Logging*						
95	Data information recording	52w	1-Jan-98	1-Jan-99	62	As Soon As Possible	PhysU[.125]
96	*Calibration*						
97	Level 2, Level 3, Host routines	78w	1-Jan-97	1-Jul-98	56,59,64	As Soon As Possible	PhysU[1]
98	*DAQ monitoring*						
99	Global monitoring process	52w	1-Jan-98	1-Jan-99	10,46,48,52	As Soon As Possible	PhysU[.5]
100	*Event monitoring*						
101	Event monitoring processes	52w	1-Jan-98	1-Jan-99	66	As Soon As Possible	PhysU[4]
102	*Accelerator interface*						
103	Communication procedures	52w	1-Jan-98	1-Jan-99	10	As Soon As Possible	PhysU[.25]
104	Data monitoring and recording	52w	1-Jan-98	1-Jan-99	11,103	As Soon As Possible	PhysU[.25]
105	*Control and monitoring*						
106	Data logging	52w	1-Jan-98	1-Jan-99	62	As Soon As Possible	PhysU[.25]
107	*System Operations*						
108	Electronic log	26w	1-Jul-98	1-Jan-99		As Soon As Possible	PhysU[.25]
109	DAQ process monitor	13w	1-Oct-98	1-Jan-99		As Soon As Possible	PhysU[.125]
110	Store monitor and logging	26w	1-Jul-98	1-Jan-99	11	As Soon As Possible	PhysU[.25]
111							
112	**Data taking**	**91w**	**1-Jul-97**	**1-Apr-99**	**78**	Must Finish On	
113	*General tools*						
114	Debugging and maintenance	39w	1-Jan-99	1-Apr-99		As Soon As Possible	PhysF[.25],CProfF[.5],PhysU[.25]
115	*Level 2*						
116	Level 2 filter code operation	65w	1-Jan-98	1-Apr-99	85	As Soon As Possible	PhysU[.5]
117	*Level 3*						
118	Level 3 filter code operation	91w	1-Jul-97	1-Apr-99	51	As Soon As Possible	PhysU[2]
119							
120							
121							

Table 1b. Run 2 Online Software Task Schedule

Software Resources

From the Software Task Assignments and Schedule information in Table 1, one can determine the human resources needed to carry out software development. Figure 5 shows the requirements in FTEs as a function of time.

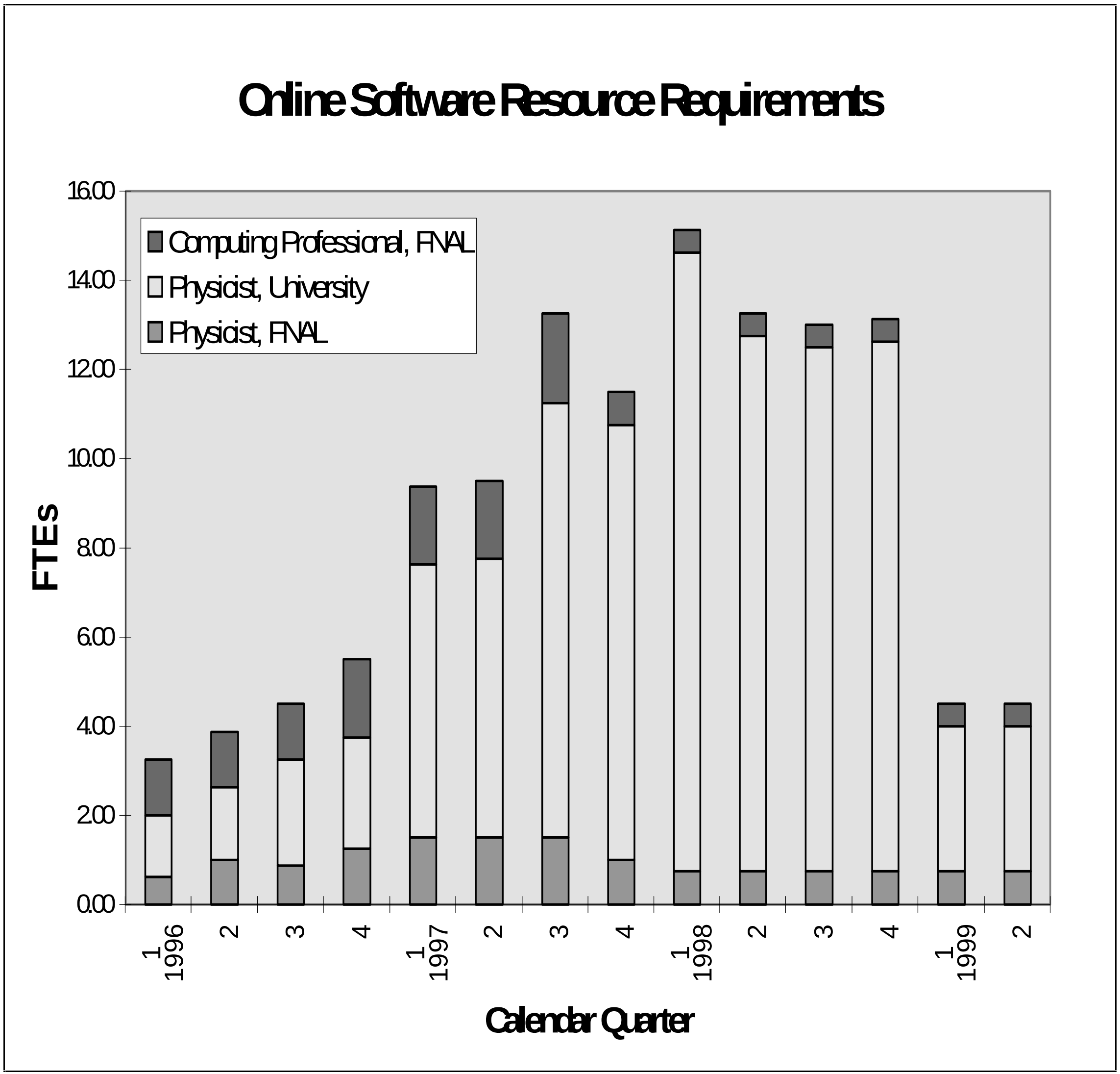

Figure 5. Online Software Human Resource Requirements

Current Needs

- **System management**
- **User interfaces: decisions and expert**
- **Level 2 filter management**
- **Level 3 filter management**
- **Event Data logging**
- **Calibration**
- **DAQ (Global) Monitoring**
- **Event Monitoring / EXAMINE**
- **Accelerator interface**
- **Control / Monitoring**
    - High Voltage
    - Parameter page
    - Data logging