AN ACQUISITION SYSTEM BASED ON A NETWORK OF MICROVAX'S RUNNING THE REALTIME DEC VAXELN OPERATING SYSTEM

I.D'Antone, G.Mandrioli, P.Matteuzzi, G.Sanzani
INFN Sez di Bologna and Physics Department, University of Bologna, Via Irnerio 46, 40126 Bologna, Italy

C.Bloise, A.F.Grillo, A.Marini, F.Ronga
INFN Laboratori Nazionali di Frascati, C.P.13, 00044 Frascati, Italy

A.Baldini
INFN Sez di Pisa, Via Livornese, San Piero a Grado, 56100 Pisa, Italy

G.Mancarella, O.Palamara, A.Surdo
INFN Sez di Lecce and Physics Department, University of Lecce, Via per Arnesano, 73100 Lecce, Italy

E.Lamanna, S.Petrera
INFN Sez di Roma and Physics Department, University of Roma, Piazzale A.Moro 2, 00185 Roma, Italy

Abstract

We describe an acquisition system based on a network (Ethernet/DECNET) of MicroVAX's running in the VAXELN environment. VAXELN is a Digital Equipment software product for the development of dedicated, real time systems for VAX processors.
A central VAX running under the VAX/VMS operating system is used as file server and as an interface of the acquisition system with respect to the user's world.

I.INTRODUCTION

The MACRO detector [1] has been designed to search for rare phenomena in the cosmic radiation (GUT monopoles or supermassive charged penetrating particles), to perform a survey of cosmic point sources of high energy gammas and neutrinos, to study the penetrating cosmic ray muons, and to be sensitive to neutrino bursts originated by gravitational stellar collapses in our Galaxy.

The detector, located at a depth of about 3,600 meter water equivalent in the INFN Gran Sasso Laboratory (LNGS), has an acceptance of about 10,000 m^2 sr for an isotropic flux of particles. It consists of plastic streamer tubes, liquid scintillation counters and track-etch detectors arranged in a modular structure, the basic element called "supermodule" having dimensions 12x12x5 m^3 (see Fig.1). The present implementation plan includes 6 supermodules with a possible extension to 12. The first supermodule is operative and is taking data since February 1989.

The data channels related to the detectors are:

Digital readout (streamer tube subsystem):	100,000
Wave form digitizers (scintillator subsystem):	580
Wave form digitizers (streamer tube subsystem):	720
ADC's (scintillator subsystem):	1,160
TDC's (scintillator subsystem):	580

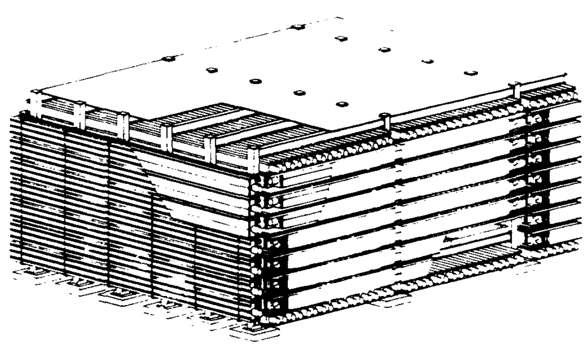

Figure 1: General view of MACRO supermodule.

The most important requirements to the acquisition system of the experiment are not originated by the cosmic ray muon rate (~1 Hz/m^2/hour), but they are dictated by the other physics items, since the background due to the local radioactivity gives a substantial contribution to the total trigger rate. In particular, this holds for the GUT monopole search in which particles with low velocity (down to ~ 10^{-4} c) are expected; this implies that the trigger gates are opened for a considerable amount of time (several hundreds of microseconds), and, therefore, that the trigger needs sophisticated circuitries in order to obtain trigger rates compatible with the acquisition system capabilities. The expected trigger rates for MACRO are of the order of several tenths of hertz.

The total amount of information that is read from the first supermodule equipments is of the order of 20 Kbytes/event. Data compaction is performed before the data logging in order to skip over the not significant values. For the first supermodule this reduction is performed 50% in the front-end readout circuits and 50% by software algorithms; for the other supermodules the electronic has been designed to perform this operation only in the front-end readout circuits.

II.THE MACRO ACQUISITION SYSTEM

The acquisition system for the experiment has been designed taking into account the following requirements:

1) The system must be modular and distributed in order to match the apparatus modularity; in this way the system size can be tailored to the real needs.
2) The system must be integrated in a network in order to allow an easy access from remote locations to each computer or microcomputer; in this way the control and the monitoring of the apparatus can be performed even from large distances (e.g., from USA).
3) The system must be largely based on commercial products, both for software and hardware, in order to be easily maintained.

Taking into accounts these requirements, we have chosen a system based on a MicroVAXII network (Ethernet/DECNET). The MicroVAX's run a VAXELN system. VAXELN is a Digital Equipment software product specific for the development of dedicated, real time applications for the VAX processors.

The "CERN-Fisher CAMAC System Crate", that uses the CES 2280 Q-bus/CAMAC interface and the CES 2281 DMA controller, has been chosen as the main hardware standard. In the future we will probably use some VME or Fastbus electronics; in both cases the readout will be done via the DEC DRV11-WA parallel interface, whose software driver is included in the standard VAXELN distribution kit.

The general layout of the data acquisition system for MACRO is shown in Fig. 2. Three MicroVAX's control three groups of two supermodules. Each MicroVAX is connected to two CAMAC parallel branches via a System Crate. A fourth MicroVAX acts as the supervisor for the neutrino events originated by a stellar collapse. A central VAX8200, running under the VAX/VMS operating system, is used as file and network server and as an interface of the acquisition system with respect to the users. The raw data logging is performed on the VAX8200 disks and the amount of mass storage (3 Gbytes) allows the data to be available for a period lasting few weeks for a fast analysis or for copying.

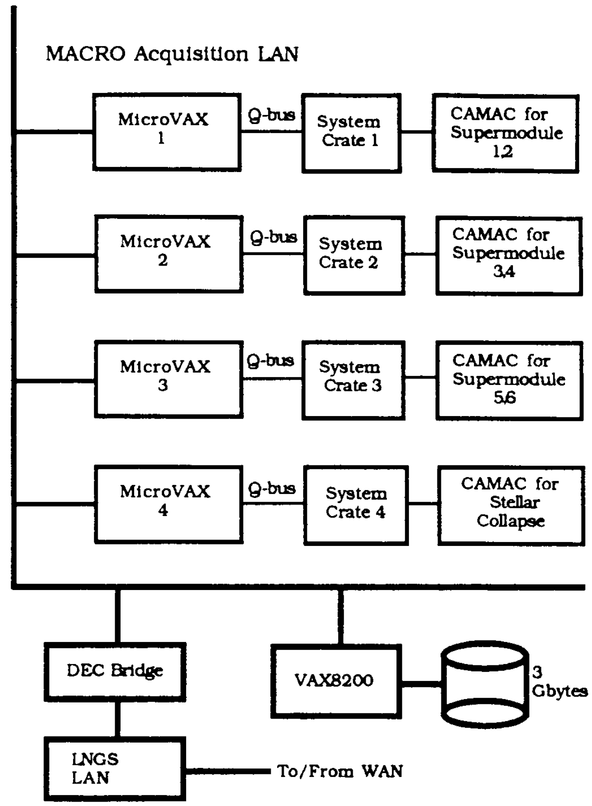

Figure 2: Layout of the MACRO data acquisition system.

The tasks performed by each MicroVAX are: electronics readout, event filtering, data reduction, data forwarding to the VAX8200, apparatus monitoring and calibration.

All the computers are connected via Ethernet under the DECNET network protocol. This choice does not imply any limitation on the data throughput that, in our case, is due to the power of the CPU's we are using: in absence of high load on the CPU's, we have obtained a data throughput bigger than 300 Kbytes/sec between two MicroVAXII's operating under VAXELN, and bigger than 150 Kbytes/sec between a MicroVAXII under VAXELN and the VAX8200 operating under VAX/VMS. A big advantage of having an acquisition system distributed over the network consists in the reduction of the cabling, since all the electronic crates are not forced to be located at the same place. The Macro Local Area Network (LAN) is connected to the General LNGS LAN by a DEC Bridge. The LNGS LAN is composed by segments linked via optical fibers running for about 6 Km in order to cover the distance between the underground laboratory and the external laboratory, where the general computing facilities are located. From the external laboratory it is possible to have access to the DECNET Wide Area Network(WAN).Utilities have been implemented to help the control and the debugging of the apparatus; in fact, during the acquisition, a user from any DECNET node can:

1) Request a copy of the raw data buffer from a given MicroVAX or from the central VAX.
2) Require the execution of CAMAC operations by the use of Remote Procedure Calls.

III.THE VAXELN SYSTEM FOR THE MICROVAX'S

VAXELN has the following advantages with respect to the VAX/VMS operating system:

1) optimized performances for real time operations, due to the small load of the operating system;
2) fixed time response to external events;
3) powerful and efficient message exchange facilities that are very useful for building multi-jobs applications, even if the jobs are running on different machines;
4) easiness to write device drivers and to work with peripherals; device drivers can be written using high level languages (e.g.,EPASCAL, C, FORTRAN)

The development of a VAXELN system is done in a machine running the VAX/VMS operating system. The result of the development is a file containing the application code and the kernel code. This file is then downline-loaded to a target machine using Ethernet. The target machine can be a VAX processor of any type, not necessary equipped with online disks. The devices needed are the Ethernet interface, the processor and memory boards. There is a powerful debugger that can be run from a VAX/VMS host; this facility is particularly useful in the debugging of multi-job, multi-process applications.
A machine running a VAXELN system becomes a machine dedicated to a particular application. There is a limited possibility to load dynamically programs, but, from this point of view, VAX/VMS is a more general purpose operating system.

The VAXELN system for MACRO consists of six user jobs running with different priorities. Four system jobs are included in the system (the Ethernet driver,the console driver, etc.). In each job many concurrent sub-processes can run with a different priority; in our system more then fifteen sub-processes run concurrently; the exact number of sub-processes depends on the number of activated DECNET connections, since each connection is handled by a dedicated sub-process.

All the software has been written in EPASCAL, the PASCAL native language of VAXELN.
The data flow after the occurrence of a trigger is shown in Fig. 3. The exchange of data between jobs is done using the VAXELN message facilities. This facility is very useful for the handling of the queues of buffers.
We have chosen a multi-job architecture instead of a single-job architecture in order to have the choice of redistributing the jobs in the network, if this will be necessary in the future. However this architecture implies some overhead. The overhead, which is due to the message mapping, message sending and message receiving, sets a limits on the maximum acquisition rate. In our system this limit is about 90 Hz, for buffers of 100 Kbytes and a MicroVAXII CPU. At lower rates the effect of this overhead on the acquisition dead time is negligible.

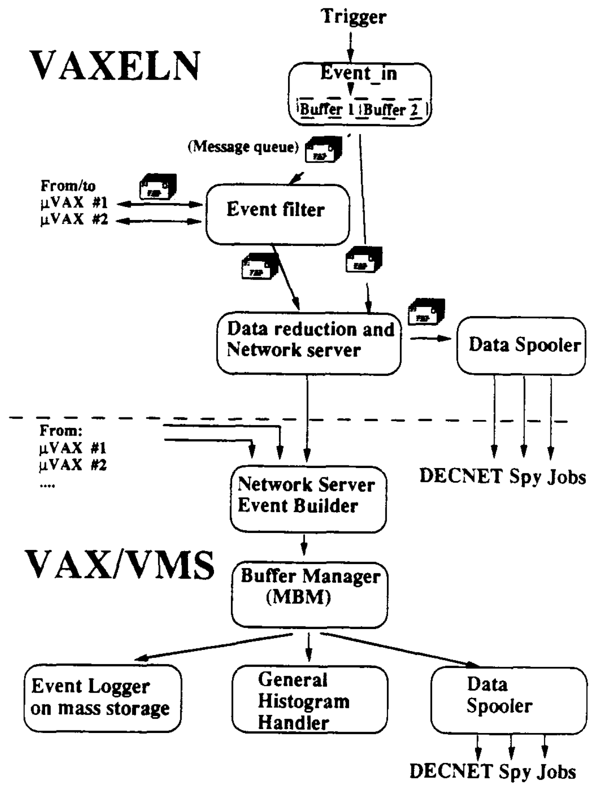

Figure 3: Event flow from a MicroVAX to the VAX8200.

The CAMAC input/output can be done using a list of CAMAC operations, loaded from a file resident in the VAX8200; this is the normal mode of handling the CAMAC during the data taking. Moreover, the access can be done using a library containing the standard ESONE CAMAC routines and a set of fast VAXELN routines [2]. The CAMAC routines can be called either directly inside VAXELN or from a computer in the DECNET network using remote access routines.
Some I/O performances are reported in Table 1 for three different MicroVAX processors. A CAMAC I/O in a given branch, crate, station, takes two MicroVAX read/write operations; if the CAMAC I/O is repeated on the same location of the last transfer, only a single MicroVAX read/write is needed.The units are microseconds. Note that for processors with virtual memory management (MicroVAXII and MicroVAX3xxx) some figures could change (up to 25 %), according to the different memory mapping.
The rt VAX processor is similar to the MicroVAX II but it has no virtual memory management (virtual memory

management is not necessary in VAXELN); this feature improves slightly the performances and makes the time response independent from the allocation of the program in the memory.

CPU's:	MicroVAXII	rt VAX	MicroVAX3xxx
Read the first word in a CAMAC module:	10	-	5
Read additional words inside the same CAMAC module (System Crate):	5	-	2.3
Read the first word in a CAMAC module (using CAMAC List with Q test):	55	38	13
Read additional words inside the same CAMAC module (using CAMAC List with Q test):	29	27	11
Start a LAM interrupt routine (typical value):	30	30	20
Resume a process waiting for LAM:	290	290	110
Read CAMAC register by the CSSA ESONE routine:	74	-	-
DMA transfer of one word(*):	700	650	350

(*) The DMA transfer of additional words takes about 2.3 microsec/word in the System Crate and 3.3 microsec/word for a crate in a branch.

Table 1: CAMAC I/O performances for different MicroVAX's

The performances obtained using the Remote CAMAC Routines are reported in Table 2. The server was a MicroVAXII. We have done measurements using as hosts a VAX8200 and a VAX8650 connected to the MicroVAXII in Ethernet/DECNET. The VAX8200 is equipped with a DEUNA interface. The units are milliseconds.

Hosts:	VAX8650	VAX82000
Single CAMAC I/O using CSSA:	12	16
I/O of 10 CAMAC words using the ESONE list routine (CSGA):	13	20
I/O of 10 CAMACwords using the ESONE DMA routine (CSUBC):	27	33
I/O of 5000 CAMAC words using the ESONE DMA routine (CSUBC):	68	85

Table 2:Performances for CAMAC I/O (Remote CAMAC routines)

In this case the performances are limited by the Ethernet throughput and by the DECNET overhead. Obviously the throughput increases if many CAMAC I/O's are requested in the same call as in the list routine (CSGA) or in the DMA routine. However, this kind of access to CAMAC is very useful for the debugging of the electronics, since it is possible to write and to run simple programs on a VAX/VMS host without any knowledge of VAXELN, or for the monitoring of slow devices (HV systems, etc.).
If two programs are running at the same time on two different hosts the response times are typically 20% higher.

After a trigger interrupt (see Fig.3) the data are read inside the Event_in job using a memory region mapped as a VAXELN message. A concurrent sub-process, running at lower priority, provides the message mapping and the message routing (the time needed for these operations is 4 msec for a buffer of 100 Kbytes); two memory regions are allocated as data buffers in order to minimize the dead time.

The data can be sent either to a job, dedicated to the data reduction and data forwarding to the VAX8200, or to the Event filter job. The Event filter job makes the decision if the event should be discarded. This decision is taken in a master MicroVAX, defined on a event by event base according to the trigger topology. In the case that the master MicroVAX should request some information to the other VAXELN MicroVAX's, messages across the Ethernet network are used. The decision will be communicated to all the MicroVAX in the acquisition network.
This distributed scheme for event filtering has a limitation due to the fact that the maximum rate of the Ethernet packets between two MicroVAXII's in VAXELN is of the order of 200 Hz.

If the data buffer size is greater than 32 Kbytes, the data are divided into segments that are sent to the VAX8200. This is necessary in order to cope with the VAX/VMS QIO limit of 32 Kbytes for the buffer size.

The data are also sent to the data spooler job, which runs at low priority. Users from remote DECNET nodes can ask for a copy of the data buffer by requesting a DECNET connection to this job. A dedicated sub-process is created for each connection and the message flow control facility is used to avoid interferences between the sub-processes and to avoid interferences with the main acquisition.

A centralized job handles the commands coming from the central VAX and routes such commands to the appropriate VAXELN job (Fig. 4). This job also supports the ESONE Remote CAMAC Procedures calls coming from the main acquisition program.
Another job provides support for Remote Procedure calls coming from secondary users. A simple CAMAC crate booking is included to avoid interferences during the data taking. The number of concurrent users at a given time is limited only by the amount of available memory in a MicroVAX.

The network buffer size used in our application is 576 bytes (the default). A higher network buffer size allows better performances for the exchange of big Ethernet messages. The memory occupation of our VAXELN system, with no secondary users and with data buffer sizes of 100 Kbytes, is 2 Mbytes.

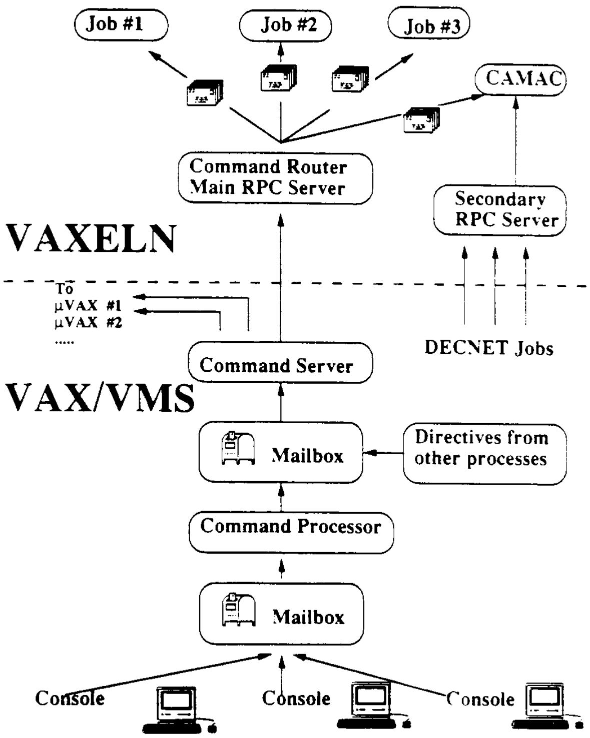

Figure 4:Command flow from the VAX8200 to a MicroVAX.

IV.THE VAX/VMS ACQUISITION SYSTEM FOR THE VAX8200

This part of the acquisition system performs the following functions:

1) Collector of the data coming from the MicroVAX's of the acquisition network;
2) logger of the data on the mass storage;
3) router of the commands to the MicroVAX's in the network;
4) router of the user directives to the other components of the VAX/VMS acquisition system;
5) handler of the general histogramming;
6) spooler of complete event data and of general information on data taking to other VAX requesters in the network
7) collector of the alarm conditions.

This system is constituted by a core composed by several concurrent sub-processes, generated by a single parent process running in batch. The sub-processes are scheduled according to a prefixed scheme of priorities and are synchronized via Event Flags. Other processes, that can run in multiple copies and that are not tightly coupled to the core, are the interfaces toward the user's (Consoles, Histogram Presenters, Event Displays).

The data sharing among the sub-processes is performed via Mailboxes and Global Sections, while the raw data flow among the different components of the system is arbitrated by the MODEL Buffer Manager (MBM) [3].

The network service is performed by the use of the DECNET communications. The system implements asynchronous DECNET nontransparent communications in the sub-process (Network Server, see Fig.3) that handles unsolicited data coming from the VAXELN systems (messages containing event data, alarms, periodic monitor buffers, etc.). This sub-process includes a procedure that builds the event data structure starting from the information (fragmented by the transmission protocol) received from the network and originated from different MicroVAX's (more than one MicroVAX can be involved in the acquisition of the same event). After that the building operation is completed, the received data are injected inside the Data Stream controlled by MBM.

Several consumer sub-processes are linked to different Ports on the Data Stream. Among them, the Event_out sub-process obtains the totality of the collected data and performs the writing activity on the mass storage by means of the FORTRAN unformatted I/O. A consumer sub-process is the Histogram Handler, that makes use of ZEBRA [4] data structure management and of HBOOK version 4 [5] for the filling of the live histograms needed for monitoring the data taking; this operation is performed at low priority and on an event sample depending on the currently available resources (CPU,Memory).

Data Spooler is an additional sub-process, sampling the events like the Histogram Handler, that makes use of DECNET nontransparent communications in order to accept connections from remote jobs and to ditribute on request event data and general information on the data taking status.

The system implements synchronous DECNET transparent communications in the sub-process Command Server (see Fig.4), that routes the directives to a companion job in the VAXELN systems. The commands are originated by a Console process linked to a Mailbox accessed by the Command Processor sub-process.

The Histogram Presenter is a program, making use of HBOOK, ZEBRA and GKS [6], that allows the presentation and the manipulation of the filled histograms. The histograms data are shared via a Global Section (local Presenters) or via a disk file containing a periodic dump of the histogram content (remote Presenters).

The Event Display is an interactive program that, using GKS, performs the 3D visual presentation of the event data. The data sources for the Event Display are either the Data spoolers of the components of the acquisition network (VAX8200, MicroVAX's) or the raw data files stored on disk.

The synchronization among the VAX/VMS acquisition system and the VAXELN systems is performed by message exchange over the synchronous and the asynchronous links: when a command of change-of-state is issued (e.g., PAUSE), the VMS system makes a transition to an intermediate state (e.g., PAUSING) for avoiding the issuing of another change-of-state command; the command is sent to the VAXELN systems, that performs the change-of-state activity; when this activity (e.g., the flushing of the event buffers) is finished, a change-of-state record is sent over the asynchronous link and when such record is received from all the VAXELN systems the VAX/VMS system makes a transition to the stable state (e.g., PAUSED).

OVERALL PERFORMANCES

The performances of the system, measured making use of a random trigger generator, are reported in Fig.5 (Fraction of dead -time vs. Event frequency).

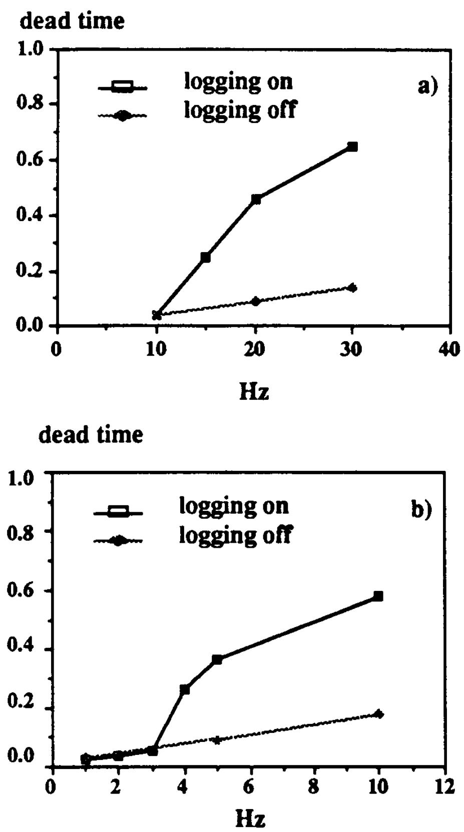

Figure 5: Performances for random triggers.

The flow of data after a trigger occurence is shown in Fig 3 and the reported performances are related to events involving one MicroVAX of the network.

In Fig 5a the CAMAC List caused only one DMA transfer and 40 word program transfers for a total buffer length of 2.18 Kbytes. In Fig 5b the CAMAC List caused 10 DMA transfers and 80 word program transfers for a total of 13.07 Kbytes. The data logging has been done on disk..

A throughput of about 40 Kbytes/sec has been obtained (for a buffer length of 13 Kbytes) at 10% of dead-time. This value corresponds to about 30 tapes/day, supposing to fill each tape with 120 Mbytes of information.The limitation is due to the VAX8200 CPU that saturates for this data throughput. With think that the current Ethernet hardware could support a data throughput at least a factor three higher, if a CPU more powerful than the VAX8200 is used. However this is not needed for the MACRO experiment.

We conclude that Ethernet provides a very useful support for a distributed acquisition system, if the required performances are not exceptionally high.

REFERENCES

[1] The MACRO Collaboration, "MACRO, a Large-Area Detector at the Gran Sasso Laboratory", *Nuovo Cimento,* vol.9, pp. 281-292, March-April 1986.

The MACRO Collaboration, "The MACRO Detector at the Gran Sasso Laboratory", *Nucl. Instrum. Method,* vol.A264, pp.18-23, January 1988.

[2] A. Giannasca et al., "A new library of CAMAC routines for the control of the System Crate by a MicroVAX in the VAXELN environment", Internal Note IFU 893, Physics Dept., Univ. of Roma "La Sapienza", 1987.

[3] P. Vande Vyvre, "The MODEL Buffer Manager", *CERN Mini and Micro Computer Newsletter,* No. 17, pp. 13-15, Geneva:CERN, December 1987.

P. Vande Vyvre, "The MODEL Buffer Manager User's Guide", CERN, Geneva:CERN, May 1987.

[4] R.Brun and J.Zoll, "ZEBRA User Guide", CERN Program Library Q100, Geneva:CERN, January 1987.

[5] R.Brun and D.Lienart, "HBOOK User Guide - Version 4", CERN Program Library Y250, Geneva:CERN, October 1987.

[6] GTS-GRAL Gmbh, "GKSGRAL-3d User Manual", Darmstadt, May 1988.