



**HAL**  
open science

## Evaluation of a Linux based control systeme for GANIL

L. David, Group The Control

► **To cite this version:**

L. David, Group The Control. Evaluation of a Linux based control systeme for GANIL. Personal computers and particle accelerators, Oct 2002, Frascati, Italy. in2p3-00019958

**HAL Id: in2p3-00019958**

**<https://in2p3.hal.science/in2p3-00019958v1>**

Submitted on 20 Nov 2002

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# EVALUATION OF A LINUX BASED CONTROL SYSTEM FOR GANIL

L. David and the control group, Grand Accelérateur National d'Ions Lourds,  
BP 5027 14076 CAEN CEDEX 5 FRANCE

## *Abstract*

For almost ten years, the Ganil control system has been based on VMS workstations and Camac crates running Vaxeln on RtVax controllers, with Ada as common language, and Ingres as relational DataBase.

When Digital Equipment (now HP-Compaq) gave up with RtVax processors, we decided to move to Vme crates with PowerPC controllers running VxWorks.

After that, we have also wanted to try some use of Linux, to get rid of the links remaining with Vms in the beam tuning programs and to be able to use some free software tools that were not very powerful in a Vms/Motif environment.

This paper describes the milestones we are underway to perform :

- Graphical user interfaces using GTK instead of Motif with XRT widgets
- Data Base access with Ada / SQL requests
- TCP/IP communication with VxWorks real time crates

## **1. NOWADAYS CONTROL SYSTEM**

When it was defined in the early nineties, the control system was based on Vax, then Alpha workstations and VMS servers, programmed in Ada.

The front-end level was realised with Camac crates using RtVax microprocessors.

When Dec, now HP-Compaq, decided to stop the Vax family, we began to replace the Camac crates by Vme with PowerPc processors.

At this time, we moved from VaxEln system to VxWorks, using Ada95 (object oriented) instead of Ada83.

## **2. SOFTWARE UPGRADES**

### 2.1. Graphical User Interfaces

Until now, all the beam tuning programs use graphical interfaces programmed with Motif and Dec Ada bindings. Linux offers two different graphical libraries : QT used by KDE and GTK used by Gnome, as desktops.

We chose to try GTK, because an Ada binding was necessary for us and was only available with Gtk/Ada.

As both Motif and Gtk are built over the X11 library, the functions and names are different but the resulting displays can remain almost the same.

The Callbacks are still there, with procedures linked to each graphical object, and Gtk gives an object oriented approach.

### 2.2. Commercial widgets

Graphs and sheets are not easy to draw with Motif, so we bought extra widgets Xrt/Graph and Xrt/Tab, that we use to display the variation of beam intensity during the day, or the different turns of the beam inside the cyclotrons.

With Linux, the package GTK/+ contains Plot and Sheet widgets, with similar functions as the Xrt package.

### 2.3. Data Base

All the characteristics of the motors, power supplies and diagnostics used to control the beam are stored in the database, which also contains all the optic parameters for the different available beams.

In the tuning programs, the requests to the database are done with Sql commands, included in the Ada source code.

In these files, before compilation, a commercial tool replaces the Sql requests by calls to the Ingres API database, to provide only Ada source code.

At the beginning, this tool wasn't available for Linux and we tried to translate our Ada with Sql procedures in C with Sql equivalent procedures.

As there are a lot of database requests, the translation of all of them would mean a lot of time. By the time, a Linux release of the tool became available, giving us the ability to keep all of our existing procedures without any modifications.

### 2.4. TCP/IP communications

The communications between the workstations in the control room and the Vme crates are based over Tcp/Ip.

In the real time crates, VxWorks provides the IPC procedures, with an Ada binding included with the compiler.

With Linux, the same procedures weren't available with the Gnat compiler, so the Florist library, issued from the public domain, was tried at first. Recently a more complete package was delivered by Gnat, with all the needed functions.

### 2.5. Beam tuning programs

The first milestone was the rewriting of one of the ECR ion source program. This program calculates the HT voltage, or Dipole current, requested to obtain a specific ion in the source output, without accessing to any device.

This step showed us the difficulties linked to the Gtk graphical library and the database access, but without any network communication with the front-end crates.

We also begin to check all the Vms calls that have to be replaced by Ada code or standard system calls.

After that, we'll check all of our 40 tuning beam programs under Vms, to give them the same multi-tasking architecture.

So when we'll have moved two or three programs, it'll be easier to change all the others.

### 2.6. Development Tools

Our object oriented programs use a lot of common procedures and libraries, organised in different levels, with dependencies controlled by the Vms/Ada toolkit. The Linux GPS development toolkit manages the different levels of source files as hierarchical projects, including classes and dependencies graphs with a specific browser, a documentation generator and a GUI builder.

## **3. HARDWARE UPGRADES**

### 3.1. X terminals

In the control room, there is always an X terminal associated with each Vms workstation. This X terminal displays the individual controls of the devices, and the workstation runs and displays the beam tuning programs (beam losses, transverse matching, turns separation, emittance, beam injection centering...)

As the X terminals won't have any more support from Compaq, we'll have to replace them next year.

We'll try to use Linux PCs instead, with the Vms workstation using remote X display on the Linux screens.

That will be another way of integrating Linux in the current control system. We also hope to discover there the difficulty of maintaining several separated PCs, because our Vms cluster has only one system disk for all the workstations, allowing us to upgrade the Vms versions in only few hours every year.

### 3.2. Disks servers

All of our sources files were on the Vms cluster disks. For the Linux compilations, we tried to use an NFS server on Vms, but we discovered bad access times.

We moved the common files to the faster Linux disks, and we reached almost the same compilation times for Linux as for Vms.

As those files are also common with the real time Vxworks development, working on Windows NT, they must be available for NT.

On Vms, the Pathworks server shows them to the NT PCs, and on Linux, the Samba server does the same, but with slower access times that we hope to improve.

## **4. KNOWN DIFFICULTIES**

### 4.1. Knob boxes

Each workstation manages a knob box, connected with a serial link and a special driver generating Motif event.

On the Linux PCs we'll try to keep these knobs, and we'll have to develop a specific driver with a third part company.

### 4.2. Endianness

On Intel/Linux, the bytes representation in memory is "little endian", but is "big endian" on PowerPcs.

As Alpha/Vms processors are also "little endian", the problem has already been discovered and solved when we began to install the PowerPcs. The network messages coming from the workstation are stored in an Ada stream, which is an unformatted buffer associated with special procedures that changes the representation of data before using them.

### 4.3. Ieee Float

Ganil is still controlled by 10 Camac crates, and 30 Vme. Even if we try to replace 2 or 3 Camac crates a year, we have to deal with some Camac crates left for our Linux trials.

Our Camac processors are RtVax using only Vax Float, and even if our Vms workstations, VxWorks crates and Linux PCs can manage Ieee floats, we still have to send and receive floats in Vax representation between Linux PCs and Camac crates, until all the Camac crates will definitively be shut down.

## **5. CONCLUSION**

The aim of this test is to get an evaluation of the time necessary to move one tuning program to Linux, and to list the main difficulties of the simultaneous use of both VMS and Linux stations, connected with a single database to a single generation of Vme crates.

One control program is now available, and after, we'll need two years to move all the beam tuning programs from Vms to Linux.

A more important use of graphical objects, easier to design with Gtk and the Glade generator rather than with Motif, should limit the time spent for programming and testing the graphical user interfaces.

After that we'll have to replace our Vms server, now used for the Database access, Data collector and Alarms display, by a Linux server.

In the same time we begin to upgrade our communications routines to introduce CORBA, first with the Data Collector process as server for Linux clients.

## **6. REFERENCES**

- [1] Integration of PowerPc processors in the commands & controls of the Ganil accelerators  
M. Ulrich, P. Duneau, J.M Loyant (Ganil internal report )
- [2] Software trends for the control system (Ganil A98.03)
- [2] Evaluation of Linux in the Control System.  
L. David & the control group (Ganil internal report)
- [3] ACT documentation about Gnat, Gtk, Glade, Gps
- [4] CORBAda/OrbRiver documentation from Topgraph'X