A SEAMLESS CONTROL SYSTEM UPGRADE FOR A CONTINUOUSLY RUNNING ACCELERATOR FACILITY

W. Busse, C. Rethfeldt
Hahn-Meitner-Institut Berlin, Glienicker-Str. 100, 14109, Germany

Abstract

The control system of the Ion-Beam-Laboratory Berlin (ISL-Berlin) dates from the late 1970's and was originally designed for the HMI-VICKSI accelerator complex. It is based on a single server processor (PDP11), a CAMAC fieldbus and software which was developed in-house. The strategy for a control system upgrade has to face the request for continuing accelerator operation and the limited available manpower. The choice taken is the commercial product Vsystem by VISTA Inc. Accelerator specific applications can easily be generated on multiple OS-platforms in a heterogeneous network by supplying transparent APIs as programming interface between the different control system layers. As Phase 1 of the upgrade, GUI's for operators or machine physicists have been implemented along with a Linux application server, collecting and archiving machine data, generating 24h trend charts, which are accessible via internet by standards browsers or WAP-service, and sending alarm messages when requested.

1 MOTIVATION AND POLICY

The old ISL control system dates from the late 1970's and was originally designed for the VICKSI accelerator complex [1]. Despite of its still modern internal design and its extensibility, it has encountered considerable draw-backs: poor performance when after the extension of the accelerator facility several operating teams wanted to have independent but simultaneous access, poor graphics and the fact that all the control system software is home made.

Therefore it was decided to move to a new system with distributed processors and data bases, with modern graphics and as much commercial software as possible, but to maintain the CAMAC fieldbus, at least in the first approach. The move had to be seamless to allow for continuing accelerator operation and it had to be achieved with the limited man-power available.

The choice taken in the course of the 1990's consists of an OpenVMS cluster with workstations and X-Terminals for the presentation and human interaction level, with servers for the distributed data bases and transaction libraries, with rt-VAX's as front end servers for concurrent access to the fieldbus hardware, with PVWave (Visual Numerics Inc.) to provide graphics for data evaluation as an intermediate step and with Vsystem (VISTA Inc.) as the basic control system software [2].

Vsystem comes along with a general set of application programs which control systems have to supply anyway. However, it goes without saying that accelerator specific applications and field hardware specific drivers and conversion routines are at the customer's responsibility.

Both versions, the old and the new, of the ISL control system will be operating in parallel for some interim time. As the porting of the old to the new platform does not foresee any co-ordination or inter communication between the ISL and Vsystem databases, Vsystem applications will only be allowed read access to accelerator parameters in the Phase 1 Approach. Active accelerator control is foreseen for Phase 2, by moving adequate parameter groups or accelerator sections from the old to the new platform in complete parts.

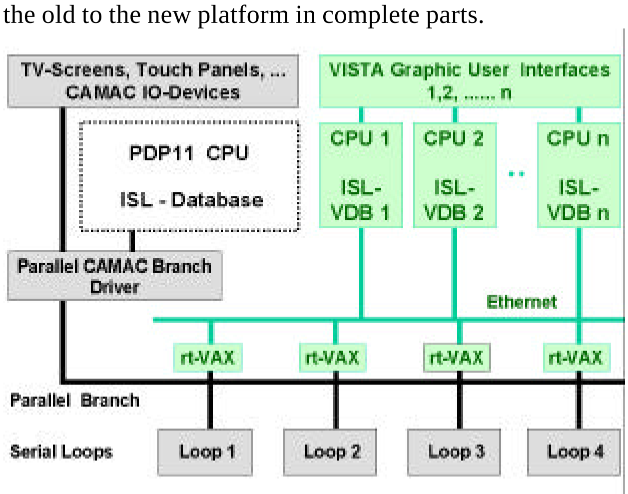

Fig. 1: *Schema of the ISL-Control System Upgrade*

VISTA Databases (VDB) are distributed among several CPU's, supporting the Graphic User Interfaces on X-Terminals and handling the CAMAC communication by remote procedure calls to Serial Highway front end servers (rt-VAX's). Simultaneous access to the Serial Highways from either the Parallel Branch or the rt-VAX's is co-ordinated by hardware semaphores.

2 NEW CONTROL SYSTEM APPLICATIONS

The common Vsystem layer in figure 2 contains distributed data bases along with access and transaction libraries. The API's, supported on various OS, like OpenVMS, MS-OS, Mac-OS and UNIX-derivatives, allow transparent application programming with regard to accelerator parameter access which in fact is tied to local or remote databases in a heterogeneous network. Thus Vsystem provides a uniform access to centrally driven hardware and local Front End PC's, gluing together long standing and modern software or hardware into a uniform and consistent control system.

Generally speaking, the Vsystem application layer incorporates a drawing tool for graphical applications and a bunch of general application programs such as scripting and alarm handling, all of which can be live linked to respective accelerator parameters. In addition, the ISL controls group has provided adequate API's for the coherent access to these parameters from other commercial tools, such as LabVIEW and PVWave to complete the advantages of GUI's for machine analysis and beam physics.

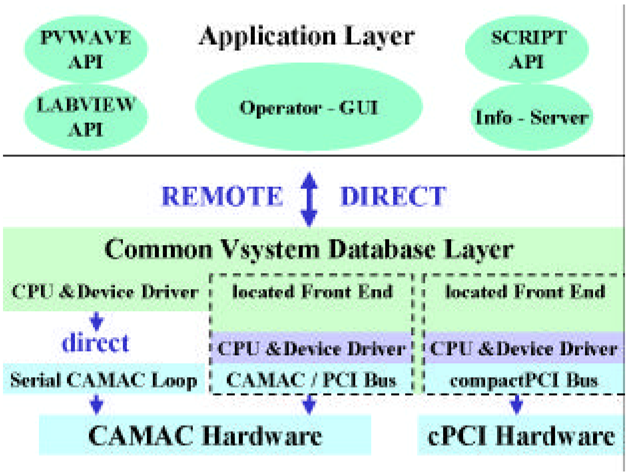

Fig. 2: *Tool integration into a uniform control system*

Vsystem features the unified access of various control applications to a scaleable heterogeneous CPU configuration driving a wide collection of long standing and new hardware.

2.1 Graphical User Interfaces

The new control system allows the implementation of graphical information displays as shown in figure 3, giving a live overview of the status of accelerator sections of interest. Special graphical features such as a selection of standardised colours for 'normal', 'warning' and 'alarm' states, trend charts and live text or bars support the ergonomic information being presented to the operators or machine physicists.

In fact, a standardised layout of such displays with regard to character size, to the meaning and choice of symbols and colours, to the placement of items of the same type of information as well as to the position of interaction areas within various displays largely meets human nature and is highly appreciated by the operators.

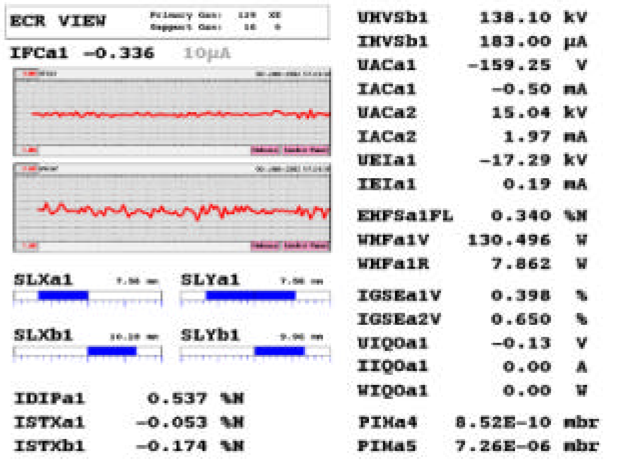

Fig. 3: *A typical Vsystem Graphical User Interface*

Trend Charts, Live Text and Bars display the actual status of the ECR Ion Source of the RFQ injection beam line.

2.2 Linux Application Server

As Vsystem is a multiple-platform implementation allowing remote access in a heterogeneous network through a common transparent API, useful remote applications can be developed for the control and monitoring of a large facility like the HMI Ion Beam Laboratory. For remote visualisation and access to archived data a first Linux Application Server was installed (Figures 4a,b). It collects and saves device data to generate 24h trend charts which can be viewed via the internet [3]. It also supports WAP data exchange with mobile phones, runs predefined checks of the data and sends alarm messages when requested. The latter service might become important when ISL will have to run the facility in standby but unattended over night.

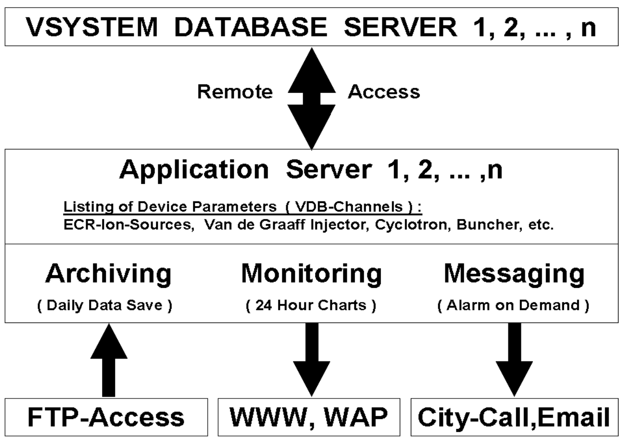

Fig. 4a: *Communication Topology of a Linux application service archiving long term monitoring data, keeping them available for various presentations upon request or running predefined checks to generate alarm messages as required.*

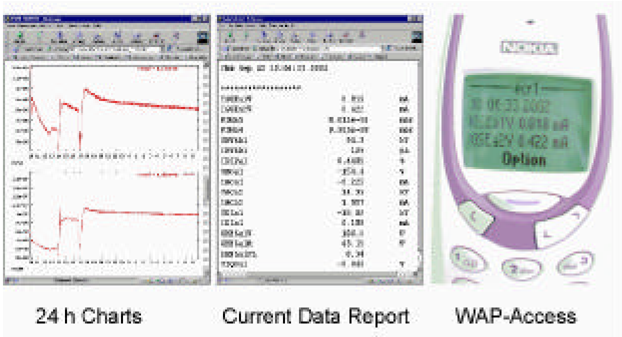

Fig. 4b: *A Linux Application Server provides remote monitoring of the ISL-Facility. The example illustrates the monitoring of the ISL-Ion Source.*

2.3 Analysis Tools for Machine Physicists

As modern graphical tools often allow rapid prototyping machine physicists get more and more interested in developing their own analysis programs for machine study. Such programs require control access to vary setting values. The common Vsystem API's allow rapid programming, i.e. the required change and the visualisation of the parameters of interest for evaluation. Figure 5 illustrates a typical example of a VISTA driven ECR Ion Source Output plot by varying the setting of the analysing magnet, a tool which became a fixed part of the ISL operator GUI's.

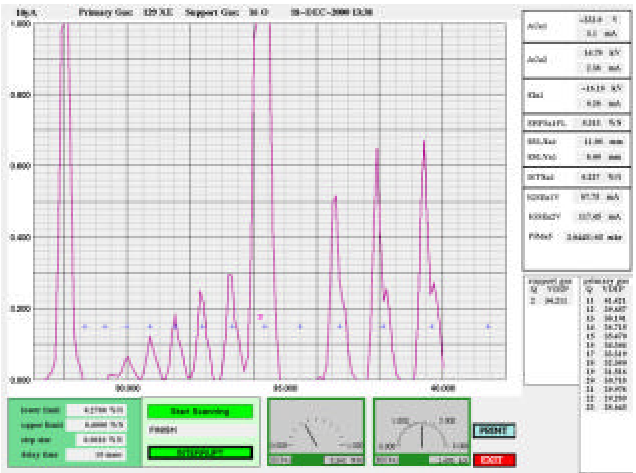

Fig 5: *VISTA tool for the analysis of the ECR Ion Source performance with underlying C-programmed data acquisition and evaluation.*

2.4 Control Access for Experiment Software

Furthermore, physics experiments have ever been asking for access to beam line parameters for logging purposes or even for control access to be able to cycle the experiment through required beam variations. Most of such experiments are run by local PC's equipped with individual experiment specific software products. In the case of LabVIEW, the controls group provides the VI-library for the support of remote control system parameter access. Figure 6 is an example for the LabVIEW integration of a fast local DAQ on a Windows2000 PC and the remote access to the radial probe position as reading and setting parameters of the accelerator control system.

Fig 6: *A C-programmed LabVIEW VI-library allows standardised remote access to parameters of the main VISTA control system. This example is part of the ISL control system illustrating the turn-pattern acquisition of a beam in the ISL cyclotron by moving a differential radial probe.*

3 SUMMARY AND OUTLOOK

Vsystem is a powerful commercial control system kernel with a collection of integrated tools for implementing and upgrading control systems under the hard conditions of a running facility and of restricted manpower. Our positive experience with transparent API's in heterogeneous networks and on multiple processing platforms leads to continue the distribution of control to front end PC's where it reasonably applies, working as PLC's with Vsystem kernel and database, and generally use remote access by operator GUI's at operator locations.

4 REFERENCES

[1] http://www.hmi.de/isl/index.html
[2] http://www.vista-control.com/
[3] http://www.hmi.de/isl/ics/datonl/index.html