RE-ENGINEERING CONTROL SYSTEMS

Peter Clout, Robert Westervelt, Mark Geib, and John Sinclair
Vista Control Systems, Inc.
134 B Eastgate Drive, Los Alamos, NM 87544, USA

Abstract

The pace of hardware and control system software development forces periodic, extensive upgrades to accelerator control systems. This paper explains why such upgrades are necessary, what aspects of an accelerator control system must be upgraded, several methods of managing an upgrade, and the challenges of migrating application software. The paper also briefly describes our experience re-engineering Vista Control System's core real-time database, Vaccess, using modern software engineering techniques.

Introduction

The rapid evolution of electronic, computer, and software technologies forces extensive, repeated upgrades in accelerator control systems. Though upgrades can be time-consuming, costly, and disruptive to operations, there are compelling reasons to upgrade before your current system becomes obsolete and unsupported.

Our experiences and those of our customers prove that you can effectively manage the process of re-engineering your accelerator control system. Drawing from these experiences, we have developed a few ideas that may help you manage a smooth control system upgrade. This paper contains the following information:

- Reasons to upgrade before the end of your control system's life cycle.
- Aspects of your control system to consider upgrading.
- Methods for managing the transition between old and new systems.
- Challenges you may face as you port software to the new system.
- A brief case study of our experience with software application migration as we re-engineered Vista Control System's core real-time database, Vaccess, using modern software engineering techniques.

Reasons to Upgrade Your System Now

Many production and research accelerators have useful lives extending well beyond the technology cycle of computer hardware and software. Many systems also have operational lives that extend beyond the maintainable life of interfacing electronics. Such extended life spans pose a management problem: Should you invest in keeping the control system current, or should you try to squeeze yet another year of operation out of the old system? Our view is that upgrading before the end of your current system's life cycle is preferable to the problems accompanying system obsolescence.

Perhaps the most compelling argument for upgrading before the end of your system's life cycle is the insurmountable obstacles you'll face when hardware or software is no longer maintainable. Just when product development and support is no longer available and when the pool of personnel qualified to maintain your system has dried up, you may encounter a barrier that limits your system's capability to accommodate growing operations. For example, software designed to support a limited number of I/O points might break when your growing organization exceeds the software's limits. With no product support and little in-house expertise, you may be forced into an on-the-spot (and extremely costly) upgrade.

Changing out a complete control system that has been developed and extended over the years is time-consuming, expensive, and disruptive to regular operations. But putting off the inevitable may only compound the problems and expenses associated with software upgrade.

On the other hand, an upgrade to a recently developed, state-of-the-art system offers possibilities for easy maintenance, support, and expansion. Production and research facilities can increase efficiency by taking advantage of new operating systems, software development facilities, and core control systems software now on the market.

Upgrades can also add new features that lead to better operations: protection and development environments isolated from the running system, for example.

Aspects of the System to Upgrade

Once you've considered the alternatives and decided to perform the upgrade, you're likely to begin extensive planning and budgeting. Here is a brief list of aspects of your system to consider upgrading:

I/O Subsystem

If you are thinking of re-engineering your I/O subsystem during the control system upgrade, think very carefully about the costs involved. These costs can skyrocket to $1,000 per channel when you include engineering, installation, and checkout. A small accelerator can have 2,000 channels, and larger systems one or two orders of magnitude more. Though you'll probably redesign the interface between the I/O subsystem and the control system software, the expense of re-engineering the I/O subsystem is usually prohibitive.

Computers, Networks, and Consoles

Also included in this category are updated user interfaces that may increase productivity among users of the control system software.

Core System Software

Core software often consists of the operating system and software development environment, with the particular I/O subsystem support added.

Application Codes

Designed to meet the special requirements of your particular machine and its users, application code is an ongoing expense. Custom applications can easily represent an investment of tens of man-years.

The last three items—computers, core software, and application codes—usually change together as a result of the decision to change the computer family.

Two Types of Transition

Upgrading your control system requires careful planning and a massive investment of time and money. But it is not always necessary to take such a drastic step all at once. Our customers have used two strategies to manage control system modernization: continual upgrades and project upgrades.

Continual Upgrades

A continual upgrade involves running parallel operations. This type of upgrade requires continuous effort on the part of the controls group; such an effort will only be successful if the caliber of people who originally designed, built, and installed the system can be retained in the controls group.

In practice, the continual upgrade approach works successfully only at major accelerators, where substantial new additions to the facility and investment in controls projects are required to keep the overall facility competitive.

To maintain continuous operations during a control system upgrade, the accelerator must conduct dual operation during the last part of the transition period, when the facilities are moved over to the new system one-by-one and the old system is eliminated. This transition period creates difficulties for the controls staff members who, regardless of good intentions, will spend more time on the old system than they did before the project was started. The most difficult continual upgrades involve porting from application software written in a now obsolete language.

Upgrade Projects

The second approach simply exploits the current system until a barrier of some kind appears and management initiates a project to upgrade the control system. A problem with this approach is the increase in staff required to implement the upgrade while maintaining existing operations.

Upgrade projects usually involve keeping the I/O hardware intact by having the existing computer system and the new control system share the I/O subsystem. There are three ways of sharing the I/O subsystem functionality during the dual operation phase of the project:

Dual I/O Subsystem. The accelerator can maintain dual I/O subsystems during the transition to a new control system. The key to success for this type of upgrade is to choose the point of dual access. If the I/O hardware is to be changed out, the dual access point is the wiring from the devices of the system (see Fig. 1).

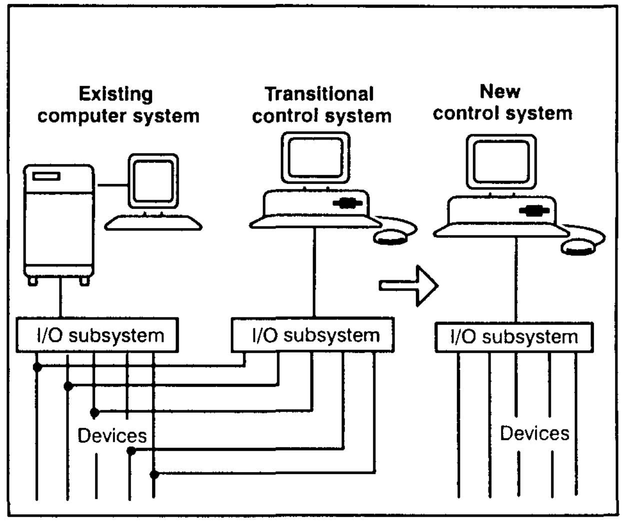

Fig. 1 Dual I/O subsystems

Shared I/O Subsystem. The two systems can actually share the I/O subsystem, if sharing is allowed by the existing I/O subsystem (see Fig. 2). The new computer system is simply plugged into the existing I/O subsystem and the change over can begin.

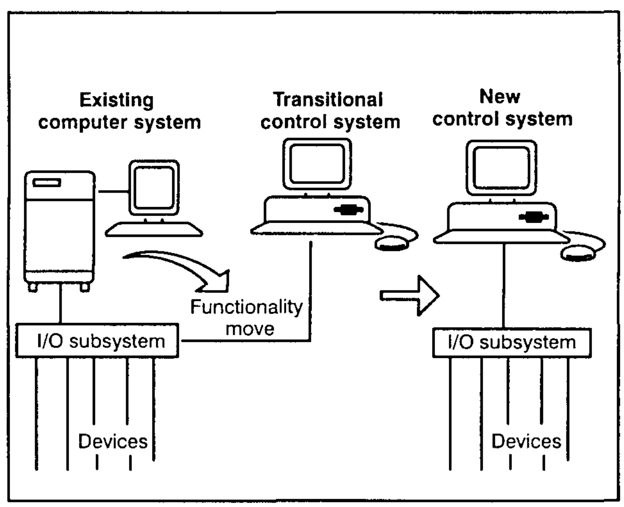

Fig. 2 Shared I/O subsystem

For I/O subsystems that do not allow I/O subsystem sharing, the transition can be accomplished by unplugging the old computer, plugging the I/O subsystem into the new computer for testing, and then restoring the old system for production again. Alternating use of the I/O subsystem is continued until the new control system is judged ready for operations.

Shared Computing Resources. Another technique for dual operations is making the I/O subsystem available to the new computers through the old computer system (see Fig. 3). As the transition progresses, the I/O subsystem is moved part-by-part over to the new computers.

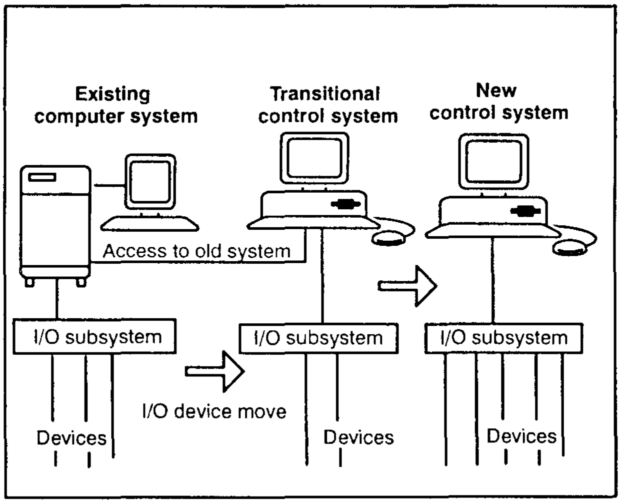

Fig. 3 Shared computing resources

Challenges of Application Software Migration

Control system software architecture should separate the various functional components logically, with well-defined interfaces. Such separation provides the benefit of simplicity in development, testing, and maintenance. Valuable, facility-specific application software (such as the physics and engineering models) separate from the operator interface code and the facility I/O code is easily ported to the new control system.

We have encountered a few challenges as we ported application software. Among these challenges:

• Extracting domain knowledge from the man/machine interface and the I/O access.

• Converting the untangled source to one of the current languages. Examples of programming languages now seldom supported are RTL/2, P+, and Forth.

• Converting the operating system access when the new operating system has radically different facilities.

Here are a couple of scenarios our customers have encountered:

• An application from a single computer control system assuming fast access to any I/O point is moved to a distributed computer control system. In the new, distributed system, each I/O access has a time overhead and the I/O accesses have to be organized for efficiency, whereas previously they did not.

• New facilities have been added to the control system using a variety of completely different technologies: There is a PC is sitting next to a PDP-11 and a Data General Nova computer. Here the job also includes control system integration to overcome the expediencies of the past.

Re-engineering Vaccess

To support new computers and operating systems, we decided to re-engineer Vaccess, our core real-time database, rather than port the existing code. Our reasons:

• The maintenance load of the current code was becoming large.

• We felt that porting the current code to a threaded environment would be difficult.

• Since the current code was tightly coupled to the VMS/VAX environment, the portability of the current code was low.

• New functionality planned for Vaccess would have been difficult to implement without massive changes to the current code.

Our first task was to develop a specification for the behavior of Vaccess. The current software had evolved over many years, and no precise description of behavior existed. The behavior was defined in terms of code tightly coupled to the VMS operating system. We will phase out the existing implementation. To replace it, we generated a precise, implementation-independent specification of behavior, using a hybrid functional/object-oriented methodology. This decision resulted in a high-level description of data structures and function, a description that facilitated the proper partitioning and isolation of platform dependencies.

The design phase placed special emphasis on supporting interoperability among diverse platforms. We included in the design an improved ability to test the system. Our new design also increased maintainability in anticipation of many future enhancements.

Finally, a strict review process ensured coherence between all phases of the development process.

The early results of implementation are high initial costs justified by significantly lower long-term maintenance and porting costs.

Conclusion

Upgrades to control systems can go smoothly and with little disruption to the operations of the facility if they are carefully planned and the appropriate additional effort is exerted. We have seen such upgrades successfully implemented as complete control system change outs, as well as simple, successful computer and software upgrades. It is vital that research facilities are kept current and useful for research in order to continue to receive adequate funding. The control system plays a role in this process by helping the operators maintain efficient and flexible operation both directly and through facility-specific application software.