The Case for Commercial Software

R. T. Westervelt, R. B. Rothrock, P. N. Clout
Vista Control Systems Inc.

Abstract

This paper examines the current trend in the particle accelerator community to use commercial rather than in-house software to develop process control systems. We will analyze why this trend has developed and discuss specific applications suited to commercial, rather than internally developed, software. We will conclude with features to consider in choosing commercial software for control projects.

Introduction

As a vendor of software suitable for particle accelerator controls, we advocate the purchase of commercial, as opposed to in-house, software solutions. However, our staff and management are long-term members of the particle accelerator community; many of us have lived on both sides of the fence first as in-house developers at laboratories and in industry, and then as commercial developers for Vista Control Systems. In this paper we examine reasons why commercial software provides an increasingly attractive option to home-grown control software. We analyze the trend toward purchasing commercial software and we discuss reasons why commercial software is often chosen over software developed in-house. Along the way we address some of the issues to consider in choosing a commercial approach.

Current trends

The use of computer control systems for particle accelerators began during the very early stages of computer technology development. No software was available, so accelerator personnel developed everything from scratch internally, from operating systems to programming languages to networks to databases. This situation resulted in many internal developments and enabled remarkable progress in accelerator and experimental physics. Today many accelerator controls groups have worked hard to maintain this position at the cutting edge of software technology.

As the computer industry matured, however, many of the fundamental building blocks for control system development have become available commercially. For example operating systems and networking technology available commercially have almost completely replaced home-grown solutions and hardware in both scientific and industrial computer applications.

To examine the trend toward commercial software implementation in particle accelerators we have studied the number of references to commercial software made in proceedings of previous ICALEPCS conferences. We count only those references to commercial software when it played a crucial role in the accelerator control system; we count only once any paper reporting the use of several commercial software packages. The dramatic trend toward use of commercial software reported in the proceedings is shown in figure 1.

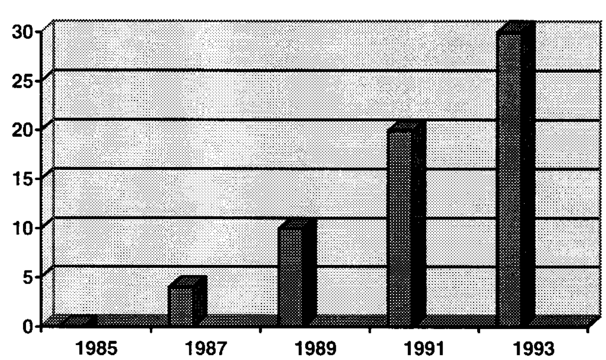

Figure 1 Commercial Software Usage Reported in ICALEPCS

Because ICALEPCS represents a worldwide community of control systems developers, the graph shows the broad base of this trend. The continued increase in references to commercial software over the past few conferences also shows that this trend has not yet reached its limit.

Software Requirement of Physics Control Systems

The topic of the requirements for software for physics control systems has been extensively discussed. One good summary is the panel discussion on software sharing from the last ICALEPCS (1). That panel discussion and the work being done on software sharing have identified the key application areas that are common across control systems and represent the areas that might benefit most from investment in commercial technology. The types of applications needed in the modern control systems and the characteristics that good components must have are listed below:

- User interface development
- Console manager
- On-line help
- Database access
- Real-time database management
- Sequence builder
- Simulation access
- Data logger
- Archiving system
- Alarm system
- Equipment access
- Startup/recovery mechanism
- System configuration tools

Some of these application areas are common to both industrial and scientific control systems. Industrial systems have the same general requirements as scientific ones for user interface development, help, real-time database management, data logging, and alarm management. This overlap in functional requirements contributes to the synergy between commercial applications and scientific control systems. This synergy in turn drives the trend for using commercial software in scientific control systems. Some other application areas provide common ground for commercial and industrial control systems, but the degree of overlap is not so great as in those applications mentioned above. Equipment access on an industrial level tends to be centered around access to various types of commercial PLCs with less use of faster and more diverse data interfaces, such as VXI, VME and CAMAC. On the scientific side this equipment mix tends to switch more toward high-speed interfaces, reflecting the need for gathering data on the time scales associated with acceleration cycles. Like scientific control systems, industrial ones need access to simulations, but in the industrial case the model codes tend to be built directly into the control system itself and are routinely executed as part of such a control system. For industry there is less human interaction with the models as part of the control process.

Why commercial software abounds

Economics: Economics plays perhaps the greatest role in the move toward commercial software in the accelerator and physics communities, where fundamental changes are taking place. In the United States, for example, the SSC has been canceled and funding is tight at other laboratories. Commercial software can reduce the total cost of the basic software by at least a factor of ten (2).

One of the arguments for in-house software development has been cost control. But the reality is that one underestimates the budgets for in-house projects, and inflates them with expanded requirements for commercial ones. We at Vista Control Systems have experienced first-hand the impact of misjudging in-house project duration and effort required: twice we chose to develop tools to re-engineer a core product, rather than purchasing commercial tools to facilitate the re-engineering process. We felt we could develop such tools for half the cost of a commercial product, but in the end our costs to develop them more than doubled because we underestimated the manpower required and because we expanded our requirements as we began the re-engineering project.

Personnel costs can be underestimated also. Consider the development of user interface software. Most modern graphical user interface systems such as Xwindows/Motif or Microsoft Windows contain enormous capabilities accessed through large and complex APIs. Teaching a staff to use all of the capabilities requires a large investment in time–time better spent developing control system requirements, managing the system and expanding applications. Because many commercial packages provide user interface management systems there is no need to invest time developing expertise that is difficult to retain after operations begin and that will someday be outmoded.

Don't think personnel costs will be eliminated, though, with commercial software. Because they are by necessity generic solutions for a variety of industrial and scientific controls, commercial systems do not often address the specific needs of the particle accelerator community or any particular facility. Staff must take advantage of the extendibility built into these products to tailor control system software.

Compatibility: Often, in the name of expedience, internal software packages fail to provide compatibility between software versions. However the pressures of the marketplace have made backward compatibility an essential component of a good software release for commercial vendors. At Vista we have more lines of code in the regression test suites for our database software than lines of code in the Vaccess database product. We also perform extensive beta testing–a luxury not often afforded in-house developers.

Support: Commercial software packages generally provide professional support, training and documentation. These elements enable a smooth, speedy implementation of control software, ensure controlled start-up costs and provide continuing support despite years of operator and staff turnover.

Low risk: In application areas common to industrial and scientific controls commercial software is a sound, tested alternative to that developed in-house.

Source code arrangements: Customers of commercial software are often concerned about access to the source code. The need to fix bugs quickly and to add new functionality is important to industrial and scientific customers. And all customers want to guard against the software vendor going out of business or abandoning the software product.

Standard software engineering and legal practices, though, allay fears concerning source code access. First, good software is extensible so that users can expand it to suit their needs. An example is the IDL data analysis package which interfaces easily to other programs. In the PC market most major software vendors provide scripting packages to extend functionality. A favorite example of this extendibility is the adding of a new device driver to an operating system. Access to the operating system source code may be useful, but if that system is well documented and adequate examples are provided, having the source code is not necessary.

As a vendor we do not recommend that customers modify source code due to the danger of introducing errors and making it difficult to provide support. Further, we provide legal and administrative mechanisms to address concerns about our company abandoning a software product. Source code escrow agreements provide customers with the source code in the event that the company does cancel maintenance of any product.

Conclusions

Our conclusion is that carefully chosen commercial products provide a sound economical alternative to in-house software. With a wide variety of flexible, well-supported, extensible commercial packages on the market, in-house development is now only one option, not a necessity. When beginning to look at the commercial options, we suggest considering a software package that provides

- much of the functionality you need in order to control your process effectively.

- mechanisms for your in-house staff to build additional functionality into their software controls.

- backward compatibility between software versions.

- professional support, training and documentation.

- a history of effective implementation in both scientific and industrial applications.

a• n acceptable source-code escrow agreement.

If one can find these qualities in a commercial software package, the chances are that the package will greatly benefit his control systems project and allow his organization to devote greater effort and more resources to its prime tasks.

References

(l) B. Kuiper et.al., Panel Session on Software Sharing, NIM, A 352 (1994), pp 501-515.

(2) P. Clout, "Sharing Control System Software," *Proceeds of the 1993 Particle Accelerator Conference,* Vol. 3, 1993, pp 1801-1805.