



Title:GUI builders require application portability trade-offs. (graphical user interface) (includes directory) (Buyers Guide)

Pub:*Software Magazine*

Detail:Emily Leinfuss. 13.n5 (March 15, 1993): pp15(5). (2060 words)

Abstract:

Users seeking to build portable and complete graphical user interface (GUI) applications face some trade-offs on the benefits they get, because GUI development tools offer differing database, network and operating system portability. The two types of GUI builders in the market are: GUI-only tools, that provide a runtime environment for most GUI platforms, and GUI tools that offer both portability and an application development environment. The latter type of tools are more useful and more integrated than the GUI-only tools. While these programs are intended to free the programmer from the complexity of user interface technologies; and management, network and operating system technologies, actual products in the market do not have the capability for providing all those technologies. Some users' experiences with GUI tools are described.

Full Text:COPYRIGHT Sentry Publishing Company Inc. 1993

GUI-ONLY TOOLS EASILY CROSS GUI PLATFORMS, BUT TOOLS THAT BUILD GUI APPS MORE LIMITED; SUPERNOVA HELPS INTECOM DEPLOY ON UNIX

Client/server developers dream of an open environment with easy, seamless and transparent access to all platforms, systems, networks and databases. Building that environment is a different story, however. Customers face a trade-off between the benefits they get from tools that provide application portability only at the GUI level, and their desire to build complete GUI applications that are also portable.

Client/server development tools -- particularly graphical user interface (GUI) builders -- vary in what they provide for database, network and operating system portability.

GUI builders on today's market fall into two classes. The first, GUI-only tools, offer a runtime environment for most GUI platforms. Supported platforms typically include Microsoft Windows from Microsoft Corp., Redmond, Wash.; IBM's OS/2 Presentation Manager; OSF/Motif from the Open Software Foundation, Cambridge, Mass.; Open Look from Hewlett-Packard Co., Cupertino, Calif.; and the Macintosh from Apple Computer, Inc., Cupertino, Calif.

The second class of GUI tools offers an application development environment along with GUI portability. These GUI environment products also vary widely. Typically, these tools are based on third-generation, fourth-generation and object-oriented languages.

The purpose of both classes of tools is to isolate the GUI developer from the complexity of the underlying user interface technologies, as well as management, network and operating system technologies, said Scott Sargent, associate partner at the New Age Systems Group of Andersen Consulting, Chicago. However, he said, "None of the products available cut across all those technologies."

Sargent added that the GUI building tools that offer their own environments are more integrated and cover more ground than GUI-only tools.

LIMITED LIFE SPAN

In the first GUI tool class, products like the XVT Portability Toolkit from XVT Software, Inc., Boulder, Colo., and Open Interface from Neuron Data, Palo Alto, Calif., offer a GUI shell, noted Andrew Mahon, senior analyst

for advanced software development services, at consulting firm New Science Associates, located in Westport, Conn. None of them can write calculations or queries. However, they have a graphical field in which developers can write limited program logic, said Mahon, who is based in Boston.

Mahon believes that the XVT and Neuron tools have a limited life span because they only cover a certain amount of function. This type of tool is ideal for commercial software vendors who want to capitalize on their product by porting it to other graphical environments. However, Mahon said, corporate developers "want a higher level, not just the ability to write the GUI, but also the application."

Not everyone agrees. Some corporate developers want the ability to use their existing development tools with the GUI builders, and are pleased with the functionality of the GUI-only products.

General Physics Corp., in Columbia, Md., develops expert system-based custom software. The company originally targeted most of its development for Unix because of the operating system's processing power. But now, with the power of 486 PCs and the new Macs, the company wanted the ability to develop applications on those platforms.

Eric Bishop, technical project manager at General Physics, chose Neuron Data's Open Interface because of the product's ability to develop on DOS and Macintosh platforms, and because he felt that GUI building tools for Unix platforms were more expensive. He also did not want to give up his favorite code editor, VI (Visual Editor), and other tools. VI is the original standard full screen editor for Unix that comes with the operating system.

Bishop uses Open Interface to develop on Macintosh IIfx and PC 386 or 486 compatibles running Windows 3.1. He deploys the applications on Macintosh and Windows platforms.

Open Interface provides a good framework to develop GUI applications, he said. "In most development the GUI drives the application. What is neat about OI [Open Interface] is that it gives a robust architecture in which to develop an application. Then you can go out to your normal programming environment. So I'm not forced into some environment, hoping the editor is full-fledged. I don't have to relearn commands," he said.

Bishop said he mistrusts tools that have what he calls "monolithic" tool sets. "When vendors claim to be the be-all and end-all for development, you had better hope they have everything you need to use. If not, you get thrown into the lowest level of machine interaction." As a result, this does not allow the developer to use any high-level tools. "You would have to have a guru programmer who knows how to deal with the low-level machine code on a primitive level," he said.

SELECTION BASED ON NEEDS

GUI builders that offer development environments differ in their underlying technology. They provide varying levels of integration with databases, network transparency, and operating system and application logic portability.

Customers should base their selection of a GUI portability tool on application needs and developer preferences, said Andersen Consulting's Sargent. That means developers must be aware of what capabilities their tools have for object-oriented development, database portability or GUI independence. With that knowledge, he added, they can determine the best environment in which to work, based on their needs.

Bob Millen, senior staff software engineer at Intecom, Inc., Allen, Texas, a network management software developer, had two main requirements for developing a Unix-based network management application -- database and GUI platform portability. He also wanted to avoid having to code at the native GUI level.

Intecom needed to develop its application to run on HP's Open View for both Unix and DOS. Millen explained that HP has one division that provides network mapping GUI capability for Unix and another division that does the same for DOS. "We wanted the Intecom network mapping application to have the ability to interface with the databases in the two different environments," he said.

After evaluating both XVT and Neuron Data's tools, Millen chose SuperNova, a fourth-generation application development tool, and SuperNova GUI Builder, both from Four Season Software, Edison, N.J. The GUI Builder offers front ends to MS Windows and Motif, and Four Season will be introducing an Open Look version shortly. Versions for Presentation Manager and the Macintosh are in the planning stages.

Millen develops on a 386 PC platform running Windows 3.1, and deploys applications on Motif platforms on workstations from HP and Sun Microsystems, Inc., Mountain View, Calif., and soon on the IBM RS/6000.

SuperNova lets an application sit on top of different databases without having to make any changes. Millen felt that the XVT and Neuron products, while affording GUI portability, did not operate at a high enough level for the firm. "With SuperNova I am portable, avoid the minutia of having to work with the local GUI, and the application can simultaneously access different tables on different databases," Millen said.

RDBMS INTEGRATION

Pompi Malik, manager of information technology at Brewers Retail, Inc., Mississauga, Ontario, also had a need for GUI portability and database access. When developing a Unix-based client/server application for balancing inventory, his greatest concern was for tight integration with SQL Server, the relational database management system (RDBMS) from Sybase, Inc., based in Emeryville, Calif.

Malik opted for the Universal Presentation Interface (UPI) development environment from Uniface Corp., Alameda, Calif. He liked the tool's close integration with SQL Server. "Our application is very tightly integrated with Sybase, and Uniface has a strategic alliance with Sybase. We didn't want to look at tools that weren't going to exploit all the advantages of client/server-specific Sybase-stored procedures and triggers," said Malik.

Brewers Retail deploys applications on Windows and DOS character-mode platforms.

Malik also evaluated some object-oriented products before choosing Uniface. He likes and agrees with the object-oriented paradigm, and plans to start trying it out. However, he said it was enough of a challenge right now for his assembler-based programming staff to move to a 4GL programming environment. "To actually change the way they think about data was too big of a leap," he said.

WANTED: OOP ENVIRONMENT

GUI portability was only part of the reason Harvey Kramer decided to use VisualWorks from ParcPlace Systems, Inc., Sunnyvale, Calif. Productivity gain was a key reason that Kramer, manager of applied technology for Southern California Edison Co., Rosemead, Calif., wanted an object-oriented programming (OOP) application development environment.

Another major requirement was that he wanted his users and clients to be able to choose the most appropriate platform for them. "That means that some clients are using Windows and some the Mac. And some may want to change in the future," he explained.

VisualWorks, said Kramer, allows him to "develop one application and port to all those platforms for the manpower price of zero," he said.

With VisualWorks, Southern Cal Edison develops applications on OS/2 and Windows platforms, and deploys them on Windows and Macintosh platforms.

VisualWorks is said to port to all current GUI platforms. Based on the OOP language Smalltalk, the product is also binary-compatible. This means that applications run on each platform without requiring recompilation, allowing the developer to maintain one set of source code for all platforms.

This is a major advantage, said John Bruzas, product manager at ParcPlace. "We didn't want to put the burden of work on the developer to have to, for example, create source code on a Windows machine, then have to move text files onto the target machine. That forces the developer into making sure there are the proper directories and

appropriate compilers on the system, then compiling and linking the application. That is why we provide one image on all platforms," he said.

Another proponent of OOP is Ian Wu, president of San Francisco-based Auto View, Inc., which develops applications for the real estate and pharmaceutical industry. He said C++/Views from Liant Software Corp., Framingham, Mass., met one of his main requirements -- the ability to isolate the presentation code from the application code. From a development standpoint, this lets Wu prototype and make dramatic changes to an application without having to make code changes to the GUI interface.

Aside from OOP, a prime goal was to find a GUI development tool that allowed him to get as far away from native GUI code as possible. "We have a lot of projects and didn't want to spend a lot of time doing native code, especially since we intended to go cross-platform," said Wu.

With C++/Views, Auto View programmers develop on Windows platforms and deploy applications for OS/2 and Macintosh platforms.

Wu evaluated Neuron Data's Open Interface, but said it was too expensive for what it did. He initially chose XVT's XVT Portability Toolkit because it seemed to have extensive features. However, Wu said a lot of things he wanted to do were too cumbersome with XVT. "Its support of graphics manipulations was lacking. There were things we wanted to do with Windows clipboard -- a lot of graphic imaging -- and it couldn't handle that. We would have been forced to go back and write Windows code," he said.

HOLD THE 'EXTRAS'

Just as each tool has different capabilities, users may have different needs. SpaceLab Medical, Inc., in Redmond, Wash., did not need GUI portability but did want cross-platform portability. SpaceLabs Medical develops patient care monitors and hospital information systems.

Roger Talkov, senior software engineer, wanted the ability to run Motif-based applications on any hardware platform. He chose Builder Xccessory from Integrated Computer Solutions, Inc. (ICS), located in Cambridge, Mass.

Talkov wanted portability, but he did not want the extra baggage that some of the GUI portability tools carry. "We didn't want those extra things. We have no requirement for Open Look, Windows or Mac."

He did evaluate the Jam family of products from JYACC, Inc., New York City, but he said there was too much integration with databases. "They were doing too much stuff behind our back. We wanted to stick with simple tools and be able to mix and match any tools we wanted to use -- any database, any debugger," he said.

With corporate computing environments that are increasingly heterogeneous, the desire for a product that will get users and developers to all of their platforms without much work will only intensify. Today's GUI builder tools offer a variety of features, but no one tool can do everything--yet. The available tools address different parts of an enterprise better than others, said Andersen Consulting's Sargent. The trick for developers is to figure out which tools fit their needs.

Source Citation

Leinfuss, Emily. "GUI builders require application portability trade-offs." *Software Magazine* 13.5 (1993): 15+. *Academic OneFile*. Web. 10 Nov. 2011.

Document URL

<http://find.galegroup.com.ezproxy.lib.swin.edu.au/gtx/infomark.do?&contentSet=IAC->

[Documents&type=retrieve&tabID=T003&prodId=AONE&docId=A13663723&source=gale&srcprod=AONE&u](http://find.galegroup.com.ezproxy.lib.swin.edu.au/gtx/infomark.do?&contentSet=IAC-Documents&type=retrieve&tabID=T003&prodId=AONE&docId=A13663723&source=gale&srcprod=AONE&u)

Gale Document Number:A13663723

- [Contact Us](#)
- [Copyright](#)
- [Terms of use](#)
- [Privacy policy](#)

