Determining a Best-Fit Measurement Server Implementation for Digital Design Team Solutions

Prototype analyzer customers wanted fast throughput, quick answers, a turnkey solution, an affordable base system price, connection to diverse open-systems networks and platforms, and interfaces to a wide variety of tools. An encapsulated measurement server architecture based on a dedicated workstation and a SCSI II interface best fit the requirements.

by Gregory J. Peters

Implementing an analysis and display architecture for the next generation of HP logic analyzers presented opportunities and challenges for the cross-functional design team assigned to the task. Freed from the boundaries of commercial real-time instrument operating systems and needing to offer a flexible and high-performance window-based interface, the project team took a long look at both technologies and the total envelope of costs before settling on an encapsulated measurement server architecture based on an HP 9000 Series 700 workstation.

The goal of the HP 16505A prototype analyzer team was to develop a breakthrough analysis and display environment for digital design teams involved in the integration and debug of prototype systems. The resulting product is a combination of the best of traditional instrument and system benefits. The turnkey prototyping system provides the performance and flexibility demanded by leading-edge digital design teams but its low total system cost appeals to the smaller teams as well. The tough decisions made very early in the product definition phase have paid off. The product meets market needs and has a low cost of sales and support envelope.

Genesis

The genesis of the prototype analyzer occurred with the realization that digital hardware developers, who are traditional logic analyzer users, were spending an ever higher percentage of time using electronic design automation (EDA) tools, such as schematic capture and simulation software packages. The use of workstation or PC-based EDA tools came at the expense of time spent in the lab making measurements on prototype hardware.

The workstation or PC was fast becoming home base for designers. Traditional test instrumentation was in danger of becoming less relevant to the design team, in large part because it didn't interface with EDA tools. The question the team pondered was how future real-time measurement systems would interface with the designer's workstation home base.

This concern generated an investigation into workstation connectivity, and the prototype analyzer project was born. The development team at first conducted research with hundreds of customers worldwide, in different industries and with different measurement needs. The results of the investigation drove the definition of the prototype analyzer measurement server implementation.

Measurement Servers

Measurement servers are defined as a class of solution that combines physical measurements with client/server computing. Real-world measurement data is captured, analyzed and shared with users and other tools over the network using a variety of open-systems standards.

Measurement servers use widely accepted networking standards such as Ethernet, the X Window protocol, and the Network File System (NFS) to enable customers to control and view measurement instruments and get easy access to measurement data. Standard data formats are used to interconnect the measurement server with other applications. Specialized host-based software is not required. Measurement servers can provide anywhere, anytime access to prototype measurement data via today's ubiquitous networks.

Client/server computing enables each component of the system to specialize in what it can do best. The locations and definitions of the interfaces between components are determined by the nature of the application and the network topology. A well-recognized example of a client/server application can be found in the EDA industry: a simulation engine runs on a large compute server while the simulation interface runs on the engineer's desktop. Application developers must pick the appropriate points to split apart an application to maximize a product's performance and value.

Measurement server architectures vary depending on the scope of the solution. Some architectures are completely encapsulated inside a box, while others are split apart or distributed (see Fig. 1). Identifying the appropriate point to split the application apart is critical. Splitting an application at the wrong place can limit response times or cause nondeterministic behavior. Encapsulating the wrong functionality into a system can increase the cost of the system or the cost of sales without increasing revenue. Encapsulating too little functionality may result in trivial or redundant products.

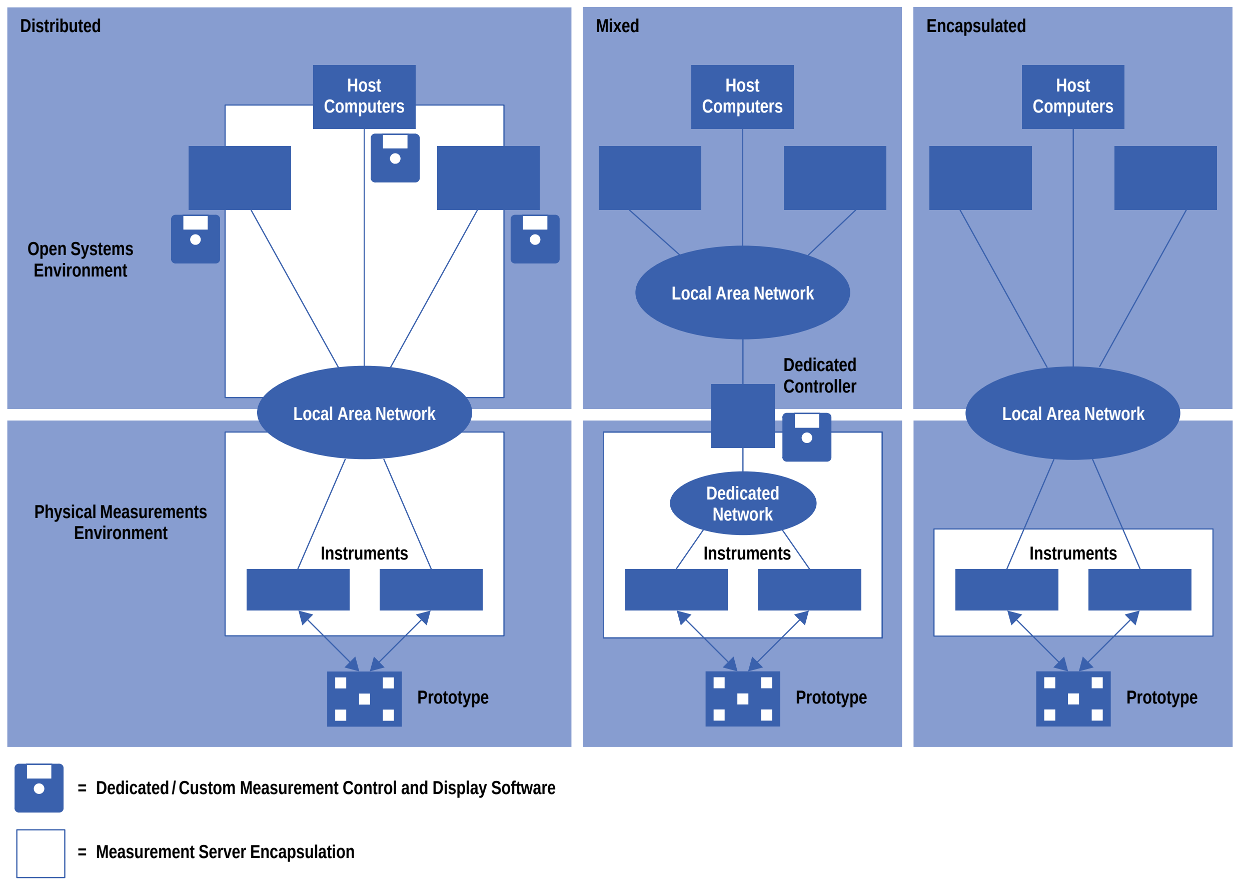

Fig. 1. *Three possible measurement server architectures: distributed, mixed, and encapsulated.*

The full use of the client/server paradigm is the differentiating aspect between traditional instruments and measurement servers. Traditional instruments can also be connected to computers via the HP-IB (IEEE 488), but this connection requires special interface[1] and control hardware and software on the host computer. Additional analysis and display software is generally custom, and can be time-consuming to develop and debug. The localized nature of the HP-IB limits its penetration into open computing environments and the need for specialized control and interface software puts HP-IB-controlled instrumentation at a distinct disadvantage in the networked design environment.

Architecture Selection

To obtain the best price and performance trade-offs, the proposed solution must be evaluated closely to determine the optimal measurement server architecture. The three examples given below highlight some of the issues that must be addressed. The prototype analyzer development team was able to gain valuable experience from reviewing each of these product forms.

An example of a distributed measurement server architecture can be found in the HP B3740A software analyzer. This software package provides software developers with a means of viewing real-time microprocessor traces referenced to source code. The HP B3740A runs on several workstations and on PCs.

The host-based nature of this application was derived from a need to provide source-viewing functionality to a large number of design team members who timeshare a single real-time instrument (the HP 16500B logic analysis system). In this case, the small amount of data transferred over an Ethernet link between the HP 16500B and the HP B3740A does not impact performance. Performance here is measured as update rate.

As a point application, this product does not require extensive support. If the same approach were taken for several applications, however, the sales and support burden could limit broad market penetration. In particular, maintaining host-based software on a platform that has frequent operating system revisions is not only time-consuming but also results in a low return on investment to the development team.

HP VEE, a visual programming and test development environment,[2] best illustrates a mixed measurement server architecture. HP VEE runs as a host-based program on PCs and workstations. This product enables engineers to quickly develop test programs that control a wide variety of instruments over the HP-IB on a standard computing platform such as a workstation or PC. The key benefit of this approach is the flexibility afforded both solution providers and end users. A choice of supported platforms means customers can use a platform they are familiar with. An open-system development environment enables value-added engineering, upon which solution providers can build custom applications.

A mixed measurement server architecture has drawbacks, however. Maintaining a reliable interface between the host-based software and the measurement instruments through operating system revisions or changes in network protocols can be challenging. Setup and maintenance of a distributed system require significantly more effort than a turnkey instrument. A dedicated Ethernet network can impose bandwidth barriers that severely limit measurement throughput.

The HP 16500B logic analysis system with an HP 16500H,L interface module is an example of an encapsulated measurement server. This system can provide both local and remote control and viewing via the X Windows protocol. The turnkey nature of this product makes installation and setup simple. Since all data processing takes place in the instrument, only X Window calls are transmitted over the Ethernet. This helps keep the update rate fast over most network topologies.

While the support burden of an encapsulated measurement server is low, it comes at a price. User-defined measurement sequences and extended data analysis and reporting are not provided. Customers must make a connection to the HP 16500B and import data and screen images into a general-purpose computer to perform these tasks.

Considerations such as customer expertise in open systems, support costs, and the use model for the product must all be factored in when choosing an architecture for a measurement system. Trade-offs must be made to fit the needs of the market and target customer. Table I outlines the advantages and disadvantages of the three measurement server architectural approaches.

Customer Requirements

The workstation connectivity investigation generated an extensive set of customer requirements. These inputs spanned the entire realm of product definition, from technology needs through user disciplines and measurement tasks. The investigation revealed:

- Prototype measurements are essential for hardware/software integration and product verification.
- There is a wide variety of use models, from the benchtop standalone user to completely networked control and viewing of measurements from a remote site.
- An analyzer might see single-person use or shared use by teams of as many as 50 engineers.
- There is a strong desire for a turnkey system, not a collection of components that must be integrated by the customer.
- Equipment setup time must be a minimum. Customers want to be able to make a connection to the prototype and make measurements as soon as possible.
- There is a broad diversity of measurement needs, from signal integrity to software, and across a variety of probed points, from serial buses to RISC processors.
- Surprisingly, we found the potential to address a broader group of customers than envisioned before the investigation, including software developers, who traditionally avoided the lab and used host-based software development tools.
- We heard a strong request for snappy displays and quick measurement results (facilitation of the debug loop explained in ***Article 1***).
- The solution must interface with a wide variety of design tool chains, consisting of EDA software programs, test development programs, and others.
- There is a desire for connection to diverse open-systems networks consisting of nearly every possible combination of computer hardware and operating system software on the market.
- The base system must be affordable to reach most users and scalable to cover both performance and mainstream users.
- The Ethernet protocol was in use at most customer sites.

Taken together, these inputs overwhelmed the project team. Scaling the task to available time and resources became a critical project goal. Reducing the dimensionality required of the architecture would be the key to delivering an on-time product that met key user needs.

Table I
Measurement Server Architectures

Advantages

Distributed
- Fastest leverage of links to other host-based applications.
- Most efficient at using customer's available computing power (MIPS). Takes advantage of idle MIPS on customer's host computer.
- Low cost of goods sold (software only).

Mixed
- Provides best mix of flexibility and power for custom system development.
- Best for integrating different measurement systems.
- Deterministic performance can be achieved over a dedicated network (HP-IB).

Encapsulated
- Turnkey solution (ready to run).
- Very high measurement throughput.
- Deterministic performance.
- Low cost of sales.
- Low cost of support.
- Low R&D resources required to support and enhance the system.

Disadvantages

Distributed
- Low measurement throughput.
- Nondeterministic performance (Ethernet).
- Requires continuing R&D effort to maintain support on new operating system revisions.
- Requires extensive and time-consuming QA on supported host platforms.
- Requires some customer operating system and I/O knowledge.
- Higher support costs than encapsulated solution.

Mixed
- Requires special dedicated network hardware and software.
- Requires customer operating system, I/O, and networking knowledge.
- Throughput can be limited (depending on the performance of the dedicated network).
- Requires continuing R&D effort to maintain support on new operating system revisions.
- Higher support costs than encapsulated solution.

Encapsulated
- High cost of goods sold.
- Fixed functionality.
- Low efficiency of using customer's available MIPS.
- Requires some networking knowledge.

Demographic Factors

Digital design teams of all sizes are found in every industrialized country. Design teams as small as two engineers expressed many of the desires listed above. Because of the broad demographics of digital design teams, a key challenge was offering a low cost of sales and support product structure to fit a sales channel that could reach these customers.

A complex product structure greatly increases the cost of sales and support. A high cost of sales makes broad product adoption difficult. This is because complex product structures typically demand a specialized sales and support organization, which in turn limits coverage both geographically and at smaller accounts. Product specialists are expensive because of their training and expert knowledge, but also because it is difficult for them to cover the same geographic area as thoroughly as a group of generalists who make daily contact with a wide variety of customers, and who sell a wide variety of products.

Product generalists are far more likely to make contact with small customers. These customers are less inclined to purchase a system they view as so complex that a specialist is required to sell and support it. Creating a product that demands product expertise and specialization of a sales force directly contradicts the fact that design teams of all sizes and budgets have many of the needs listed above.

The resolution of this issue became a focal point of the prototype analyzer product structure. The outcome of the measurement server architecture decision would be the major contributor to the cost of sales. Reducing the dimensionality of the product would help lower overall costs and allow the product to be sold and supported by generalists, not specialists.

Issues

The four issues that most affected the choice of measurement server architecture were:

- Fast throughput and quick answers
- A desire for a turnkey solution
- An affordable base system price
- Connection to diverse open-systems networks and platforms and interfaces to a wide variety of tools.

The additional requirement that was used as a starting point for the design was that the product must work with existing HP 16500B systems and measurement modules. The design team was to create a product that added functionality to the HP 16500B system but did not require customers to make a significant new investment in real-time acquisition hardware.

Work on the core software architecture began independently of the measurement server decision. The four issues were used as a basis for discussion about the pros and cons of each measurement server architecture. In the end, an encapsulated architecture was chosen, as shown in Fig. 2.

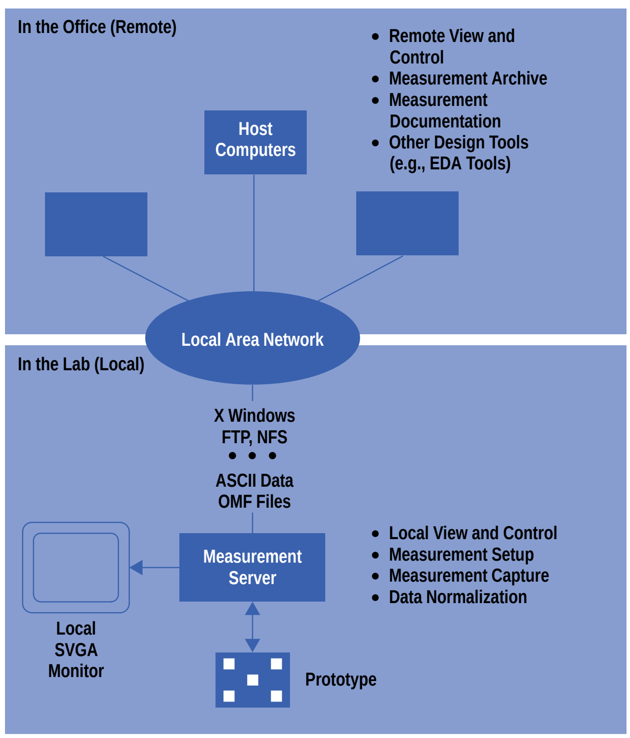

Fig. 2. *HP 16505A prototype analyzer measurement server implementation.*

Throughput

Throughput was a key element of the prototype analyzer design. A customer set a benchmark for the design team by stating the expectation that a full screen of HP 16550A data should be refreshed once per second. This became known as the "one update per second" goal and was used as the de facto throughput benchmark.

Table II outlines the amount of data that some HP 16500B logic analysis system measurement modules capture in one acquisition. The italic numbers indicate common configurations. These configurations were used in ad hoc tests by R&D to evaluate the update rate as the software architecture progressed.

A single acquisition covering several microseconds could generate as much as 30M bytes of data. Sending this amount of data over a customer's local area network would be impractical and would cause severe network performance problems. It was obvious that some sort of dedicated network would be required to move data quickly from the HP 16500B system to a workstation or PC for data normalization and display.

Table II
Typical Measurement Data Sets

HP 16500 Measurement Module	1 Module	2 Modules	3 Modules
HP 16550A 500-MHz Timing, 100-MHz State, 4K Depth, 100 Channels	55K bytes	*110K bytes*	N/A
HP 16555A 500-MHz Timing, 110-MHz State, 1M Depth, 64 Channels	8M bytes	16M bytes	*24M bytes*
HP 16532A 1-GSa/s Digitizing Oscilloscope, 8K Deep, 2 Channels	*16K bytes*	32K bytes	48K bytes
HP 16517A/18A 4-GSa/s Timing, 1-GSa/s State, 64K Deep, 16 Channels	*128K bytes*	256K bytes	384K bytes

The real-time nature of the system implied a robust and deterministic interface. Handling real-time I/O across open networks without the benefit of a well-defined protocol is difficult, and would require special drivers on both ends of the network. Porting real-time code across platforms would add complexity to the design.

Leveraging analysis done by HP Laboratories and other divisions,[3] the design team was quickly able to evaluate interface performance characteristics. Table III provides a comparison of different interfaces and the estimated throughput of each. Data handling in the HP 16500B and normalization time in the HP 16505A is not included in these figures.

Table III
Interface Speeds and Estimated Throughputs for 110K Bytes of Data

Interface	Maximum Transfer Rate	Typical Transfer Rate	Transfer Time for 110K Bytes of Data	Transfer Time for 24M Bytes of Data
HP-IB	1 Mbyte/s	240 kbytes/s	0.47 s	100.4 s
Ethernet	1 Mbyte/s	300 kbytes/s	0.38 s	80.7 s
SCSI II	5 Mbytes/s	1.5 Mbytes/s	0.07 s	15.27 s

Although all three interfaces perform the 110K-byte transfer in less than 0.5 second, the SCSI II interface offered a substantial improvement in performance, which would be needed when transferring the 30M-byte files found in high-end HP logic analysis configurations.

SCSI II was not the first choice of the design team. The HP 16500B already had HP-IB and Ethernet ports. Adding a SCSI port would take more time and resources. Some team members argued that HP-IB performance could be improved. However, the HP-IB interface would then require a corresponding connection on the other end. Since workstations and PCs don't come standard with HP-IB interfaces, the use of this port would require special hardware for the host computer. No one relished the task of designing a computer-based HP-IB card or evaluating the commercially available cards.

Ethernet was a clear winner from a user perspective and an Ethernet port was available on the HP 16500B. The two strikes against Ethernet were its performance compared to SCSI II and its inherent nondeterministic behavior, a result of the collision detection and retransmission scheme used.

In the end, the SCSI II port won out. In retrospect, the use of the fastest interface available was an excellent choice, since HP 16500B data sets continue to grow in size and customer throughput expectations constantly increase.

As the design team developed the software architecture, it became apparent that there were many areas where code optimization could improve throughput. A problem with software optimization is that it is often dependent on the architecture of the underlying hardware. Although the team was using the HP 9000 Series 700 workstation as a development station, a platform choice had not yet been made. One factor that swayed the development team in favor of an encapsulated measurement server was their feeling that significant improvements to performance could be obtained by tuning the software architecture to one computer architecture. This insight proved fortuitous because the R&D team got the chance to optimize the architecture and gain a 10× performance improvement when the coding was complete.

The decision to use SCSI II meant that a distributed measurement server architecture would not be feasible, since SCSI is not a network protocol.

Turnkey System

With the interface issue settled, the design team began investigating dedicated controllers. Both workstations and PCs were evaluated. Each platform had advantages and disadvantages, as outlined in Table IV.

Table IV
Controller Comparison

Advantages

Workstations

- Highest single-processor performance available
- Standard SCSI interface
- Good networking
- Good development environment
- Can act as X client or server
- Excellent graphics support

PCs

- Good performance
- Generally lower cost than workstation
- Excellent development environment
- Customers familiar with Microsoft Windows®
- SCSI interfaces available

Disadvantages

Workstations

- Higher cost than PCs
- Many customers not familiar with HP-UX* operating system
- Potential for file system corruption
- Requires more base system memory

PCs

- X client software not readily available
- Acceptable but not robust networking

An important factor in the decision between a workstation and a PC was the ability of a workstation to operate as both an X client and an X server. Since customers demand both local and remote viewing and control of the prototype analyzer, X Windows was a necessity. In general, PC-based operating systems did not support this function or handled it as a special case.

However, a problem with the choice of a workstation was the HP-UX operating system. Many customers were not familiar with HP-UX and did not want to learn it. Seeing an HP-UX prompt on the screen would also create problems for HP's general-purpose sales channel, who were generally unfamiliar with the UNIX® operating system.

The team decided to create a complete turnkey system on top of the operating system. This meant that the system could not be used to run other HP-UX applications or as a general-purpose computer. While the implementation of this policy was not technically difficult, explaining the concept took considerable effort. The team found that since there were many existing models of mixed measurement server systems, the natural conclusion was that the prototype analyzer would also be open for customer development. The level of communication required to explain that the prototype analyzer would be turnkey was much higher than anticipated. In the end, the team found that demonstration was by far the most effective way of communicating the product's structure and its advantages.

An added benefit of offering a turnkey system was that the team did not have to worry about operating system revisions. Maintaining an open system would require the team to put ongoing effort into supporting three operating system revisions: the last, the current, and the future.

The prototype analyzer is built upon only one operating system revision. The entire system will be revised only when customer needs change and substantial new functionality is required. This frees the design team from chasing operating system revision-related product defects.

Meeting Price Goals

Research indicated that while customers wanted a powerful analysis and display environment, their perceptions of price were influenced by the availability of PC and workstation-based data analysis packages that sold in the U.S.$1000 to $5000 range. Customers also viewed the prototype analyzer as a kind of operating system. Operating systems aren't valued as highly as the applications that run on them.

This data implied that the total system price would need to be in the range of U.S.$5000. The design team had two areas where costs could be lowered. The workstation would represent a large portion of the prototype analyzer material cost. Selling and support expenses also contribute to the total system cost.

Concurrent with the prototype analyzer development, HP's Workstation Group was designing the HP 9000 Model 712 low-cost workstation. A review of the workstation's price and performance specifications indicated that it would be an ideal fit as an encapsulated measurement server.

The base system would consist of a 60-MHz CPU, 32M bytes of RAM and a 525-Mbyte hard drive. The system would be shipped from HP's workstation manufacturing operation to the Colorado Springs Division, where a new operating system and the application code would be loaded and tested. The completed product would be shipped to customers from Colorado Springs as a turnkey system. Minimizing the extra handling helped keep direct manufacturing expenses low. As with any new product, initial inventory was difficult to estimate because there was no order history. However, the ability to order and receive semiconfigured systems with fairly short lead times helped maintain a low inventory and thus contributed to a low manufacturing cost.

Cost of sales is defined here as the effort put into customer contact, demonstration, and objection handling during the sales process. Although precise numbers on these efforts are not available, the design team had sufficient practical experience to know what factors contributed to a complex product and a higher cost of sales.

Traditional instruments generally have a low cost of sales because they can be easily explained, demonstrated, and left with the customer for an extended evaluation. The instrument model was used as a goal for prototype analyzer structure, connection, and demonstration.

A learning products engineer was assigned to evaluate barriers to installation and first measurement. Two goals were adopted. The first was that customers should be able to get the system running from a shipping box in less than one hour. This was accomplished by examining in detail the steps a customer would take getting from the shipping box to power-up.

A second goal of getting from turning on the power to a measurement in less than 15 minutes resulted in significant changes to the initial user interface design. The initial interface used the standard window bar to access instrument, analysis, and display tools. After several days of usability testing at a customer site, the learning products engineer mocked up a toolbox on the left side of the workspace using masking tape on the monitor. The toolbox contained all available tools. These tools could be dragged and dropped onto the workspace (see Fig. 3) and would automatically connect themselves.

The efforts put into these goals paid off. Both sales engineers and customers found the product easy to set up and run. The toolbox proved to be a big hit during customer demonstrations and added significantly to the ease of use of the product.

Products with a substantial software content present support problems. The EDA industry generally addresses this issue with software maintenance or support contracts, which provide the customer with software updates and defect fixes over a specified period of time. Software maintenance contracts are generally priced at a percentage of the total system software cost.

The prototype analyzer team wanted to hold to the instrument model. Most instruments do not have software maintenance contracts. Instead, software upgrades and defect fixes are usually distributed through an ad hoc process of sales and system engineers personally distributing flexible disks or occasional mailers to customers who have expressed an interest in future software upgrades.

The prototype analyzer team decided to implement a software notification process. This process would reduce the burden on the HP sales or system engineer of distributing new software and defect fixes. Defects and minor revisions would be distributed free of charge. Customers would receive notification of the availability of major revisions and related new products.

The difficulty with this approach lay in the need for many new processes within HP. These included development of a call-handling center to interface with customers and get their names and shipping addresses for the free updates. Process improvements are ongoing, but customers have indicated their satisfaction with the approach.

Standard Interfaces

Somebody once commented that the great thing about standards is that there are so many to choose from. The prototype analyzer design team faced a bewildering array of networking, data, and tool chain standards. Narrowing the choices down to just a few supported standards would be required to meet the project schedule.

Networking standards were quickly defined as a result of the decision to go with the encapsulated architecture. This choice necessitated the use of the X Window protocol to support local and remote control and FTP/NFS to get data in and out of the

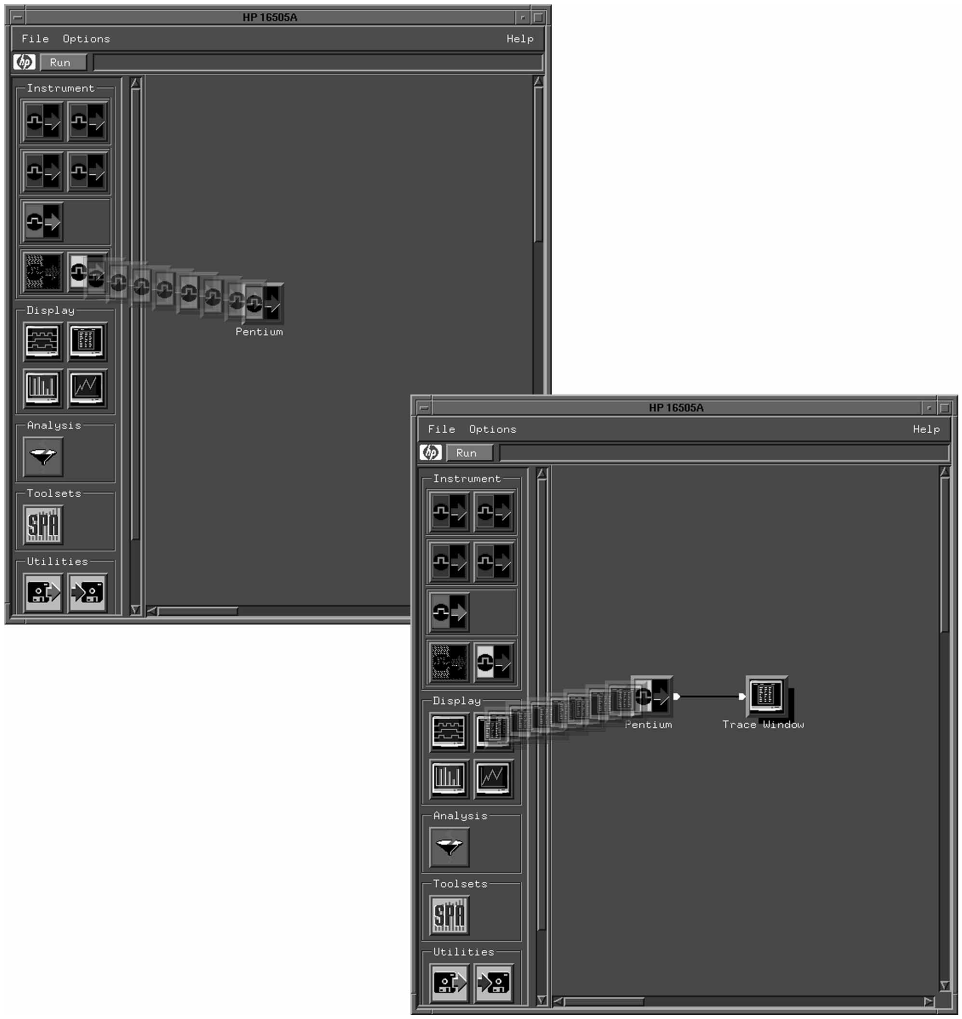

Fig. 3. *Tools can be dragged from the toolbox at the left side of the display and dropped onto the workspace of the HP 16505A prototype analyzer. They connect themselves automatically.*

system. Ethernet was the natural choice for a network connection. Fortunately all of the networking hardware and software was already present in the HP Series 712 workstation and only needed to be augmented with a graphical user interface.

Early prototype analyzer users found that the networking was sufficient but not ideal. In particular, customers asked for the ability to call up remote X Windows applications onto the prototype analyzer's local display. This feature was useful because customers could access a remote application such as a schematic or text editor from the lab bench. This capability was added in a subsequent release of the product.

Interfaces to hardware and software tool chains continue to evolve. The encapsulated measurement server approach enables the design team to maintain control over the type and manner of data exchanged with other design tools. Having control over this process provides additional robustness and stability which is critical to maintaining the low cost of sales discussed above.

The Results

The HP 16505A prototype analyzer has met its price and performance goals. The only true measure of a product's contribution, however, is customer acceptance. A wide range of customers, including all major computer manufacturers, have purchased prototype analyzers to aid in the development of their high-performance digital systems.

Measurement throughput continues to improve as the design team gains knowledge about the cause of performance bottlenecks. The ability to focus on a single platform and the simple software update process afforded by the encapsulated nature of the prototype analyzer make it easy for the design team to develop and distribute new software that contains performance improvements.

Adherence to network standards such as FTP, NFS, and the X Window System has lowered the support burden. However, the diversity of customer networking schemes does create a demand for factory-based network troubleshooting expertise. The design team will continue to apply best networking practices to the system in an effort to reduce the occurrence of network setup problems. Usability testing will be used to gain insight into networking problems.

The ability of the architecture to support additional functionality makes it a cornerstone of HP's real-time measurement solution for digital design teams. The reduced cost of incremental development and the low cost of sales and support are important product attributes that outweigh the higher cost of goods sold compared to host-based architectures.

Conclusions

The development of the HP 16505A prototype analyzer presented unique challenges to the project team. The encapsulated measurement server approach taken by the team meant there were no guideposts to follow. The success of the project was in doubt until the product was introduced.

The decisions described in this article may appear obvious in hindsight. They were not. The design team wrestled with the pros and cons of the measurement server architecture decision throughout the product development cycle. As new data became available, the team eventually rallied around the encapsulated approach. As with any new endeavor, at the time the decisions must be made, there are no right answers, only informed judgments.

The design team learned several lessons during the development process. One clear lesson was to focus on solving customer needs. Issues such as internal architecture and form factor are important, but clearly secondary to the problems the product is trying to solve. An important market research lesson the team learned is to encourage customers to describe their problems, not their thoughts on instrument architecture.

Acknowledgments

The author would like to acknowledge several people for their significant contributions to the HP 16505A prototype analyzer. James Kahkoska distilled hundreds of customer needs into an innovative software architecture and provided the vision, skill, and tenacity needed to bring the product to market. Pat Byrne acted as mentor for the project through various internal checkpoints and helped craft and present the product's benefits to countless customers. The design team of Frank Simon, Mark Schnaible, Tracy Vandeventer, Doug Robison, Jeff Roeca, Richard Stern, John Friedman, Mason Samuels, Dave Malone, Mick Backsen, Bob Beamer, and Pat Desautels added their expertise, insights, and dedication to meet project deadlines. Doug Scott added his unbiased expertise on user task analysis. Steve Shepard and Jeff Haeffele (the "Binars") provided invaluable moral support and coaching. The extended project management team spanning all functional areas within the HP Colorado Springs Division, too numerous to mention here, also merit appreciation and gratitude for their process expertise and enthusiasm in support of this project.

References

1. N. Barnholt, "Future Test Systems Will Use Standard Networking Protocols," *Electronic Design*, January 10, 1994.
2. R. Helsel, *Cutting Your Test Development Time with HP VEE*, Hewlett-Packard Professional Books, Prentice Hall, 1994.
3. J. Burch and K. Moore, *Communication Architectures for Next Generation Measurement Systems*, HP Laboratories Technical Report HPL-92-137, November 1992.

▶ Go To Next Article

▶ Go Table of Contents

▶ Go To HP Journal Home Page