The accelerating pace of technological change coupled with the increasingly widespread use of information technology (IT) has forced organizations to be more concerned than ever with making more effective use of these technology-related resources. Indeed, effective use of IT is now considered a major determinant of competitive advantage, productivity, and even personal competency (Davenport, 1993; Hammer and Champy, 1993; Neumann, 1994). Furthermore, research suggests that improvements in the way IT is used may be the principal source of future economic growth (Jonscher, 1981).

One area in particular that merits increased focus is on the end-user — how useful and easy-to-use new IT applications are. Both academics and practitioners agree that the potential of IT depends on how it is used (Cash, Eccles, Nohria, and Nolan, 1994; Keen, 1991) and, in turn, point out that how IT is used is highly affected by the interface between the user and the technology. From a cycle time perspective, the challenge is two-fold: (1) rapidly developing IT applications with user-friendly (and useful) interfaces that as a result of the “friendliness” can (2) be quickly learned and thus enlisted to provide organizational benefits. However, as good as this sounds, it is by no means an easy task. Organizational pressures to decentralize computing, stay current with technology, and derive any benefits offered by the Internet make the challenge greater. Coupled with the reality that any new systems delivered should seamlessly cooperate with pre-existing “legacy” systems, the challenge often becomes insurmountable for organizations today.

However, there are ways to address these challenges. This article is a success story of one organization’s experience with these challenges, and the development of their “Enterprise User Interface” (EUI) as one method to quickly deploy IT applications today amid the myriad pressures. After a brief introduction of both traditional and new interface architectures and user interface
implementation approaches, we will suggest a unique approach to user interface implementation for new IT platforms and legacy system extensions. We will illustrate this process through a case study of the user interface implementation efforts at Ascension Health System — a large, nation-wide hospital network — and will show how the process represents a unique strategy for effectively meeting an organization’s IT needs. The reader will find that this unique approach is more user-centered and promises to satisfy more requirements in a time-critical environment than more traditional user interface efforts might.

**The Importance of the Enterprise User Interface**

User interfaces are becoming an increasingly large and important component of most systems implementation efforts. Enterprise User Interfaces (EUI) represent what an organization’s end-users see (on their terminal or PC screen for instance) when they access an organization’s IT applications. The design activities related to developing effective EUI primarily focus on providing a useful, yet easy-to-use means for users to interact with their computer-based applications (for example, EUI developers concentrate on creating intuitive and efficient forms of user input and output). However, in addition to these details, the demand to be responsive to competitive initiatives and customer expectations requires that attention also be paid to designing interfaces that can be implemented quickly. This time challenge is amplified by the rising popularity of the Internet, the world-wide web (WWW), and electronic commerce, since EUI design must also incorporate requirements related to the need for easy-to-use Internet or Intranet applications. Finally, the flexibility of current EUI technologies has permitted interfaces to be highly customized on an individual basis. This flexibility has led to a “mass customized” user mindset (i.e., “I want it my way”). This pressure exacerbates the challenge to develop an interface architecture that allows newly developed applications to coexist and interact with legacy applications. Taken together, these demands often make the challenge of defining an effective EUI intractable.

In response to these challenges, many organizations are now defining “interface architectures” to achieve consistent, manageable interface development across their enterprises. The underlying principle is to define a flexible, cost-effective, and common architecture for interface development that results in interfaces that are consistent across platforms, effective, and easy to use. In addition, by adhering to a pre-determined architecture, many systems development and user interface decisions can be predetermined and as a result, cycle times effectively reduced.

**Centralized, Decentralized, and Internet Computing: A Continuing Saga**

Computer architectures have been continually evolving from centralized computing to distributed computing to Internet/Intranet computing (see Table 1).

Early generation computing systems were characterized by a centrally controlled host computer and were known as centralized computing systems. The host computer, usually a mainframe or large minicomputer, contained all the information system components, including data storage, business logic, user interfaces, and system interfaces (Whitten and Bentley, 1998). All processing was done on the host computer, and the locally and remotely attached terminals (or, today, a PC emulating a terminal) were simply host accessing devices.

As technology advanced over time and computer costs decreased, the processing power and
The intelligence of the nodes in a computing network became less centralized and more distributed. In the client/server environment, data storage, software applications, and both user and system interfaces could be distributed across a network of clients and servers that communicated and cooperated to achieve system objectives. This gave rise to the possibility that both data storage and data processing could be moved from the centralized host computer out to decentralized mini-computers or distributed host computers (McNurlin and Sprague, 1998).

With centralized computing, applications utilized an old character-based user interface (CUI) that was awkward and cumbersome in comparison to today’s graphical user interfaces (GUI’s). With PC’s replacing dumb terminals, old CUI’s could be stripped from centralized applications and instead use GUI’s that resided and executed on the PC’s (a.k.a. “fat clients”). There are several advantages with distributed computing. First, it can be implemented quickly since most aspects of old legacy applications remain unchanged. Second, users can get a familiar interface to existing systems (i.e., the GUI’s can be developed to look like the legacy systems). Finally, the useful lifetime of legacy applications can be extended until such a time as resources warrant a wholesale redevelopment of the application. As a result of these advantages, most organizations now tend to utilize distributed system architectures to support legacy systems.

As the technology to secure Internet data traffic evolves, on-line transaction processing capabilities are extended to the Internet. Thus, every computer on the Internet becomes a potential client with access to a world of servers. Consequently, today’s client/server architectures must also incorporate Internet/Intranet technology into the picture. With Internet/Intranet computing, most of the executed applications are first downloaded to the “thin client” from mainframes, servers, or the web, and executed. One key enabler of this “download and execute” model is the Java programming language. Small, modular applications written in Java (called Java applets) are stored on an Internet server and downloaded to the client and executed when the client accesses the application. Because Java applets are theoretically simpler (and smaller) than most client applications, organizations are using client computers called “thin clients” that are designed to run only Internet-based applications, such as web browsers and Java applets. Generally, there are two basic types of thin client computers today: the network computer (NC) and the Windows terminal (WT). These thin clients are microcomputers designed

<table>
<thead>
<tr>
<th>ARCHITECTURE</th>
<th>LOCATION OF DATA AND APPLICATIONS</th>
<th>WHERE APPLICATIONS ARE EXECUTED</th>
<th>EXAMPLE OF “TYPICAL” CLIENT DEVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized computing</td>
<td>Centralized host computer</td>
<td>Centralized host computer</td>
<td>Dumb terminal</td>
</tr>
<tr>
<td>Distributed computing</td>
<td>Distributed mini-computers</td>
<td>Distributed mini-computers</td>
<td>Dumb terminals or PC’s acting like dumb terminals</td>
</tr>
<tr>
<td>Client/server computing</td>
<td>Shared between servers and clients</td>
<td>Cooperatively processed by clients and servers</td>
<td>PC’s with significant storage and processor power (i.e., “fat clients”)</td>
</tr>
<tr>
<td>Internet/intranet computing</td>
<td>Internet and corporate intranets</td>
<td>Client devices (after downloading from web servers)</td>
<td>Diskless PC’s (i.e., Network PC’s or “thin clients”)</td>
</tr>
</tbody>
</table>

Table 1. Comparison of Three Forms of Client/Server Computing
primarily for use with the Internet and corporate intranets by clerical workers, operational employees, and knowledge workers with specialized or limited computing application needs (Moll and Aitken, 1999; Jacobs, 1998). NC’s or WT’s are typically low cost, sealed, networked computers with minimal or no disk storage. The operating system, application software, and data storage are provided by the server. According to DeGroot (1998), one of the main attractions of network computers is their lower cost of purchase, upgrades, maintenance, and support compared to full-featured PC’s. Other benefits to businesses are easier to administer, including ease of software distribution and licensing, computing platform standardization, reduced end-user support requirements, and improved manageability through centralized management and enterprise-wide control of computer network resources (Jacobs, 1997; Jacobs and Picatril, 1997).

Many companies are now concentrating more on network computing than network computers.

Burden (1998) states that in a survey of 150 user companies on their plans for the desktop, Computerworld magazine discovered that regardless of the type of PC user, traditional PC’s will account for the majority of desktop devices, at least for the next three years. Although thin-clients are still theoretically cheaper than full function PC’s, they do not deliver as much value for their price (Greenbaum, 1997; James, 1998; Meyer, 1998; Sibley, 1998). Many experts believe thin-client still has a chance of competing with the PC. Horowitz (1997) reports that Citrix Systems, Inc., has developed WinFrame, a key component of Microsoft Corp.’s windows terminal server strategy that could turn older PC’s into Windows-based terminals. Later in the case study, we will present the successful efforts of one organization to maximize the thin-client advantage of centralized manageability. WinFrame and Java-based computers are included as a part of the enterprise user interface project and help the organization overcome ongoing IT issues including the challenge of the old legacy systems.

**The Challenge of Legacy Systems**

Legacy systems are those computer systems and applications from the 1960’s, 1970’s, and early 1980’s that are based on what is now primarily obsolete hardware and software technology (Wetherbe and Vitalari, 1994). More than 70 percent of the corporate data in the world is still on legacy systems (Horwitt, 1999; Booker, 1997). Before IS executives can answer the significant question of whether to replace or upgrade legacy systems, a careful and thoughtful planning process must be initiated. Typically, this planning process analyzes all essential aspects of the legacy system’s functionality and establishes what needs to be done to solve the fundamental problem of how to maximize efficiency in the future (Bourne, 1997). Wetherbe and Vitalari (1994) discuss three strategies for dealing with legacy systems: “scorched earth,” — a complete replacement of the legacy application; “quarantining,” — keeping legacy systems intact, but completely separated from other application systems; and “cocooning” — wrapping legacy systems with “middleware” that translates between the old legacy system and new application systems. In terms of the cycle time metrics of cost-effective systems delivery, Wetherbe and Vitalari suggest that often cocooning is the more attractive alternative. In this case, part of the “middleware” that “wraps” the legacy system includes the end-user interface.

For system developers and architects, one primary goal is to develop applications that adhere to an application architecture that is flexible enough to handle the diverse needs of the different application interfacing requirements within an enterprise, while minimizing the burden of managing multiple interfaces. One negative
aspect of most legacy systems was that each application interface is developed with a different approach and functionality. In most cases, different people have developed these interfaces at different times with different resources.

With today’s Internet technologies, there appears to be hope on how to deal with the legacy issue: Running legacy applications on mainframe computers combined with web-based interfaces (Colkin, 1998; Mateyaschuk, 1998; Tucker, 1997). This enables users to make use of web browsers to do their own mainframe queries with software that they download from a web server. In the following section we will present the successful story of Ascension Health System, a health care organization that has integrated their legacy, client server, and web resources into a coherent whole using their in-house IS staff. Through the use of a coherent end-user interface architecture, many applications from different sites are merged and the infrastructure put in place to give the users a single view of enterprise-wide information resources, regardless of where the users reside in the United States.

Learning by Example: Ascension Health System

The rise of Internet generation computing coupled with the challenge of legacy systems leads us to the case Ascension Health System (AHS). Established in the U.S. in 1809 by St. Elizabeth Ann Seton as the Daughters of Charity, today’s AHS is a pioneer in the health care industry. AHS is a not-for-profit corporation cosponsored by four Ascension Health Provinces in the United States — Northeast, Southeast, East Central and West Central Provinces. For management and reporting purposes, “ministries” are organized into five geographical divisions. A ministry is a hospital and its associated clinics and physician offices.

In support of their mission and values, as well as its strategic priorities, the information systems arm of AHS, Ascension Health Information Services (AH-IS), aids member organizations in terms of selecting, implementing, operating, and improving the cost effectiveness of their clinical and administrative information technologies. AH-IS operates numerous platforms from virtually every major vendor in the health care industry and is supported by 175 staff members. The organization provides extensive consulting support to local health ministries (LHM’s) as they develop information technologies in response to the changing health care environment. AH-IS provides support for the complete continuum of physician offices, acute care, home care, and Integrated Delivery Networks (IDN’s).

Challenges Facing AHS

In response to healthcare industry pressures to improve service quality and reduce costs, in 1997 the organization decided to refurbish their old legacy system portfolio with the idea of implementing all enterprise interfaces through a common architecture. They analyzed their legacy systems and determined that there were no major application problems and that they remained quite maintainable (thus making cocooning a viable strategy). In addition to improved functionality, legacy applications were to be enhanced to take advantage of Internet technology. This included the use of web browser software and Java-enabled software applications (applets) to permit improved system access, allow new uses of system-generated output, and to provide AHS users with a means to deal more comprehensively and consistently with corporate data and with each other.

While the strategic benefits of IT were becoming increasingly clear, the proliferation of local computing (i.e., ad-hoc decentralization) had seriously hampered the ability to share information across the organization. Closing the books at the end of a quarter became more and more difficult, and the ability to compare operations from one unit to another became
Ascension Health System: An Enterprise User Interface Approach

almost impossible. Prior to 1997, AH-IS had been facing the challenges of distributed stand-alone systems that did not really form an enterprise-wise distributed system. These computing “islands” had been interconnected to allow a little data to flow, especially upward to the corporate host. This resulted in the following three major issues:

Accessibility. Virtually all LHM’s required access to multiple, traditionally incompatible systems. This often resulted in the need for multiple terminals or workstations at a given location in order to access the disparate systems. A second related issue was usability — that users were challenged with having to know how to navigate through each system to retrieve their data. Additionally, there had been little or no off-campus access (e.g., from a doctor’s office or home to the hospital).

Emulation. To date, terminal emulation has been a high-cost component of every user’s desktop workstation maintenance. Across AHS, there are many different emulation packages in use, each requiring a person to visit each workstation to install the software. With hundreds, or in some cases thousands of workstations, this had become a significant cost, time, and human resource issue. Upgrades are continually released in order to remain compatible with operating systems and other installed software. This in turn requires the installer to spend even more time diagnosing and resolving the incompatibility.

PC Overkill. Whether a dumb-terminal, thin or fat client, most organizations today use conventional PC’s as the standard desktop workstation. With that comes the continual upgrading/converting of these workstations to keep pace with technology and software requirements as well as the attendant obsolescence problem. At AH-IS, the typical life of a PC was about 12 months before an upgrade or change-out occurred. As is typical with most organizations of knowledge workers, AH-IS has found that a large majority of users typically need to do three or four functions with a workstation (e.g., terminal emulation to a system or systems, e-mail, problem ticket reporting, and/or word processing). With Internet generation technology, most of these functions can be performed via a web browser. So the question is, why do we have all of the complexities of a PC on a user workstation when only a web browser is required? AH-IS is not the first to realize that often a PC platform provides more features than necessary (e.g., Norman’s “creeping featurism,” 1988).

The three issues outlined above involving systems incompatibility, application maintenance, and creeping featurism had discouraged AHS end-users. Their perception was that the usefulness and ease-of-use from the existing systems was low. According to both AHS’s CIO and Chief Technologist, the majority of end-users did not believe that technology could help them perform their job better. In some cases, they had to invest significant effort to learn and operate the systems due to unfriendly user interfaces. In the following section we investigate how AHS chose to meet these challenges.

Meeting the Challenges at AHS: A Four-Pronged Approach

At AHS, struggling to make sense of the rapidly changing business environment, management embraced new approaches to distributed information processing and management available through client/server computing. The objective of the new approach is to provide people more control over their work by linking mainframe and personal computer technologies to preserve the benefits of both. The underlying principle of the AH-IS approach to solutions is to adopt an effective enterprise user interface architecture that is flexible, robust, and cost effective. Although several companies have benefited from adopting the most leading edge solutions for self-describing interchange formats, assured data delivery, queue-driven data replication, and change management (Cash, et al. 1994; Neumann, 1994), AH-IS adopted a more productive principle that demonstrated a relatively basic approach that relies on a sound
end-user interface architecture. The architecture was, in turn, operationalized through four approaches: synchronous data updates, common access, terminal emulation, and thin client computing.

Architectural Component #1: Data Synchronization. Many enterprise-wide interfaces require the translation of data from more than one source to more than one target system. For Ascension Health, AH-IS faced a dynamic environment that was dependent upon such variables as the enterprise’s changing customer base, a changing regulatory environment, and a volatile insurance environment. The choice of data source or data target is often driven by values within the data itself. With a technique referred to as synchronous index replication, each time data is changed; all indexes for one or more sources to one or more targets is also updated. Synchronous, real-time index replication is required by physicians that need high availability and concurrency of their patients’ essential information, such as lab results, pharmacy information, and radiology reports.

Architectural Component #2: Common Access. Through Common Access, AH-IS normalizes many different systems into a common web browser view. Consequently, a consistent end-user interface is provided to all AHS users regardless of the system they are accessing. In addition, inexperienced users can access many different systems with very little knowledge of the underlying legacy systems. Common Access involves three main architectural components:

1. A generic web server. Typically, a Microsoft Windows NT Server with an IIS web server or any Unix web server can be used.

2. CNX (connection component). The CNX function maintains hot or persistent connections to the systems being accessed within Common Access. CNX communicates with all of the involved ancillaries and with the web server.

3. An ODBC relational database. The relational database is where temporary data is kept for the duration of the patient encounter. The database contains pointers to relevant patient information that is needed to access the patient data in other systems such as, lab, pharmacy, radiology, etc.

Common Access provides a common view to any number of disparate systems which cross over function, hardware platform, connectivity, and geographic area. There is significantly less training involved for the infrequent or occasional user. One display device is needed as compared to a display device for each system. Access or... once Common Access is installed, incremental new users ... are voluntarily requesting that they be added.

To date, ten local ministries are deploying Common Access. The East alumni ministries will have the capability to install Common Access when they connect to the wide area network. The number of users depends on the site. Typically there are approximately 50-200 users per site. One interesting observation is that once Common Access is installed, incremental new users (i.e., those that never used the system previously) from across medical disciplines are voluntarily requesting that they be added.
Architectural Component #3: Terminal Emulation. Should users prefer to have an end-user interface that resembles the more traditional legacy interface screens, they can use terminal emulation. Terminal emulation has been accomplished at AHS utilizing a web browser. Java applets are loaded via a web server into the user’s browser on the workstation. The workstation can be a conventional PC, thin-client, or any workstation that has a browser. This environment provides the ability for all locations to be on the same terminal emulation application across the AHS system. To date, all requirements have been met as requested by the LHM’s. The emulation modes provided are 3270, 5250, and VT100. The overall costs are about 20% of conventional emulation packages today.

This technology extends the life of the old legacy systems while greatly reducing the effort and cost of providing maintenance on each workstation. From a single workstation, a user can log on to any system within their operational domain through a Windows-based “point-and-click” interface.

Terminal Emulation applets provide a common approach to terminal emulation across most of Ascension Health today. The support is the same throughout AHS and training is minimal for the end-user. Furthermore, there is now a common view whether on or off the Ascension Health campus. The only requirement for the end-user workstation is a browser which today costs about $30 per workstation.

Approximately two-thirds of AHS’s local ministries are currently using Terminal Emulation. It will be available to the remaining locations as they connect to the AHS wide area network.

Architectural Component #4: Thin-Client Computers. To contain the costs associated with maintaining current technology PC’s, AHS has chosen to adopt a thin-client solution. Thin clients are an alternative to the conventional PC-based workstation. The thin-client is a hardware platform that provides high functionality while providing minimal support services. While this technology saves on the initial cost of a workstation, the real savings come from the amount of backend support and maintenance needed. AH-IS has found the majority of LHM users have three areas of need: terminal emulation, E-mail, and problem ticket reporting. These functions can be performed within a web browser. By providing a thin-client with a resident web browser, users have a stand alone workstation without the need for and expense associated with backend WinFrame/Citrix servers and their respective infrastructure support requirements.

Thin-clients make it less likely that AH-IS has to continually change or upgrade hardware to run applications. There is virtually no software to license since only a browser is used. End-user training is minimal. At Ascension Health, thin-clients have become a fixed function productivity tool and are centrally administered and updated. The benefits of thin-clients are very little maintenance at end-users’ desktops. All of the configuration is done at a centralized server. This reduces the staff resources required to support a number of workstations. Thirteen local ministries are currently installing the AH-IS thin-client technology.

Conclusions

All organizations today are facing imperatives requiring that they make cost-effective and timely use of information technology to deliver organizational benefits. Prerequisites include quickly delivering systems that are easily accessible and easy to navigate by today’s end-user population. Additionally, the need to eliminate disparate application “islands” has never been greater as organizations find they need
the “whole picture” in an informational sense. Failure to deliver on these requirements often leads to end-user departments striking out on their own to find technology solutions – often at the expense of their own departmental performance, or worse, the performance of the organization at large.

Given the large investment in legacy systems and the need for reduced cycle times in systems delivery, working with the installed base of these pre-existing applications has significant merit and can provide significant benefits. On the negative side, working with these systems heightens the challenge of delivering information to users any time, place, or way they may want it. It is here that we believe the Enterprise User Interface architecture may provide direction to organizations.

Implementing an effective architecture for EUI is not trivial, however, it can yield great benefits for an enterprise that adopts one. With Ascension Health System serving as a case study, we have introduced an organization that has demonstrated how the EUI architecture can be used to deliver results to the organization. By offering familiar yet consistent interfaces that span pre-existing systems, AH-IS is finding that their AHS user base is voluntarily adopting and using their systems to benefit not only themselves, but also the ultimate customers...AHS patients. Other observations from the study illustrated that by building a successful architecture, system interoperability and knowledge transfer can be achieved without imposing excessive costs, training, or maintenance.

The presentation of this case study should be interpreted with some degree of caution. If the solutions used by AH-IS prove to be reproducible in other cases, this finding holds significant implications for both IT researchers, who prefer to study the adoption and diffusion of innovations, and IT practitioners, who want to successfully implement enterprise-wide systems.

References


Ascension Health System: An Enterprise User Interface Approach


Brian D. Janz, Ph.D., is the Director of the Institute for Managing Emerging Technology and an Associate Professor of Management Information Systems at The University of Memphis. His research focuses on how information technologies affect organizational strategy, design, and knowledge worker behavior.

Sutee Sujitparapitaya, Ph.D., is a director of Institutional Research and Decision Support at Kent State University. In 1996-2000, he was a director of Systems Administration at The
University of Memphis. Dr. Sujitparapitaya's research interests focus on the impact of information technology in facilitating a variety of organizational behaviors and on organizational efforts involved with planning, managing, and diffusing information technology.

James C. Wetherbe, Ph.D., is a professor of Information Systems & Quantitative Sciences at Texas Tech University and a professor at the Carlson School of Management at the University of Minnesota. Quoted often in leading business and information system journals, Dr. Wetherbe is internationally known as a leading authority on the use of computers and information systems to improve organizational performance and competitiveness. His specialties include cycle time reduction, e-business, business reengineering, management and computer based information systems, systems analysis and design, and interpersonal skills.

David Sammet is the Director of Research and Development at Ascension Health Information Services.