Paper # 5110

Living With Antique System Architecture - A Transitory Case Study

by James A Hepler, Senior Technical Consultant
Hewlett-Packard
Novi, Michigan, 48375-8024 USA
248-380-2207 - Fax 248-449-5645
jim_hepler@hp.com

THE CHALLENGE

The Millennium is approaching! Many businesses are realizing that their obsolete, cumbersome hardware and software architectures make it difficult to address the now well publicized Year 2000 issues. Even worse, many applications do not currently meet the changing requirements of the user community. These applications are typically inflexible, require a large amount of maintenance, and lack user friendliness. New features are difficult to add because of the antiquated file systems supporting the data and the clumsy user interfaces built in. Often, the applications have hard-coded references to non-standard hardware or networking environments. It is also common to be missing documentation or even source code on some of the more unwieldy programs. Maintenance costs keep increasing on older hardware and the risk of hardware failure increases as time passes. All of these factors contribute to reducing availability to the systems for the users.

These problems have been intensified by exposure due to the year 2000 problem. Many IT shops are striving to avoid problems expected as the millennia approaches. They are discovering how antiquated their existing systems have become as they examine them for potential problems in their data and logic concerning expiration dates and other time period calculations. Since funding is often available to address Year 2000 (YR2K) issues, many shops are leveraging this fact into a movement towards more current technology in hardware, software, and data structures. At the same time improvements in data accessibility, availability of information, and security will result. Management of the environment will be easier and allow more control and better support of the users and the business requirements of the organization. The business justification is much easier as a result.

But how can all of these things be accomplished without shutting down the business for lengthy periods of time, risking loss of data, spending several years of budget allocation, upsetting employees and customers, and generally causing havoc in the IT department? IT shops with Antique System Architecture must be determined to do this, or eventually go the way of the dinosaur when things fail around them.

ANALYZING THE CURRENT IT SITUATION

The first step in creating a viable transition plan for moving from the antique architecture is to carefully and completely document the current technology, how it is used, where the inherent problems are, what the risks in not transitioning are, and how the infrastructure and systems are currently being managed. Any IS or IT plans or strategies in place need to be discovered and analyzed for use in this and later steps. These plans often do not exist or are incomplete, so any short and long range statements, strategies or plans relating to the corporation or organization will also be needed. If a documented Enterprise Architecture exists, use it! If not, many assumptions must be made or a working conceptual model of an architecture must be created. This will make the process more risky and time consuming. Interviews with management are needed to verify any of that information and to update with current events and trends relating to the business plans or strategy.

A corporation's competitive situation should also be considered to get an overall understanding of the business climate that is driving their IT decisions. The business plans need to be understood in order to evaluate the alignment of IT and IS strategies with the business. If they are not currently tightly aligned or the existing strategies are not designed to move IS/IT into alignment, it is important to consider a target architecture and transition plan to move towards alignment in the future. This may help with management commitment and funding of the project. It will also help address employee morale issues since there will be considerable career enhancement for technical people whose skills and experience are as ancient as the architecture. Those personnel who do not see moving towards current technology as a career benefit can be re-deployed after the transition as necessary.

There are many details to be considered and issues to be dealt with in planning these types of transitions. Volumes have been written listing topics and areas to cover. There are many tools, forms, approaches, and methodologies available to organize these efforts. It is well beyond the scope of this paper to cover these in depth, but instead, a high level view is offered.

Current Technology

Inventories of system hardware and software should be gathered. CPU/SPU models and configurations are needed. Versions of operating systems may be useful in certain cases to help determine which transition approaches are feasible. In a few cases, it is possible to emulate existing environments as a way of transitioning to more technically current hardware/software environment more quickly to reduce the costs or risks associated with existing hardware/software. If installed versions are not up to date, it may be an indication that the IT staff is overwhelmed with current responsibilities and could be a risk factor for the IT department. Many other system level items should be collected including the following where appropriate:

- System Utilities
- Sorting Utilities
- Transaction Monitoring services if system level
- System Programming services and languages
- Middleware in use
- Backup Peripherals - Media types, etc.
- Network Interfaces and speeds
- Unique Peripherals
    - WORM or Optical Storage or other NearLine Storage Devices
    - Microfiche/film
    - CD Towers

Non-Standard printing devices
Real Time device access
System level security features/devices
Customized data communications interfaces
Other non-standard hardware devices

Configurations of DASD/disk space are critical to sizing target hardware. Total space installed and planned for installation should be documented. Individual files and data bases must be inventoried. Types of data and structural elements and entity relationship definitions are needed. Any documented data models should be available for the analysts. Data Base Schemas and/or Data Dictionaries will need to be reviewed. Access methods in use for all files and file systems must be determined. An often overlooked item is the types of data used. This is critical because not every data type transfers from one file system to another. It is also true that some languages or versions of compilers do not support all data types. This means that conversion utilities or programs would need to be acquired or written. All of this information must be related to the applications that access the data -- cross-referenced if possible. Some or all of the following detailed items would be desirable:

Number of relational table rows
Number and size of relational tables
Number and size of data sets
Number, Size, and type of indexed files
Number and size of raw or flat files
Record lengths
Item type and data type/length
Access types
Metadata usage
EBCDIC, BCD, ASCII or other representation

Languages and compiler or interpreter versions for all applications will be needed. Total amount of code will help to determine the extent of the effort to migrate, convert, or re-create the applications. For languages such as COBOL, lines of code would be appropriate -- for fourth generation languages, number of commands, structures, or constructs will suffice. Other information of value concerning the application coding include:

Library or Copylib in use
Lower and Upper CASE tools used
Testing/Sandbox environment
Production testing and roll-out procedures
Test Data Environment

For interactive applications, the type of user interface is important. Are they using a dumb terminal or emulator, or Graphical User Interface (GUI) such as Motif or MS-Windows? Information on the user desk top devices is also required. If a dumb terminal is currently being used (IBM 3270 or other character mode device), a Unix-based work station, or a Personal Computer with software to control the interface, the analyst must consider that as part of the transition strategy. Desktop information such as the following will be useful:

Hardware type - dumb terminal versus Intelligent desktop

Chip sets - 386, 486, Pentium, etc.
User Interface types and versions - VPLUS, Win 3.1, Win 95, Win NT, Motif, etc.

Batch operations must also be transitioned. Batch applications use some type of job control. The following batch job control items may be useful:

Job Control Language - JES II/III, MPE, korn shells, etc.
Job Control/Scheduling Software
Batch Operational Procedures
Job Recovery Procedures
Messaging and Mail-boxing facilities used

Both internal and external networking software and hardware must be identified. For a successful transition, these linkages must be replaced or maintained. Types of data communications devices and standards used are required. It is of course much easier if these interfaces are standard's based, but if they are not, they must be addressed. Examples of information to be collected are:

SNA types of communications
Local Area Network types - Token Ring, Ethernet based, etc.
Leased lines used
Dial-up access methods
ATM
Public Data Networks used
Private Intranet installed
Internet access allowed
Proprietary Network systems
Customized data communications software
Private switching devices

Utilization

Information on the user base is needed to create a viable transition plan that meets the requirements of the end users and the business units. For guidance, the five W's should be used.

Who uses which applications?
Where from and how do they access the data?
What do they do with the information?
When do they access the system?
Why do they need to access data?

The user population must be identified and quantified. Where are they located? How many users access each application, data base, or file? What type of user interface do they employ? If an application is only used by a few users at a single location, the transition strategy will be much simpler to implement than if ten thousand users scattered around North America access the data through a private network connection.

Another important factor is the response time they are now experiencing and the load a user puts on the network. It may be that the re-architected or transitioned application may make a larger network demand, or create a larger demand on the system - CPU, Disk, Memory, or some other resource. In summary, some or all of the following resource utilization information is needed:

Number of users by application
Transaction Volumes by application or function
Current response times
Current Throughput Measurements
Current print volumes
CPU Utilization levels
Disk I/O rates
Memory Utilization
Table access rates
Media request volumes

The schedule of interactive access and batch operations must still be met after and during the transition. If there is a 9 hour batch requirement nightly now, the new architecture must support that or the application design or requirements must be modified. If the backup techniques must be modified because of new technology such as data base type or schedule of availability, a new backup strategy must be created and perhaps new media or backup devices selected.

Relationships between applications must be determined. If two applications are linked by a daily batch file update followed by a data base load, the transition is much easier than if a real time update is performed across a network using customized code, utility, or middleware. It is fairly common in older architectures for a daily batch job from one system or platform to transmit a file to a second system and then a local batch job is executed to load the transmitted file into the local data base. Newer architectures are moving towards more immediate update methods - sometimes near real time.

Sharing of data files or data bases between two applications on the same system can also be a problem when the transition strategy requires one application to move first. The linkages must be maintained in a suitable way similar to the way they exist in the current architecture. For example, if a data set is shared and

updated interactively, that must still take place at all times during and after the transition. This could be a significant challenge. The alternative is to transition linked applications simultaneously which may be extremely difficult or complex.

Linkages to systems outside of the control of the IT department must be maintained or recreated. Examples of these are:

Another division or department
Customer
Supplier
Financial Institution
Government Agency
Data Base Provider
Public Internet Access

How the information is used is important. For example, if the users access data to download it to their PC and then massage it with local tools, that feature must be allowed in the new architecture. If the user has a client application where local information is collected, and then updates are uploaded after a connection is made, the transitioned application must allow that type of feature.

The criticality of the user access is also an important element in the solution design. The impact of the application or data being unavailable must be considered in the transition plan. The amount of acceptable unscheduled downtime must be determined and employed in designing the target architecture. If the business cannot function without the application, what is the cost of downtime? If the cost of downtime is high enough, redundant hardware or high availability features may need to be included in the target architecture.

Critical Problems/Risks

The trend in the industry has always been that as hardware and software systems age, the cost of support both from the vendor and from internal application support increases. It is often the case that the increased cost reflects the availability of replacement parts and trained service personnel. Internal maintenance costs go up for similar reasons. As normal attrition occurs, it may be difficult to hire or train new personnel on old technology. New hires typically do not want to learn old systems. Personnel who already know the old systems tend to be more expensive to hire because they are more experienced and less available.

New development tools allow faster maintenance than traditional tools do. This relates to internal maintenance costs also. It also means that new development takes longer and is thus more expensive. These facts mean that maintenance becomes less and less effective as time goes by in comparison to the industry trends and often cause the business to fall behind competition that is using new technology architectures. This means that old systems increase the business risk.

From a hardware point of view, it happens that parts are not available or if they are, the delay to get them is longer than it used to be because the vendor no longer exists, or has reduced support of the older technology platforms. Even vendors find it

increasingly difficult to maintain their own antique hardware and software architecture. Clearly if the hardware fails and does not come back up quickly, there is a serious business risk.

IT Management

An IT Management profile is needed. If one does not exist, analysis must be done, and results documented. There are many ways of doing this data collection and analysis, and many ways of subdividing the IT Management activities. These methods are discussed in detail in many industry documents. Whichever organizational approach is used, it is critical that the new environment is managed as well as the existing one. The processes must be transitioned. The staffing levels and roles and responsibilities defined. Personnel may need to be re-trained, hired, or re-deployed depending on the situation. A typical list of areas of IT Management categories might contain:

Operations Planning/Management
Problem Management
User Support Services
Capacity Management
Change Management

Output Management
Media Management
Backup/Archiving Management
Disaster Recovery/Contingency Planning
Performance Management
Configuration Management
Service Level Management
Availability Management
Training and Skills Assessment
Risk Management
Resource Management
Security Access & Control
Event Management
Console Control
Storage Management
Workload Management
Data Base Management
Network Management
System Administration
IT Planning and Budgeting

This list is not all inclusive, and may not be the same as your IT shop views itself, but these functions generally are accepted in the industry.

DETERMINING THE TARGET ARCHITECTURE

The most important factors to consider in determining the target architecture are the business needs of the organization and standard's based industry technology trends. The business needs of the end users in their job functions determine the type of features needed. Do they need more flexible access to the data? Better security? Do the current applications meet the business needs as they exist? Would installing an organization wide intranet improve

the accessibility? Does the competition already have an advantage in ability to respond to changing market conditions? Is development and maintenance of existing applications too slow or cumbersome?

These are the types of questions that must be asked to determine the strategic direction to take with the technology architecture of the organization. Both short term and long term strategies should be considered. When transitioning from an antique architecture it is important to avoid the same trap of inflexible obsolescence in the future. The best way to accomplish this risky task is to move towards standards based architecture as much as possible. This will allow the organization to adopt emerging technology standards in the future and quickly move in whatever direction is needed to meet competitive or other business demands.

The trends today are wide reaching and generally are towards more agile architectures. This means towards standards such as SQL based relational data bases, standard networks (Frame Relay, ATM, SNA or TCP/IP and others), open environments to enable easier access to data, and graphical user interfaces which allow point and shoot types of access and application development. Client-server distributed architectures tend to be part of the current trend although there are many implementations and definitions of what client-server means.

Most businesses that are transitioning their IT implementations today are moving towards more open client-server architectures. Any vestiges of the antique architectures that must remain after transitioning are planned for removal as soon as a method of replacing them can be devised. A viable desk top strategy is critical to the future success of most organizations. One advantage to moving towards the client-server model is that if the client desktop changes in the future, and if it is standards based, there is a good chance that the application will continue to function with minor software updates or changes. If the server technology evolves, the server applications will also evolve to take advantage of new technologies as they become available.

Balancing all of this movement to client-server open environments is the need to control budgets and not endanger data. It may be necessary to create a long term vision of architecture transitioning part way in that direction, but to plan the rest of the transition in a future budget cycle. There are elements of this in the case study to follow.

The target architecture should be based on movement towards the vision of the future that the organization needs to have. For most organizations today this means standards based client-server distributed architecture with standards based infrastructure supporting it.

CREATING TRANSITION PLAN

The transition plan must describe the essence of how to move from the current situation to the target architecture determined. It must meet the business objectives of the organization. It must support any IS/IT Strategy statements that exist. The transition plan must be based on the Enterprise principles for IT that form the Enterprise Architecture. The components to transition include:

People
Processes
Technology

The people part of the transition plan includes training the end users on the new technology components they will be using. Some of these may be:

New Desk Top hardware/software
New networking interfaces
New or re-engineered applications
New utilities for accessing data
Improved IT support
New procedures

IT staff will have significant opportunities for training. Any new IT Management tools and procedures supporting the infrastructure must be learned including all of the IT Management components described above. Because of the significant changes to the IT infrastructure, new strategies for IT Management will need to be created and employed. This must be completed or substantially underway before the user transition begins.

Help desk personnel will also have to learn any new application processes and features that the end users will learn. The training department will have to schedule and roll out training to end users in conjunction with the roll out and parallel testing of the technology components.

Risk factors will need to be considered and contingency plans created. For example, if a new application goes on line, but a problem is discovered, how does the transition backwards proceed? If ordered hardware or networking components are delayed, what impact will that have on the rest of the transition plan.

It will be necessary for two types of processes to co-exist for some time -- the old and the new ones being transitioned to. This will require extra work by the support and transition teams. The data may have to be temporarily bridged if the some users will be on the old system while others employ the new applications. This will depend on the application architecture. For example, suppose the old application accesses an existing relational data base and the new application supports a relational data base on new hardware platform with new client software. Any transactions entered on either data base must be entered in parallel on the partner data base, or some procedure created that re-synchronizes the two data bases when required. This may sometimes be facilitated with Middleware or specially written code. Of course if all users were transitioned at the same moment, this would not require the same co-ordination. In some cases, the users may function independent of each other, so data co-ordination could be accomplished with a delay rather than a real time type of update.

Operations will also have similar problems to deal with. In addition to maintaining multiple platforms, they will be learning or creating new procedures and tools. Many transition plans include transitional out-sourcing or temporary operations staff to assist with the added burden of supporting multiple environments.

If the basic network must change, it may be necessary to install multiple networks for some length of time with additional networking costs and perhaps multiple desk tops for the users. It is common to update the desk top first and add whatever emulation software may be needed so that the same desktop can be used with both the new and old network -- but sometimes this cannot be done so two desk tops are needed during the transition.

User support and system administrators must create new users and security systems to allow users to access the new systems. There will probably be much extra duty to monitor the new user accesses and adjust for oversights and problems that may arise.

The technology components may seem like the simplest, most straight-forward part to plan and schedule. However, unexpected delays can stretch the plan and resources and thus the costs of the transition. Hardware and software and perhaps services must be ordered, received and installed. Normal site preparation considerations must be anticipated such as environmental controls (electrical and air conditioning). There may not be enough space for both the new and old environments, so temporary space may be needed, and perhaps temporary network connections. Until the transition is complete and acceptance testing is done, both sites may be needed and supported. This equates to added cost and scheduling problems. Then a simpler system and network move to the final location must be scheduled, planned, and completed.

The transition plan includes all of the above and many, many other details too numerous to mention in the scope of this paper. The most important thing is to consider all possibilities and employ a full time project manager for this transition.

A CASE STUDY

This case study is high level with the most interesting and important factors mentioned. It is based on a real IT department.

The business situation at the start of the transition project was that following a series of mergers, there were disjointed applications with little or no data sharing. There were four major hardware platforms with little or no communication between them. The applications for approximately half the users were not entirely functional and they had adapted through the use of manual operations and user designed PC applications to overcome the limitations of the existing applications and systems. Operations for three of the systems were out-sourced to a third party. The one system entirely in house was a data warehouse application fed from an IBM mainframe and other data sources with an outbound link to an internet web site. This data warehouse was housed on an HP9000 server.

Two of the platforms were considered risky from several points of view. The Unisys mainframe system was aging, with poorly designed and old application architecture. It had dumb terminal interfaces with little or no flexibility in accessing the data. There was some proprietary networking involved with the interactive terminals. Application support was basic at best with the third party outsourcer

providing operations, help desk, and maintenance. Enhancements were unheard of and when they had been introduced in the past had created new bugs that the users had to work around. The users were very frustrated with this low level of support.

An other system was a true antique from a now defunct hardware vendor. As a result, the hardware was an extreme risk. Parts and maintenance were not always available. There had been as much as two weeks of downtime for repair. The applications are fairly trivial with a small number of users and less than 100 Megabytes of data on the entire system. The applications only did a small percentage of the job the users needed, and they had created their own application extensions on their PC's using a variety of PC tools. None of these applications were supported by the IT department and when personnel transferred or left the organization, there was no transition to the new users except by training of their peers.

The IBM Main Frame was stable and had several extremely viable applications that were popular with the majority of the users and were based on DB2. This data fed the data warehouse on the HP9000 systems. Much of the operations responsibility was out-sourced to a facility management organization, but application development and maintenance was in house. In general, this system was not considered a problem and was part of a medium term strategy.

The long range plan was to eventually be a 100% client-server environment in order to be more agile to adapt to changing user requirements. The short range plan was to move to two platforms and towards the long range plan by transitioning from the Unisys and Defunct Vendor systems to target platforms to be determined. To be successful, an integrated data model was needed.

The IT department was handling user support for all systems including help desk, but was understaffed in those areas. Parts of the network support were also outsourced but managed in house. Network problems were detected by the network provider however and managed by them. The data warehouse application was entirely managed in house, but was not widely deployed and used mainly to reduce the cost of generating reports on the IBM main frame since charges were based partially on utilization. The two platforms of choice were the IBM main frame and the HP9000 server.

The first step in the process of creating a transition strategy and plan was to identify all the technology components in great detail as described briefly earlier in this document. Management was then interviewed to determine business drivers and situations as well as plans for growth that should be considered in selecting the best approach. Critical Success Factors for all business entities affected were documented.

A cross section of key users were then interviewed and questionnaires collected from other users to determine the utilization and suitability of a strategy to migrate applications and data. Shortcomings of existing systems and applications were identified. In general the applications on the Unisys met about 50% of the user needs and on the Defunct system only about 25% of their needs. The remainder of the user job functions were performed through creative

work-arounds involving PC tools and manual effort. The Unisys applications used COBOL and some proprietary reporting tools. The Defunct applications used exclusively proprietary languages that were not in any way transportable. From these interviews, a functional matrix of the business processes that the organization and departments had to do to be successful was created. The concept was to look for commonality of functionality as a way of determining how to complete the business and IT mergers and end up with an integrated data model. This business function matrix was a key element in our transition plan.

The IT department management and staff were then interviewed, and all documentation on IT Management and Operations was acquired. This was a new shop with very few processes or documents on IT Strategy or Architecture in place. There was no Enterprise Architecture. They were grossly understaffed to take on their long range business plan to reduce outsourcing and take over support of all systems within five years.

A cross functional team was formed to create the IS strategy. The team consisted of IT personnel, key end user, and business analysts and first line managers. Executive sponsors were found for the team. Through a series of facilitated seminars and meetings and sub-team activities, a working model for an IS strategy and some components of an Enterprise Architecture was agreed to so that the transition plan could begin. This process took approximately six weeks. This was not a finalized or complete Enterprise Architecture or IS Strategy, but enough agreement was reached to continue.

There was not a data model of any type, although there was some documentation including schemas, and listings of data files and structures on the Unisys and Defunct systems. There was also some Data Dictionary information available. With the business function matrix and all data documentation in hand, another sub-team was formed to create a target data model for the entire organization to reflect the desired integrated data design specified in the long range strategy.

While the long range architecture was to be client-server, the short range was to include the IBM and HP platforms. This was partially because of the budgeting restrictions and the need to get return on investment on installed hardware and software development. The key issue became how to transition from the Unisys and Defunct platforms and which data and applications to which platform. The following steps were determined to be needed and a transition plan created and initiated:

All users were converted to PC desk tops using Windows 3.1 and a 486 or later chip set

A Desk Top strategy was created and a software distribution and licensing control facility and process was established.

All user networking was altered and/or installed to support connections to either the IBM or HP systems

IT Management processes, tools, and infrastructure will be created over a 12 month time frame. Key IT processes such as backup and security are already in place. A review of the suitability of those processes required some updating to handle the increase in interactive activity. This will require adding approximately 12 people to the staff and training them over the next year.

Application functionality which is available on the IBM mainframe was adapted to meet the needs of those users currently on the Unisys and Defunct system that perform functions supported by the IBM applications. This effectively removed about 60% of the usage from the Unisys system and 25% from the Defunct system.

Remaining Unisys data was migrated to the HP systems and stored in a combination of SQL compliant data bases and flat files depending on the data types and access methods.

Remaining Unisys applications in COBOL were converted to COBOL on the HP system. The user interface was converted to a windows based tool. This allows the users slightly more flexibility of data access.

Other Unisys applications were rewritten in client based SQL tools selected to access the relational structures on the HP system.

The Defunct applications remaining were only used by a small number of users. The data was transferred to an SQL data base on the HP system and new client based applications have been created to replace the functions performed on the Defunct platform.

All end users had to be trained on the applications that were new to them. Many had to be trained on Windows usage and PCs in general. Users also required training on the SQL tools to access the relational data bases on the HP system.

Not all of these things happened simultaneously, but many of them took place in parallel. The Defunct system was attacked first since it had the largest risk of failure. The long term strategy of having an integrated data model was not realized by this project, but a large step in that direction was made by reducing the platforms to two, and consolidating some data and moving it to a relational model. This should facilitate the completion of the strategy in the future. The applications did not share data to begin with, so they could be moved one at a time and as long as the users could access the systems from their desk tops, they were only slightly inconvenienced by multiple logons to the various platforms. Over a period of two months, all Defunct system data and applications were moved and recreated on the HP system.

The remaining Unisys applications were recompiled in COBOL on the HP systems and data was converted as required. This process took about six months including the parallel testing. The user interface was changed, but the application logic remained the same. This was not a trivial exercise, as the two COBOL's were not 100% compatible. and much coding effort was involved. Mapping of the data structures and an SQL data base was designed. Load utilities had to be written and tested. There were many minor glitches in the process, but the tasks stayed on schedule and were successful.

The application functionality that existed on the IBM main frame which was similar to certain major functions on the Unisys and Defunct platforms had to be modified slightly for unique requirements of those users that were slightly different than the existing user base. The data in the files on the Unisys and Defunct systems had to be mapped into the DB2 data base on the IBM system. There were a few problems with differences in data type and field length, but these were overcome. A temporary DB2 data base was created for testing of the data migration process so that the production data base was not touched until testing was complete.

This brief summary is clearly enormously condensed. There were literally thousands of steps to schedule and accomplish and the project manager assigned and all transition team members were

challenged to say the least. Contractors were used for much of the coding, but were always led by internal staff so that the support of the applications or processes created could continue after the transition was completed. Some of these contractors are still on board while IT staffs up and training is completed.

The need for data sharing was partially fulfilled with the integration of certain functions and data to the IBM applications. Other data is available to all users on the HP system from their client utility programs, but integration is not 100% complete. There is a plan in place to create this integration and integrated design of the data in the future.

The primary goal of removing all users from the Unisys and Defunct platforms is now completed, and the Defunct hardware has been scrapped. The Unisys hardware remains until a three month production period is completed. Then it will be sold to a re-marketer. Down time is at an all time low because of the viability of the two platforms remaining.

There is an improvement in user satisfaction because the IBM applications are much more functional than the Unisys and Defunct applications. The data is more accessible than ever and users can share files easier because they are all located on the HP server or in the DB2 data base.

Design of an in house intranet is proceeding, but not completed. It will take at least another year before this is in place.

CONCLUSION

A transition can take place if careful planning is done before hand. The rewards are enormous! Users can access data more easily. IT staff can have careers enhanced and extended. The organization is more able to adapt and agilely react to changing business needs.

Antique architectures can be removed!