Deploying Distributed Computing: Parking gets a New System

Steve Sawyer
School of Information Sciences and Technology
Pennsylvania State University
Rider 1 Building
120 South Burrowes Street
University Park, PA, 16802 USA
o: 814-865-5603
f: 814-865-5604
i: sbs11@psu.edu

William Gibbons
School of Information Studies
Syracuse University
4-291 Center for Science and Technology
Syracuse NY 13244-4100 USA
o: 315-443-2911
f: 315-443-5806
i: wjgibbon@syr.edu

Keywords: packaged software, IS impacts, systems acceptance, IS implementation approaches.

A slightly modified version (and without the teaching note) of this was published as: Sawyer, S. and Gibbons, W. (2000) "Implementing Distributed Computing: Parking Gets a New System," in Khrosowspour, M. (ed.) *Annals of Information Technology Cases, 2*, Hershey, PA: Idea Group Publishing, pp. 24-39.

Deploying Distributed Computing: Parking gets a New System

Executive Summary

This teaching case describes the efforts of one department in a large organization to migrate from an internally-developed, mainframe-based, computing system to a system based on purchased software running on a client/server architecture. The case highlights issues with large scale software implementations such as those demanded by enterprise resource package (ERP) installations. Often, the ERP selected by an organization does not have all the required functionality. This demands purchasing and installing additional packages (known colloquially as "bolt-ons") to provide the needed functionality. These implementations lead to issues regarding oversight of the technical architecture, both project and technology governance, and user department capability for managing the installation of new systems.

Background

The parking department of a large southwestern university is the setting for the deployment of a new client-server based system that uses purchased software. This effort is discussed from inception through post-implementation. Issues covered in this case include project management, implementation planning, vendor management, and balancing responsibilities between centralized and distributed information technology groups. This discussion is set within the context of the university's decision to change its computing infrastructure and to purchase an enterprise resource package to replace its aging, and home-grown, mainframe-based systems.

Learning Objectives

This case is intended for either senior-level undergraduates or early-career master's students. Upon completion of this case, the learner will be able to:

1. Determine, and be able to discuss, both the major issues and key stakeholders affecting deployment of distributed computer-based systems. This includes the:
* technical constraints
* organizational levers and
* change management concerns.

2. Determine, and be able to discuss, issues regarding the emergence of multiple centers of technology expertise inside organizations (due to the distribution of computing).

3. Determine, and be able to discuss, the issues with purchasing, instead of building internally, computer systems.

Preparing for the Case

It is important to have an understanding of distributed computing and client/server concepts to assist in developing this case. If you do not, or would like a refresher, please read Harvard Business School teaching note 9-195-211 (by Geoffrey Bock and Linda Applegate) on Client/Server Computing. This note provides an overview of client/server computing (the dominant form of contemporary distributed computing systems) and issues with both the business value of client/server computing and some of the managerial aspects of these types of computer systems.

Deploying Distributed Computing: Parking gets a New System

Having finished preparing his presentation, Ed Morales sat in his office late that Thursday night in June, 1998. As director of facilities at a large southwestern university, he was to meet the university=s CIO tomorrow to go over the recent implementation of a new computer system supporting the Parking Department=s operation. As the first of the new externally developed and client-server-based systems being deployed campus-wide, the CIO wanted a full report of the lessons learned by the Parking effort. Ed had mixed emotions: with little computing experience or support, amidst massive organizational changes, they had made the transition.

Ed manages the grounds crews, the physical plant, and parking departments (see Exhibit 1). He explains that "the first and last experience when visiting our campus usually involves parking. How that experience goes impacts attitudes toward our school and effects our business in terms of enroll-ment, employment, and community support." Ed acknowledges that this aspect of how people interact with a large organization is considered mundane: AThe importance of operations like parking is often overlooked when considering critical university functions.@ Ed is pleased that this is not lost to the department=s employees and their professionalism and care for the customer is known across campus.

Reflecting on the implementation effort, Ed said: "After one year of operation our expenses are up and revenues are down. Primarily, this is because our new computer system is 100% over budget. It’s a state-of-the-art, client/server based system that has taken more effort than we expected." As an example of how this happened he said: "At first, I was very impressed to see the vendor in our office every day working on post-cut over problems. Then I discovered he was billing us by-the-hour for these consultations!" His report contained a full accounting of the process, including these mistakes, and ended with a set of suggestions for others who planned to deploy client-server-based systems.

Transitioning to Client-Server Computing

Harold Gomez is both Chief Information Officer (CIO) and director of research at the university. He became CIO after the previous director of administrative computing left. He had initiated the transition from mainframe to client-server computing infrastructures in 1992, a year to the day after he took over as CIO. Harold states: "conversion to this new paradigm <client-server> will eliminate the expense of a dual-mode environment and direct resources formerly associated with the mainframe to a newer and more powerful approach to computing." After a distinguished career as a computational linguist, Harold had moved to administrative duties by becoming first the director of research, then CIO. Harold's career as an academic had made him familiar with technical aspects of computing, his time as a department chair had exposed him to organizational politics, and his reputation as a fair, thoughtful, and intelligent peer had made him the first choice of the selection committee.

Harold's expects to complete the client/server infrastructure transition before the turn of the century. This is important since existing systems were not Year 2000 compliant. In making this change, Harold's strategy has been to retain present staff and retrain them, move from building systems to buying them from the market, minimize their reliance on consulting and distribute the costs of computing to the units and departments where the computing is being done. After six years in the job he is seen as a visionary by most of the senior administrators and his staff adores him. Around campus, his aggressive pursuit of client-server computing, and the limited shift of budget from central computing to distributed departments, is not seen as cheerfully. Still, the university's senior administrators are solidly behind Harold and his efforts. And, his personal credibility is unassailable.

The reasons behind this transition center on how the university conducts itself. For example, the university, like all businesses, faces an escalating cost of computing. Mainframes and their databases have become central to conducting business. However, the present systems are internally developed, expensive to maintain, rely on outmoded technologies and often inhibit analysis and reporting. Harold believes that the central computing group (CCG) cannot keep paying for the mainframes while also investing in a parallel infrastructure based on networks of personal computers (PCs). These PCs are everywhere on campus and in both administrative units and academic departments. The PCs are also cobbled together in an ever-growing interconnection of local-area networks (LANs). His response to this dilemma is to embrace a client-server/network architecture.

The fundamental premise behind client-server computing is that the total capacity of the distributed servers and workstations exceeds that of the mainframes they replace. This new architecture also moves transaction processing closer to the user and provides the opportunity to streamline existing business processes. The concept is premised on the host (server) and client (PC) sharing processing and storage. This is enabled by high-speed data commun-ication connecting clients to servers (see the Addendum for more on client/server computing).

Harold understood the transition would not be simple. Technically, the transition demands acquiring extensive knowledge about client-server technology and its incorporation into departmental environments. Administratively, this transition shifts responsibility and funding away from the centralized computer organization to the individual departments. Organizationally, a successful transition demands balancing the technical expertise between centralized and decentralized operations. That is, CCG must provide and maintain the network infrastructure. But, the individual departments must maintain the software and hardware they need for their operations .

The Parking Department

Having earned a CPA and worked for years at the university in accounting, Ed moved to his current administrative position five years ago. He left accounting to gain ›line= managerial experience in the hope that he could move up the administrative ranks. While Ed is not very comfortable with computing he is proud of his Ahands-off@ management style and attention to Aprudent financial controls and oversight.@ As he says, AI try not to meddle in the day-to-day. As long as the numbers look good, I=m pleased.@ He relies on the weekly staff meetings with his three direct reports to keep him updated.

The Parking Department=s mission is to provide excellent customer service while maintaining and enforcing rules and regulations associated with the limited space available for parking. It is a large operation with annual revenues of about $3.5 million and expenses of $1.5 million. The department operates metered and attended parking lots, issues permits and citations, and collects fees and fines. This demands sophisticated data input, query, and storage requirements. The database enabling the department=s operation contains records of more than 50,000 permits along with the associated citation and collection records for the university=s staff and students. This database is a 1960's vintage indexed file system structured so that the only key is the permit holder's social security number.

Deciding to Change

Gretchen Juarez, the present Parking Director, says the "new system was chosen by what it can do like we do today and not what it can do for our overall mission and into the future. . . .@ Since the 1960's, the parking database, running on the university's mainframe computer, was an account-based system, typical of technology of that period. Each customer, student or staff, is assigned an account that contains a running balance of money owed. Permit and citation fees, along with payments and credits, are posted to this account. Each account is, at present, measured by its "bottom line" and the success of the department is determined by summing account balances against operational expenses.

Tracking specific activities within accounts (or across accounts) is difficult and time consuming. For example, partial payments to individual accounts and successful appeals to individual fines within multiple citations cause problems in account verification and auditing. The parking system is linked to other university systems for employee payroll deduction and for "holds" on student registration if there are unpaid balances. Customers are routinely inconvenienced by the time required for payment verification and audit checks. Conversely, fines by graduating students often go unpaid because the system is not set up to easily isolate these.

This existing, mainframe-based, batch processing, operation had problems beyond its account-based design. According to the previous Director of Parking, Wayne Fricks, "the problem was that within CCG, Parking was always a low priority for 'enhancements'. We just could not support our customers with the archaic batch processing system." Gretchen adds, "Simple change or modification requests often took months to investigate. We weren't part of the 'in crowd' and if we were to wait for CCG, we'd be waiting forever. The decision was obvious, if we were to modernize, and offer better customer support, we had to forge ahead on our own."

In 1992, and with Ed's blessing, Wayne began to search for a more modern system. Ed sold this to his boss, Joanne Nunes, because they were to use internally generated funds. This action preceded the overall university client-server migration plan. This leadership position was a further catalyst – a chance for Parking to be a focus of attention. Throughout the effort, many other department heads on campus contacted Ed with interest in how the migration was progressing. Several members of CCG would also chat informally with members of Parking to see what they might learn (often providing useful advice in return).

Wayne (and Ed) expected the move to client-server technology to simplify data access, improve data record structure (primarily by allowing for relational database format with multiple keys) and expand the functionality beyond what the mainframe system provides. Increased functionality was centered on more easily accessing permit-holders records and dealing with appeals to tickets and streamlined billing. Further, this new system would allow the use of both hand-held computers (to help automate the ticketing process) and electronic ticketing in the metered lots (to improve billing and cash flow tracking). Finally, both Ed and Wayne believed that the distributed computing platform would reduce the time-delays of waiting for batch-processing on the mainframe.

New systems to help in the operation of municipal and private parking operations began to appear in the late 1980's. These systems included individual ticket-based software coupled with computerized, hand-held, ticket-issuing devices. In these ticket-based systems, fines, payments, and audit trails could be associated with individual components of the account and not simply the account balance. These new designs could provide improvements in efficiency, accuracy, and customer service.

Although the Parking Department's employees shared a positive attitude toward new technology, none of them had much technology expertise. Said a senior clerk AIt feels like we=ve no choice other than to modernize the operation." Until the decision to modernize their computing system, the Parking Department had relied heavily on CCG for technical assistance. Having decided to replace the mainframe system and move to some form of PC-based system, they turned to CCG for advice on how best to modernize. Although CCG encouraged Parking's modernization, and seemed quite interested, they chose not to be responsible for, or be directly involved in, software selection and database design. This responsibility shift was not recognized by Parking personnel as a fundamental characteristic of distributed computing but was considered ". . . a lower than desirable level of support" caused by a unique physical separation of the department from the main campus. In the late 1960's, during periods of campus unrest, for security reasons, a new office building was built off-campus approximately one mile away from the main campus. CCG=s offices are in the remote office building. These people are noticeably separated from the main campus where CCG does 90% of their work functions.

Evaluating Alternatives

In late 1992 Wayne began investigating available software and associated computing platforms. After reviewing the products of four vendors, the one most local to the campus was chosen as "being most supportive." The vendor suggested use of a dedicated server, PC workstations, and a shared database platform. A review of this proposal by CCG led them to recommend a more economical shared server arrangement. The CCG review also cautioned against adopting a software system based on a shared, PC-only, database technology. Parking was now in a very difficult position, not able to modernize without accepting responsibility for the design and decision, while needing advice and support from those most knowledgeable of the technology and its implications. They were caught between the university's computer experts and the vendor. Says Gretchen, "It was obviously becoming a separation but we could not risk a complete divorce. CCG was about to lose some of their control and we were about to accept additional responsibility. We could not afford to lose their support. Besides the in-house operation, several interfaces to other systems had to be addressed. We decided to compromise by accepting the idea of a shared server to keep CCG involved."

Parking's new responsibilities in technology acquisition quickly dominated implementation. Other important issues such as redesigning work flow, assessing organizational effects, and developing a change management plan were Asidetracked.@ As Ed recalls, "our focus was to just get the transition accomplished.@ "We wanted a system better than the existing one but we didn't want this much responsibility for its selection and design" said Lenny Pietsch, the designated technologist in Parking. Lenny=s had spent years as a hobbyist learning how to put PCs together, make them work, and was seen as very knowledgeable (about computers) by the rest of the Parking Department. However, he had never had any formal training and had never worked on either a network or a networked system.

In early 1993, after some delays due to the series of discussions regarding server support, Parking reached a compromise with CCG. Central Computing would provide and maintain the shared server and Parking would be responsible for software selection and its impact on their operation. Interface requirements between the chosen software platform and other campus systems would be a joint responsibility involving Parking and CCG. Still, no one person at CCG became a liaison for the effort and Lenny dealt with whomever answered (or returned) his phone calls and e-mails. Lenny relied more on informal contacts because he was well-liked by many of the CCG members (for example, he was the power hitter on their intra-mural league softball team).

Deploying the New System

The phased implementation started with the operation of hand-held ticket writers in January, 1994 (See Exhibit 2). These devices made the issuing of citations and the associated data entry much easier than before. The Ahand-helds@ produced a paper ticket and accumulated data for batch downloading to the mainframe. This required a special (rush) modification to the current mainframe system which Parking had not counted on paying.

Both Ed and Wayne were surprised by this expense: neither the vendor nor CCG had mentioned this during the discussions earlier in the year

The second phase, finally implemented in May 1997, put Parking's new client/server system up on a shared server which was maintained by CCG. The server was located at the remote campus and it also supported other departments at the remote location. The extended interval between project phases was due to unplanned organizational change within the Parking department. During 1994, both Wayne and the other manager (Jay Beloc) left the department. Wayne left for a similar position at another university in February. Jay left in April after he was found to have been rigging parking tickets. Responsibility for continuation of the client-server implementation went to Gretchen. She had been a member for 15 years, though this was her first managerial experience.

Being moved to manager was a significant promotion (several steps) which moved Gretchen past Lenny, who became her assistant. Until that time, Gretchen had not been involved in the computing project.   Many of the project and change management plans were not yet complete. However, Gretchen had neither the skill or the time to complete these plans and the rest of the project was characterized by rather haphazard decision-making. In fact, from this point on, both Lenny and Gretchen relied on the software vendor to help them answer questions with the university's network, and to keep the parking system running. Lenny and the vendor got along very well since their children both attended the same high school.

The new system went on-line after May graduation activities and had a three-month window of reduced loads before students and faculty returned for the Fall, 1997, semester. This interval proved insufficient for accomplishment of all on-line testing and training requirements. Training for the six full time and 24 part-time employees began in July with everyone receiving the same four-hour class (in groups of ten). Training was completed about three weeks before the old system was turned off. This training splintered the department. Some felt training was too simplistic, other felt it was too complicated. The vendor and Lenny, Parking's training coordinator, feel that they did sufficient testing in the six months before cut over. Gretchen, and most of the full-time employees, feel this was not so and point to difficulties in the period after cut over as an indication of improper pretesting.

The Vendor

Parking's 1993 contract was the vendor's first big success. Maxine Systems, the vendor, had been started two years previously by Al Horter and named after his wife. Al had left IBM and begun his own small company designing database systems. Soon after building a system for a local parking lot company, he realized this was an under-served market and began turning his system into a parking management software package. In early 1993, after 14 months of development, Parking was the first contract for Release 1.0.

The next few years were heady times for Maxine Systems. Three new releases and dozens of major contracts led to spectacular growth. Parking's system implementation slowly slipped to a back burner. The contract was structured so that 50% was paid up front with the rest due upon implementation. Further, Wayne and Al structured the contract to provide

100 hours of implementation support at no cost, with any additional support charged at $100/hr.

When Lenny contacted Al in late-1996, it had been 17 months since anyone at Parking had spoken with Maxine Systems. Al tried to get Parking to upgrade to the newest release but this would have added 20% to the purchase price and both Gretchen and Lenny felt that was impossible. This meant that Parking's new implementation would be done with a three year old system. Further, Al had to be the lead on this project since no one else at Maxine knew anything about Release 1.0.

Post-Implementation

Phase two's aggressive project schedule, because it was tied to the academic year calender, caused problems in system testing, training, and documentation. Lenny, the vendor, and Gretchen felt that a May cut over would enable on-line testing and problem correction before the return of students and staff in August. The two systems ran in parallel for three months (May to July). However, Parking Department people felt there was no reason to test the new system extensively while the old system was running. People were very busy and confidence was high that the new system would resolve some of the problems with the mainframe-based system. Upon cut over, problems emerged. For example, in the fall semester, Parking could not figure out how to bill students for unpaid fines, to collect faculty and staff payments through payroll deduction, and to automatically search for license plate/owner matches. Further, it was only after employees started using the system that they discovered certain functions were not yet operative (and required additional programming). Fixes did not come easily as the system was now on line and handling "live data.@ In order do any updates, the system needed to be taken off line.

To further complicate the cut-over and post-cut-over phases, problems occurred with CCG=s operation of the shared server. Indications of insufficient memory, multitasking time delays, and failure of data back up procedures were common in the first six months of operation. Pinpointing the cause of these problem was difficult and this made it hard to assign responsibility for corrective action. That is, given that the shared server, software, local and network operating systems, and local PCs were all from different vendors, and supported by either third-party vendor contracts, CCG, or Lenny, just establishing what was not working became a complex task. Says Ed: AWe now realize that it would have been better to train more on a non-live system. Data entry mistakes on the live system have caused serious problems. Also, system documentation was not made available during the testing and training phases of the project adding to difficulties in learning the new system.@

Poorly defined project and change-management plans meant that there had been no collection of baseline data for comparison of pre-cut over and post-cut over operations. As Ed says: AThe bottom line is down and we really don=t know how much of the change is attributable to the new system. If we had been tracking and projecting each of the revenue and expense components, it would be much easier to pinpoint a source of change.@ For instance, the person responsible for appeals to issued tickets says, "It now takes me ten times as long to process an appeal as I now handle the whole appeal from inquiry to resolution. Obviously some other people are no longer involved but I don't know if total process time and associated expense are better or worse.@

By May 1998, a year after the system went on-line, it had begun to work well. Employees were adjusting to new ways of doing their jobs, and the relationships among departments improved due to more frequent meetings and formalized procedures. For example, the Parking Ticket Management System User Documentation, provided by the vendor, was supplemented by a formal Methods and Procedures Manual written by Parking Department personnel. Still, implementation took three years (instead of one) and ran 100% over budget. Ed was particularly concerned about this. He did not learn of the cost overruns until the vendor billed him for consulting. Gretchen and Lenny had never read the clause that said that, after the first 100 hours, the vendor billed at $100/hour. The consulting costs soon climbed to be equal to the system’s purchase price. And, Gretchen and Lenny did not feel they could operate the system without 5-to-10 hours of the vendor=s time per week. What had seemed great support was costing them almost $1000 per week.

Meeting Harold

In June, 10 months after the new system went live, Ed received a call from Harold. While Ed reported to a different VP, Harold asked if Ed would prepare a brief on Alessons learned@ from the Parking Department=s experience. At first, Ed felt good to be asked. His report -- and presentation -- detailed the ups and downs of the transition. Still, the cost and schedule problems concerned him. Earlier this evening, on her way out the door, his VP, Joanne Nunes, had said she looked forward to his presentation to Harold tomorrow. This was the first time he had heard she was to attend.

Addendum: Client/server Computing

Client/server computing is a phrase used to describe a model for computer networking. In client/ server computing, processes are divided between the client and the server. This relationship is based on a series of requests and responses. For example the client requests services or information from another computer (the server). Then, the server responds to the request by sending the results of the request back to the client computer. In a client/server setting, the client computer runs a software application called a client program. The client program:

1. Allows the user to send a request for information to the server.
2. Formats the request so that the server can understand it.
3. Formats the response from the server in a way that the user can read.

The server program receives a request from a client and processes the request and responds by sending the requested information back to the client. Most transactions that occur on the Internet are client/server based. One server application can support numerous clients, giving each client access to all of the application data. Properly implemented, a client/server application appears to be a stand-alone application. Client/server systems operate over LANs, WANs, or a combination of the two.

Client/server computing is typically used in situations where an organization has one or more central data repositories and many user access points. The user access points may or may not be co-located with the central data repository. Because of the flexibility inherent in client/server systems, they tend to be much richer in functionality than the mainframe systems that they replace.

State of the Technology

Client/server hardware and software technology has reached a mature state, however hardware performance enhancements continue to be realized. The flexible nature of client/server architecture allows old obsolete servers to be replaced by new servers running the same applications. Network and programming standards are at point where applications can quickly be ported from the machine of one vendor to another. Because of the "open" nature of client/server computing, significant competition exists among the major workstation/server manufacturers.

There are two primary modes of organizing a client/server infrastructure. A *two-tier architecture* is where a client talks directly to a server, with no intervening server. A *three-tier architecture* introduces a server (or an "agent") between the client and the server. The agent plays several roles (such as hosting programs or data). These architectures can support various processing models such as cooperative, distributed and intranets.

The connectivity inherent in client/server computing allows applications to transparently communicate with other programs or processes, regardless of their location. The key element of
connectivity is the network operating system (NOS). NOS provides services such as routing, distribution, messaging, file and print, and network management services. NOS rely on communication protocols to provide specific services. The protocols are divided into three groups:
media, transport and client-server protocols. Media protocols determine the type of physical connections used on a network (some examples of media protocols are Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), coaxial and twisted-pair). A transport protocol provides the mechanism to move packets of data from client to server (some examples of transport protocols are Novell's IPX/SPX, Apple's AppleTalk, Transmission Control Protocol/ Internet Protocol (TCP/IP), Open Systems Interconnection (OSI) and Government Open Systems Interconnection Profile(GOSIP)). Once the physical connection has been established and transport protocols chosen, a client/server protocol is required before the user can access the network services. A client/server protocol dictates the manner in which clients request information and services from a server and also how the server replies to that request (some examples of client/server protocols are NetBIOS, RPC, Advanced Program-to-Program Communication (APPC), Named Pipes, Sockets, Transport Level Interface (TLI) and Sequenced Packet Exchange (SPX)).

A client/server environment is typically heterogeneous and multi-vendor. The hardware platform and operating system of client and server are not usually the same. Moreover, the NOS and applications may also come from multiple vendors. Client and server processes communicate through a well-defined set of standard application program interfaces (API's) and RPC's. An important characteristic of client-server systems is scalability. They can be scaled horizontally or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means migrating to a larger and faster server machine or multi-servers.

Strengths Client/server computing facilitates the development of applications with enhanced access to data. The implementation of client/server systems has enabled organizations to achieve breakthroughs in performance and significantly higher levels of customer satisfaction. The flexibility inherent in client/server architecture allows for more competitive procurements, and easy upgrading of hardware. Client/server implementations force organizations to focus on how information moves within their organization, and how to best use that information.

Weaknesses Client/server computing is a completely different approach from mainframe computing. Using client/server to implement a major system requires a large and very expensive development and conversion effort. Client/server systems are often not implemented until the mainframe system that they are designed to replace is obsolete. Existing information systems (IS) personnel must be retrained on the new technology.

Exhibit 1: Partial Organizational Chart of Big Western University

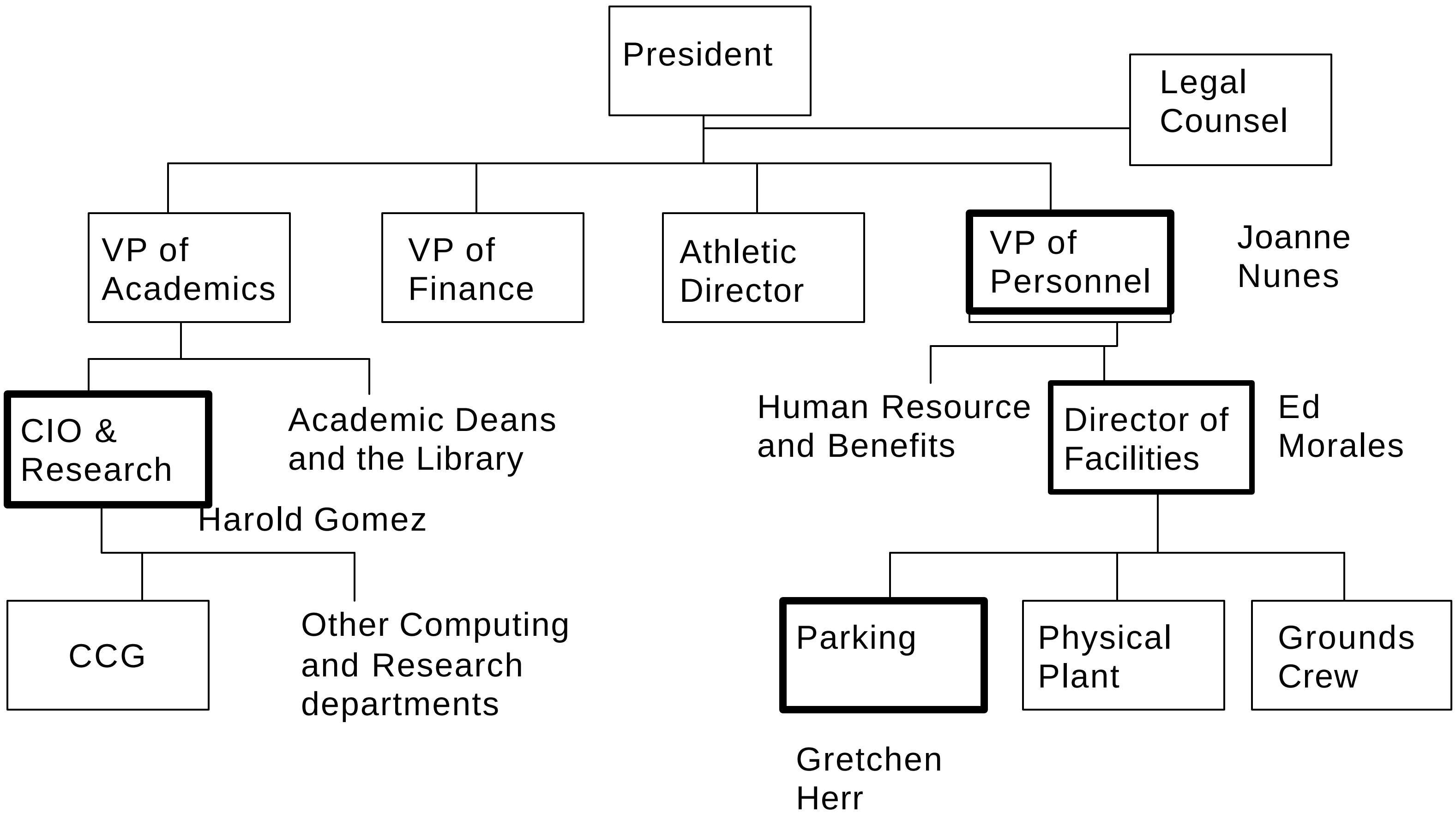

Exhibit 2: Original Parking Implementation Plan

Planned

Fall '92	Winter '93	Fall '93	Spring '94	Summer '94	Fall '94
Vendor Selection	Fitting vendor package to Parking needs	Implement "hand-held" units	Parallel Run of new and old systems	Training and cut over	Fully Oper-ating

Major Players in Implementation:

Parking Director	Oversee implementation and manage project.
Vendor	Fit package to Parking needs and install systems.
Parking IT person	Support implementation and maintain new systems.
CCG:	Liaison as requested.

Deploying Distributed Computing: Parking gets a New System

Executive Summary

This teaching case describes the efforts of one department in a large organization to migrate from an internally-developed, mainframe-based, computing system to a system based on purchased software running on a client/server architecture. The case highlights issues with large scale software implementations such as those demanded by enterprise resource package (ERP) installations. Often, the ERP selected by an organization does not have all the required functionality. This demands purchasing and installing additional packages (known colloquially as "bolt-ons") to provide the needed functionality. These implementations lead to issues regarding oversight of the technical architecture, both project and technology governance, and user department capability for managing the installation of new systems.

Background

The parking department of a large southwestern university is the setting for the deployment of a new client-server based system that uses purchased software. This effort is discussed from inception through post-implementation. Issues covered in this case include project management, implementation planning, vendor management, and balancing responsibilities between centralized and distributed information technology groups. This discussion is set within the context of the university's decision to change its computing infrastructure and to purchase an enterprise resource package to replace its aging, and home-grown, mainframe-based systems.

Learning Objectives

This case is intended for either senior-level undergraduates or early-career master's students. Upon completion of this case, the learner will be able to:

1. Determine, and be able to discuss, both the major issues and key stakeholders affecting deployment of distributed computer-based systems. This includes the:
* technical constraints
* organizational levers and
* change management concerns.

2. Determine, and be able to discuss, issues regarding the emergence of multiple centers of technology expertise inside organizations (due to the distribution of computing).

3. Determine, and be able to discuss, the issues with purchasing, instead of building internally, computer systems.

Preparing for the Case

It is important to have an understanding of distributed computing and client/server concepts to assist in developing this case. If you do not, or would like a refresher, please read Harvard Business School teaching note 9-195-211 (by Geoffrey Bock and Linda Applegate) on Client/Server Computing. This note provides an overview of client/server computing (the dominant form of contemporary distributed computing systems) and issues with both the business value of client/server computing and some of the managerial aspects of these types of computer systems.

Teaching Notes for: Deploying Distributed Computing: Parking gets a New System

This case explores issues regarding the governance and management of distributed computing. The rather mundane venue B a small and non mission-critical department, like Parking, of a large organization B provides a tactical view of larger strategic issues. In support of the case this note has two sections: preparatory questions for students to prepare and a set of lessons learned.

Preparatory Questions for Students

1. There are at least four groups of stakeholders involved (the vendor, the CCG people, the managers of Parking (including their technologist), and Parking employees). Can you draw a time-line of events that reflects each stakeholder group's perspective?

2. If you are Ed, what are the key points to make in tomorrow's meeting with Harold?

3. Why do you think Joanne is attending this meeting?

4. What are the good aspects of this implementation?

5. If you had been called in after the first manager left and the second manager was fired, what would you have done? Frame your response as a series of action items.

Lessons Learned

1. Computing system implementations are still very difficult to do well. The history of computing shows that new systems often suffer from being over budget, being late, and missing functionality. Worse, occasionally the newly implemented system does not meet minimum users needs and/or expectations. In a distributed environment these problems are even more apparent for at least two reasons. Firstly, the technical aspects of a client/server computing system change is often nearly invisible to the user since the infrastructure is mostly hidden. Often, a major system change may appear as one more icon on the workstation's screen. Secondly, the goal of distributing the computing is to move it (and its support) closer to the user. This distribution has a twofold effect, both of which are evident in the case. The first effect is that more people are affected by the system if the system does not work. The second effect is that more of the responsibility to keep this system (typically comprising many vendors' hardware and software) is now the responsibility of a (typically junior) technologist (Lenny in this case) who is often physically isolated from the resources of the vendors and from central computing. All this reinforces the importance of developing, and adhering to, a project management plan. This also highlights the need to develop specific change management plans to assist the range of people involved in this effort.

2. Deploying client-server systems heightens the importance of several well-known consequences of computing implementation. Since more of the processing (in terms of querying databases, analyzing data, and producing reports) is in the hands of the users, they are expected to grasp what their work means. This leads to the heightened effect of at least these issues:

2a. Like any large project, a new systems deployment needs project leadership. The project driver, or champion, should be at a high enough level and with enough interdepartmental power to make things happen. Also required is a knowledgeable project leader, a dedicated implementation team, and sufficient organizational stability so that project focus can be maintained throughout the planning, design, and implementation phases. Since these efforts take several years, this team must be maintained throughout the process. Certainly the five-year journey of the Parking Department is due, in part, to a lack of stable leadership.

2b. Gathering requirements for a distributed, and vendor-supplied, system is even more diffi-

cult than trying to build a system. One of the most difficult properties of this undertaking was the complexity and interrelationship among the components of technologies, business processes, and organizational boundaries. Typical of organizations implementing new information technologies, Parking concentrated on the technology itself and not on what it would do for the function of the department. This was both time-consuming and demanded extensive integration of various vendors= systems along with the support of existing computing resources supplied by CCG. So, changes in methods and work requirements came as a surprise to employees of the department.

2c. Decomposing a large change effort into manageable stages leads to a higher probability for success. New technology implementations should be broken into manageable pieces that build on each other toward total business solutions. This way each piece can be accommodated and adjustments made incrementally to better adapt to the next phase. Parking did an excellent job in this area with the early introduction of Ahand held@ ticket issuing devices before the major database and client-server system transition. In the early stages of the project, output from the Ahand helds@ was downloaded to the mainframe system. This allowed time to gain expertise on this phase before moving onto the next. These well defined and manageable chunks provide intermediate assessment points and build confidence in the transition plan as it progresses. The cut-over after dual operation was not as successful, but the intentions were well-placed.

2d. Change-management planning is critical in order account for the organizational and social effects of new systems. Planning for change is an important and often overlooked component of the transition process. It seems to be even more critical for distributed computing efforts. In agreement with the literature on implementation and management of technology associated change, this case supports that managing the change is at least as important as bringing to technology to the organization. To accomplish such management, people must be trained in the change process that takes into account the unique challenges presented by the interaction of existing organizational processes and the new technology.

3. Distributed computing places extensive demands on technical leadership. Parking=s migration from a centrally focused network environment to a distributed client-server orientation placed inordinate pressure on a junior technologist (Lenny) to comprehend sophisticated technologies with little support. Without clear guidance from CCG, Parking did not know that it did not possess adequate knowledge of the technology or the implementation/change management process. This lead them to rely on the vendor. Since the vendor does not have responsibility or accountability for the operation of the business, such a situation could be untenable. In Parking=s case, their technologist was isolated from the resources of CCG. It may be that CCG should have provided project/change management and training support. To ask CCG to help manage this transition may be difficult but they may be the only location which could provide this service.

4. Informal social networks are important to successful implementation. The informal interactions between staff members and across departmental boundaries were critical components to success. For example, the rapid turnover of management and the dissolution of the informal ties that had developed between Ed and Wayne took some time to rebuild. This is one factor in the mis-communication between Ed and Gretchen relative to the vendor consulting. The lack of informal ties between CCG and Parking in the early stages led to problems. Lenny=s rapport with CCG and with the vendor may have made the transition possible. Finally, the informal socializing among the parking department staff made it easier to deal with the new system=s glitches in the Fall of 1995. The point is that, independent of the control strategies and change efforts, this informal socialization plays a key role in making a transition occur.