Implementing “Best-of-Class” Client/Server Systems at a Small College

Rick Peterson
Randolph-Macon Woman’s College
Lynchburg
Virginia

Rather than using a single vendor system, Randolph-Macon Woman’s College (R-MWC) elected to integrate diverse client/server software in a distributed environment for finance, human resources, student (registration, admissions, degree audit/advising, financial aid), and development/alumnae, completely replacing the College’s current mini-computer system. This paper will provide a history of the project (including how we chose to integrate diverse systems), and then discuss the specific systems selected, our experience in implementing the systems, rough costs, timeline, and the technical infrastructure. Issues specific to integrating “best-of-class” client/server systems will be discussed in detail.
Implementing “Best of Class” Client/Server Systems at a Small College

“One of the main tenets of the database crusade was the creation of a single and therefore central database for the entire organization. The idea that it was a net plus to have a single database turns out to be exactly the problem. Both from a perspective of design feasibility and from a perspective of performance feasibility, what we want is not one database but many. And then we want a mechanism for integrating those databases, tying them together, and providing senior management with a corporate-wide view. It turns out, however, that providing that corporate-wide view does not mean that every last piece of operational data has to be in the one universal repository. And in the final ironic twist of fate, as we start to think about empowered employees and self-managed teams, the last thing we want is a monolithic database.”

Beginning in 1992, Randolph-Macon Woman’s College began taking major steps to improve its strategic technology systems. In the past two years, the college has concentrated on replacing its core mini-computer system by evaluating, selecting, and implementing a diverse set of “best-of-class,” client/server software packages. We chose the “best” software system available at the time to meet the needs of each major functional area. We had evaluated several single-vendor systems that have traditionally provided small colleges with software solutions. However, the maturation of client/server technologies, the implementation of a new campus-wide network, and the growth in use by our faculty, staff, and students of GUI-based personal computers and standardized, PC-based productivity software tools provided the network infrastructure and software tools necessary to benefit from deploying distributed client/server systems. As we continue our implementation in each major area, these individual systems will integrate to form a complete system. In this paper, I will briefly discuss the history of our systems project, why we chose to pursue a “best-of-class” strategy, and discuss the software packages we picked, including approximate costs. Finally, I will discuss in detail the unique issues raised by integrating diverse client/server systems, sharing our learning to date.

Background history of the project

It is important to know from the start that the groundwork for implementing these “best-of-class” systems was laid as long ago as 1992. At that time appropriate, long-term decisions were made at the Board and executive levels to support improvements in the college’s technology environment. Specifically, the Board, President, and senior staff supported the creation of an integrated Computer Services department (overseeing both academic and administrative computing) and the creation of representative computing advisory committees (encompassing both academic, administrative, and college-wide computing). They also supported the implementation of a new, campus-wide network, and the establishment of a “fleet replacement plan” for faculty and staff personal computers. A single computing organization has allowed us to better manage computing resources; the fiber/copper 10/100BaseT-Ethernet network connects all faculty and staff offices and student residence hall rooms and supports a high-speed, dedicated connection to the Internet. A “fleet replacement plan” allows us to maintain a technologically current fleet of personal
computers and replace it every three to four years. In 1994, Computer Services advocated the move to a standard, GUI-based PC environment for all users. By accessing legacy computer systems, sending/receiving electronic mail, and working with standard office productivity software (Microsoft Office) in a GUI-based environment, staff, faculty, and students have gained a base of knowledge necessary for them to use the new software systems. Without a single, coordinated Computer Services department, a new, solid, campus-wide network, a strategy and budget for replacing PCs in an appropriate time frame, and a user community with a knowledge of GUI-based software, the implementation of these distributed systems would NOT have been possible in the time frame estimated.

In late 1994 a consultant was hired to examine our current systems and work processes and help us migrate to a new campus software system. The college’s legacy system (an IBM AS/400 running home-grown applications in every major area except Payroll and Development) was recognized across the college as unable to meet the college’s information needs and support its future strategic directions. In June of 1995 a representative task force was formed and determined that the college would be best served by systems with the following characteristics (cost was considered but it was not a primary consideration):

- Easy to learn, easy to use functions;
- Easily integrated with GUI-based office productivity tools;
- Easy to use ad hoc reporting and data access;
- Integration capabilities between and among, and close integration within, the major areas of student, finance, human resources, and alumnæ/development;
- Windows, LAN-based, client/server architecture;
- Relational data base (preferably Microsoft’s SQL Server);
- Good vendor track record.

Considering preliminary demonstrations and acquired knowledge, the task force recognized that at that point in time a single vendor might not have the ability to provide systems solutions that met the above criteria in all major areas. Therefore, the task force anticipated that the College might be best served by selecting the best available system in each major area, taking a “best-of-class” approach. Success of this approach would rest in no small part on the flexibility of each system in receiving and sending data from other systems. A request for information was then issued to selected vendors, and proposals and systems were evaluated and demonstrated. Since no single vendor meeting our criteria could supply adequate solutions for all the major areas in a reasonable time period, in October of 1995 the task force unanimously agreed to pursue a “best-of-class” approach and integrate the best software systems in each major area.

**Vendors, implementation timeline, and cost**

After further evaluations and demonstrations, in February of 1996 the task force recommended the purchase and implementation of the following systems (under the noted time frames):

<table>
<thead>
<tr>
<th>Software</th>
<th>Architecture</th>
<th>Go-Live Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABT PowerCAMPUS</td>
<td>PowerBuilder / MS SQL</td>
<td>Admissions 12/1/96</td>
</tr>
<tr>
<td>(admissions, registrar, billing, accounts receivable,</td>
<td>Server</td>
<td>Student 3/1/97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Billing 4/1/97</td>
</tr>
</tbody>
</table>
Costs for purchasing and converting systems are detailed in the table to follow. The table reflects total project costs for the major budgetary items over a five-year time frame. Certain costs already related to operating expenditures such as ongoing software maintenance and PC “fleet” replacement after year two of the project are NOT included in the project totals.

<table>
<thead>
<tr>
<th>Major Budget Line Item Description</th>
<th>Cost / 5 years:</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Software Systems</td>
<td>$208,000/project</td>
</tr>
<tr>
<td>Training, Travel, Support</td>
<td>$124,000/project</td>
</tr>
<tr>
<td>Server Hardware (four, dual or single Pentium-based servers, RAID5, 6-10 GIG HD, 64-128 MB RAM running Windows NT)</td>
<td>$40,000/project</td>
</tr>
<tr>
<td>Conversion</td>
<td>$33,000/project</td>
</tr>
<tr>
<td>Yearly Ongoing Maintenance</td>
<td>$48,000/year</td>
</tr>
<tr>
<td>Upgrades to Campus Network Equipment</td>
<td>$48,000/project</td>
</tr>
</tbody>
</table>
backup unit, 100mbps switches, etc.)

<table>
<thead>
<tr>
<th>Category</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC Upgrades (years 1 and 2)</td>
<td>$124,000/project</td>
</tr>
<tr>
<td>Network Printers</td>
<td>$18,000/project</td>
</tr>
<tr>
<td>Temporary Staff Support</td>
<td>$25,000/project</td>
</tr>
<tr>
<td>User Training, Consultative, and Other Support</td>
<td>$160,000/project</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>$20,000/project</td>
</tr>
</tbody>
</table>

**Total Cost of Project:**
(includes year 1 maintenance on the software systems)

$800,000/project
Computer Services staff changes

Without going into details, it may be instructive seeing how the configuration of the Computer Services department staff changed in the course of the last three years. Notice how a back-office operations position has become a front-office user support position and that a strict programming position has evolved into a systems administrator position.

<table>
<thead>
<tr>
<th>Previous Position</th>
<th>Current Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Director of Computing</td>
<td>Director of Computer Services</td>
</tr>
<tr>
<td>Network Manager</td>
<td>Network Manager</td>
</tr>
<tr>
<td>Senior Programmer/Analyst</td>
<td>Senior Programmer/Analyst</td>
</tr>
<tr>
<td>Senior Programmer/Analyst</td>
<td>Client/Server Systems Administrator</td>
</tr>
<tr>
<td>Computer Operator</td>
<td>User Support Coordinator</td>
</tr>
<tr>
<td>Help Desk Coordinator</td>
<td>Help Desk Coordinator</td>
</tr>
<tr>
<td>Microcomputer Technician</td>
<td>Microcomputer Technician</td>
</tr>
</tbody>
</table>

Distributed Client/Server Systems

Distributing client/server systems offered Randolph-Macon Woman’s College an opportunity to meet its selection criteria (as noted above) for campus software systems. Specifically, they provided our institution with enticing advantages and many new challenges as contrasted with single-vendor systems.

Advantages

Each functional area/office is free to pick the software package THEY desire. It may be almost a cliché now to note that customer “buy-in” is an essential element in the long-term success of any process/project. What better way to encourage and establish buy-in than to give department heads the opportunity to pick the system THEY want? They do not have to choose what the computer department wants, and they do not need to gamely participate in a selection committee “least-common-denominator” solution, one that often forces the office to pick a system for the good of the institution rather than for the good of the office.

Vendors who concentrate their resources and efforts in a particular functional area will be better able to develop and maintain systems than vendors who try to supply all things to all institutions. Who can develop and maintain a better general ledger or accounts payable package: a single system vendor with 100 or fewer clients or a financial software systems vendor with thousands of clients? How many “single-system” vendors are already looking to partner with companies who “do” one piece of software and do it well? What kind of research and development money can the vendors who only concentrate in a single area expend compared to the research money available to single-system vendors? How fast will vendors “web-enable” their current applications and provide the desired functionality in that area? Which type of vendor can develop and bring new products to market faster? We felt that the vendors with modular or discrete software applications could develop and maintain better applications faster.
These systems are all true, windows-based systems and because of that we gave them high marks for ease-of-use. These systems look and feel like the personal productivity applications they already know and to a valuable extent look and feel like each other. Because these systems are developed with windows-based, “off-the-shelf” packages like Visual Basic, PowerBuilder, Microsoft SQL Server, Watcom SQL, etc. they are easier to maintain (one can, for example, go to a local bookstore like Barnes and Noble and get books on how to manage SQL Server databases or write add-on applications in Visual Basic).

From a technical standpoint, these systems are easy to configure, maintain, and backup. Configuring and managing a database engine like Microsoft SQL Server is relatively straightforward as compared to old-style databases. Set up and configuration is, again, windows-based (for example, Microsoft SQL Server 6.5 contains a “wizard” to aid in administering, configuring, and executing automatic backup and database consistency procedures). Backup agents exist from off-the-shelf backup software packages like Arcada’s BackupExec to automatically schedule and backup distributed databases. Data dictionaries are often available on-line, inside the software package itself. If you want to see what fields compose a table, or what a table’s keys are, that information is often available directly off the menubar. Information in these systems is incredibly easy to access using standard query tools. Since these databases are standards-based and are popular technically, any standard query tool (Microsoft’s MS Query, Access, and Visual Basic, Crystal Reports, R&R Report Writer, PowerBuilder, etc.) can easily access the distributed data.

If you make a bad choice in a given functional area (remember you’re picking the best system at that point in time) or if another vendor releases a “killer app” in a given functional area, the college does not have to migrate all its systems to remedy the bad choice or take advantage of a great new application. All that needs to change is the software (and training) in that functional area and the integration routines that tie it to the other systems. Since the integration points have already been defined at a process level, redoing the links at the technical level is straightforward. This same logic applies if a vendor goes out of business (that never happens, does it?). You only have to make a major change in the functional area affected.

These systems need less ongoing maintenance/staff support than more monolithic systems. We estimated that we would need AT LEAST one more highly technical staff position to accommodate some of the bigger systems due to the technical nature of those systems and our unfamiliarity in-house with them. The fact that users can use their favorite query tool to generate reports relieves pressure on Computer Services staff while bringing the user close to their own data.

Challenges

Each office is indeed free to choose the system they desire. Of course, given the technical need to integrate diverse systems, there are overarching technical guidelines to which each office must subscribe. For example, in our project, we wanted to standardize as much as possible on systems that were written in open, “off-the-shelf” development packages (such as PowerBuilder) and which accessed open, off-the-shelf database engines (such as Microsoft SQL Server).

Consequently, each office was not absolutely free to pick the system they wished. This turned out to be a challenge in areas where it was arguable which system was the best functionally, but where one of the systems better met our technical criteria. What if an office does not like ANY system?
Functional-area vendors may not know the higher education environment. In many ways, a general ledger is a general ledger, but is that general ledger flexible enough to accommodate the specific needs of an educational institution? As always, caveat emptor. Know what things you need a system to do and CHECK that the system which interests you does them. Assume nothing, especially with vendors who do not traditionally sell to higher education.

You may have to deal with installation, training, and support of a discrete system through a third-party reseller. Since the software vendor may not be your direct contact for installation and support services, this adds another layer to the process of getting questions and issues resolved. Questions may arise. For example, is the software bad or did the reseller do a bad job installing it? Did the software vendor promise functionality to the reseller who then promised it to you, or did the reseller promise you something without the knowledge of the software vendor in order to sell you the product? In our estimation, it is best to avoid working through a third-party reseller unless you absolutely have to or unless you really know the reseller’s history, staff, and capabilities for supporting you.
Implementing a “best-of-class” client/server systems strategy

“…when a small college investigates a new technology, organizational structure, or fiscal strategy, it enables other colleges to evaluate the results without putting themselves at risk. In the words of a familiar truism, ‘it’s a dirty job…but somebody’s got to do it.’”

We are “doing it” and in some ways it is a “dirty job.” We have learned a lot from our efforts and this more colloquial section of the paper will discuss what we have learned, some challenges yet to be met, and some potential “gotchas.”

Distributing systems requires distributing knowledge

First and foremost, distributing systems requires distributing knowledge. All system conversions require that staff members embrace change. With these systems, you must hug the life out of change. You can NOT train enough. Users moving to new systems must not only learn their new business applications, they must learn how to work in a distributed environment. For example, how do you reply to your controller when she emails you with the following question: “The general ledger is $87.02 out of balance. Please look into it and let me know when you have fixed it. If there’s anything I can do to help…”? This may have been a very appropriate question in a legacy system (since the users may not have been able to easily access their data, let alone dig deeply into it where needed to fix problems). Actually, it still is a very appropriate question to ask if the controller has not assumed “ownership” of that system. What happens when multiple users in an office have to manage their office’s new networked laser printer? Before, their jobs were printed, collated, burst, and distributed to them by a computer center operator. Now those users have to agree which printer tray holds special forms, which holds regular white paper, AND they must be trained to manage their own print jobs and redirect printed output. To the end user learning how to perform these kinds of tasks, “simple” items like this can make the difference in how they perceive the quality of their new business application. These new systems bring new end-user responsibilities, ones that certainly can be empowering. To many end-users, they can be just plain frustrating. Training users to problem-solve for themselves must be a priority. Find a user in each office who wants to “own” the new application and be an application problem-solver / office trainer / technical go-between. Users are now in control, and computer services staffs must help them in realizing it.

Move the data as “close” to the end user as possible

We fully believe that users will feel more powerful if we can put the data they need “closer” to them. To that end, we have physically placed workgroup servers within the major offices. This much more than symbolic gesture will worry most I.S. managers, but most of these systems are designed to be managed in a distributed fashion. If you can just keep the users’ children in those offices from pushing the ‘power’ button on the weekend (which did indeed happen to us -- with no damage) you will be all set.

Timing is everything

Timing is everything. If you pick any best-of-class system too soon in its development cycle you will not be able to wait for the vendor to finish incorporating the certainly promised functionality that enticed you to buy the software in the first place. You also may find that you are educating
vendors about higher education practices in the hopes they will get it right in time for you to meet your major milestones. Yet, with a lot of newly (re)designed client/server systems coming to market now or in the very near future, I believe many colleges will find the risk in undertaking a “best-of-class” strategy to be less than the risk we assumed. I do not believe we would have been successful starting our project a year earlier than we did. It’s also worth noting that there may/will be a better system out there a year from now, but if it will not be ready when you need it, it’s no good to you. If you pick the right vendor, you can negotiate a migration strategy to the platform you desire while using their current architecture.
Do your homework

Perform your “due diligence” on each system you intend to purchase and on the vendors themselves. Make sure that the software does what you need it to do. See it demonstrated on the platform you are choosing! For example, let’s say you like a certain application that uses a proprietary database engine. The vendor knows you want to buy a system that uses a different database engine, and tells you the front end is database independent. See the system work on the database platform you want before you buy it. Given that advice, you still must pick the best system you can find at the time under a “best-of-class” strategy. If you end up picking a bad application, you can replace it alone (of course, minimizing the risk by discussing and budgeting up-front for the possibility of making a bad choice is wise).

Don’t sweat the technical stuff

From a strictly technical viewpoint, integrating these various systems is relatively easy. The most asked question I have had about our project is something like: “Isn’t integrating all of those systems going to be a nightmare?” or something like that. Actually, it is not. For example we have moved data to and from an IBM DB/400 database, a Foxpro for Windows database, and a Microsoft SQL Server database using Access and PowerBuilder. With clear data dictionaries, knowledge of the systems to be integrated, a talented staff member or two, and some helpful technical contacts at the vendors, we have been very successful in integrating our various systems. Obviously, there are and will be work process issues to overcome as we tie all of these systems together. We must decide how best to propagate name and address changes, among other data items, through the various systems. Yet there always seems to be a good technical solution and we have learned to pick the best one available to us at the time.

GUI-based systems are a different animal

PC/GUI-based systems have their own unique requirements. Clients and servers for these systems eat RAM for lunch. Do not underpower your PCs or your servers, no matter the client/server environment. As for user PCs, we WANT our users “in” an email package, in their particular business application, in Word, maybe in Access as well while they are working. Sixteen megabytes of RAM in a 486/66 PC will not cut it. Start at a Pentium 150 with 32 MB of RAM for your power users and make sure everyone in the offices has at least a Pentium 75 with 32 MB of RAM (remember the PC fleet replacement plan?). In our experience, if your campus is Macintosh-oriented you will have trouble finding systems in all areas to support Mac clients. Also, GUI-based systems that we have tested have an incredible capability to store the most granular items of information for tracking. Is this a “good” thing -- or a “bad” thing? Make sure there is a batch data entry screen or subset of screens/tabs where your back-office staff can quickly and efficiently capture all that new information. Staff who are used to “old-style” terminal entry screens do NOT want to move the mouse around while they are entering a large batch of information; they want to keep their fingers on the keyboard for speed. As for PC servers, we have purchased clone Pentium-based dual and single processor servers with 128 MB of RAM and 10 Gigabytes of RAID5 SCSI storage for under $10K. Be sure if you do not purchase a name-brand server that you test the configuration well in advance. We run Windows NT Server on all our application servers and you never know when a hardware device or some bad RAM may cause unforeseen difficulties for the operating system. Once we settled on a working configuration, these clone systems have worked extremely well for us at a significant hardware cost saving.
No customizations for one year

Our selection committee agreed that NO major customizations would be attempted on any of these systems for at least one year after installation (and hopefully never). We want the vendor updating these systems and we want to avoid paying for costly customizations that changes in expectations or work processes may more easily solve.

Prepare for staff changes

Prepare for staff changes in Computer Services. No matter how good your legacy system administrators/analysts are, they may not want to learn the new skills necessary for them to do their new jobs. What may have once been a behind-closed-doors, back-office style of user support (users call in with a request and the analyst programs it when the schedule permits) now becomes more up close and user-focused. Build a staff of mixed skills who do not mind getting their fingers dirty in these new systems. There is no room for territory battles in a “best-of-class” strategy. Everyone must help everyone else do everything. Pay to re-train those who wish to take on a project like this. It always helps to recognize that a vendor may have trouble finding and retaining quality staff with the skills necessary to implement these systems (like you may be). Would one of your vendors outsource programming to a third party? Under what conditions?

If it’s Tuesday, we must need to call ‘Vendor A’

Vendor support is always an issue, and it is even more of an issue in a “best-of-class” strategy. We are dealing with six vendors in supporting eight different software packages. Now that we have distributed these applications to the various offices and are encouraging end users to “own” their particular applications, we must ask them to call in problems directly to their vendor as well. Again, more responsibility is handed to the end user. They have to practice the new skill of asking vendors why systems do not work the way they think they should, and in brainstorming possible work-arounds. We have only one full time computer services staff member devoted to supporting users of all our systems. We want the end users to eventually grow into solving their own problems.

If at all possible, deal directly with the software vendor. These systems are new to us, new to the vendor, and are certainly new to third-party system integrators who may be a software application’s reseller. For example, Great Plains sells only through a third party reseller. We have found that our staff ends up learning new systems faster than the reseller’s staff (who may be busy selling and supporting many other applications). Make sure you have direct access to the software vendor’s technical support staff.

Otherwise good technical guidelines may limit system choice

The overarching technical guidelines you apply in choosing systems may limit your choices. For example, an office may want a particular application. If that application does not “fit” with your technology direction, an office may not be getting what they really want, which undercuts a major benefit of a “best-of-class” strategy: departmental choice. Even worse, one particular office may not like any of the systems available. This may not be as big an issue currently or in the near future due to the development and release of a wider variety of client/server systems.
Conclusion

As our college moves further along its implementation cycle, I become more and more convinced that the strategic direction we took was indeed the correct one for our college. That is, of course, an easy statement to write, given its self-justifying nature. Nevertheless, I am continually surprised at how “nice” it is for users to have great flexibility in these systems, for users to be “closer” to their data, for users to operate in a fully Windows-based environment, for users to pick the system they want and be given the opportunity to “own” their system, for users to print to standard, networked laser printers, and for me to see how quickly new tools are developed which help us integrate these systems and which mesh well with our standards and needs. These systems are slick and are maturing quickly. Technology-oriented staff members will never be bored implementing these systems, and with the right staff that means an exciting, fun-filled office full of people learning as fast as they can to stay current. Technical staff will want to work with these new tools for higher-education wages because they are new and “hot,” accessible and fun.

We will know a LOT more in a year or two as we meet the challenges of fully integrating and working in a production environment with all systems running concurrently. Farther down the road there is the opportunity to radically improve work processes, using these new systems as levers for improvements. For now, we are doing the dirty work and finding a wealth of challenges, surprises, and rewards.
2 Randolph-Macon Woman’s College is a private, liberal arts institution located in the foothills of the Blue Ridge Mountains enrolling 732 students.
3 I prefer the term “campus software system” to “administrative software system” given that these systems are increasingly becoming easily accessible for use by faculty and students, not just the administration.
5 I would also like to thank Richard Kesner and Rick Mickool at Babson College, who did a lot of the dirty work prior to us, and who were kind enough to share their experiences with us.