



Treasury and IT Integration in Plain English

Part I

Background and Vocabulary

by
Richard J. Poje, President
R. J. Poje and Company

Introduction

This is a true story about a consulting project that took place in the early 1990's. The client was a Federal government agency that employed 18 clerks to key-enter banking-related data into its general ledger system. The clerks, each earning about \$34,000 a year, were not very productive; error rates were high and re-work consumed much of their day.

Agency management decided to correct the problem by hiring 18 additional clerks who key-verified the first group's work. The second group also earned, on average, \$34,000 a year but did not solve the problem. It only produced different results, and management was unable to determine which were correct. In its wisdom, management decided that "integration" was the answer and hired yet another group of eighteen clerks to reconcile the work of groups one and two. Our firm was later engaged to figure out why data entry costs tripled in less than twelve months.

Defining the solution proved easier than implementing it. The underlying data had been transmitted in machine-readable form, then printed and prepared for key entry. There were several treasury workstation products available with functionality to reformat the information for automated input, eliminating all key entry requirements and any need for 54 high-priced clerks. The conundrum was the clerks were unionized, and once hired, work rules prohibited the elimination of their jobs because of automation. The agency finally implemented the workstation, but redeploying the clerks took more than two years and cost nearly four million dollars.

The moral of this story? "Integration" is a word with many meanings. Dictionaries define it as "a combination and coordination of separate, diverse elements into a complete, harmonious whole." When applied to computers and

information processing, it often implies multiple, complex processes and outcomes, many with unforeseeable and costly consequences.

"Integration" is the IT buzzword of the new millennium, but it is neither a panacea nor a slam-dunk solution. The sad fact is an alarming percentage of integration projects fail, at least in the minds of senior managers intent on reducing expenses or improving the quality and timeliness of corporate information. Typically, the broader or more ambitious the project objective, the more likely its outcome will disappoint. This may result from unrealistic expectations, the relative immaturity of integration-specific development protocols or flaws in the implementation process, probably a combination of all three.

Disappointment, however, is not inevitable if objectives and expectations are realistic, situation-appropriate tools are employed (tools and processes have improved significantly in recent years) and projects are well managed. The series of white papers beginning here will explore why so many integration projects fail and how they can succeed. This first paper explores the vocabulary of IT integration, examining objectives, technology approaches and common reasons for success or failure. The second paper will focus on the corporate treasury function, the challenges of integrating diverse treasury-specific information sources and those of melding single-purpose software packages into unified "systems." It will also focus on getting the most out of software procurement processes and managing implementation efforts. The third and final paper will focus on company-wide integration initiatives and their impact on treasury, especially when treasury is a team participant, not the project's leader.





**Payment Solutions to Grow...
Not Just Survive**

Integration Objectives

IT integration is about making information more valuable and proactively reducing IT-related costs and risks. Broadly, it comes in four flavors: the first focused on process automation, the second on process transformation, and each having either "ad hoc" or "strategic" characteristics (Figure 1). The difference is more than semantic for executives charged with the success of major IT integration projects. Process automation can be complicated and costly, but its goals are well understood with "best practices" and performance metrics to guide implementation. Process transformation can be even more costly and complex,

but we lack the experience to know how much so because it is the new, new thing. Goals remain fuzzy, "best practices" scarce. There is a body of evidence, moreover, that line managers are at least as skeptical about transformation as their antecedents had been about automation, especially when conversations turn to strategic decisions about high visibility applications like Customer Relationship Management (CRM), Supply Chain Management (SCM)

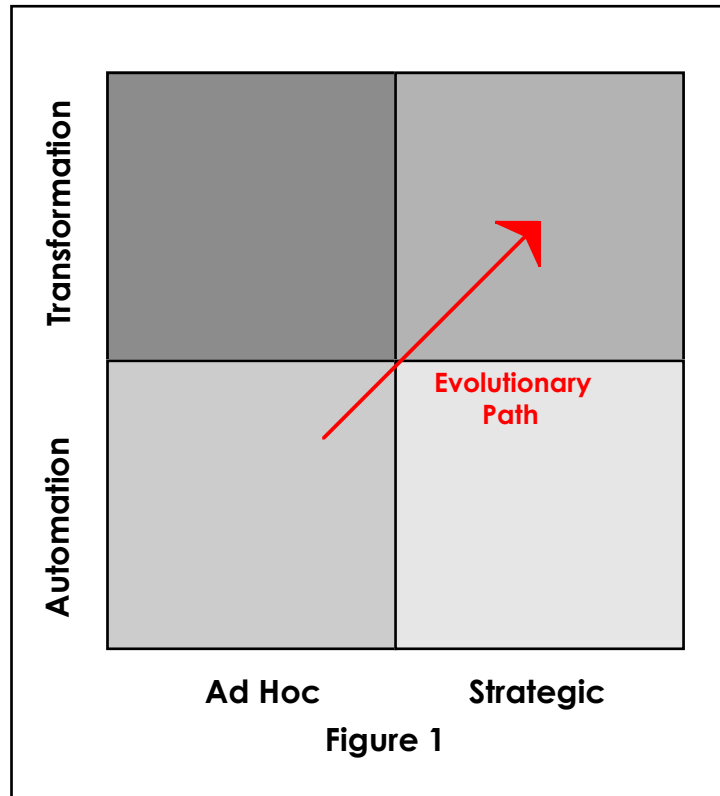
and Enterprise Resource Planning (ERP). It is important that integration managers understand the different challenges involved with automation/transformation as well as the implications of ad hoc/strategic solutions and direct their efforts appropriately.

Process automation has been an IT objective for centuries. Leonardo DaVinci contemplated it in the fifteenth century and Blaise Pascal demonstrated it in the seventeenth. As a practical matter the IT community has vigorously pursued process automation since the introduction of "3rd generation" computers and operating systems in the mid-1960s and the emergence of desktop computing in the late-1970's. Early process automation dealt with discrete "tasks"

like summing numbers read from key-punch cards. As technology permitted, tasks were integrated into "programs," programs into "jobs," and jobs into "systems" or "applications," each a combination of automated and manual work steps. The focus was almost always on using computers to replicate existing processes and perform them better, faster, cheaper, not on challenging or changing processes in more substantial ways. Automating residual manual work steps ultimately became a priority, but that has been a rela-

tively recent event. Computers, operating systems, data storage and communications capable of achieving complete process automation were either unavailable or not cost effective for many situations until recently.

As IT proliferated, more complete automation became a requirement for sustaining application growth. Older "apps" needed to "feed" and "be fed" by newer ones and man-





ual work steps, unnoticed when mainframe turnaround was typically 24 or 48 hours, became disruptive in a world of desktop processing and instant answers. New systems were written with automation in mind. Application bridging, that is, getting otherwise isolated systems to communicate with one another, became a critical IT goal; likewise, building bridges "straight-through" without manual work steps. The underlying business objective may have been getting different departments or divisions "singing off the same page" by sharing information about customers, vendors and corporate operations. It may have involved building global linkages to facilitate 24X7 operations or incorporating technologies beyond imagination only a few years earlier. The ultimate driver, however, remained largely unchanged, and that was the automation of existing work processes.

The idea that IT could add value beyond process automation had been acknowledged in academic and military circles as early as World War II, but the gap between theory and reality was huge. IBM, for instance, conceived a cradle-to-grave, fully "integrated" system for manufacturing control in 1968. It was called COPICS and while an elegant conceptual solution to a string of messy problems, implementation would have required 27 of the largest mainframes then being built attached to nine football fields of disk drives. The project was shelved. There were similar outcomes with early transformation experiments in other industries, among them, the Chicago bank that sunk \$50 Million into an integrated database project the sole productive result of which was publication of the daily cafeteria menu. The fact of the matter was technology was not good enough to enable automation, muchless transformation, in the 1960's, or for that matter, in the '70's and '80's either; and even had it been good enough, the business community lacked sufficient experience with computers and automation to effectively manage change at that level. Transformation was more a cultural event than a technological event and culture changes more slowly than the speed of microchips.

Culture really did not begin changing the advent of Y2K when necessity proved itself the mother of invention. Of course, change was facilitated by decades of successful automation projects, enormous advances on almost every technology area, not least of which involved the Internet, and productive results from corporate reengineering efforts focused for the most part on organizational transformation and human work products.

Process transformation brought the prospect of IT reaching an entirely new level and defined new challenges for technology implementers. It pointed to non-business achievements like the "smart" weaponry that played a decisive role in the Gulf War, "smart mathematics" that facilitated the birth of derivatives markets and "smart" communications technologies that brought the Internet into half of all American homes. More important, it at least whispered the possibility computers would no longer be just tools to facilitate problem solving but fundamental components of problem solving processes themselves. The IT community's rush to adopt process transformation had an unfortunate side effect, however, and that was its love affair with "big" solutions. Times were heady, after all. IT staffs and budgets ballooned with Y2K, CEO's and CFO's became inured to eight-figure price tags on vendor proposals, and no one wanted to be left behind as the technology juggernaut rolled across the economy.

Enterprise Resource Planning systems, ERPs, became popular by promising to solve Y2K problems, and having been written from scratch with Y2K in mind, they achieved that goal for the most part. They also promised process transformation with functionality to eliminate organizational silos, data incompatibilities and other integration barriers. ERP ideas were not new. They were remarkably similar to IBM's COPICS and other centralized database concepts that had been discussed for years (Figures 2 and 3), though technology had improved and implementation no longer required 27 mainframe computers and football fields full of disk drives. Still, process transformation was a promise on which ERP's never really delivered for two reasons:

First, true ERP implementation demanded more change than many corporate cultures, still focused on process automation, were prepared to accept. New software was either modified to retain the "look and feel" of what it had replaced, absent Y2K problems, or "legacy" applications were retained, and retrofitted with enough ERP functional-



Payment Solutions to Grow...
Not Just Survive

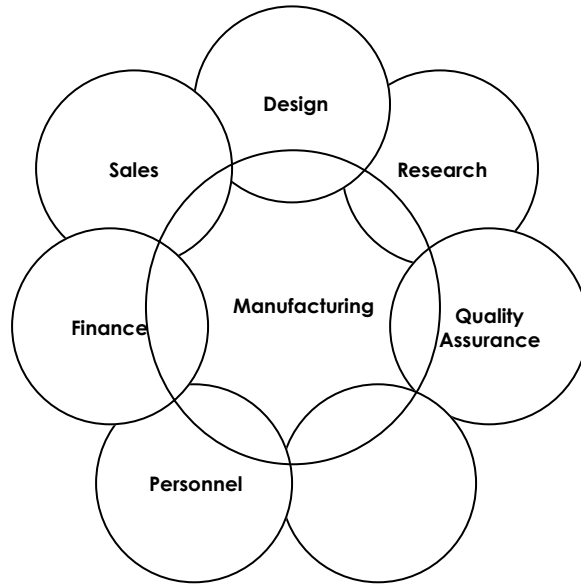


Figure 2

COPICS

IBM Corporation

Communications Oriented Production Information and Control System
Volume I, Management Overview: 1974



Figure 3

Enterprise Resource Planning System Structure (Example)



ity to declare implementation a success. IT managers who justified ERP business cases on achieving process transformation were disappointed and often replaced when CEO's and CFO's realized they had gotten less than they had paid for, or thought they had paid for.

Second, ERP modules, early versions in particular, were inconsistent in quality and functionality. For instance, software originally developed to support manufacturing operations might have world-class cost accounting and inventory control functionality but weak general ledger or HR support. As a result, it was not unusual for companies to pick and choose modules for implementation, sometimes even buying from multiple ERP vendors to meet their needs. Inevitably, a lot of "legacy" software was retained to fill the gaps. This was particularly true with treasury applications. ERP vendors addressed treasury requirements relatively late in the game. Initial offerings reflected little understanding of the treasury function, and treasury was usually low man on the totem pole when it came to implementation. Treasurers in many cases became the rugged individualists of the integration movement, forced by circumstance to find their own solutions.

As things turned out, even when their Enterprise Resource Planning implementations were behind them, companies continued operated in silos. It was possible to move data from silo to silo, but you still had to know it was there in the first place; you had to be certain it was available in a meaningful form, and access required permission. If anything, integration became more difficult with ERPs in the picture because they added layers of complexity to corporate data management, another "thing" requiring interfaces and maintenance, one more place to reinterpret and/or replicate information. After the Y2K dust settled, many corporations realized, seven and eight figure investments notwithstanding, they were still running sixty days behind in closing their monthly books, they knew little more about customer behavior than they had a decade

earlier, and IT had failed to deliver streamlined business processes, more efficient supply chain management and lower operating costs.

Integration Comes of Age?

Though ERPs were initially disappointing as transformation agents, the notion of computers and IT becoming fundamental components of problem solving still gained currency with many intriguing and highly useful "proofs of concept." Among them, treasury workstations demonstrated transformation when they began reformatting bank data into general ledger transactions, as our government agency belatedly figured out (Figure 4). We learned about "straight-through" transformation when antivirus software on our home computers began autodialing the Internet to update virus definitions and American teenagers discovered they could transform music "ripped" from CD's or over the Internet by accessing services like www.ccdb.com for instant delivery of information about recording artists, song and album names.

New solutions for transformation began emerging in the business world, some focused on specific functional objectives, others on more generic goals.

Customer Relationship Management (CRM), and Supply Chain Management (SCM) tools had specific outcomes in mind, the former centralizing information about customers, their buying behavior and servicing requirements across business and organizational lines, the latter on streamlining procurement, production and related processes.

Other solutions focused on improving workflows, optimizing business processes and "pure" integration, i.e. "[tying] together a company's back-end systems and ...[enabling] systems to be added with minimal customization" ([Information Week](#), January 6, 2003). Still others focused on "cleansing" data resident in disparate systems before loading into transformation solutions, tacit acknowledgment that "garbage in – garbage out" remained a major obstacle. All solutions have four things in common:

- (1) They add one or more "layers" of code to achieve process transformation, in effect, using automation to circumvent silos, data incompatibilities and other integration barriers, but creating new implementation/maintenance challenges as they did so (a simplified example appears in Figure 5).



Payment Solutions to Grow...
Not Just Survive

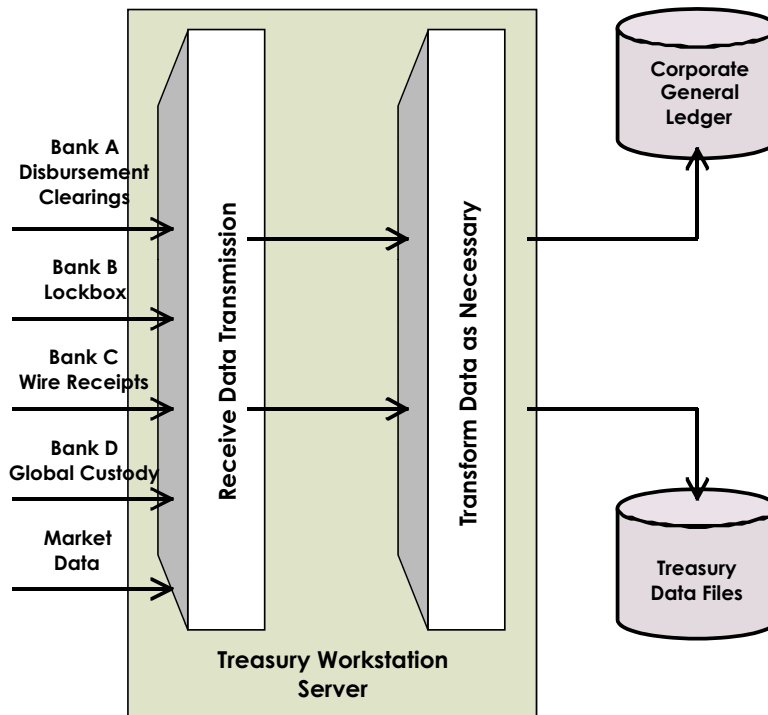


Figure 4
Treasury Workstation as Integrator

- (2) They are expensive. According to *Information Week*, integrators can expect to pay \$250,000 to \$600,000 in software licensing fees (other sources suggest as much as \$250,000 per server for software licensing), 18% - 25% of licensing fees a year for maintenance and 150% - 400% of licensing fees for vendor support. Waste Management Inc. recently reported its IT "transformation" project will cost more than \$400 million (*Information Week*, December 2, 2002).
- (3) Solutions are morphing faster than retroviruses. Application-server platform vendors like IBM and Microsoft are adding integration functionality; ERP providers like Peoplesoft are adding CRM or SCM; workflow software providers like FileNet, Identitech and Staffware are adding integration and "business process management" functionality, etc. *ad nauseum*. About the only thing that is certain is that tomorrow's solution will be different from today's solution and further development will be required.

One area of particularly significant change has been the integration community's recent move from server-based solutions to "web services" as the newest new thing. Collectively described as "Enterprise Application Integration" or EAI, server-based solutions have proven expensive, both in terms of implementation and maintenance. "Web services," a term used to describe a variety of technology approaches, promise faster, cheaper implementation and greater operational flexibility by replacing physical with "virtual" (i.e. web-based) servers. The idea is intriguing but we have even less experience with "web services" than we have with EAI.

- (4) Outcomes remain unclear and transformation success elusive. Some surveys report half or more of CEO's and CFO's who have sponsored process transformation projects are dissatisfied with results, and at least one survey found 74% of senior



**Payment Solutions to Grow...
Not Just Survive**

managers unclear about the expected returns from process transformation efforts.

CIO's should not be deterred in developing business cases for process transformation, but they should be cautious, challenging plans and assumptions on all fronts, starting with whether or not transformation should even be their objective of choice. They should also be getting their hands around factors that are precipitating success and failure in transformation projects. IT may not yet have distilled the wisdom that will make transformation work, and throwing ever more sophisticated technology at the problem may not be the answer. The

answer just may lie in identifying less ambitious but more achievable objectives, low-budget experimentation and analysis to identify critical factors for success or failure, and it may lie as well in having the patience to let someone else spend the big bucks sorting out technology approaches that may or may not work, may or may not survive. These are lessons anyone who has "loved and lost" in the past, particularly with programming languages like PL/1 and operating systems like IBM's OS/2, should already have learned.

Strategic vs. Ad Hoc Integration

Earlier, we mentioned that integration comes in four flavors: the first focused on process automation, the second on process transformation, each having either "ad hoc" or "strategic" characteristics. Our discussion thusfar has focused only on strategic elements, arbitrarily defined as anything that is risky, expensive or has a long implementation lead-time. Ad hoc solutions lack the elegance and sophistication of strategic systems and are unsuitable for many integration applications, but they can be effective as interim solutions or as prototyping vehicles to substantiate more aggressive plans and business cases. They can

also be vehicles for accessing the "low hanging fruit" so popular in corporate circles today.

We have been re-formatting files and writing "extract routines" for as long as we have used computers to achieve "ad-hoc" integration. We have likewise used spreadsheet macros and "linked" spreadsheets very effectively for two decades to achieve similar "near-seamless" results. Not that strategic solutions are unnecessary or without value. Large scale in-

tegration would be impossible without them. A multibank holding company would invite disaster if it attempted to integrate all its deposit systems with extract routines and spreadsheet macros, though doing so is theoretically possible. The result would be a costly nightmare for anyone charged with maintaining the end product, something one expert likened to a junkyard operator replacing his pit bull with 700 chihuahuas because the chihuahuas ate less and were easier to control.

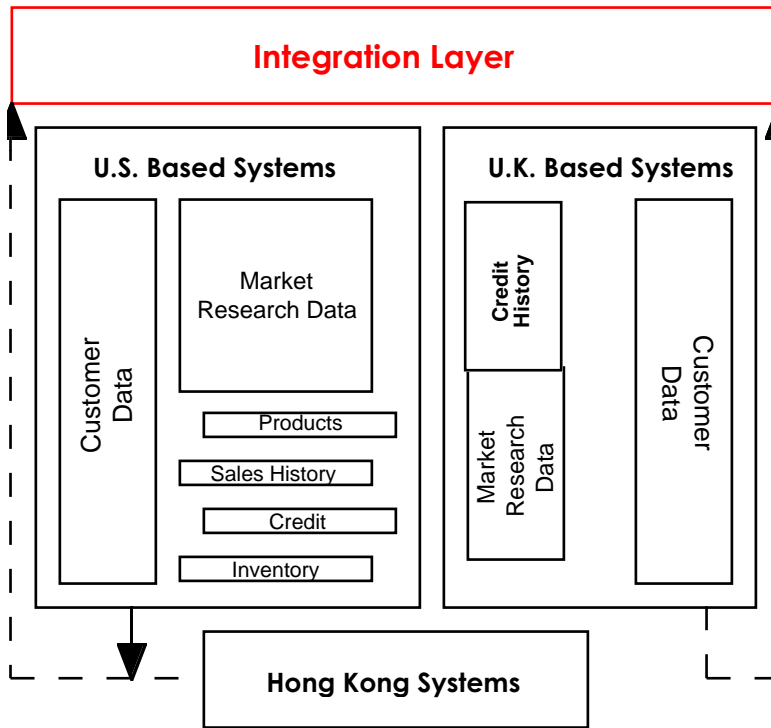


Figure 5
Integration System Structure - Simplified



One of the historic problems with IT integration projects, transformation projects in particular, has been selecting appropriate solutions: chihuahuas when pit bulls were indicated and vice-versa. Simple solutions like spreadsheet interfaces, extract/reformatting programs, even changes to job control instructions (JCL) that define mainframe computer workflows are often dismissed because they are inelegant, geared to addressing specific, not generic, integration problems and are usually implemented on "end-user" computers rather than on IT-controlled server/mainframes. There are also no web conferences or magazines dedicated to ad-hoc integration, and ad-hoc approaches seldom justify bigger staffs or budgets.

Still, "ad hoc" solutions come with much less planning overhead, require fewer development resources and cost less than strategic alternatives. Properly managed, they add value when the task at hand is eliminating a wasteful manual work step, when discrete bits of information must be gathered from disparate sources or when permanent, more generic, solutions are not achievable. Take, for instance, the situation described below.

A client firm used its lead bank's securities safekeeping services for more than thirty years. The service was not automated and involved little more than the bank periodically putting inventory reports into the mail, the client's treasury staff handling record keeping on its own network with a patchwork of purchased software and spreadsheet applications. When the lead bank abandoned its safekeeping business, the client's treasury department decided to re-think this aspect of its operation and chose another provider with more sophisticated, robust, automated custody services. Treasury anticipated the new service would change many business processes, not least of which were elimination of keypunch data entry and replacement of an undocumented, error prone interface to its general ledger system. Unfortunately, it did not anticipate the severity of the corporation's Y2K problem,

staff cuts in the IT maintenance group or problems with the new software. The result, a month before going live the new system was unable to communicate with legacy applications, including the general ledger. Even worse, anticipating the new functionality, treasury had initiated staffing and other changes that made returning to old methods cumbersome if not impossible.

The Treasurer considered a number of solution alternatives, including a lawsuit. All involved spending money that was not available and implementation delays of a year or more. The ultimate solution was a simple "ad hoc" integration process: a series of linked Excel spreadsheets were placed "between" the custodian's software and the client's legacy applications (Figure 6). The spreadsheets "deconstructed" custodial system transactions in multiple steps to produce recognizable inputs for legacy systems that had not been modified as planned. It was not a long term solution but it was written and tested in less than a week, it achieved the project's "straight-through" objective, facilitated implementation of desirable new custody functionality and avoided the need to modify poorly documented, operationally unstable legacy applications. Coincidentally, the lawsuit was never filed.

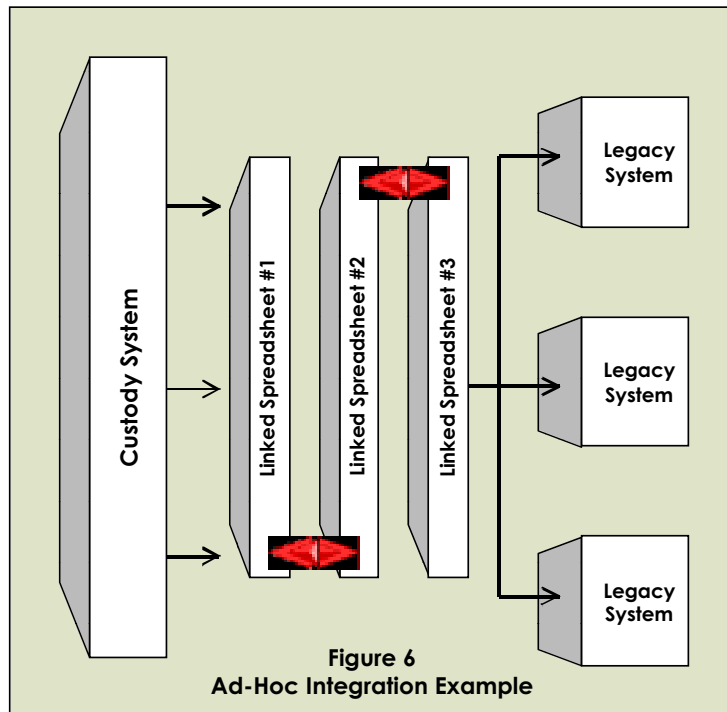
While the value and potential for "ad hoc" integration cannot be ignored, spreadsheet applications, extract programs and job control language modifications, as noted earlier, are not always appropriate solutions. Integration managers should know such approaches exist, however, and not assume every effort has to start with a million dollar software purchase. "Strategic" integration still requires investment, risk-taking and patience. It may take years to realize benefits, perhaps the reason so many CEO's and CFO's voice disappointment with recent investments in areas like CRM and SCM. Strategic integration, especially when it reaches beyond process automation, requires the right goals, the right tools and solutions, the right leadership and the right culture. Easy goals seldom justify complex solutions, but a yen for complex, state-of-the-art solutions may precipitate setting goals that are unattainable. Leaders who misunderstand this conundrum may expect, even demand, miracles but more likely will experience failure, and organizations not ready for change or unaligned with respect to how, when and where change should occur will likely subvert the very thing they hope to accomplish. *The McKinsey Quarterly* (2002 Number 4: Technology) summarized the situation in a recent article entitled "How to Rescue CRM":



**Payment Solutions to Grow...
Not Just Survive**

Treasury still has to live with the consequences of decisions made elsewhere, as many learned when ERP procurement decisions were made at high levels and treasurers were stuck with less than adequate new functionality. Treasurers need to understand how corporate decisions can impact their organizations. They need solid information to make decisions specific to treasury and to provide meaningful input when decisions are made at a corporate level.

Perhaps the most important diagnostic question is whether the executives who originally decided to implement the CRM system hoped to use it to effect more change than the organization could really support. Even with the right alignment of incentives and training - and it is rare for companies to get both right the first time - radical changes in work processes can take time and effort to gel. If the culture and practices of a company won't support its stated goals, the answer should be intuitive: scale back aspirations and build the organizational capacity needed to achieve early results.



There is absolutely no indication that integration at corporate levels has or will diminish the need for treasury specific solutions, and consequently, treasury specific IT integration. In fact, there is every indication to the contrary. Treasury

workstation software is becoming more sophisticated, with new functionality constantly being added. Major workstation providers like Sungard and Selkirk are rapidly integrating their own deliverables with those of banks, investment houses and other software providers. Additionally, a plethora of "standalone" treasury software has entered the market recently in diverse areas like risk management, performance measurement, foreign exchange trading and investments. And phenomena like "web services"

Ad Hoc, one step at a time, integration may well be the way to "scale back aspirations" and "achieve early results."

Summary and Conclusion

Our purpose in assembling this paper has been to explore the origins, challenges and vocabulary of IT integration. Treasury managers may think this information lacks relevance to their situations in a business climate where major integration decisions are made by CIO's, CFO's and CEO's. That would be a serious misconception for several reasons.

and other process transformation tools are rapidly emerging in treasury-specific contexts. In many cases, integration of "standalone" and workstation systems will be required, as will integration with corporate IT infrastructures. Decisions will also have to be made within treasury about web services, ASPs, outsourcing and similar phenomena. Treasury managers must prepare themselves to make these decisions and to understand the implications of their decision-making.

Our next two installments will examine integration from a treasury-specific perspective. Installment 2 will consider



the implications for process automation/transformation within treasury itself and between treasury and external sources, including corporate IT and the Internet. It will focus on challenges and decisions in which treasury is central and treasury personnel project leaders. Our third and final installment will examine automation and transformation when treasury is part of a broader effort like ERP implementation, CRM/SCM, or major changes in core technologies and platforms.

About the Author

Dick Poje formed his own firm, R. J. Poje and Company in 2000 to provide consulting services in the areas of payments strategy development technology integration. Previously, he had been a Partner and Director of Treasury Strategies, Inc. since 1985. Dick has been active as an entrepreneur and management consultant in the treasury and payment communities for 35 years. A co-founder of Phoenix-Hecht and its Chief Executive Officer for many years, Mr. Poje brings a unique, entrepreneurial perspective to client engagements. He advises senior managers of client organizations on strategic matters related to cash and treasury management and the payments system. He assists financial services firms in the formation of strategic alliances in various product areas, advises the Federal Reserve on its continuing role in the payment system and works with a number of private and public sector clients in re-engineering treasury and related operations.

Mr. Poje is a frequent contributor to industry publications and a respected commentator on developments in the payment system and Treasury Management. His article, "Avoiding the Pitfalls of Electronic Commerce," appeared in the January

1998 issue of the TMA Journal. In 1997 he co-authored "Electronic Commerce and Financial EDI: An Examination of Innovation in the Electronic Payments Arena". His article, "A Blueprint for Leadership: the Federal Reserve in the U.S. Payment System" was published in the American Banker. His most recent work, a series of essays on payments-related topics, are available at www.poje.com

Mr. Poje is a graduate of Fordham University and the University of Chicago and is currently writing a book about the roots of treasury innovation.

**R. J. Poje and Company
312 Ridge Road
Barrington IL 60010-9602**

**847.304.9961 (voice)
847.304.9976 (fax)
708.764.6768 (cell)
richard@poje.com**

www.poje.com