RE-ARCHITECTING AND RE-ENGINEERING TRADING AND TREASURY SYSTEMS IN THE MERGER OF CHEMICAL BANK AND MANUFACTURERS HANOVER TRUST -- An Interview with Brian Slater, Vice-President, Chemical Bank

by

Katherine A. Duliba

Robert J. Kauffman

Department of Information Systems
Stern School of Business
New York University

Working Paper Series
STERN IS-93-38

RE-ARCHITECTING AND RE-ENGINEERING

TRADING AND TREASURY SYSTEMS

IN THE MERGER OF CHEMICAL BANK

AND MANUFACTURERS HANOVER TRUST --

An Interview with Brian Slater, Vice-President, Chemical Bank

September 3, 1993

KATHERINE A. DULIBA
ROBERT J. KAUFFMAN
Stern School of Business
New York University

Forthcoming as a chapter in "Section 2 -- Trading Platform Design and Automation in the 1990s," *Information Technology and the Financial Markets: Designing and Managing Today's Trading Platform*, Robert J. Kauffman, Robert M. Mark and Bruce W. Weber (editors), Business One Irwin Publishers, Homewood, IL, 1994. The authors appreciated logistical support and access granted to the member firms of the United States Council for International Banking, obtained through its president, Dan Taylor. In addition, Rob Kauffman benefitted from financial support as an NEC Faculty Fellow from the U.S.-Japan Business and Economics Research Center, Stern School of Business, New York University, the Nippon Electric Corporation, and the Simon Graduate School of Business Administration at the University of Rochester. All errors are the responsibility of the authors.

CHAPTER

RE-ARCHITECTING AND RE-ENGINEERING TRADING AND TREASURY SYSTEMS IN THE MERGER OF CHEMICAL BANK AND MANUFACTURERS HANOVER TRUST --

AN INTERVIEW WITH BRIAN SLATER, VICE-PRESIDENT, CHEMICAL BANK

Katherine A. Duliba
Robert J. Kauffman
Stern School of Business
New York University

This chapter presents an edited transcript of an interview held in August 1993 with Brian Slater, a vice-president in the Global Bank at Chemical Bank. Slater is responsible for the firm's U.S.-based trading and global risk management systems. The interview was conducted in the context of Project 1990s -- the U.S. Council for International Banking's Study on Information Technology Investment and International Banking Performance. The purpose of the discussion was to develop background information on the issues the bank's senior management team faced in merging the trading and treasury functions of Chemical Bank and Manufacturers Hanover Trust. The new Global Bank faces challenges in the areas of global risk management, cost-effective delivery of information technology-based products, trading infrastructure application functionality gaps, and human resource management that

can be best addressed by a shift to a new paradigm for trading and treasury application software development: reuse of object-oriented representations of the bank's systems. We now turn to the details of the discussion that develops these themes.

THE MERGER FROM THE GLOBAL BANK'S PERSPECTIVE

Q. Good morning! We would like to begin by asking you about the big picture. What were the reasons for the merger between Chemical Bank and Manufacturers Hanover Trust (**MHT**)?

A. The merger occurred during a time of wide-spread consolidation in the financial services industry. Both organizations were weakened by the credit crises in Latin America and in the real estate sector, so there was the impetus to strengthen the capital base of each firm. In addition, both organizations had gone through four or five years of significant down-sizing. By creating a single stronger, healthier firm, we would be able to compete even more effectively. Both banks wanted to become more significant players in the global capital and credit markets.

Q. When did the merger occur?

A. The intent to merge was announced in July 1991, along with the plans to merge operations by January 1, 1992. These dates were a significant challenge within the Global Bank. In the trading environment you can't go into the market with two faces, you know. So time was very critical. Chemical and MHT would need to trade as one unified operation. Doing it separately would have been confusing for both, especially in view of how large each is, I would think, but also you are not going to obtain any efficiency, scale-size or risk management benefits if you don't bring things together.

The result was that we had six months in which to go from 0 to 100 miles an hour, while managing to keep things under control!

Q. What were some of the other initial challenges that you had to deal with when you contemplated the consolidation of Chemical and MHT's trading operations into the new Global Bank? Did you have different businesses? Different products? Different technologies? Different budgets? Different risk management approaches?

Different attitudes on the part of your senior management teams?

A. All of the above actually, so let's consider them in sequence.

I think that between the two organizations there were places where there was substantial overlap. But in other places there was no match at all; the merged organization would reflect the strengths of two quite different organizations. We began by thinking about the merger as a "1 + 1 = 1.3" proposition. It would take a lot of work to eliminate the inefficiencies of both trading functions, but each would gain.

But, in fact, the merger created an organization that was greater than the sum of its parts. In some businesses we found that we even ended up with "1 + 1 = 2.5", which was outside of our expectations by a considerable degree.

By January 3, 1992, the changes were dramatic. In some businesses we went from what we expected to be a low-volume, low-intensity operation, after adding two low-volume, low-intensity businesses together, to something that went completely the opposite way -- to something that became high-intensity, high-volume.

Where the strengths of the businesses in the preceding organizations were in different places, things did not change as rapidly. Take derivative trading and foreign exchange, for example. MHT's strengths in derivatives far exceeded Chemical Bank's, whereas Chemical's strength in foreign exchange surpassed MHT's.

But there were also certain places where there was no clear match in the operational infrastructure, though there were similarities in the businesses. You couldn't plug two systems together and go from there.

In other cases where there was a more direct match, the questions then were: Which way would we approach the problem -- from the Chemical or the MHT perspective? Would we be able to support the merged business from both an operations point of view and from a systems point of view? And what kinds of personnel issues were going to be associated with that?

A major effort was devoted towards working out how we would approach each of the problems. But we were also interested in moving the pendulum of technology investment back in a direction that more closely matched the strategic goals of the firm. The years following the crash had been hard on banks, and Chemical and MHT were no exceptions. We looked forward to a big swing of the pendulum -- from a maintenance-oriented organization back to a development-oriented one -- one that senior management would agree was appropriate.

Our challenge now is to return from this state to one that can provide a platform upon which the merged bank can move forward. As with any company, the business has to be the driving factor behind what we are doing today in technology.

MANAGEMENT APPROACHES AND TECHNOLOGICAL CHOICES

Q. What approach did the Global Bank's senior management team take to make the most of the merger?

There are many ways to accomplish a merger, but I think of three classic forms. There's the *pure merger model* where you spend a lot of time studying the merging organizations to see if you can create a unified front out of the pieces. Then there's the *acquisition value model*, which involves selecting those pieces that are the best, and discarding everything else. In other cases an acquisition gets done with systems and operational changes that are mandated across the board: everything just gets turned over. This *clean sweep model* was used when the Bank of America acquired Security Pacific. Even though we call it a merger, of course, what really happened is that B of A acquired Security Pacific. The systems and operations were switched over to the Bank of America standards within days of the decision being made, whether it was the best or not.

Chemical Bank used the first two viewpoints with respect to the Global Bank's operations. The acquisition value model was especially useful for us. If a system provided a strength in the business which was predominant for one of the organizations, then it was hard to justify doing anything else but keep it.

Q. There must have been some tough calls that had to be made. Personalities, different opinions of what was best for the bank, and different views of how well the system would either scale up or integrate with the other organization's application architecture would all play a role. Did you have any outside help when it came to trying to sift through all the systems and work out a solution that would fly in terms of the politics involved?

A. No, outside help was very limited. Without being too rude about the consulting business, that would probably have made sure that the Global Bank's merger process would not have been successful. You bring in consultants for several reasons, but we felt that consultants would not add any value here. We had to make some tough decisions and the forces that supported making them overrode some of the political issues that naturally arose. Plus, there was so little time.

Q. At what level were decisions made about systems that would support the businesses that made up the new Global Bank?

A. About two to three months after the merger announcement occurred, there were also announcements about the organizational structure of the merged Global Bank. That made the process a lot easier.

By removing the uncertainty in terms of the organizational structure, and giving people the power to make decisions, then the decisions can be made. So, it wasn't too long after that we reached a position where we were able to say: OK, we've selected these systems. These people are who are going to run the operations. And this is where the operations are going to be located.

Q. Can you give us a frame of reference for the size of your systems operations, the information technology (**IT**) budget and how it has changed since the merger?

A. Let's see ... Which numbers would you like me to use?

I really can't divulge the amount of money that we used to spend or that we currently spend on IT, but I can give you an idea of how large we are in terms of staff -- project management, planning, development and maintenance -- and how our budget is changing.

We have 120 people involved in trading and risk management systems in the Global Bank, and if we don't include the management structure, then the number is less than 100. And that is still too big.

Our total IT budget has probably gone up over what it was before. That's because the world that we support has expanded. There are more branches in Global Banking than there were before in Chemical, so again, I'm measuring against what Chemical's original investment was.

Even with the growth in the Global Bank's budget, the business is much larger, we're supporting more business units, and we're in more locations. Overall, we've done fairly well to keep expense growth low. We have obtained significant merger-driven savings.

THE OLD WORLD AND THE NEW WORLD OF GLOBAL BANK SYSTEMS

Q. Can you think of some adjectives to describe the old world systems at Chemical and MHT? For example, what were the technical qualities, the level of integration, and how did the systems at the two firms differ? What were the software development methods used to deliver them?

A. Good questions.

From a technology standpoint, in certain places the two banks had very strong overlap. They both made use of AS/400 and DEC's VAX hardware in international branch operations. We were a very close match. On the domestic side though, we were mostly different: MHT used AS/400s in places where MHT used VAXes.

As I mentioned earlier, down-sizing during the years following the crash shifted the focus of IT spending within the Global Bank from development to maintenance. In some cases perhaps, we had moved to a level of minimum maintenance. Chemical's approach to strategic systems goes all the way back to 1983, when the bank formulated a plan to support the its core businesses. Major investments during the middle years in the 1980s produced quite a bit of synergy in the bank's systems, but thereafter we found it hard to keep on track, in view of all the other pressures.

MHT's systems needed fresh investment. Because of the maintenance perspective that had been adopted, both banks' system portfolios had reached a point where they were delivering a level of functional support that was less than was needed in the business.

Think about the way IT executives go about selecting packaged solutions. My rule of thumb is this: If the package delivers 70% or more of my requirements, then a packaged solution is a good answer. When your own systems fall below 70% of the functionality you think you need to do business, you've got an issue to deal with there. You've got to decide whether you're going to invest to get them above that level or whether it's worthwhile to restructure the entire business process. New investments will need to be made to bring the "fit" back between the systems and the business.

Q. How did you go about gauging the extent of the functionality that you did have? Your 70% rule is an interesting one, but how would a person be able to look at the application inventory and figure that out? Wouldn't you need to do it at a level of the organization that was higher than an application system?

A. Here's how we did it. Similar operating groups in each of the banks conducted independent reviews of the businesses its systems supported, the complexity of the operating environment, and how the business would be affected when operations were merged. We hoped to identify the constituents of the systems portfolio that would best support the merged operation.

In certain cases, it was a "no-brainer". For example, if you combined the banks' derivatives sales and trading operations, then it was quite clear that the systems environment that supported MHT's derivatives business was going to be taken forward. There really wasn't anything on the other side of the fence. Likewise, for foreign exchange: Chemical Bank had scale. You could add MHT's business to it, and it wouldn't make any difference. In other cases, we tried to identify the key systems requirements of the business. The system that better supported the new business requirements of the merged Global Bank would be the one that survived.

You'll be interested to know that on the international side we ended up choosing the oldest system that was implemented. Even though it offered us less functionality, it had features that covered more businesses than our other alternatives. It was the only way we really could approach it.

Q. You earlier indicated that time was tight. Can you share some of the specifics with us?

A. All those decisions were made and completed by January 1, 1992. Keep in mind that we operated with two important dates related to the merger. There was the need to have the two organizations trading as one entity, and then there was the financial merger of the organization which came later, in the middle of 1992. The systems that handled the trading function's books had to be done by January 3. Those that handled the financials of the merged bank had to be done by July 18, 1992.

By January 1, we had chosen all of the systems that we would take forward. They were converted and consolidated too. This enabled us to manage risk through one system on a consistent set of books. But for reporting purposes, we were still separate entities until later.

In many places the pace was set by how fast other areas of the bank could merge their operations. In New York, Global Bank is just another part of the total bank, whereas internationally the branches are themselves miniature banks. In terms of people, the Global Bank is 20% or less of the organization. On a revenue basis, it's even greater than that. But we depend on the rest of the organization for the capability to make payments, to manage account relationships and financial reporting and accounting for the operations. If they can't change things, we can't go ahead. Instead, we have to wait for them. So each of the units really went into the merger process together. In general, we felt pretty good because a lot was achieved, and it was done without any unexpected problems.

Q. The merger of Chemical and MHT had the potential to become one of the biggest re-engineering efforts that the industry had seen in a long time. But another way to look at this is that you weren't re-engineering at all: you were struggling just to tread water, to try to bring your businesses together by the dates that you had targeted.

A. I think our guide at that point -- because of the time pressure -- was you just do things in the most effective way possible, within the bounds of the resources that you have available. Trying to come up with a completely new solution in six months is taking too much risk. You know what you've got. Pick the best of what you've got and use that.

Q. And how about once things had settled down following the merger?

A. After January 1, we began to more closely inspect our systems environments with the future in mind. As a result, we are doing three things:

1. *re-architecting and replacing* systems that still work, but are no longer sufficient to meet the bank's needs;

2. *re-engineering* systems that have become fractured as the business processes have changed -- we have the pieces but they no longer fit into a consistent model, so we need to bring them back; and,

3. *back-filling* gaps where the appropriate functions didn't exist before or where a system has totally broken.

We've broken down our operations into three segments: the front-office trading operation, the back-office processing environment, and the MIS functions that enable the businesses to manage risk and identify where the real business opportunities lie.

Unfortunately, the systems in the *front office* are largely not going to get us where we think we need to be. What they do cover is insufficient and the technologies -- their very basis -- that we have in place are now outdated, so we can't go forward with them. I'll tell you more about that in a moment when we discuss the potential of object technologies.

In the *back-office*, we've come up with a number of solutions, but what we need to do is come up with one that is consistent, allows us the flexibility of covering everything that we can do in the front office, and does it in an efficient way. We are looking for a platform that will allow us to regionalize or deploy new businesses without suffering undue delays or unexpectedly high costs.

Our MIS operations are charged with taking the information that comes out of those environments, and *re-packaging* it for management use. This doesn't always have to be done in real-time -- *optimal time* is more like it. However, some of the applications we are building will have the capability to support risk management on a real-time basis in those businesses where real-time really matters.

ANALYSIS AND DESIGN OF TRADING AND TREASURY OPERATIONS

Q. What implications does this have for your vision of the way that software to support your domestic and international operations is designed?

A. We are shooting for a consistent processing environment that will be split into two separate units: domestic and international. The domestic side has to have broad-based systems capabilities. Domestic operations in a money center bank are largely New York-based operations. However, there are lots of kinds of processing that we don't deliver ourselves, but we nevertheless make use of the bank's services. Internationally, all of these services are considered to be part of the bank's environment.

Let me give you an example to illustrate.

In New York we trade domestic securities instruments which differ from those traded in our other international locations. We are in the process of re-engineering our securities systems so that we can come back to a model which says this: Any traded instrument that possesses the features or the behavior of a security will be processed by that environment.

So instead of having multiple copies of systems, each of which contains a permutation to support very similar kinds of processing, you build a base which has the flexibility to encompass the permutations. This is all done with the goal of getting things done with less staff.

From a business analysis and application software portfolio development perspective, people are saying: "No. What we need to do is go back to a single original model -- *an object-oriented blueprint of trading and treasury system functionality* -- and pull everything into it." Our design ultimately is not broken, it's just that some new life needs to be breathed into it again, and we have to do this keeping an eye on the costs.

We are hoping that at the end of this process, the Global Bank will be left with a vision of securities trades processing systems design so that when the market produces new permutations of the products, then it's just a matter of our turning on some permutation of the processing functionality that we've already got. But to do this, you have to have a very viable securities model going forward. Once we have that in place, we think we can add things like foreign currency processing and equities, and so on.

Of course, this isn't the only way that we are going to do it. We are re-architecting. We are re-engineering. And we are bringing the capabilities of the new technologies to bear on our business -- whatever works best to achieve the technological flexibility the bank needs.

Q. In effect, what you are doing is re-engineering your software development environment.

A. That's right.

Q. A strategy that emphasizes reusable software is a key to unlocking productivity and flexibility there.

A. We've decided that this whole process has to lead to a software development environment that is entirely new, where applications consist of between 65% to 70% of reused code. When we reach this level, it will be as though we can deliver packaged software in-house. The 30% to 35% new code will extend the software functionality that we have on hand at the bank, and that too will later be available to be reused.

To reach that point, our techniques have to change also. We would like to be able to design a system for a specific platform, and then drop it into a different platform when the need arises. Taking the variety of models that we have for the systems that process our business and bringing them down to one is not easy: We liken it to performing open-heart surgery when the patient is walking around, when we ought to put the patient to sleep and then re-evaluate what we have when he wakes up.

Since January 1992 Chemical's credit rating has improved significantly. This gave us more room to create profitable business. In the Global Bank, we went from a business which would normally run in about 2,500 deals a day to one that generates 4,000 or more. And that was just in foreign exchange trading. Recently we've been running 5,000 plus, and the per trade dollar volume is getting larger also, so you can see the business changes that are occurring on top of our systems.

Unfortunately though, we're not making changes to a single operating system. That's a significant issue. We're too big. In some cases, we need to migrate both older systems into our newer model in a controlled fashion and see how they scale up for the new Global bank. But you can't just decide you're going to have two systems that do the same thing; at some point you've got to cut over. You take lots of risk with this. We are looking for low risk solutions. It takes longer, but we think it means that we will have a more powerful, more flexible processing environment as a result.

We have a good illustration of this in our international businesses. We have to be able to support foreign exchange, securities, derivatives, loans and the standard sort of corporate treasury business. We would like to be able to do all of them through one environment. But what we're doing at moment is taking the systems' environments which we had, and then bridging the gaps to build a complete chain of trading and treasury software functionality. This chain is the backbone of our design philosophy and linking new pieces to that chain is how we put it into action.

Q. Tell us more about the object-oriented design perspective that the bank has now.

A. Our goal is to come up with a design view which allows us to be able to support and process any instrument. We're doing this with an interest rate system now. So many of our applications require interest rate-related processing and updating, it makes sense to exploit the generic processing we have there. We can reverse-engineer the pieces that exist to bring them up to standard, which will yield code for a more generic model of interest rate processing.

When we do this, it works out so that we probably 35% reuse and end up with 65% to 70% complete replacement. This is a significant effort. It could be called "Chemical's NEWARC".

(Note: Slater is making an analogy here to another well known re-engineering effort conducted at the First Boston Corporation. The effort led to the development of a "new architecture" of investment banking software applications for the firms investment banking and trading operations, and was known in the industry as "NEWARC 2000".)

We're going to take it slowly across the whole spectrum of application functionality so that, at the end, we'll have one system model that covers all the varied international operations. In the process we probably will have turned off four systems -- five actually -- there will no longer be a mainframe system that is doing the IS processing.

STANDARDIZATION VS. CUSTOMIZATION, REGIONALIZATION VS. CONSOLIDATION

Q. How will this vision apply to your international branches? Don't regional differences in the content of the business create constraints on the applicability of building from a single model though the trend is towards global operations?

A. True. Our international branches have specialties, so there isn't a standardized model. In addition to funding, you've got foreign exchange and derivatives, international treasury and then securities markets. They each have some level of representation in our branches, based on the importance of the business in that part of the world. Hong Kong and London are the major sites for the international treasury business. London and Singapore, and to some extent Tokyo, are the centers of the foreign exchange business. But the derivative business is larger in Tokyo and foreign exchange is larger in Singapore. Lately there has been a move towards centering our support functions in Singapore because of the cost structure of operating in Tokyo. Securities is predominantly London, New York, London, and some in Tokyo.

It's definitely not one size fits all. The history of software development within the bank is regional with local customization.

From a back-office point of view we'd like to build a generic capability to support all of these businesses. Of course, at the present moment we are some way away from that. There are different systems solutions applied to many of these. In some places, the only place that we really consolidate is into the general ledger, and that's not good.

Some of the businesses are global as well, so they have a notion of consolidation coming back to New York. And it will probably stay that way. It's possible that you might even see more of that happening: it depends on issues such as cross-border data flows.

There is an ongoing tension here. On the one hand we are trying to consolidate our systems to reduce costs and improve effectiveness. On the other we are pushing to develop regionalized business strengths, by building on the local ones. We are trying to re-engineer systems to fit a consolidated model, but we're regionalizing at the same time, even though the businesses are moving towards a global perspective. So you can see we have several parallel tracks that we are moving on at the same time.

SOFTWARE DEVELOPMENT WITH OBJECT TECHNOLOGIES

Q. How will the tools that you have available support that?

A. We're attempting to obtain some of them from the marketplace. We are looking to packaged solutions, but recognize the importance of broadening the spectrum and including flexibility for the future. We believe that many of the systems development tools that are required are generic to the industry, so rather than developing them from the ground up, we believe that we can purchase them off the shelf.

For the moment though, it won't be easy to implement this approach. The packaged solution business is changing. It's playing more along the lines of tools to support an object-oriented view of the world. And what we would like to put into place is a complete object model. Even though it is still too early, we were able to identify several financial toolkits sold by companies that have solid experience in building systems. Instead of buying languages, we wanted to buy a model from which we can build more refined solutions and that's what we've done. We are using the model as the basis for building the first level of new systems. And we are working with the vendors to make their toolkits object-oriented.

Q. Is this work being done within the Global Bank?

A. We're doing it with them directly, yes. We've committed ourselves to a technical solution, and we're taking the toolkit and deploying it as it is. This is a big step to prove to everybody that this technology is really viable. I have business problems I have to resolve, and I can't really say, "Oh, philosophically, we should do it this way, just hold off for 2 years, and trust me."

Q. But the vision is really to re-engineer the software development process?

A. Yes. This is an investment for the future.

If we do this properly, then for the next few years, at least -- if not for a long term -- we'll be using this. It raises the level at which we develop systems and solutions. The problem is we still think of systems by their names so I can't sell a business unit a generic, this-is-a-widget application. No, our business people tell me: "I want *HedgePro* or I want *RiskMan*, or whatever we're going to call these things."

And they think that by putting a name on a system, the project will get funded. I've got to accept that.

It's no different than what happens in manufacturing. Ford uses the same frame, the same engine, the same drive train as the ones that were used in designing a Taurus when they design other cars. But would a buyer say she wanted a generic car? They put a name on it, and make it a new product that is based on the old one.

The packaged solution software marketplace is moving in this direction. We're buying the pieces now, and the whole will come into clearer focus in the future.

Q. So reusable designs and reusable software objects will come from both inside and outside the firm.

A. We've got lots of things that are right there on the shelf.

We already have implemented dollar-based yield curves for fixed income instrument out there, and they are used all over. So, why do I have to have generate yield curves in three different environments? I have one that sits there, and if I really want to have a real-time yield curve, every time one of the factors changes, I can change it! And whenever anybody asks for it, I've got the current one. If you're willing to throw some reasonable piece of hardware at this, which these days would cost you about $20,000 and not much more, you could do that.

So, that's idea, and you don't need to have multiple layers to do that. We maintain a consistency. That's the approach we're trying to take here.

Q. And the tools will be adopted across all the businesses in the Global Bank?

A. The styles and pacing of businesses change from location to location. We're trying to use consistent tools across the entire business, it's just that we're not there yet.

There are certainly several other banks thinking along the same lines as we are, but some of them are just going right back to the vendor. They want to buy the *entire object model*. However, I just think: Why do I need to do that? This looks like a long-term approach with no delivery in between. Our approach, by contrast, is buying in that capability, and then taking it forward. It's a massive process. Even though object technologies have been around for twenty years, it only has been a philosophy: you couldn't really apply it because the technology didn't let you. Now the technology is able to deal with the whole issue. The picture has changed.

We want to take these tools and define a generalized object model that covers trading, and see where the toolkit fits in with the object modeling approach.

Yes, we'll need to bring small software development units into this early on. You can't have everybody sitting around and saying: "We'll wait until we're all done, and then deliver the software to the repository." If we all wait to reach that single point, it's like we're going into a funnel. There's the potential for a back-log -- maybe a breakdown. People will get nervous.

Some people are building small, well-defined products. Do a model of that, and then come to the core group, and integrate it into the overall model. They decide who does the work. And when it is done, it goes into the pot -- a repository for our future software.

In the meantime, we'll continue to have a need for people who can develop specialized things -- yield curves, figuration, mathematical types of things or whatever. We'll also have people who specialize in managing the objects that are produced, just as we've seen the need for people to perform data management in the past.

HUMAN RESOURCE ISSUES IN THE TRANSFORMATION OF THE GLOBAL BANK'S SYSTEMS

Q. This last segment of our discussion goes a long way towards pointing out that re-engineering trading and treasury systems at the Global Bank has an important human element. One expects that as people change what they are doing, their skill set also must change. And then, as they develop a new skill set, they also have to be ready to accept new career paths. The education that prepares people to work in this industry also has to change.

A. Certainly we are on the leading edge of this -- hopefully not the bleeding edge. For an application developer, the future will be very different from the past. The developer will not be a programmer. The developer will be somebody who understands the business process and enough of how things go together to be able to be able to manage the process.

Q. One of the important hypotheses we've heard among firms that invested early in computer-aided software engineering **(CASE)** technology is that the curve of software development effort across the phases of the software development life cycle -- planning, analysis, design, construction, testing, implementation and maintenance -- is shifting. Traditional software development methods lead to a curve which is skewed towards the right, requiring relatively greater efforts to go into construction and coding. CASE pushes software development effort towards the left, into analysis and design. Here is how these relationships look.

INSERT FIGURE 1 ABOUT HERE

The goal is to get the system right the first time around, to search out opportunities to reuse application design materials and software object, and also to make the resulting application flexible enough to incorporate the possibility of future changes.

A. That's a big issue. And, of course, this perspective assumes that you can plug everything together, and it will automatically cooperate. But we're a long way from "plug-and-play" at the moment. The intervening layers between the lower levels of technology, the operating systems and the networks are still too visible at the level of applications development. We have the whole business of "middleware", all the software functionality that comes between them to make them cooperate.

There is also a massive re-training issue that we have to deal with. The problem, of course, is that human beings are often miles away from the knowledge centers for new software development techniques in a firm as big as ours. As a result, we've had to be realistic, rather than purely futuristic.

And we have to be careful. A technology may look interesting, but it can't really deliver value for you if you can't find people who really know it. Most people still view the whole design philosophy as specification-oriented, so we have to change people's attitudes towards the way that systems should be built. The way I look at it is in thirds:

* One-third of the people will be able to move, you really have no difficulty with them. They'll be able to learn C^{++}, they'll learn to deal with relational and object databases, and they'll come to understand network technologies.

* Then there's another third of the people who can, but they're going to need a little bit of pushing to move in that direction.

* Finally, there's a third of the people who just won't accept the baseline changes in the environment.

For example, in that last group there will be people who believe that COBOL is the best thing that's ever been invented on the AS/400 platform. They know it inside and out, and so long as you are using it, they have a role in the firm. But then one day you turn the AS/400s off.

In fact, there's an inverse bell curve relationship between the value of the people who know a specific technology very well and the length of time you have that technology in place. Let me illustrate.

INSERT FIGURE 2 ABOUT HERE

When the technology is new and few people have mastered how to work with it -- C^{++} going back several years is a case in point -- they are very valuable to you.

Then the technology matures, and there are many people who know it, so you value them less. Finally, the technology gets old, perhaps even obsolete, yet it's still there filling out an important part of your trading and treasury platform. The rub is that in the end you have to hire these people back as consultants, just to be able to maintain your systems, and you pay them a fortune because there are so few left who know this language or that operating system.

A second thing that hits you is organizational in nature. As the Global Bank changes its software development methodologies, it will go from being a *vertically-oriented support* group structure which is focused on a narrow band of instruments, to one that involves *horizontally-oriented support*. The result will be an increasing need for the bank to achieve software development that delivers consistent processing. Then instead of having people understand all of the hardware and software details, the focus will shift: you will need people who understand the securities business or the derivatives business. The software development tools -- the code generators, the application diagrammers and so on -- will make knowledge of the procedures for how things get done less important.

What emerges is a recognition that there will be business processes that can move from a vertical support structure to a horizontal one. Your centers of knowledge, your people in a hierarchical organization structure will need to be replaced by people who are comfortable working within a matrix organization -- *a virtual software development organization* -- that can react as swiftly as the market requires.

That's the theory anyway.

Q. That a closing statement if ever there was one. We wish you the best in trying to make the most of your vision. Thank you very much.

Figure 1. The Case Development Life Cycle

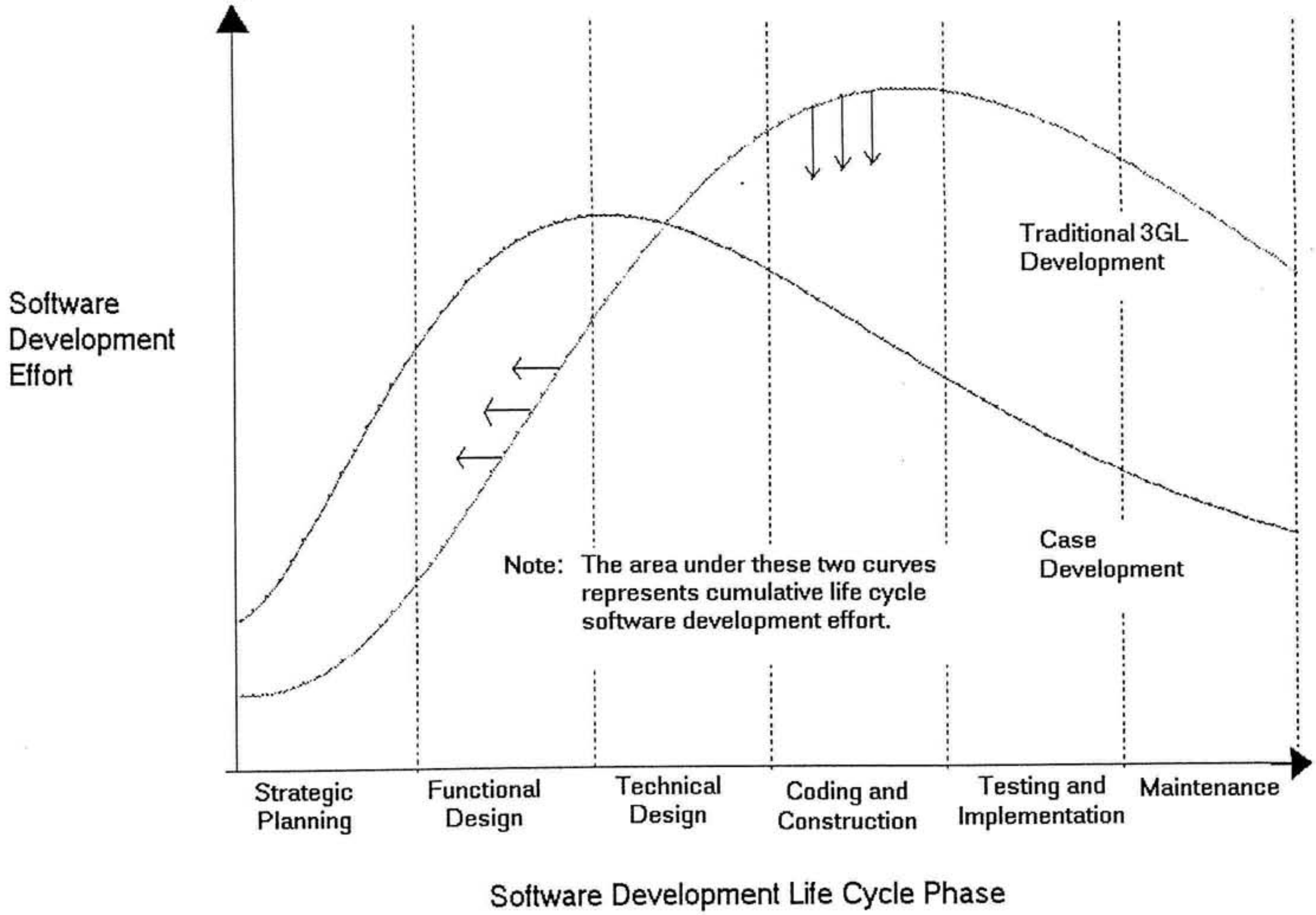
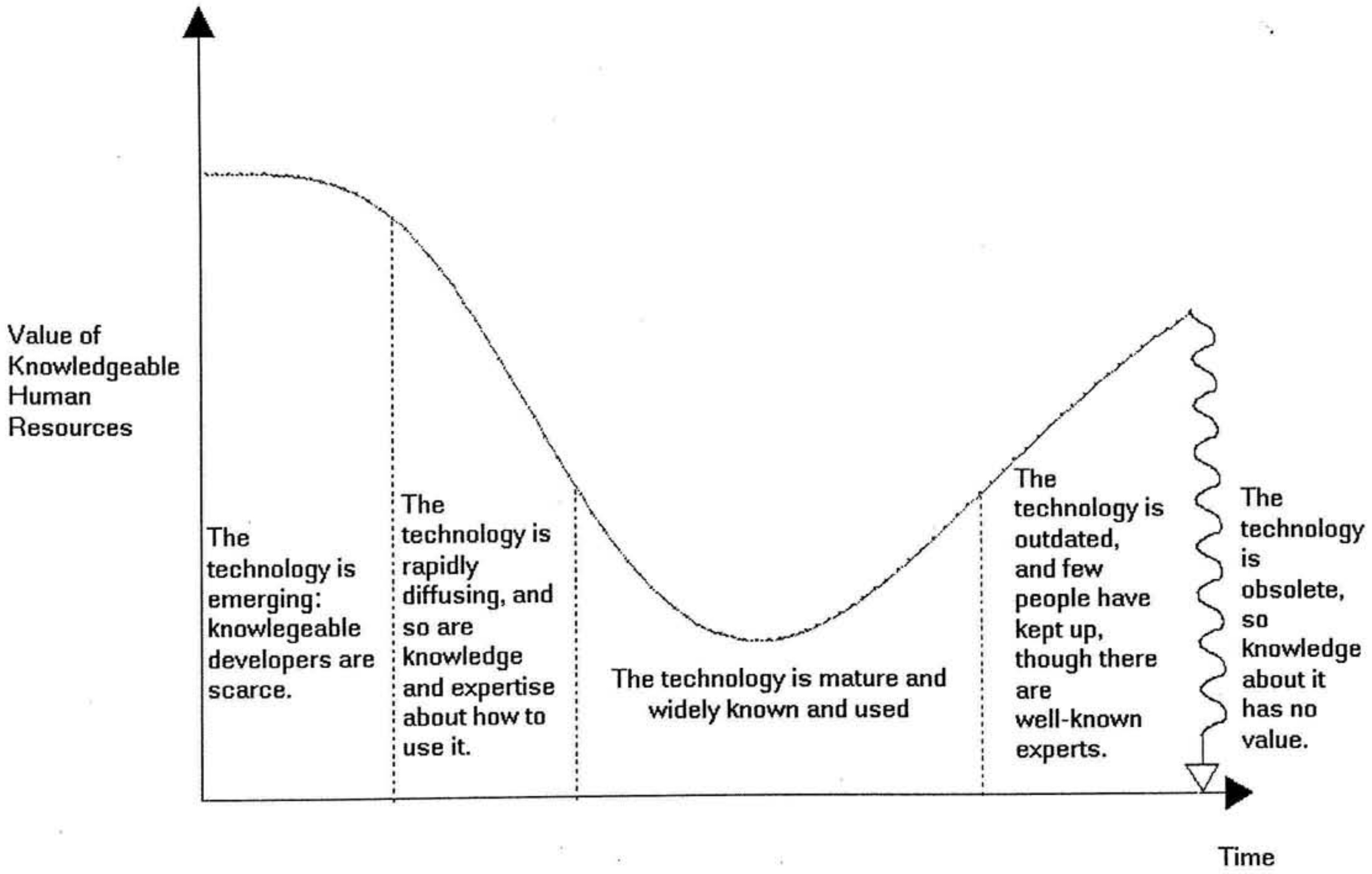


Figure 2. Slater's Human Resource Value Curve


Value of
Knowledgeable
Human
Resources
The technology is emerging: knowlegeable developers are scarce.
The technology is rapidly diffusing, and so are knowledge and expertise about how to use it.
The technology is mature and widely known and used
The technology is outdated, and few people have kept up, though there are well-known experts.
The technology is obsolete, so knowledge about it has no value.
Time