RECORD OF SOCIETY OF ACTUARIES
1994 VOL. 20 NO. 1

HOT TECHNOLOGIES

MODERATOR: GREGORY M. MATEJA
PANELISTS: VINCENT J. GRANIERI
WILLIAM RAICHLE*
WILLIAM SCHEEL†
RECORDER: GREGORY M. MATEJA

- We're not talking stolen secrets, but rather what new emerging technologies have people implemented recently?
  - Throw out those mainframes
  - Step into the client/server arena
  - Geographic technology for insurers
- What are their experiences, benefits, problems, and pitfalls?
- How do you keep the implementation of "bleeding-edge" technology from leaving you bathed in red: red ink, red in the face, or sometimes even a pink slip?

MR. GREGORY M. MATEJA: My feeling is that technology has totally revolutionized both our personal and business lives over the past few decades. The pace of change has been, and continues to be, totally overwhelming. Our panelists are going to give us some insight into some past, some emerging, and some current technologies that are of general interest to insurers. More importantly, I think, they're going to provide us with some general insights into the developments and use of technology to solve real world business problems.

Our first panelist is Vince Granieri. Vince is vice president and CFO at Midland Mutual. Prior to working with Midland, he spent time at Conning & Company, Hewitt Associates, and Nationwide. Vince is a Fellow of the Society of Actuaries, a member of the American Academy of Actuaries, and an Enrolled Actuary. He also holds a master of business administration from Harvard Business School. As part of a company reengineering effort in 1989, Midland decided to throw out all of its mainframe computers. He's going to tell us about some of the trials and tribulations encountered during that process.

Our second panelist is William Scheel. He is the director of actuarial system integration at Price Waterhouse in Hartford, CT. Bill has a Ph.D. in economics. He is a certified property/casualty underwriter, a certified employee benefits specialist, and an associate in risk management. Prior to joining Price, Bill was with Tillinghast, and he has also spent many years as a university professor. Bill will provide us with an introduction to client/server technology.

---

*William Raichle, not a member of the sponsoring organizations, is Assistant Vice President of Insurance Services Office in New York, NY.

†William Scheel, not a member of the sponsoring organizations, is Director of Actuarial Systems Integration at Price Waterhouse in Hartford, CT.

Our final speaker is Bill Raichle, assistant vice president at the Insurance Services Office. He is responsible for bureau ratings, property information, premium comparisons, and a variety of reporting developments. He is also the project manager for the geographic underwriting system. His talk will be about the applied use of geographic information systems in the insurance industry.

MR. VINCENT J. GRANIERI: I'll tell you up front that I am not a computer guy. I don't know gigabytes from trilobites. I'm going to give you a user-oriented perspective, as some people have said. If you want evidence, I think we could find three or four in the room here who can attest to the fact that I'm quite opinionated. My opinions may or may not apply to your particular situation, so I don't mean to be offensive in any way. I think what you'll see is a specific situation where we came up with some pretty specific problems and recommendations on how we might fix those problems. In any case, I hope it's entertaining for you and somewhat insightful as you try to apply it to your situation.

First, let me say that it's my firm belief that systems and networks and hardware are really not the solution. It only makes sense to view them as part of the larger entity. What are you trying to do as a business? They're a means to an end, just like your staffing, just like capital, your distribution system, your management. I look on how systems fit together with all these other factors, and how they come together to sort of shape that organization and make it successful. The important thing is to align the inputs and then see how they work together. That's what we did at Midland.

Midland is a company undergoing tremendous change. We'll get into that a bit later as we go forward, but Midland, among other things, is demutualizing, and that has to be factored in to the whole capital crunch. This means that we are focused on the bottom line. In my opinion, we've got to use systems and these other inputs to focus on four things: profitability, service, capital preservation, and managing operational growth.

- Profitability: to provide a competitive return after demutualization, as we have provided significant operating gains as a mutual company.
- Service: to support a revamped work force in its efforts to maintain extremely high standards of performance.
- Capital preservation: to maintain a solid balance sheet until we receive our capital infusion postdemutualization.
- Operational growth: to survive, and I do mean survive, the move from January 1992, when we were getting 100 applications in a week, to the first quarter of this year, we've been averaging tripling new premium income with a disproportionately low increase in staffing.

We'll try and focus first on Midland as a backdrop for my comments regarding our move to throw out the mainframes. Table 1 presents some basic information about Midland Mutual. You'll note that somewhere between 1989 and 1990, there was a period of severe capital crunch. We had to sell some blocks of business to maintain our desired capitalization; that threw us into the realm of demutualization. We quickly approached that important capitalization crunch again at the end of 1993. As you see, our policies in force have grown from 145,000 to 166,000. In-force business is

over $20 billion, but you'll notice that the salaries and the numbers of associates do not follow that same pattern of growth.

TABLE 1
TOTAL COMPANY INFORMATION
(In Millions of Dollars)

	1988	1989	1990	1991	1992	1993
Assets	$886	$1,003	$939	$1,080	$1,135	$1,204
In Force	$14,842	$17,635	$19,900	$19,250	$18,820	$21,991
Policies	145,492	159,465	164,438	162,853	157,521	165,941
Salaries	$9.4	$8.5	$8.1	$8.1	$8.3	$8.8
Number of Associates	328	276	231	241	224	240

What products do we sell? We are best known for competing in the brokerage distribution channel with our level-term policies of 10 or 15 years. Of course, that introduces other uncertainties as to *Guideline XXX* and what's going to happen after 1994. We sell mostly single-premium deferred annuities (SPDA). We have a fairly broad universal life (UL) portfolio. We sell back-end-loaded products now. We used to sell some front-loaded products as well. We have pioneered what we call our bio-age underwriting, which is based on the customer's biological age rather than chronological age. I do note that many of our employees have probably jumped in biological age 10–15 years in the last three or four years through all these processes. We do sell some pre-need insurance. We have some simplified underwriting products on the UL and the traditional side, and we sell single-premium whole life (SPWL), so we have a broad spectrum of the individual life arena.

Let me tell you a bit now about what we did. Really, we are talking about total corporate renewal. We changed our distribution systems. We went from a career system to a brokerage system, and that occurred beginning in the mid-1980s and was completed in the early 1990s. Our career system had more fixed costs; the brokerage system has more variable costs. What we wanted to do was pay for what you get. If we get production, we pay more. If we don't get production, we pay less. As you can see, we're into rocket science at the Midland, a very simple concept, but very important. As we try to run lean and mean, we felt it was most important to cut down the fixed costs and put ourselves on a more variable cost structure, and that has something to do with why we did what we did with the mainframe as well.

For our internal structure, we moved from a functional organization to a product organization. We felt that it was very hard to get our arms around the entire insurance realm, all lines of business, all functions. We thought it was better if we had to pick between going functional and having someone know everything there is about policyholder service, or know business and know everything you can know about annuities, then we'd take the latter. So we set up what we call the A–Z teams. I guess if you've ever read management material on Volvo and how the company sends 6 or 12 people out to build a whole car rather than working the assembly line,

like General Motors (GM) or Ford, you know a bit of the approach that we took to managing our business. We thought it would be easier for our associates to see the impact on the bottom line if they were able to look at it from a product perspective rather than a functional perspective.

I'll try to get in some of these juicy, computer-guy kind of numbers and things like that, although I'm not a computer guy. I might gloss over some of the key details, but I'll be happy to answer any questions that I recall the answer to at any time. We had an IBM 4381 mainframe and a System 36. We had two HP-1000 minicomputers. We had a DEC minicomputer doing our accounting. We run ULIS and LMS and McCormick and Dodge accounting software. That's where we started.

What we ended up with was, and I'll give you more details about our network system, basically one local area network (LAN) that replaced the host of five or six computers, and we went to a Macola System. Our operating software was Life Pro out of PDM in Carmel, Indiana. Macola, I believe, is based in Columbus, Ohio, which is the home of Midland Mutual as well.

Chart 1 shows that our 1988 actual data processing (DP) expenditures were about $4 million. In 1993, we were under $2 million, for a reduction of a bit more than $2 million. If we would inflate our DP costs in 1988 at 5% a year, and that's the bottom half of the chart, you'd see that those costs would be over $5 million by now. Having those same actual expenditures of under $2 million means that we're $3 million under where we were before. I'm not the author of the conversion on the operating side, but I was in charge of the financial conversion on the accounting system, so I can give you a bit more detail there.

I should also say at this point that I still, to this day, believe that the McCormick and Dodge accounting system is certainly the Cadillac system. It provides much information, and for the right company, it certainly could be the right solution, which is consistent with my discussion earlier on the systems being part of the entire whole. For the Midland, the size company we were and where we were, it didn't make as much sense as the microbased solution from Macola. As a cost comparison, Table 2 will tell you the hard-dollar cost comparison.

Look at the software licensing, consulting fees, and programming tools to implement both of those systems. When I took over as chief financial officer, we were just completing the first conversion from our old, home-grown general ledger to the McCormick and Dodge ledger. It cost us $700,000 to convert to the McCormick and Dodge system in the late 1980s. In the early 1990s, it cost us $200,000 to convert to Macola.

A soft-dollar comparison indicates that our McCormick and Dodge system took eight people 16 months to convert after a five-month selection period. The Macola system took us basically nine months after a two-month system selection, although the system selection was greatly simplified. There really weren't that many accounting packages available for microcomputers that could handle or even had a hope of handling the volume of transactions generated at the Midland.

CHART 1
EXPENDITURES COMPARISON
(In Millions of Dollars)

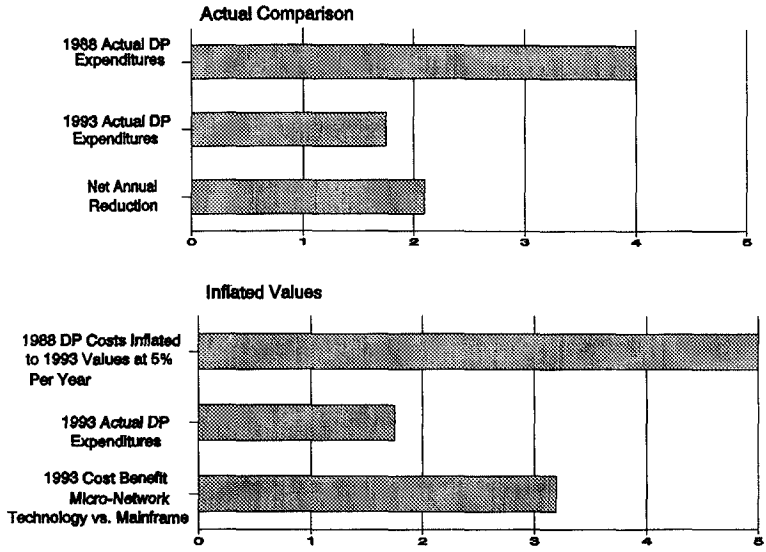


TABLE 2
MACOLA/M&D COMPARISON
(In Thousands of Dollars)

	Implementation Cost		
	Macola (New System)	**M&D (Old System)**	**Savings**
Software License	$24	$313	$289
Consulting Fees	149	399	249
Programming Tools	32	0	(32)
Total	$205	$712	$506

Summarizing what we did, we built real consistency into all the above elements: the distribution, the internal structure, and the computers and software. They all work in concert to provide us with what we feel is a competitive advantage and a better way to operate our little insurance company. Let me describe the computer environment for you.

As I said before, it's a local area network. We have approximately 300 workstations, primarily 386 and 486 technology. We have 25 servers, primarily 486 and 586 technology. The computer guys tell me we have 86 gigabytes of disk space. I have not independently verified the accuracy of the 86-gigabyte number. As a postscript, the computer room that we once had on the third floor of our building has been converted to a workout facility for our lean and mean associates. It's much more regularly frequented than the old computer room ever was.

What are the advantages of this system? The major advantage, as I said, is that bottom-line approach. I'm going to talk a bit about the number of advantages and where they impact our bottom line that I can see. First, lower cost. Because the maintenance is much lower and the hardware requirements are much less, we have less need for the programming resources; so we're looking at about a $200,000 cost advantage running our new microbased accounting system versus the old system.

Before we reorganized into our product alignment, we ran much less information through the old system than we're running through the new accounting system, because now we perform full income statement and balance sheet accounting for all four of our operating companies. But if you've got to lower cost, it's easy to see how your profitability is enhanced, how your capital is preserved. Another important aspect is that of managing growth, because as we were growing before, you get to a point on a mainframe computer where you're running out of space or your response time is slow and you have to make a big decision, usually one of those $2 and $3 million decisions that CFOs heads roll over if they're not made correctly.

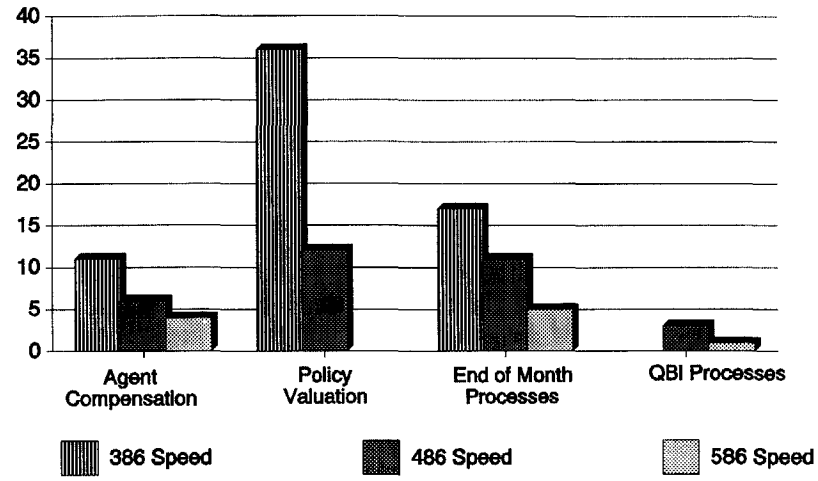
Here we had the opportunity to take on expansion in a much more orderly fashion and in more bite-sized pieces. There's an old Sicilian saying that asks, how do you eat an elephant? You eat it one bite at a time. I liken buying a new mainframe to trying to eat the elephant in one fell swoop; whereas under the microbased technology, you can take it one bite at a time. You add a server. A new server costs $35,000. We put up our servers with about 35–96 megabytes of RAM each, and it costs about $1,000 for each additional 16 megabytes. So we can accurately tailor our operations and expand as the business expands.

We don't have to buy too much early on and have overcapacity and then have that underutilized which, of course, wastes capital. We can take it as it goes, and we can match the power of our computing system. Each of our teams actually has much different levels of computing power, even though they're configured with the same basic computers. But if you take a look at the computing power, the interest-sensitive product group that sells UL, with many more transactions, is going to have the 586 technology throughout its server room. For our accounting system we have one 586 and two 486s. You can really tailor the expenditures to the needs of your users.

I'll talk about some other benefits, I believe, of this technology. Efficiency. This is almost a serendipity, as we found. We started up when the 286 technology was dominant and 386 technology was on the horizon. As it turned out, not for input/output functions but for actual processing speed, the 386 was faster than our old mainframe application. What you see in Chart 2 is a number of different applications: the bonus system to the left; policy valuation, which is setting reserves;

end-of-month processing, which is basically closing the general ledger; and then another bonus calculation.

Chart 2
PROCESSOR SPEED COMPARISONS
(Duration in hours)



The most dramatic difference you see, the first block, is the 386 technology. The 386 speed for our valuation system on UL used to run at 36 hours. We decided that we needed monthly financial statements, so this was 36 hours every month just to run the valuation system. We've got that down to 12 hours on the 486, and we're currently exploring whether 586 is going to give us a benefit. We haven't tested that completely yet. But on all the different functions that you see, basically the technology is leaping forward and allowing us to process. Even as we're growing, we're processing more and more quickly, which has really helped us out.

We turned out our quarterly statements. We had preliminaries by the seventh working day this year for statutory just this last quarter. By the eleventh working day of January, we had our preliminary, 1993 year-end figures. That is a week to a week-and-a-half faster than we were ever able to do with the mainframe.

What else does this do for us? It basically increases the associates' understanding. In a decentralized environment, where they can see the whole business and not have to see the whole company, they understand their business better. This has an obvious impact on service and an obvious impact on profitability. Our decentralized technology also fosters independent operations. In the old days, if a problem occurred in the DP area, the whole company was shut down for a period of time. Now the problem is isolated to only one team and not the entire operation.

Furthermore, each operating manager (we call them general managers) can decide how to deploy resources, and they, in fact, make policy on how they're going to run

their shop. A common platform is another advantage. Running with the same hardware and the same software throughout the company, as opposed to that conglomeration that we had, allows the same screens for policyholder service. Therefore, we can cross-train our associates. If we get an overload in term one week, we'll shift some annuity guys over there to help out and carry the load.

All in all, we're really talking about increased flexibility, and we're going to need that as we go forward. Not being a large company, we need that flexibility. If we were to buy a block of business, instead of having to convert that through some of the older methods—getting tapes and bringing them in, etc., and trying to put them on our mainframe—what we can do now is send a team out with a file server and some computers, and they can do that entire conversion on-site, bring it home, plug it into our network, and they can be operational the next day after they bring it in.

Helpful hints and the practical stuff? Here's where I get really opinionated. Judgment is vital in this process. You need to know when you can take a risk, because you're going to be taking risks if you do this. Planning is important, but judgment is vital. You can talk yourself out of a lot of efficient performance through planning. In the first year, we identified 34 separate tasks that we completed. If, at the beginning of the year, we would have decided how many we could do, we probably would have come up with 15 or 16 things that made sense.

The fact is, you really don't know what's down the road. I guess I used to be one of those guys—and being a numbers guy, we're a bit more conservative—so I was thinking maybe you'd plan 75% of the way, and let the rest fly. What I found out is maybe you stop at 50% on the planning, because the other 50% is unknown. You're not privy to that. They say that you're only given in life as much as you can bear. You can't bear to know 100% of what's coming down the road, so it's good that you stop at 50%, start down there, and see what happens.

You never can plan for everything. We couldn't plan for that technological improvement. Was it a bad decision that we implemented 286s instead of waiting for 386s? In hindsight, maybe so. We wrote off lots of depreciation very quickly when we switched to 386s within the first year. However, we were a year ahead of schedule on our conversion too. You can always look back on things and say, if you had known this and had you known that, but the fact is, you don't know this and you don't know that.

We started up with 12 megabytes per server, and I told you now the biggest server is up to 99 megabytes, so we didn't really understand exactly what it would be like to run under that environment. However, when you're buying a 16 meg at $1,000 a crack, that mistake is easily rectified and it's not material, except for the one cost center for the computer guy who gets stuck ordering everything. The point is that we can recover from that.

Second, and this is getting into "mom and apple pie" kind of stuff, it makes a lot of sense when you hear it. However, you've really got to do this stuff. You can't put it out there and pay lip service to it and then watch it, because there are lots of good reasons not to pick a date and stick with it. We've got vacations. We're understaffed. We're working hard. But you pick a date and you stick with it. You have

to go back to my first point. Judgment is vital. You better know what you're doing when you pick that date. If you let the thing slide one time, it's going to slide again and again and again. From the looks of the room, I see that that's happened in some other places. We picked a date and we stuck with it.

In fact, thanks to some of the people in the room, like Mr. Christians and his efforts. This comes to my next point: find people that have done this sort of thing. If you're going to try this, you need the judgment. Find somebody to help you who has done it before. That really means you leave theory for the theoreticians and the textbooks, and you get in there because you're going to get dirty. I can't remember the amount of plan codes. I hazard a guess, but I think we had 5,000–6,000 plan codes. Al probably knows better than I. Does that sound right? No, 1,000. It was a lot, anyway.

But you need somebody who's been there before. Another very important thing: somebody has got to own it, hopefully somebody high up, and not a committee. Committees can't own things. People have to own it. We split this thing into two parts—the operations and the financial. I owned the financial and our operations guy owned the operations. Both of us were senior managers reporting to the CEO. Yet even at that level, we had to manage the details. I don't know if it's human nature or what the situation is, but when you ask the question and you get the answer, you better make sure that it is exactly the way you think.

Trust is a misnomer in conversions. You can't trust. Someone says, "Well, did you make sure that the accounting codes are aligned?" "Yes." "Prove it to me." Manage the detail level. Inspect what you expect. Next, you've got to take a step backward to go forward. You've got to expect that you're going to lose your mainframe guys if you make the announcement that you're moving to micros. There might be a few loyal ones who stick around to the end, and they're worth being rewarded in a very special way. But you're going to go backwards first. You're going to give up something. In our case, we gave up, much to our actuary's chagrin, the jumping-juvenile reserves, which weren't quite as good under the new system as they were under the old system. Since it was a $75,000 reserve, I didn't get too excited about that. Our total reserve is more than $800 million.

It doesn't work perfectly from day one, and it's never really over. There are a lot of postconversion events—cleanup, suspends went crazy, names and addresses. We had a centralized name and address file because that made perfect logical sense. The only problem was when one team went down, the whole centralized name and address went down; so we had to split that up into lots of pieces.

Lastly, kill the curmudgeons early on. Usually they want to keep something that they love, something from long ago, because the new system is not "capable" of meeting our needs. Of course, neither did the old system. Someone told me that we should keep our McCormick and Dodge system because it could be run on a DEC micro-computer. I asked him to prove it. Three months later, it still wasn't running on a DEC microcomputer.

The reinsurance system wasn't good enough in the operating software, and they wanted to build a side system. We only allowed the side system to be built in one

area, the death claims area. What we did instead was modify our reinsurance system and that worked out better. It's always great to have that common platform where things are linked together.

In summary, this is the bottom line. It's profitability. It's service. It's capital preservation. It's managing growth. This is our business. The ideal system is one that contributes to your bottom line. Now Bill Scheel is going to discuss client/server technology.

MR. WILLIAM SCHEEL: I'm thinking, as I set this up, about Vince's 86-gigabyte hard disk drive. When I get back to Hartford, I'm going to ask for one of those to be attached to my PC. Consider your PC. Consider your 20-megahertz PC. Let's assume that a 20-megahertz machine is roughly working at 20 cycles per instruction and is equivalent to a one MIP computer, one million instructions per second. Let's argue that it takes perhaps 500 instructions to service a single, line of source code. That means there are about 20,000 source lines per second that can be processed. Think of that like a loop with a single statement that maybe assigns a variable, and that's roughly equivalent to say five or six of my source lines. That's reasonable. You could make a PC comb through 5,000 of those iterations.

Now that would result in about 6.5×10^{11} source lines per year that PC could process. How many of you are using your PCs right now? Very good. Most of you aren't. In fact, my guess would be that probably none of you are using your PCs right now. In fact, I think it would be generous to say that any given PC is utilized perhaps 10% of the time. If you think of the world's stock of PCs being on the order of 100 million, that idle time works out to 5.8×10^{19} idle source lines.

Now let's suppose that the cursor up there takes about 1,000 source lines to implement, to execute. That works out, in terms of idle, wasted PC time, something on the order of 5.8×10^{19} repetitions of the code it takes to draw a cursor on a screen. In other words, it is the equivalent of 60 million trillion cursors a year of idle time on your PCs that aren't used. I would argue, people, that I have one message I want to leave you with regarding client/server. That message is simply that client/server technology really means tapping into all that wasted computer time.

Now what I've put together here is a client/server application and it is a joke. It's a small insurance company, but it has sort of the fodder of an insurance operation. It has an underwriting module that looks at different risks, writes those risks, generates premiums and commissions for them, has a general ledger attached to it. It also keeps track of different distributions of things, and I think is a reasonable example of a client/server application running on just an ordinary old PC.

The second message I want to leave you with is that, to me, the focal point of client/server is not so much what Vince has described to you. That's going on in the background, that little insurance company. And it is, by the way, a miniature insurance company. I would call it kind of a factory, a mill. They just had an unusual loss. Is someone submitting phony claims? That's an event and the machine captured that event, and it captured it by filtering a transaction; in this case, a claims transaction. That's an example of client/server technology in operation.

But really the main thrust of client/server, the thing that we usually think of, is a network on which there is one or more machines, such as Vince's 86-gigabyte hard disk, operating on several different servers. The insurance commissioner says underwriting profits are too high for the company. I have a very low loss ratio, and unfortunately, I have to pay premium taxes of $56,000. Again, another event, the main thrust of client/server technology, are the things that most people think about—networks. That's really not, in my judgment, the important point of client/server technology.

These are examples. You have file servers. You're all familiar with those. You have database servers. Some of you are now using database servers to do relational queries. Probably few of you, if any of you, are dealing with something called an Internet server. But I would submit that within a half decade, all of you will be using Internet servers. These are devices that will talk to the outside world over all those high-speed information highway lines, which you read about and hear about, that are being scrambled for by companies like Sprint and MCI.

I think that the concept of client/server is embodied in what we all think of as one computer talking to another. Your CEO wants to look good and make a charitable contribution, another event being captured in the background. It may well be that client/server is the idea of really having different executable activities talking to one another, communicating with one another. That is what I would call the essential essence of client/server technology.

Here are some examples that you can relate to in various ways. File transfer. All of you have done that. You are all querying information. Probably you haven't thought too much about client/server technology in the context of the inner-process communications that is going on over a network. That's an example of client/server technology. Graphical user interfaces (GUI), these kinds of things that you see here, are often viewed as client/server. But I would argue that really the essence of client/server technology is what I call event-posting, this messaging thing. As I was interrupted periodically here by the machine, it advised me that certain occurrences had happened. That means that your computer systems, your networks, your clients, and servers will be constantly monitoring transactions, filtering those transactions, and causing you, your company, your bookkeeping or staff to be notified of those events. That, to me, is really what it means when we talk about client/server technology.

Now, at the very heart of it, it means that there is something called the "client" that is making a request out on the system somewhere, another showstopper. You just threw $4,256 down the rat hole. I don't know when these are going to happen, so it sort of breaks the train. We were talking about the client side.

The server side, on the other hand, is offering data. Often this is done on an advice. You set up a server device to advise you when something important has happened. A classic example of that would be in a server device where you have things called triggers. A piece of data comes in, that server is filtering the data, and it triggers an event, which can mean that a different executable will be launched somewhere in the system. The idea of client is that you request. The idea of server is that you get the data fed to you. It's not a big deal, but really the communication channels are very important.

Part of the problem with client/server today and the implementation of these kinds of things is that they're operating in the background and that is a background process. This is not trickery, people. This is actually going. There are messages flying around here. This is not canned, so to speak. This is not two concurrent processes that I am task-switching back and forth. This is going on as I am doing the slide presentation. This is kind of an activity here, even though it's trivialized in the sense that it's all located on one computer.

Consider a general ledger. A general ledger could be something that is an event recipient. It could be a client that's being served by accounts that are filtering in transactions. They may be premium receipts. They may be commission payments. All the things that are commonly thought of as a general ledger can well be thought of as a series of accounts, each of which is a client. Servers are moving stuff into it. Conversely, each of those accounts can be thought of as a server itself.

For example, right now that premiums thing that you see iterating up there with a $2,500 transaction was a result of a message being sent by the underwriting algorithm to the general ledger saying, "I just wrote a premium and here's what it is." Now look at that one there. That's a paid loss distribution that's being generated on the fly as the transactions are being processed. How do you generate your loss distributions today as actuaries? You probably go in and run some program on a periodic basis against that enormous 86-gigabyte transaction file. Who knows how long it runs; it generates a distribution.

Now why can't you just move the transactions into a whole series of slave devices that will give you those data in real time? Again, that's highly trivialized, but I want you to sense that there is some purpose to this madness, and that is that you can get information on a more timely basis. If client/server means anything to you, it probably means better access to your data and certainly faster access to the data.

Let me talk to you a bit about two of the problems. I've kind of given you some sense of the idea of windowing data. Filtering transactions on the fly would be one thought I'd want to leave you with. Another one is to kind of reiterate the obvious, and that is, what does that idea of an actuarial workstation mean? What we're doing at Price Waterhouse is to do, as a rule, all kinds of actuarial development in the form of what we call dynalink libraries. It so happens these are written in C, but that's not particularly important.

What is important is that these models, these sort of throwaway things, become "pasteable" servers. They can take an object of data, calculate on that object, and move it back. In the form of a dynalink library, which is something that, if you want to become client/server literate, you're going to have to begin to look at code in a different way. We think of programs as executable blocks that are chugged into memory, executed, go out and fish around through all the 86 gigabytes of information, and give you a result. You should, instead, think of them as things that can be brought in on a messaging basis, and that's essentially what dynalink libraries are.

Another thing that I think actuaries will benefit from in client/server technology is the whole idea of database replication. Perhaps some of you use Lotus Notes. That amounts to a replication of mail, if you will, throughout your organizations on different

devices. The interesting thing about that example is that a Lotus Notes server is also a client when it's being equalized. That server is calling up another office and saying, "Hey, let's balance our data sets," and the data is being passed back and forth.

From a very client standpoint, you are using or beginning to use kind of replicated data sets down on your LANs. So you take your transactions, maybe aggregate them at a relatively high level, bring them down, and start doing your experiments on them from a small or subsetted data standpoint. You have taken and used the client/server concept, where in this case the server was your main data and you brought it down into a smaller kind of client thing. Then that client, that local area device, is being used and tapped by your PCs. Again, what was a client is now the server, and you're down at the client level.

How do you become client/server literate? Well I would suggest that there are not really very many steps. Besides becoming somebody like me, who's sort of a hard-core computer nut, I think there are some perceptual things that can assist in your discussions with your programming teams that are worth emphasizing. The first one is to think of word processing, think of spreadsheets, and think of all these GUI things as one, highly integrated set of client/server devices, where one thing may sometimes be a client and sometimes a server.

For example, let's go back and look at this process that's going on in the background. There's just a word here. This is getting messages in from the underwriting algorithm. It's getting policy numbers. It's getting insureds' names, sexes, rating classes, and limits of liability. But that could be a policy generator. Again, it's kind of a trivialized idea, but think of word processing as something more than word processing. Think of it as something that is part of an overall system in which the data are being captured and moved into it using, again, client/server devices. Also, let's take note that on this side, which happens to be the underwriting thing that's being displayed there, I'm randomly generating.

On the other side here, like this cash-on-hand thing, what's happening there is it's beginning to receive and keep track of the general ledger from my cash account. I believe that when you look at client/server technology from the standpoint of a collection of small, third-party vendor programs that can be hung together and can be made to talk to one another, you will begin to see new insight, new possibilities, new ways of thinking about what your computing environment is really like. That's where I've used Excel in several different ways and Word in several different ways.

I've already talked about, and certainly Vince has underscored, the importance of cutting the umbilical cord to the mainframe. You've got to do it, people. That beast has seen its day. Do you know why? It's not because of its size and power. It's because all the data are locked in it. It is not well-adapted to this messaging thing we've been talking about. I think that the benefits of that at the start will be somewhat obscure, but over time certainly the detachment from the mainframe and moving the data into a more fluid messaging environment will have a great impact on your organization—impacts that none of us are, at this point, prepared to envision. They will probably take the vantage point of harnessing all of that unused, computational potential that's sitting on desktops around the world; it's an enormous sea of computational capability that is simply being wasted.

If I could send a message to it or if you could send a message to it when you're away from the shop, if I were to send an executable to your machine and say, do this for me, then that machine suddenly becomes serviceable. So that's where I think we're going, and it's important that you begin to think of all these little tools that are hanging out ready to be used.

I would strongly encourage you to build a company-wide depository of actuarial definitions of data objects. It's going to become imperative for you, because it is the common denominator around which client/server mechanisms can be made to work. They are very important. I would also suggest that if you want to become somewhat client/server literate, that you think a bit about the idea of integrated suites of software. Maybe you hadn't thought about the fact that Lotus 1-2-3 really shouldn't be on any actuary's desktop. Why? Because it doesn't do arithmetic very well. Why? Because it doesn't do client/server all that well. Why even more? Because integrating it with word processing, or with other kinds of client/server mechanisms is somewhat difficult.

I don't have stock in Microsoft, but I must say they've done a very good job, and I will give them a plug for providing a kind of environment for the development of client/server that is dynamite. You've already seen how companies are beginning to merge, like the idea of Borland and WordPerfect pairing up, so that you have a kind of a suite capability. There's a reason behind that. It's not just the economics of software. It's the fact that software is going to have to start talking to one another in a common thread. That's very important as you move into it and you might want to investigate that.

Also, I would suggest that you need to study the mechanics of messaging. There's a really steep learning curve, and it's not entirely obvious that the benefits to actuaries are all that great. It is obvious that they're great to your organization as a whole. What about things like structured query language? Are you all getting data out of databases using structured query language? Well, yes, you want to be, so that's part of this literacy test.

How many of you are doing APL programming? Whoa! Bad news! Because APL isn't really very client/server literate. It may be someday, but it isn't now. Better that you think about other programming languages like C or C++. Also, think about dynalink libraries. I know that perhaps most of you don't tune into that, but it's really like a set of object modules all put in one thing that's hanging out. Windows can go and grab these things any time it needs it. Suppose you have two things, like my Excel and my Word. These actually share a dynalink library, which serves as a bridge of communication between them.

By the way, you may be thinking this would only work on one machine, right? Wrong! Every one of those things could have been a separate instance done under a network dynamic data exchange (DDE), which is another buzzword that's sort of important. That transaction messaging doesn't just have to be on one computer. I could have had all these events operating differently on all your respective machines. So an important concept is that you begin to understand the tools that we want to use, this whole set of tools that we're moving into, not so much because you're

going to use them on a day-to-day basis, but that you begin to see kind of the "forest" of where this is moving.

Finally, a really futuristic thing, what about the idea of Internet and the idea of selling business? Some of you are direct writers, and maybe you do use the phone once in a while. What about the idea of an online catalog? I was working with a company in Texas the other day that actually has a very interesting concept on the Internet, and this would be through CompuServe. For example, you dial up and you say, "I see there's a catalog of ball bearings. I'm interested in ball bearings." Online through the Internet, through an Internet server, is downloaded an executable file and it's fired up on my PC. In it are all kinds of graphical images, sound bites.

This could be the kind of thing you find in the pension area, where you're doing a plan illustration for a prospect, and you want to give your participant some idea of his or her position. It's coming, people. The idea of serving stuff out to the community in a global network is upon us, and I think that is something that all of you, at some point are going to get very close to. You need to learn, I think, about the broader scope of Internet, Internetwork capabilities.

As a wrap-up and summary, I think first that it's messaging. Second, things like common macro languages for writing this stuff are going to be important. I think that all of you will eventually get into this GUI stuff, such as these little controls that you see. For example, there are a bunch of separate controls, each of which can be thought of as a client receiving messages, which do certain things. There are messages going amongst those things, so that when I press this H button, there's a series of messages going out that told Word to close down, to go down into the background, and it brought up a different document in Word. That is what GUI stuff is all about. That would be an example of a GUI-ism.

That's my final thought. Don't ever use the term client/server. It means so many things that it has become a cliche. I hope that I have impressed upon you the idea that client/server is somewhat in the eyes of the beholder. It can range, again, from very high-level things down to things like grids, and single things that are operating in the context of a single executable. It's a fascinating area. It's going to revolutionize what we did and are doing on a day-to-day basis. I speak because I've had over 20 years of programming experience.

What I am doing now using client/server tools has increased my productivity beyond my wildest imaginations. This little toy up here took on the order of three days for me to put together. Yes, it's a toy, but it has a lot of neat stuff. If I hadn't had the tools that are available today, I couldn't have done that in a year. Simple things that you're using on a day-to-day basis, like spreadsheets, word processing systems, GUI front-ends, and database requesters are within your grasp, and I predict that they will revolutionize how you run your business.

MR. WILLIAM RAICHLE: Before I get started into the bulk of the presentation, let me just ask a few, simple questions. How many of you out there have ever seen or somehow been made aware of geographic information systems (GIS) technology? You can just put your hands up. A relatively small group. That's good. You won't know when I make mistakes then. How many of you work for organizations that

use GIS technology right now? Fine. How many of you have actually used it yourself? I'm going to talk about a relatively new technology. From that perspective, I will step aside from my colleagues and talk about an application rather than hardware and software itself or other technologies like that.

I'm talking about taking some of the systems that these guys were talking about and building applications around it, using brand-new databases and brand-new technology. I got very interested in GIS technology myself because I manage a product that is built with GIS technology. Insurers became interested in GIS technology not too long ago. This over here is probably the crowning event in the relationship between property/casualty insurers and the technicians in the GIS or the geographic information systems field. Does anybody know what this is? It didn't happen too far from here, and that's Hurricane Andrew. Every time I make this presentation, I must update this screen over here because loss development keeps on occurring.

In the last four years, the property/casualty industry has had more losses than in the previous 20 years altogether. I'm told by some of my communications people that the last five years has had more losses, inflation-adjusted losses, than the last 25 years. It has been significant. During this four-year period, from 1989–92 over here, in addition to having tremendous catastrophic losses in the insurance industry, the GIS business was starting to get really good. The U.S. Census Bureau put out a file called "The TIGER Files," which is a street and road database that has longitude and latitude coordinates appended to it; it created a tremendous ability in the U.S. to locate addresses.

In addition, a lot of visualization software came up; Map Info, Strategic Mapping, ARC Info, things like that. At the same time that the industry was experiencing tremendous losses, the ability to deal with that information was also coming to light in the technological innovations that were coming up.

Insurance Services Office (ISO) began researching how insurers could use GIS technology. We came up with scores of applications, from underwriting and rating to disaster response. We have actuaries using some of our products right now that we developed for claims, strategic planning, and reinsurance. We spent about three or four years, and we're still in the midst of research. What I'm going to do is tell you about that research.

The benefit of geographic technology (and this kind of sums up the research before I get into it), is that it provides more complete hazard information. That's a major benefit. Let's say somebody was underwriting an automobile policy and had a number of questions. Number one, what territory is it in? Well, maybe you rely on your own information. Maybe you look it up on a map. Maybe you rely on an agent to tell you for rating purposes.

You get more complete information from a geographic information system than you ever had before. If you're underwriting an automobile policy, you're also interested in perhaps the drive distance from the insured's residence to a work address. How do you get that information right now? Rely on the agent. Rely on the information the insured gives you. With the geographic information system, the technology gives it to

you. You'll have more accurate information, again, similar to the drive-distance experience.

Finally, it reduces expenses. As part of the research that we were involved in, I personally sat in an underwriter's office up in Hartford, CT. At about nine o'clock in the morning, we first started off. A potential insured policy came across and it involved underwriting earthquake risk. I wanted to follow this because it was one of the things I was interested in. What the underwriter did was take a look at the location of the risk, identified in fact that it was in California, and didn't have enough information to go from there. He made a phone call at noon when the California office opened. The person they were trying to get in touch with wasn't there, so we got back to them a couple of hours later and told him about the policy, and what kind of information was needed. The underwriter in California said, "OK, I'll take a look at it and I'll get back to you." He got back the next day.

In terms of expenses, you had three phone calls and what I estimated to be about one hour's worth of staff time involved on either side in Hartford and California. We estimated the expense to be around $75. Right now, you'll be able to get extensive earthquake information (including estimates of expected losses, maximum probable losses, maximum foreseeable losses, and a comparison between ISO loss-cost-by-territory and a site-specific loss cost, which many actuaries in the property business are starting to become very interested in), in a matter of three to five seconds over a network. If you go all the way to the probable maximum loss, it would be at a cost of about $35.

In the course of our research, we tried to summarize all of this because we started making presentations on it. I found about seven major components to what we call insurance geographics. I'll spend the rest of the presentation talking about each one of them in turn.

First, point geolocation. The consumer products industry doesn't really care that much about point geolocation. Point geolocation means the ability to take an address, 123 Main Street, Anytown, U.S.A., and find the exact, or as close as you can get to an exact, longitude and latitude coordinate for that address. Colgate Palmolive doesn't care about it. Colgate Palmolive cares about census tracks. They want to know where they're selling their products right now, because there's an endless variety of social characteristics associated with the U.S. Census Bureau and the census file, and they can use that in their marketing. But insurers need to know where a risk is for a whole slew of purposes, we'll talk about that.

The second component of insurance geographics is mapping. By the end of 1994, ISO will have mapped over 50,000 territories that property/casualty insurers use every single day, such as distance calculations—the distance between two points, let's say a residence address and a work address, (I'll talk about that later.) The distance between a point and a line, let's say, the difference between a resident's address and the Atlantic Ocean. These are the kinds of distances that insurers are interested in—the distance between a resident address and the nearest fault trace.

Geological features. Again, fault traces for earthquake. You've got soil type, liquefaction, the fault itself, landslide potential, these kinds of things. Databases that

you folks never thought you would use because they were the realms of other disciplines will be used in insurance geographics specifically for the insurance industry. Insurance answers versus maps. Right now, many underwriters use maps. I haven't been around forever, but I still remember people using Sanborn maps, for example. It's time-consuming. It's expensive. The maps that many people use right now are not ground true, but insurers will still make use of maps, much better maps in the future. Sometimes, though, you won't want a map at all. For example, if I want to know what territory a homeowner's risk happens to be in, certainly a geographic information system can produce a map. But I might not be interested in looking at that map, because right now I can go to my own map and look at it. Just because a computer screen prints the map out for me, as opposed to my going to a Haystrom map; that doesn't really do anything for me. I need a system that comes back and tells me that territory equals 102. Sometimes you need maps and sometimes you don't, and our research hopefully shows the difference between the two.

Finally, hazards and other databases. As I mentioned before, you've got earthquake hazards and environmental hazards. We're in a partnership now with a company that has geo-coded the environmental risk locations, culling over 600 databases from the federal, state and local levels.

Finally, risk concentration analyses. How many risks do I really have in Homestead, FL? The claims people will be interested in that. Actuarial folks will be interested in that. The underwriters will be interested in that. The marketing people will be interested in that. So there's a whole slew of uses. We could spend a couple of hours talking about nothing but risk concentration analysis.

Point geolocation, 123 Main Street, Anytown, U.S.A. There's a thing in property/casualty called public-protection classification. Usually, at the fire-district level of observation, you can have up to one, two, or three public-protection codes associated with it that are used exclusively in rating. Almost every carrier that writes in the U.S. uses this information, but it's difficult to get at. You have to believe people. For example, you have to believe them about a lot of things. You're sitting in an office in Hartford, CT writing a policy and the risk happens to be in Florida. First of all, what district is it in? How do you know what fire district this happens to be in? How do you know what codes are associated with that district?

The drive miles between the risk address location and the fire station location is very important, because, in most states, if it's fewer than five miles, it's a potential for one code. If it's greater than five miles, it's a potential for another code. The relative proximity of the risk address to water, like a hydrant, is very important. If it's within 1,000 feet, you get one code. Sometimes if it's greater than 1,000 feet, you get a different code. Then sometimes you have an adjacent fire district, that has an automatic aid agreement with another fire district to the extent that they might cover some of the property that's located in there and some of the risks. You need to know all this. In the U.S. alone, there are some 36,500 fire districts that are being geo-coded right now, which insurers will have access to because of GIS technology.

Territory mapping. We can go on and on about this. I mentioned that there are over 50,000 territories that ISO alone is mapping. If you put all the insurers together, you've got hundreds of thousands of separate territories. Right now, we're only

mapping the PPC, or the fire districts, and the ISO territories. We've got 50,000 of those alone, anywhere from homeowners to automobile, to group-two commercial and, as I mentioned, the PPC codes, etc. We can also estimate crime at the census-track level of observation, so there are some more territories.

In Florida and eight other southeastern states, you have wind pools or CAT plans or beach plans. The wind pool that exists here in Florida is at the street level of observation, and so you have a thick manual that you can deal with, or you can go into a GIS system and get the information out of it. You've got territories associated with earthquake and with flood. There are over 100,000 FEMA flood maps that will be digitized, probably not by ISO, but will be used in an ISO system, and will be digitized in the near future.

You've got brush-fire zones out in California. The California Department of Forestry is doing what has been done in southern California for territories throughout the entire state: public protection classification, as I mentioned before, and finally for crime, geographic distances; I mentioned this before.

ISO's statistical database for personal auto shows that about 70% of the risks are rated for pleasure use. They drive three miles or fewer to work. However, the Insurance Research Council recently published some reports showing that about a third of public respondents to survey admitted that they thought it was acceptable to underestimate reporting mileage on insurance applications. Our internal research shows that the insurance industry alone is probably leaving about $500 million in premiums on the table every year because of underreporting of drive distance information on insurance applications.

We project that two, three or four years into the future 30% of commercial lines risk will involve some assessment of environmental hazards. The distance from the risk itself to the various environmental hazards will be important. The lenders and the banking industry are already regulated on this by the government, so most commercial properties are already purchasing information from vendors.

Geological features. I'll talk about earthquakes. Right now, you need information such as fault traces, soil types, soil liquefaction, the potential for soil to take on water and have ground failure. All this information used to be in the realm of forecasters. You've got people like the American Institute for Research (AIR) out there, EQE for earthquake, RMS, and all these companies forecasting. All this information will be readily available. As I said when I started, actuaries, claims adjustors, and underwriters will have available to them what might have taken a day or two and a number of phone calls going around the country to acquire. They'll have that available in three to five seconds over a network.

So far, I've been talking about geographic information in these kinds of systems, but you can go far beyond that. This also pertains to hurricanes, so I can talk about them simultaneously.

For earthquake, take the geological factors—the soil types or liquefaction, the distance to fault traces and that kind of thing. On the hurricane side, take into account the specific location of the property, the distance to ocean, the potential for surge

damage, and that kind of thing. Then add to it specific property information—like the age of the building, construction type, number of stories, some minimal information about that—and you can produce a probable maximum loss estimate, or a maximum foreseeable loss estimate, or an estimated loss estimate.

You can go any route you want. From an actuarial perspective, I know different people have different trains of thought in terms of what they like to see. In addition to the PNL-type estimates, you'll also be able to compare territory loss costs, say ISO territory loss costs, to site-specific loss costs. Once again, all this information is available in three to five seconds over a network.

Insurance answers versus insurance maps. A couple of months ago, I completed a study for a company that writes a lot of business in New Jersey. There was a little internal debate going on within the company about the exposure to wind-blown water or hurricane damage. One camp was saying we were fine, and the marketing camp was saying, I think we're overexposed and we need to get more agents on the interior part of the state and not concentrate on selling so much along the coast.

When we completed the analysis, and we created a fairly decent visual representation of what we found—where the red boxes indicated a high concentration of insured premiums, the yellow boxes indicated a moderate concentration, and the green boxes indicated fairly low concentration—that it was red/yellow/green going from the coast inland almost all the way down the entire state. As a result of conducting this analysis, which cost about $1,200, the entire marketing strategy for the state of New Jersey was changed.

Let's step back for a moment. Here's an example where you need maps. Maps are crucial for insurers. Without this kind of ability to visualize the data, they would not have changed their marketing strategy for the state of New Jersey. On the other hand, an underwriter probably doesn't care about maps. An underwriter wants to know, for example, this happens to be a risk in Myrtle Beach, SC. They've got a wind and hail pool there.

The underwriters need to know, is the risk in or out of the underwriting and hail zone? In this case, it's in. The underwriters want to know the extended coverage zone for the personal lines territory code. They give it their beach code of 003. They want to know the group-two zone or the commercial territory code. They want to know the distance to the ocean or gulf, which in this case is 0.06 miles to the Atlantic Ocean. Table 3 is an output screen from a product that ISO sells, called the Geographic Underwriting System. Finally on this screen we've got information about distances to various historical storms. This is a product that's widely used in the industry right now for hurricane exposure. Again, it's not a map; it's an answer.

Hazard and other databases. Here's where it gets kind of fun, because you can say to yourself, "I can buy a simple GIS system. I can go out and buy one or two, like Map Info." By the way, if you haven't bought stock in Map Info already, I think it's probably too late. It's been shooting up like crazy lately. The insurance industry is going crazy over it. The banks are going crazy over it. I think the college kids are starting to get there, too.

TABLE 3
GEOGRAPHIC UNDERWRITING SYSTEM
WIND-DETAILED

RISK ADDRESS: 5429 N OCEAN BOULEVARD MYRTLE BEACH, SC 29577-2350			
		<CONFIDENCE>	<REVISION> <DATE> 07/01/89
Wind & Hail Eligibility:	Yes	High	
Your Company Extended Coverage Zone—Personal:	Beach	High	
Your Company Personal Territory Code:	003	High	
Your Company Group II Zone—Commercial:	Beach	High	
Your Company Commercial Territory Code:	269	High	
Distance to Ocean or Gulf:	0.06 miles Atlantic Ocean	High	
Historical Wind Events as of:			
DISTANCE TO STORM CENTER	MAGNITUDE	TYPE	DATE
3.6 miles	92 mi/hr	Tornado	04/1978
3.6 miles	92 mi/hr	Tornado	09/1979
4.3 miles	95 mi/hr	Hurricane	08/1960
4.5 miles	135 mi/hr	Tornado	08/1968
9.6 miles	135 mi/hr	Tornado	09/1979

The ability to take a file like the U.S. Census Bureau's TIGER Files. I think if I remember what TIGER stands for, it's topographically integrated referencing and encoding files. Basically, it gives you a line segment, and on that line segment you've got a longitude and latitude coordinate on one end, and a longitude and latitude coordinate on the other end. You've got address information, such as the name of the street, and the line segment represents part of the street. You've got the zip-plus-four on either side of the street, and some address information. You can find out where things are.

In addition to that, you can add all sorts of other databases. For example, you can add a database that contains the polygons or the territories that I talked about, all 50,000 plus of them. You can have a database in there that contains the longitude and latitude coordinates and an impact analysis on each environmental site that happens to exist in the U.S. You can have another database out there that has the hurricane information and the earthquake information. If an insurer wants to build something like this, you certainly can. It's an extremely expensive thing to get involved in, but if you're really going to do a good job on it, you need all those different databases in there talking to each other. I guess we're going to have to change our tune because, right now, we've got that running on a mainframe, but we'll get a little better as time goes on.

Interestingly enough, we have one major carrier right now that said that it doesn't want to use our system. This carrier wants to port all our software to run on its own client/server technology because it's afraid of response time, but you need to link all these databases together. There will be vendors that will do it. ISO is doing it right now. There will probably be other vendors that will crop up and do similar things in the future, but the industry is big enough. You will be served very well by vendors, or if you choose to do it yourself for certain applications, you'll be well served by the tool vendors, the mapping companies and the data companies, that sort of thing.

Another database that's very useful, and that most of the people in your marketing departments will be familiar with, is the census files. It has, as I said before, this endless variety of social characteristics associated with it, where you have home values and income levels, household counts, the number of persons per house, the percentage of people who own cars, and so on and so forth. All that data should also be ported into a GIS system for the insurer's use.

The very success of GIS in industries outside of the insurance industry is contributing to data that will be available within the insurance industry. Government at the state, local, and federal levels is working on disaster preparedness GIS systems, so that information will filter into the insurance industry, including fire services, and health services. I've got a friend who works for Metropolitan, and they're using Map Info right now to geo-locate all the hospitals and all the physicians. I see some heads nodding over here. GIS will make real inroads into that side of the business.

There's sanitation and building codes. I'll show you what I consider to be the most effective picture I've ever seen in the property/casualty industry about property assessments. All these databases are just becoming greater and greater, and they'll all just filter through the insurance industry through uses of GIS technology. Here's a picture of Homestead, FL. If it weren't so sad, you would almost get a laugh out of the irony in that name. You've got two sets of homes over here, and you've got a street running up the middle. On the top, these homes have been completely demolished. There's relatively nothing standing over there. The roofs were all blown off. The walls have fallen down, etc. On the bottom part of the screen over here, you've got very little damage. It looks like a little roof damage over here. All these homes in here look fine.

There was one variable that separated these two groups of homes, and it wasn't that the storm bypassed the bottom homes. What was that variable? Building-code enforcement, exactly, a tremendous variable. For those of you who are familiar with public protection classification, ISO and its subsidiary, Commercial Risk Services, survey fire departments for their effectiveness in putting out fires and their response to fires. We have a class system based on a scale of one to ten, with ten being the worst.

We're doing the same thing now in conjunction with the NCPI for building-code enforcement, so you'll have a PPC code associated and you'll also have, for rating purposes, a building code associated with it that will be held to administer if you don't have a geographic information system. The kind of gerrymandering that goes around in setting up zones for building-code enforcement puts fire district zoning to shame.

That kind of information will also be built into a GIS system as soon as it becomes available for your use.

I guess to sum the whole thing up, geographic information system technology is a very simple kind of a thing, if you've got some simple uses. You can buy some software and some data. Let's say for the state of Florida, I could get you set up doing some simple kinds of analyses for about $2,000. However, if you want to build in all the databases that are of interest to you, all the things that we've been talking about so far, and also build all the modeling, the PNL modeling that we're talking about, that kind of a system costs many, many millions of dollars. It's a very expensive kind of operation.

The GIS system that we envision being most appropriate for the insurance industry not only combines all that information into a GIS, but also spits it out or ports it to whatever application you need, whether you need the information to go right back to a workstation, and many underwriters are using something like that right now. Point-of-sale underwriters are doing things where they're entering in work addresses and home addresses. They're getting the drive-distance information. They're pulling down possibly some claims information at the same time in an integrated fashion, and they're doing point-of-sale underwriting.

You might want to have it spit out the reports or maps, as in the case of the New Jersey company that I talked about. Finally, you might want to update information as PPC codes change quite a bit. Some things don't. Distance to ocean probably won't change for awhile, but many things do change. It might happen sooner than we expect, at least in our lifetimes. You might want to have your policy files updated on a continual basis. The GIS systems that we envision will also be doing that sort of thing.

I think I covered risk-concentration analysis already. I think it will be used increasingly in treaty negotiations and market penetration analysis, sales and territory planning. It's just such an easy thing to do, and this is something you could probably do yourselves or with the assistance of a vendor. That's probably it. More disaster planning, generating maps, rapid response for claims folks, locating claims-handling facilities, that sort of thing. There are some things you can do very easily by yourself; there are some things the vendors will help you out with, but the technology is here to stay.

MR. MATEJA: We have two or three minutes left for questions.

MR. CARL HERMAN ROSENBUSH, JR.: I have a question mainly for Mr. Scheel. I'm basically an actuary, and I use programming a lot. I don't feel like I have the time to get into much of this subsidiary material, like how do I program in such a way that it can be brought up in Windows. I know there are some strange types that like to have a cute little picture associated with your name, so do I need to learn icons in order to do this? Just what should I do in order to get a system working within some of these modern, initial products that would be part of a large system?

MR. SCHEEL: That's a very good question. Let's say the sandbox that one would want to start in is to work very carefully with a spreadsheet environment, like Excel,

because you're able to write and develop things in what would be a very familiar environment. It will be very soon that you will be able to take things as you develop them in that spreadsheet environment and mount them directly into these things I called dynalink libraries. It's very difficult. Your question is quite germane. It's very difficult to give you a learning path to proceed on.

I would suggest that it may not be your responsibility to do that, but rather your responsibility to guide those who do have those skills in assisting you. I think that was the kind of theme I was trying to express. I have to confess. I don't have a pat answer to what you can or should do to throw off the learning curve. It's an enormous one. Your question is well put, and I'm sorry I don't have a convenient answer for you.

MR. ALBERT K. CHRISTIANS: I have a related question. You advocated the use of suites. I was curious about your also advocating the use of programming. It seems to me that the suites might, in fact, make programming obsolete. I've conjectured recently that you could build the systems, like the ones Vince was describing, using entirely the things that are included in the suite—a spreadsheet, a word processor, a programmable database—and that you really don't need programming any more if you're willing to pay a little bit of a price on performance, but that's not going to be significant very much longer. Do you agree with me that that's true? When do you think that will become a cost-effective way to build systems?

MR. SCHEEL: Fortunately, I will be retired by the time that happens. Programmers are a dinosaur-like commodity at this point. You're absolutely correct. The tools that programmers shroud themselves with and make mysterious to the rest of the world are disappearing very fast. The suites are going to accelerate that. The comment related to things like Visual BASIC for applications harkens back to probably the kind of programming environment many of you studied when you were back in your early college days. BASIC was the language many of you were exposed to, and it's a very easy thing to use.

The interesting thing is that much of the need for that language is disappearing because of recording devices. You click on this and that and so on, and the system will remember those kinds of things. You're absolutely correct. Programmers, particularly the definition of a programmer that I think of, where you find somebody who's given a task and who then goes off and goes through the enormous chore of schematicizing it and getting it in the mold for actual production, that whole paradigm of development is disappearing for fast, high productivity.