RECORD, Volume 30, No. 1*

Spring Meeting, Anaheim, CA
May 19–21, 2004

Session 3PD
Hot Technologies

Track: Computer Science

Moderator: Charles S. Linn

Panelists: Jerry Goedicke
Neil Raden
Lori Martin†
Craig Hunneyman‡

Summary: This session covers a variety of hot technologies, including wireless business solutions, data management for analytical applications and grid computing.

MR. CHARLES S. LINN: The Computer Science Section sponsors sessions at all of the Society of Actuaries' meetings. We periodically present a generic hodgepodge that we call "Hot Technologies." We solicit feedback from section members regarding what topics are of interest to them. This year we've come up with three topics that are rather diverse. Maybe some of our speakers will find common links among the three of them.

Our first speaker today is Jerry Goedicke from Mobitor Corp. He has more than 20 years of experience in the software industry, most of it in financial services. He cofounded Mobitor after a successful tenure at Application Partners, Inc., where he served as president and was responsible for the company start-up through the profitable launch of the company's products. Previously he was vice president of Business Development at Trident Systems, Inc., and held executive, operational and business-development positions at Impell Computer Systems.

MR. JERRY GOEDICKE: I'm going to talk about wireless technology. We'll do a little Wireless 101. What exactly do we mean by wireless? Hopefully some of you have been exposed to it; you might have some wireless devices with you. Why are

†Ms. Lori Martin, not a member of the sponsoring organizations, is with Platform Computing.

‡Mr. Craig Hunneyman, not a member of the sponsoring organizations, is with Platform Computing.

companies looking at wireless applications? I'll try to point out what types of applications specifically apply to the insurance business.

Typically we talk about three types of wireless technologies. Some of you may use these technologies in your own devices. The first is Bluetooth. Bluetooth is really a wireless technology for personal-area networking. I'll discuss why that is and what types of applications use Bluetooth. Then there's Wi-Fi or 802.11. Many of you have home networks that are based upon this technology. There also are the cellular-data networks. Those involve sending data over existing cellular-data networks that are being enhanced for greater and greater speed. We use them for wide-area networking (WAN). So we will cover personal-area networking with Bluetooth, local-area networking (LAN) in your home or business with 802.11, and the WANs of cellular-data networks.

How does a Bluetooth device work? Typically you get a Bluetooth adapter, which usually connects via a USB cable. The advantage of Bluetooth is that it's a very low-cost, low-power, small radio transmitter. It can be incorporated into a lot of peripheral devices. It has a range of tens of feet; you aren't going to get great distance out of this transmission. And the speed is fairly good: you get about one megabit per second out of it as a maximum, but most of the time we see that it's more like 700 or 800 kilobits per second. We use wireless Bluetooth to connect things like peripherals to our computers; we eliminate the cable. You can get rid of the cable to your keyboard on your home computer, you can get rid of the cable to your printer, you can get rid of the cable going to projectors that have been Bluetooth-enabled. A conference could be set up with various Bluetooth-enabled devices, and you wouldn't have to run wires under the carpet. That's the principal application of Bluetooth. Again, it covers tens of feet, so we call it a personal-area network.

With Wi-Fi technology or 802.11, we jump up in range. We get hundreds of feet of transmission from this technology. Many people have gone to the store and purchased home networks. There's a variety of technology types. The 802.11B technology is probably the most common one out there; it has a theoretical maximum speed of 11 megabits per second. There is a different protocol with 802.11A and 802.11G. Most of the time now you're looking at 802.11G because it's faster. It's like five times faster than 802.11B.

The primary application for this is LANs. Now you can set up a network in your home without having to pull wires through the basement to the attic. On a business campus you can set up a network, and by bridging various access points, you can get coverage across a very broad campus. You can get a very wide-ranging network. The same technology is used at what we call "hot spots."

The third type that I mentioned was cellular-data networks. There was a lot of hype about this three years ago when you heard the terms 2G, 2.5G and 3G. All of the cell-phone makers were going to build out their networks using 3G technology.

Well, that build-out hasn't happened as fast as we had hoped, but it now is starting to come online. In fact, Verizon is the first company to have a true 3G network starting up in some metropolitan areas. You can see the advantage of that. The speed is much faster than the original 2G networks. Most of our cell phones are 2G or 2.5G.

The speed comparisons are from 14.4 kilobits per second (which is fairly slow) up to a potential of close to five megabits per second. There is a great speed improvement as the 3G networks are built, but we have very limited access to 3G now. A lot of upgrades have been made to 2G: we call it "2.5G." Wireless phone carriers mix the terms a bit and use their own terms. The most important thing is that, with the cellular-data networks, you can start to build broad applications in various industries, including the insurance industry. You can start to extend the corporate systems out, rather than just use them in your campus setting.

You can extend them through the cellular-data network and get national coverage, because you're broadcasting data, sending and receiving it on cell towers. We all know that roaming works. It works on our cell phones, and it works on these data networks. So you actually get wireless coverage and can connect to the Internet, connect to your business, connect to your business applications as you move about regionally, in a city or within a state or even across the country. We'll come back to that and talk about some of its insurance applications.

Here is a comparison of transmission speeds. I use kilobits per second and megabits per second. Let's try to relate this to some things that we're familiar with. Almost all businesses have an Ethernet-based network. Probably most of you are on a 10-megabit-per-second Ethernet network. Now a lot of companies are deploying hundred-megabit-per-second Ethernet networks. There are actually gigabit-per-second Ethernet networks out now as well. But a huge disparity exists between what we have in our wired connections compared to some of the other wireless technologies. I can remember when I got my first modem. It was a 9,600-baud modem: that's 9.6 kilobits per second.

If you go back to those speeds that I talked about for 802.11 and 2G, they're in this low-speed range. Some of you may have ISDN networking in your home, which is about 56 kilobits per second. That's the speed we're talking about. As we move up the ladder from 2G, 2.5G, 3G to 3G-enhanced, we're getting a factor-of-10 improvement.

As wireless networks are built out, they can be used very effectively to extend data. But we have to start thinking about speed differentials. We can't just take your existing applications—whether they be in the insurance company, your underwriting applications, your claims applications, your financial applications, etc.—and expect them to run wirelessly the same way that they would in the office over a wired connection, because there's a huge speed differential. Therefore, when you start looking at wireless applications, you have to architect the wireless applications to be more effective in the transmission.

Bluetooth is suitable for a personal network; 802.11 is suitable for a LAN; and the digital-cellular networks are more for WAN applications. If you start reading about wireless data networks you might read about real-time enterprise making everyone constantly in touch—mobile collaboration—moving the point of service and point of sale to where the customer is. Some of these things would be important in the insurance industry: giving field claims adjusters instant access to information so that they are able to write checks on the spot, having their computers connect to your financial systems. This would allow you to get mobile assistance quickly to people in cases of catastrophe. All of these issues are important in the insurance industry. Even on the sale side, you can have your agents make rate quotes on the spot in the customer's home at their kitchen table. Or your wholesalers can go out and meet with your channel partners and be very responsive.

So again, you see Bluetooth in connections to peripherals, etc. We see that 802.11 is obvious: we have the various cards that connect into our laptops. There are also Personal Digital Assistants (PDAs) and pocket PCs coming out onto the market. Some of these new pocket PCs and PDAs have built-in 802.11. Many laptops today have built-in 802.11. So you don't have to add adapters. Cellular-data networks now have pocket-PC phone additions. There is a cell phone with built-in wireless data for e-mail and applications, etc.

Why do we go wireless? There are clearly a lot of mobile field resources in sales and service. If you look at the job functions of mobile people, they have more wasted time than any other function in your company. Many of your back-office operations have been reengineered, and you've squeezed the last bit of productivity out of the resources in the office. They're spending all of their time at the desk. But staff with mobile resources spend more time away from their desk than they do at their desk. Many of them wait in the lobby to see their next appointment. Clearly we can gain more productivity in the field force by deploying these wireless applications and making the users more productive and efficient.

When talking to companies, their people and resources, what's the biggest complaint from the mobile field force—whether it be on the sales or service side? It's the administrative burden. Perhaps I've made five appointments today. I have to come back to my home office and type in all of the information from my customer meetings, service events or claim meetings. That's an administrative burden. We can move that administrative burden out to the field and take care of that on wireless devices.

What's the biggest complaint on the customer side—whether it be independent agents that your wholesalers are calling on, or customers that your agents are calling on directly? Most of the time it's unresponsiveness. This just happened to me recently. My agent retired, so I was contacted by the new agent. The new agent told me that I had to come down to the office to go over my portfolio and so forth. I had to go to their office? I'm a very preferred customer in the insurance industry, with a fairly good net worth. I had an umbrella policy, homeowner's policy and three car policies with the company, and they wanted me to come to the office? I told them that I was researching other insurance companies. I got other people to come to my house to give me quotes. Three different agents came to my house, they opened up their briefcases, they had applications sitting in front of them, they had PDAs that contained only their calendars and appointments. (They didn't have any type of connectivity to their corporate system.) And they had a cell phones.

They each spent a half-hour with me, filling out paperwork. They each then said, "I'll get back to you in a few days with a quote." Get back to me in a few days? Of course, you know what happens with that. Two or three days later, I get a voice mail: "I'll try to get hold of you, I'd like to talk about the quote." I called him back. Guess what? He was out of the office. Two-and-a-half weeks later, I finally got a quote. And, clearly, I'm one of the people that insurance companies would want to be very responsive to and get to switch policies over to their company. Those are issues that we can take care of with wireless technology. We can remove the administrative burden from people so that they can make more appointments in the field and be more responsive when they're interacting with channel partners and companies.

Insurance companies have deployed laptops, trying to make their people more effective in the field. But I'm a road warrior. If I can go out on the road and stick a PDA in my pocket, my laptop's not going to come with me. That's exactly what's happened. The people taking their laptops on the road are the ones who have to fly on airplanes. The regional salespeople, the people who drive around in their cars, keep their laptops in the office. Insurance companies have spent a lot of money deploying laptops, and they're sitting in the office. It creates a Catch-22. More face time with the PC equals less face time with the customer, and vice versa. So where do you spend your time? Fulfilling the administrative burden in front of the PC, or talking to your customers?

You expect to get a return on investment (ROI) and increase productivity with the deployment of laptops. You can get that now by deploying real mobile devices, a real mobile computer. They have a lot of computing power these days. The new pocket-PC devices coming out have 400-megahertz processors. My old laptop doesn't have a 400-megahertz processor. You can get a gigabyte of memory in these things. A lot of them have cameras, built-in cell phones, built-in wireless e-mail. They are very powerful devices. We can extend these applications and really get the productivity and ROI that we expected from laptops.

There are significant opportunities across many industries, including financial services, to recapture that time in the field. We've seen a lot of studies saying that 40 percent of a person's time is wasted in the field. Obviously there is an enormous amount of opportunity to increase productivity in the field. By being more responsive and having data literally in the palm of your hand, you can be more responsive to your customers and improve those meeting outcomes and minimize the administrative burden when returning to the office.

How does that translate to the insurance industry? We've talked to a number of companies, and here's what we envision as wireless applications. I talked about equipping wholesalers so that they can meet with reps and describe products and get them information and feedback instantly. And the representative can request literature overnight.

What might some of these applications look like on a device like this? I'll give you some examples from our system. We don't really need to go into the architecture. In the hands of a life agent, they could have, literally in the palm of their hand, a complete understanding of their customer, their complete portfolio. What types of products does my customer have that are from my company? What kinds of products do they have from other companies? What are their financial investments? And then they can go through a whole questionnaire. Let's start the straight-through processing application of getting a life quote or do some financial analysis by asking a few simple questions. That's taking that administrative burden out into the field when interfacing with that customer. These PDAs also have college calculators. So it offers more than quotes. All of this can be run on these types of devices.

You might have a marketing manager calling on independent agencies, reps in the field and so forth. In this way you can extend product information that you're deploying. You might want your agencies to sign up for a call center that's manned by your company. You can take away customer-care issues from the agencies and provide a centralized service. What does that product mean? What does it mean to the agent? How much time does it save them? Again, all of this information can be delivered: What are the productivity statistics or production statistics of that agency? What's the premium in force? How many cancellations did they have last year? How many claims did they have last year? It can help you with all of the things that you need to know to refine your products and be more competitive in the marketplace.

In the hands of a wholesaler, there are very similar types of applications. The device gives you a 360-degree view of your customer. Then you can do something about it, capture information, deliver it and upload it to the systems that are run in your back office that other people need to see.

When it comes to claims adjusters, you could be out in the field and have a programmed process that you want your field adjusters to follow when giving relief to people who have just had some horrible event happen to them. Your claims adjusters can be out in the field, delivering comfort and the checks, understanding everything about that particular customer.

In terms of the wireless technology, the bottom line here is it's a technology that's now moving beyond wireless e-mail. It's an opportunity for insurance companies to adopt this technology and deploy systems out in the field, in the hands of their field resources, and get a competitive advantage.

MR. LINN: Our next topic is going to be grid computing. We have two people here from Platform Computing. Platform Computing is one of the premier companies addressing grid enterprise computing, with over 1,500 customers worldwide. Our two speakers are Lori Martin and Craig Hunneyman.

MS. LORI MARTIN: The first layer of grid networking is called load balancing. Platform Computing is a 12-year-old company. Our main product is a "middleware" that sits between the hardware and the application. In your business it would be various financial applications.

You would put the load-balancing software onto the hardware; it can work on any platform. Our software can transparently detect if one of your central processing units (CPUs) is idle. You might have a few pools of computers in your department. The guy next to you might be working on a server that has 16. You might be working on a desktop, and you're doing simulations or modeling. You're trying to get results, and it's too crowded. You want to get it done faster. Our software allows you to go into the idle computer and send work there. So if you've got a return that takes 10 seconds, you could distribute the work and get that back in one second.

The next layer would be the analytics. Reporting is important to you: you require more complex reporting; you have to get the results back faster. There's another layer that sits on top of that that will give you all of the various reporting on the work that you're doing, to the point where we use technology to do a correlation against your work. All of this is done in real time.

There's a final layer that allows you to share two servers. So let's say that the servers are in some person's office. Let's say they're across the hall. Let's say one's in Paris, one's in San Diego. It is just two servers. It doesn't matter where they are. The third component allows you access to other CPUs.

J. P. Morgan, for example, has grid-enabled 1,000 nodes. They had seven applications that we could hook them to; they had seven groups that said they wanted to participate. They want to have access to those 1,000 nodes, and they don't really care which nodes they get. They just want to have access. Based on that, they shared the cost of using it. They doubled the speed of their application.

J. P. Morgan had a process that took 18 hours, but by grid-enabling that process, they got it down to 32 minutes. By being able to access idle resources, having your applications integrated and spread out, you can really get reliable data back faster. Most people start out with one department application and take baby steps; they don't go "hog wild." By implementing the grid, they see a linear improvement in terms of performance first.

MR. CRAIG HUNNEYMAN: Instead of just working on your own PC, you want to have access to high-powered machines, and maybe your neighbor's desktop. Your neighbor may go on vacation for a week. Well, his or her computer is just sitting there doing nothing. We can access all of the computers on any network, whether it's a network internally on the same floor or different departments across different offices. You may have an office here, you may have one in San Jose, you may have one across the world in Paris. We can access all of them at the same time. When people in Paris are sleeping, people over here may be working.

Those machines aren't being used all of the time. We access every machine in what we call a cluster. We want to show you from where you can perform your workload. Our software is a layer that hooks up between the hardware and software application. We make sure that the data travel wherever the compute node is, because, obviously, your data sets could be on one little PC. You've got to get that data set over across the network to whatever machine it's going to run on.

We access all kinds of servers and PCs—Windows, Linux, Mac OS X or any of the Unix operating systems. We make sure that all of your work gets done as fast as possible. It could be a two-week job. We make sure that we pick the fastest machine that's available for your workload, and we try to match your workload to the best-suited machine in the network. It's up to our software to determine where that is. You, as a user, don't even care; it is transparent to you. You just submit your job like you normally would.

We have the grid manager. With a very simple command, you have access to this grid manager. Our grid manager takes care of computing, even if it's dedicated. If you have silo machines for one application, we can break that up to make sure that it goes to all of the other machines out there.

If your job goes sequentially, you have to have one job run right after another. They're independent, but they can be run on only one machine; you can run only one at a time. Or if you run on a PC, you may submit a whole bunch of them at a time, and your operating system has to break up maybe 50 different jobs. You have to break up the data and the operating system memory, because the memory may not fit everything. Then the operating system is really managing your job. We have a layer on top of the operating system. We intercept that job and break that down. This machine is going to have this work; this machine's going to have that work. Even on a single machine, some grid technology is such that you want to break your code down and run only one part at a time, because you may have only one CPU and it may only fit in memory. So you don't want to have the operating system take the burden and the overhead of doing all of the work to make sure that your jobs get done.

What about a case where you run sequentially? Before you're grid-enabled, you run one back to another. So it takes *n* times. After a grid is enabled, you can submit your job and have them all running at the same time. Instead of *n* times, you've really done 100 jobs within one set time, because it's all run on 100 different machines. So that's what the basic concept is: doing your work but doing it much faster with grid enabling and using every resource within your network to do it.

More and more people are computing more and more data. Data are getting much cheaper. The megabyte rating is going down; you get more gigabytes at a time. Memory is going up on CPUs as well, and so are the CPU speeds. What you could do 10 years ago with a "supercomputer," you can do on one PC now. We're finding out that more people have all of the CPU resources. We're getting bottlenecks in data, because a lot of people have to have data. You have to move data around. We have some customers that use gigabytes of data at a time. Transferring that from one machine to another machine could create a big bottleneck.

Smart grid computing makes sure that you run the job that's appropriate to the machine and the policies that you may have, to make sure that the job gets completed in the fastest time possible. You don't want to transfer all of your 10 gigabytes across the network, across the world to Paris, when it's going to take you much longer just to transfer the data before you actually run the job. Then you've got to transfer data back. You always have to be aware of where your job is running and what kind of information has to be sent back and forth.

We have client nodes, management nodes and compute nodes. The concept of grid computing is such that any machine can go down in any of your networks, and your job will still run. Redundancy is built in. The fault tolerance is built in, so that it doesn't matter if you have 1,000 machines out there. A hundred could go down, and all of your work will still be done. It may get done slower, but you don't have to worry that your PC is unavailable, for whatever reason. You're accessing all PCs, all compute nodes.

When one goes down, any workload manager in a grid-computing environment will take over and figure out that a machine is down. It won't send any more work there, and the jobs that were running there will be sent to some other machine. A lot of redundancy is built in. When applications are distributed (like in a silo), we can group everything together. Instead of all machines running one at a time, we now can access fewer machines and get the same amount of work done. Instead of buying more machines, you use the machines that you already have. You could actually reduce the number of machines that you have and still get the same workload done in the same amount of time, sometimes even quicker.

You could use different services and applications. While the applications are running, you could have a server pool in another place. It could be your redundancy pool. If you have a workload that needs to be done at the end of a quarter, when you do more work than you normally do on a weekly or daily basis, you can have machines doing other application work that's not as critical as what you need to be doing. We employ those machines as part of the server pool. We can access those and complete your workload more efficiently and quickly than if you had your normal machines working on it.

Grids are built; they're not bought. You can't go out and buy a grid, you have to build it. And most people take baby steps by hooking up just a few machines, making sure that their applications work. Once they work, they want to start deploying more. After a while you start relying on the fact that everything is transparent. Things get done faster. It becomes the normal way of business.

Voluntary participation is another issue. There are grid technologies out there that we call "push," and some are called "pull." You have heard of programs for which you can allow another party to use your PC to find things out in the universe or see if there are extraterrestrials. We have that kind of technology. To engage in that program, you would open your PC to become part of this network. The outside party would input code on his or her hardware, to let the network know that it's available to do some work. But at any given time, you can say that you want your computer back. That is a system through which you voluntarily allow others to use your PC. The problem with that is, suppose nobody volunteers. That's the problem.

So there's something called "push" technology, for which all of the workload is given to a grid manager, and the grid manager determines that a machine is available. With push technology, every machine is aware of every other machine, what is going on and how busy it is.

As you scale, fault tolerance is important. If you have 1,000 nodes and 100 go down, you want to make sure that the work still gets done and none of the jobs get lost. If you submitted a job yesterday, you don't want to wonder where it went. You want to know where your results are. Good grid technology will make sure that you manage all of the paperwork necessary so that you know where your job is at any given time.

You also must think ahead. Sometimes you can't just combine one machine with another. You have to think about it. We're talking about moving data. You have to take into consideration how you get the workload exactly where you need it to be. That's important. For each application that you put on the grid, obviously, there is more and more payback. The more things you can do to make the grid aware of other machines on the network, the greater the payback will be.

Not that many people are aware of grids, but competitors may be using it. We know that a lot of companies out there are: we have over 1,500 customers. And "grid" is one of the big words today when it comes to going forward with technology. We can help do that. When you employ grid technology, you have a linear speedup. It's not really a one-for-one; there is a little bit of overhead, but it does scale linearly. You have to realize that when you start adding things, you're not really adding 100 machines to get a 40-times improvement. It's more like a 70 or 80. You've got to be aware of that. And the more machines you put on your grid, the better off it is.

For example, Royal Bank of Canada reduced a two-and-a-half-hour job down to 10 minutes by enabling a grid. Then they reduced an 18-hour jobs to 32 minutes. So you can see that this is compelling. By enabling a grid, you can actually get a lot more work done faster. Now, this leaves a lot of time to do more. By reducing computer time, you can do more complex work. Or you can do it with fewer machines and get the same amount of work done.

MS. MARTIN: And the grid doesn't have to be 1,000 nodes. You can grid-enable 100 machines or 50 machines.

MR. LINN: Our final speaker today is Neil Raden, founder of Hired Brains, Inc.—consultants, system integrators and providers of syndicated and custom research in the field of business intelligence, including high-performance data warehousing, advanced analytical applications, data mining, closed-loop decision-support systems, performance management and enterprise information integration. Mr. Raden is a practicing consultant and implementer and has hands-on experience as a designer, architect and modeler. He has been an author, speaker and industry thought leader. He began his career as a casualty actuary with American International Group, Inc. and has implemented actuarial data warehouses for life- and health-insurance companies in the United States and Canada. He considers it a privilege to have been a speaker at various SOA conferences since 1996 because actuaries are a terrific audience. They are oppressed by terrible IT systems, acutely aware of what they lack and possess a professional pride that is often lacking in other functional areas.

MR. NEIL RADEN: I speak at a lot of conferences, and I speak to consultants and vendors and industry people, and we're all just there to shoot each other down. But here I always run into people who have specific problems that need to be solved. And they don't really care about data warehousing or even technology per se. They care about being actuaries and doing a good job for their company and getting work done. So that's what I really want to talk about. I'd like to poke holes in some of the current conventional wisdom in my industry, so that you're aware of it.

I've always looked at the management of data and the integration of data for data warehousing as the smelly swamp of the business. I've tried to stay away from it and take the high ground and concentrate on the analytics, because that's what I like. Well, I have to tell you that in the year 2004, the integration and management of data are the most important thing. Tremendous things are happening that I want to try to explain. I've always looked at it like making sausage or violin strings. It has a nice end result, but you don't really want to know that much about the process that goes into it.

Who's familiar with Moore's Law? Gordon Moore was one of the founders of Intel Corp. Many years ago he suggested that the number of transistors that could be put on a silicone chip would double every 18 months. It turns out that he was right, and it's still happening. When it comes to something like data warehousing and business intelligence (BI), the application of Moore's Law, the pickup of that technology boost, is running at only about half speed. So I've renamed this law "Demi Moore's Law." The reason that it's not being picked up is that we have all of these sticky technologies—and what I mean by "sticky technologies" is relational database standards, or stupid ways of defining meta-data.

The concept of meta-data is one of the dumbest ideas ever invented. It *was* a wonderful idea, it *was* a wonderful purpose, but the way it's been implemented has been useless. It's just dead data in a drawer collecting dust. It doesn't do anything; it's not active. It's barely a catalog; it's barely a data dictionary. You know something about earthquakes: a fault line doesn't move all the time. Pressure builds up on a fault, and then it just kind of lets go. And that's what we're starting to see in meta-data.

Certain companies have thrown together a bunch of cheap CPUs (which you can get in cereal boxes now) and a bunch of cheap memory, and designed their own disk controllers (using plain old disk drives). They call it a data-warehouse appliance. They've used an open-source database that they've modified for their architecture, and they sell it as a big data-warehouse black box. It's massively parallel. When you run out of processing, you just buy another box, plug it in and "daisy chain" it. You just turn it on and start data warehousing. Of course, you have to figure out the data. But the point is that their nearest competitor is massively expensive. I just saw one proposal to a company for $60 million. One of these other companies runs their storage at one-tenth to one-fifteenth of that price and about 100 times faster. That's Moore's Law. That's what it does.

There's another company out there that can take any data-management process and parallelize it, no matter what your architecture is. And they're running 10 times, 100 times faster than the extract, transform and load (ETL) software. When it comes to data transformation, data cleansing, data profiling, they're running against mainframes, AS 400s, PCs, and you name it. They're a Moore's Law company. That's what's happening in this industry.

What you have in data warehousing that's so amazing is everything is based on physical data models. There's a thing called a data model, and there's a thing called a business rule. They're both relational-database concepts. The world is not made up of relational databases. God didn't create the universe with relational databases. Don't tell that to a data modeler, because they think that's what happened. A real business model is not a business rule, and it's not a data model. It's what you think about. It's what I used to think about.

We used to deal with problems back in 1978, 1979 and 1980 that were conceptual. What's the value of our company? What's our concentration risk? What should we charge for this new line of casualty insurance for petroleum drilling? How should we reserve for that? How should we handle portfolio transfers? These are business models. Rows and columns and entity-relationship diagrams and foreign keys and primary keys—those aren't business models. Data have been hijacked by relational-database people. My proposal is: we fire them, we get rid of them. Actually I want to retrain them to be good software engineers to build a new class of software to solve these problems, which is what I want to get to. I call this careless design. And the meta-data are DNA. All I really mean by that is, it has to drive everything.

We really need smarter software. It's happening. The databases are so much better than they were five or 10 years ago. Five years ago, trying to build a data warehouse using a relational database was like sticking needles in your eyes. The optimizers didn't understand the schema. They were not designed for those kinds of queries. You couldn't load data into them. Oracle was famous for this. They'd say that they had a parallel load process, except that you can't do indexing when you have a parallel load process. And the whole point of a data warehouse is indexing so that you can get queries answered. It took us only an hour to load the data, but it took us four days to build the indexes, because the two didn't work together. All of those things have been smoothed out, and the database vendors have vastly improved their tools. And that's a good thing.

We're in the third phase with ETL vendors. The first-phase products were just code generators. The second generation of tools were somewhat better. Now a whole host of different things are coming out. BI tools are the worst part of data warehousing. While the ETL tools have innovated, while the hardware has "gone off into the stratosphere," the BI tools are atrocious. They haven't gotten any better in 10 years. I walk into companies, and they're still using ancient toys that don't solve any problems.

I want to see data modelers working on real foundational problems instead of tuning databases for performance. It's just a waste of everyone's time. And because of Moore's Law, we really don't need to do that. Data models are like poured concrete: they're fluid while you work with them. And then by the next day, if you want to change anything, you have to bust it up with a jackhammer. The problem with these data models is that there isn't just one model. There are dozens of them, and they're all linked together. This impedes any progress in changing things.

How many of you have looked at a complex problem and figured it out and gotten it 100 percent right the first time you tried it? But that's what you have to do with the data model. You have to get it right the first time. It's an impossible problem. These things have to be solved incrementally. They can't be solved by a couple of genius data modelers talking to business users and getting requirements and building a data model that's going to last five years. It's ridiculous. It's just a dumb idea.

I did a survey last fall of 300 BI users in 66 companies in the United States. The survey was about how they felt about their data warehousing and BI implementations. These were people who actually had mature projects. We asked them only seven questions. Now I've done a much larger study of 100 questions for some vendors. Maybe some time later this year I'll publish some findings from it. But this much I can tell you right now: it's horrifying.

This is how much time a market analyst spent with the following tools. The BI tool: nothing. Excel, Access and what I'm generously calling data administration: a great deal. I call this kind of work "shadow IT," and it's terrible.

What about the suitability of the BI tool to the work they do? Eighty percent of respondents said that it's either "not useful at all" or "I could live without it." Now, this is after the companies have spent millions and millions of dollars on it. By the way, the IT people have congratulated themselves on what a success it is, and they've gone to data-warehousing conferences and gotten blue ribbons for these data warehouses. But look at what the users think of them.

Here's another thing we found. The percentage of the total budget spent on training had an almost-perfect correlation to how people felt about the project's success and usability. Well, does that mean that training is the key? Does that mean that there's a causal effect? No, because we found that the relationship between how people felt about the project and how they felt about the quality of the training had an almost-zero correlation. So it wasn't that the good training made them happy about the project, it was that the company spent the money. What does that mean?

In some follow-up interviews, you could take the cynical attitude that they don't really care about the project at all, they just care that the company spent the money. That wasn't it. It was that the companies that spent 15 percent or more of the budget on training were precisely those kinds of companies that were desperately interested in getting people to use information to inform their decisions. They created the kinds of environments in which people could be successful.

So here's the message. If you work in a lousy company, it doesn't matter how much money you spend on data warehousing. You're going to have a lousy result. I'm sorry to tell you that, but it's the truth. Now, there may be isolated areas where you can do some good work in some departments, but if you've got the kind of company where midlevel managers' jobs are basically to justify themselves every day, that's not going to be a good environment for sharing information. And that has nothing to do with technology.

These are the shortcomings of the people surveyed, cited by percentage. We got 251 respondents out of 300 to answer these questions. And 61 percent of them said that the worst part about it was that it just wasn't relevant—and in decreasing order: integration, understanding work flow and, surprisingly, ease of use and performance. The two things that we've knocked our heads against the wall over for the last 15 years turn out not to be big concerns, maybe because we were successful. I'm not sure that's the case. I think that when you give people things that aren't relevant and don't solve end-to-end problems for them, it doesn't really matter if the products perform well or they're easy to use.

Shadow IT is a disaster in some companies. This is the cost of people like you doing IT work, instead of IT doing it. It's a double whammy, because it makes your IT department look like geniuses because their costs are so low, and it makes you guys look like disasters because your costs are so high. You shouldn't be doing IT work; you should be doing actuarial work. You're happier when you do that. This has been my experience. When I've worked with actuaries, the minute we get them out of writing computer code, they feel much better about their jobs. The turnover drops. You see them working together in groups instead of huddled in their cube typing at the keyboard. That's not what you were trained to do.

What's wrong is how your IT organization looks at a data warehouse. They get to buy a really big machine; they get to build a really big database. There's a macho thing going on here. I know a lot of the IT people in your companies. They come to the conferences, and they brag (like fish stories) about how big their databases are. And what are you guys? You're "seats." That's the legal term in a software contract for how many people used the software. You are just a bunch of seats. What does a data warehouse look like to you? There's no technology in it at all. It's what you have to do for a living. How do we bridge this gap? It's not going to be easy. There's going to be a push from the technology side, and I'm going to show you what that is in a minute.

The way that data warehouses are designed comes from a bloated, out-of-date, obsolete approach, often called the corporate information factory. If you look at the architecture, there are at least seven different relational schemas in it. There might be 20, 30 or 40 different models in this corporate information factory, but it's sold to you as a data warehouse. If one thing changes anywhere in that process, everything is affected. That's why, when we say that because of Sarbanes-Oxley, we need to take a slightly different view of the way we calculate this, maybe 12 months later that will happen.

Can you spell "backlog"? That is no way to build a data warehouse in 2004. Very simply, this is based on an out-of-date notion that the computing resources that we have to work with are insanely expensive, and they're all running at capacity. We've got to squeeze the last little bit of performance out of them, because it's too expensive to get anything else. What I'm telling you is, they're giving away computers in cereal boxes now. You don't have to do this; it's careless design. Data modelers spend 70 percent of their time doing performance tuning. Data modelers need to spend 100 percent of their time building durable data models. To hell with performance, buy another computer. They're cheap, throw one on the grid: that's what I'm getting at. This is your enemy; this is the problem. These guys are nothing more than poured concrete. In the future all of the physical structures are transparent. You won't know anything about them. And guess what? The data modelers won't even know anything about them, because their smart software says that they are going to materialize this data warehouse on Tuesday, because it's the first of the month. We're going to put up 200 gigabytes of financial data in these structures and index them. We're going to leave it there for four or five days, until they close the books, and then it's going to evaporate until next month. Try doing that in a data warehouse, because everything that talks to these structures is talking through a conceptual model.

I like to use the example of an accelerator. I just got rid of a car, and it had a very interesting accelerator. It didn't have an accelerator cable, it had electronic sensors. When I stepped on the gas it sensed how far I stepped and how hard I stepped, and it sent this message to the engine management system that then figured out what to do and sent dual messages to the servomotors on either side of the engine to turn the throttles exactly the same way. Well, you don't have to know anything about how that throttle worked to drive the car. The accelerator is a conceptual model. All you know is step on the gas, step on the brake. That's where we want to be. Step on the gas and step on the brake. I don't care how the accelerator works. It's like steering: I don't care why it works. I hold the steering wheel, I turn it this way, or I turn it that way. But you can't do that in a data warehouse now.

How many of you have worked with a data warehouse tool, a front-end tool, where you actually have to program? It's idiotic! Why do you still have to do that? It's 2004. We are so done with that. This is all going to be a volatile transfer. Grids are a Moore's Law effect of data warehousing. As the capability to add computing resources to the mix becomes transparent, there's absolutely no reason to build these crazy models that can't be changed. Everything can become volatile—the tools that map data in and the tools that bring data out for analytical purposes.

Data integration is where we're heading. What's interesting about it is this notion of abstraction. The way that data integration works in companies now is, a bunch of people go out and try to figure out what the data mean in different systems. Then they try to pull the data together into some common definition. The big problem with that is that everybody knows what the data mean in their own domain, but they don't ever understand what the common domain means. Nobody understands it. Or worse, they think they understand it, and then they get it wrong. We don't want to have one version of the truth, because there isn't one version of the truth. I mean, even the Pope in Rome admits that there's more than one way to heaven now. There isn't one version of the truth; it's a fantasy. It's a contraption that was invented by IT people who have blinders on.

The conceptual model that I was talking about allows everybody to have their own version of the truth. That sounds kind of chaotic. But because of the way you would implement this, those versions are going to be managed so that when you generate a report or you do a piece of analysis, it's individual. It's not the company's analysis. So the things that work are all based on some kind of role base. What we're seeing now is fascinating: companies that are based on underlying, expert-system-rule engines that are driving ontology, modeling domains, using semantic Web concepts to tie things together in a way that makes sense.

Google what Tim Berners-Lee wrote about semantic Webs back in 1998. Go see what's going on with semantic Webs. The whole idea is to make things like the real world. Definitions aren't always so strict. In data warehousing we have put everything in neat little boxes. It hasn't served a lot of purposes. For proof, I offer you the results of that survey.

MR. LINN: Does anyone have questions?

MR. ROBERT KELLER: On grid computing, how do you prevent things like viruses from infecting all of the machines at once?

MR. HUNNEYMAN: That's really not what a grid is all about, because each individual computer is still an individual computer. There's an easy way to distribute the antivirus software across the grid. You can do that easily at one area. We can probably enable the antivirus software to be distributed more easily than if you didn't have a grid. But the point is that there's still that problem out there. If a person logs onto a network that gets a virus, there's not much we can do about it, except just give them the antivirus software.

MS. MARTIN: One of the things that we deal with is the security issue. There are a lot of fraudulent things that go on. We have a customer in government, and obviously they have strict policies.

MR. HUNNEYMAN: Obviously when you enable grids, that's a very big concern. You don't want just anybody getting onto your network and using your grid. We do have a lot of security built in, and we work with various agencies and companies to make sure that we get the proper authorization. When people log on, we want to make sure that our software is not going to allow them to get the data that they shouldn't be getting access to. There are a lot of different levels of security built into the grid. That is a big concern with grids.

MR. RICHARD LEAVITT: This is a question for Mr. Raden. First of all, I very much applaud you for your characterization of data managers as set in concrete, because that's certainly been my experience. I'm not sure that I understand your model enough to really phrase this question correctly, but I think that I spend a lot of time—when I do my analytics as an actuary—figuring out data-quality issues. What are the problems with the data? What are the holes? Because that can screw up your analytics easily. It sounds like your model is putting another sort of layer of interpretation between the data and me. How, as an actuary, would you deal with data-quality issues?

MR. RADEN: That's a really good question. I think that data quality exists at a couple of different levels. Are the data accurate? There are a lot of levels of data quality. It could be that the data are just no good. God knows we've seen plenty of that, like when we looked at cardinality of genders and got seven in an administrative system. We see that all the time. That's just a pure data-quality problem. But there are other issues for which the data may make sense within the application system, because the logic is buried into the program. It makes sense of data that don't seem to make sense. And when you extract the data without going through the logic of the program, it doesn't make any sense. That's another kind of data quality; we have to deal with that. There's a third kind of data quality, which is that data may make sense standing on their own. But when you try to marry those with data from other sources, they doesn't make sense anymore for referential reasons or for timing differences or a host of other problems. All of these really should be dealt with as a transformation problem in a data warehouse. And whether that happens just once and the data are stored in some persistent storage or it happens on the fly every time the data are queried from your perspective doesn't matter.

I'm suggesting that we get away from the physical storage of the data. Maybe we use the data; maybe we don't. But it no longer is a requirement, because we have the computing resources to grab data on the fly. The reason for it is simple. There may be data that are so rarely used that we're cluttering up a simple data model by pulling in these data and building them into a data warehouse. Or the data warehouse takes too long because we have too many sources of data that are only rarely used. So being able to grab the stuff on the fly is really a great idea. Now to your point: you're doing data-quality analysis for your analytics. I think that's a different kind of data. You're looking at whether the data are credible on their own, regardless of how we've defined or transformed the data. And I think that, in the best of all possible worlds, those first three types of data-quality problems should be solved by the software and solved in the background. You're still going to analyze your data and see if they make sense, but not because it's wrong the way the data were coded or there was a problem with the way the data were transformed.

If you like to get your hands on the raw data, that's not going to go away. To have a common way of identifying insureds, a common way of identifying coverage, if that doesn't suit your particular kind of statistical analysis and you need to go back to the source, then there's no reason for you not to be able to do that.

MR. CHARLES LINN: I have a question for Mr. Goedicke as a follow-up to the first question. What is the wireless industry doing about security and making sure that as the transmission is occurring, people aren't getting access to data?

MR. GOEDICKE: I think that what pops into everyone's mind when they think about wireless technology is that their data are less secure. I think that sometimes the data on these devices are even more secure. I'll just give you an example. A lot of people carry around laptops, and they have letters, Excel spreadsheets, etc., that are not encrypted. They're just sitting on the device. And you read about people's laptops getting stolen out of their cars. The data are largely not protected on laptops. There's been a lot of scrutiny on these devices, a lot of attention to security, simply because they're very mobile devices. They can get left somewhere. If you're looking at wireless solutions, make sure that all of the data are encrypted.

In fact, for one of our customers, we encrypted the data with 128-bit encryption. It's encrypted during transmission. If anyone intercepted the signal, all they would get would be useless bits and bytes; there's no way they could put it back together. And a lot of these devices also have, for example, biometric log-on. Some of the Hewlett-Packard devices have fingerprint readers on them. These devices have a built-in biometric signature; it's not just an image. It actually measures the speed and all sorts of vector positions doing it. It's impossible to crack. If someone loses this, no one else can't sign on; it's a dead machine. There's been a lot of attention to security. And I would bet money that these devices are more secure than most people's laptops.

MR. BRIAN GOEBEL: I have a question for Mr. Goedicke on the little devices that he is talking about. I don't have one, but my wife does. It seems like one of the more difficult things is trying to get a lot of text typed into it. She does something called graffiti. Is there anything that's coming out that would be a better methodology of getting text in, like voice recognition or something else that's practical?

MR. GOEDICKE: The Palm devices were the first ones that used the graffiti. It made you relearn how to handwrite because it had to be in their own recognition format. My current pocket PC device has two different methods. One is called a transcriber, and it doesn't make you relearn something. You can handwrite in your own block letters or script, and software teaches the device to recognize it. And there is voice recognition, for example, in the dialers and for getting data in the field. For many people out in the field, 80 percent of their job is filling out forms, either when they're meeting with someone or when they get back to the office. We were just talking about quality of data. You want the data to be the same. You don't want someone filling in a freehand note. We found that it is better to put the forms on these devices that allow for limited selections. That's how we largely handle all of that.

MR. WILLIAM DIMMOCK: I have a question for Mr. Raden. I've experienced frustration with our IT organization and data modeling, but that's an entirely different structure within our company. What do I do about it? From the actuarial side, where does that leave me? They're spending millions of dollars on systems and models that I may not like or agree with, but I'm stuck with them. What would you recommend? What have you seen companies do?

MR. RADEN: Stick your neck out. The most successful actuarial project I ever worked with was run by a chief actuary who was a true visionary. He ran with an almost altruistic zeal in that he really wanted to improve his company (which was failing, by the way). It was in big trouble. The regulators were after him. They wanted to go to quarterly valuations, and we could only do them annually. We had all sorts of problems. But more importantly than just solving his problems, he really wanted to change the nature of work for actuaries in his organization, because they came to work and they hated it. Turnover was about 50 or 60 percent a year. It was just dismal. And he ran a stealth operation. He called it a valuation system, but it was a data warehouse. He ran around IT and did his thing. He hired me. In fact, I was speaking at one of these spring conferences, and he came up to me and said, "I need your help." That's how it all started. One thing that I know about actuaries is that if they assert themselves, they can get pretty far. They can get away with things that other departments can't get away with, because they're so integral to running the company. I think that you have a lot of power, and you just need to flex it wisely and simply do what you have to do. Don't call it a data warehouse, and don't call it a BI initiative. Don't call it enterprise anything, just do what you have to do. And incrementally, you can get to where you need to be. That would be my suggestion.