RECORD, Volume 23, No. 2*

Montreal Spring Meeting

June 19–20, 1997

Session 73OF
Managing the Actuarial Systems Function

Track: Computer Science

Key words: Computer Systems, Management Information

Moderator: LARRY A. CURRAN

Panelists: MARK D. J. EVANS

ROB GLAZIER†

Summary: Modern actuarial practice depends heavily on computer systems, so the topic of managing actuarial systems support is an important one. In this session, forum leaders discuss the organizational approach to the actuarial systems function that has been applied in their organization.

Mr. Larry A. Curran: I am the actuarial administration officer at Northwestern Mutual Life (NML). I am the moderator and a co-panelist. We also have Mark Evans, assistant vice president and actuary at the Providian Corporation, and Rob Glazier, who is a guest of the Society. He is the manager of systems development at NML.

We are going to try to tell you how our two companies handle doing much of the management information system (MIS) function within the actuarial department, as opposed to being done in the information services (IS) department. I think all of us on the panel, feel that we have significant advantages in this area, and we're going to try to cover that, as well as all the other implications of our approach.

I'm going to start by giving something of a high level overview of how things are done at NML. I'll also talk a bit about the history of what we have done. Next, Rob will go into more detail about how the operations work.

†Mr. Glazier, not a member of the sponsoring organizations, is Manager of Systems Development at Northwestern Mutual Life Insurance in Milwaukee, WI.

At Northwestern Mutual, we have a full-fledged actuarial systems division; that is, it's like any other MIS division within the company, even the centralized IS department. We have about 25 people in our area. It is the only major full-service IS division outside the IS department. We report to the chief actuary, who reports to the CFO. We find, again, that this works well because it gives more control to the actuary.

I want to cover the definition of the function and scope of what we really do. Of course, the biggest part is to provide MIS support for the actuarial department. By that, I’m talking about valuation, reinsurance studies, retirement plan, and, one of the major tasks, providing the rates and values routines for corporate use. I think this is something that's a little different from what most companies do. We do not have rate factor files. We do all calculations of rates and values on the fly in all our systems, including our insurance-processing system. You will be hearing a good deal about these rate routines. They are very integral in our being a separate division and in how we operate. Both Rob and I will be touching on this in more detail as we go along. We also provide some specialized support for the operating departments. I'm going to let Rob cover that. It is an important function, and there are many issues associated with it.

The original purpose of our group was to provide programming support for the actuaries who did the actual systems function. At that time it was a pretty small group. We primarily worked with the actuarial rate routines, did mortality studies, and published the rate book. Changes over time were driven by many factors. The most important change was in the type of staffing we had in the division. We became more akin to an IS professional-type staff. Another of the big changes, however, has been the increased scope and level of support. It wasn't long after we were formed that the valuation routines drifted back to the actuarial department. The IS group did not have an understanding of the valuation function. They did not want to understand and, consequently, would assign who that they thought, perhaps, could be spared for something less important. I came to my cubicle once and found a shopping cart with the listings of the valuation programs piled on it. That's how we accomplished the transfer! We're a little more careful about accepting things now. Originally our agent and home office employee retirement plans were done as tab service functions. When it came time to automate those, we did it within the actuarial department. We also have a dividend inquiry program, a dividend history program, and quite a few other programs that service the actuaries.

Next I'd like to talk about the evolution of the type of employees we hire. Initially, the group was made up of students. In the very beginning, quite a few of the ASAs were involved when we bought our first computer. It was really a big event in

Milwaukee: *NML Buys an Electronic Brain* was the big headline in the newspaper. That was 1955, I believe.

In any case, we had actuaries working on the rate routines. The reason we had the rate routines to begin with was because we didn't have the capacity to store our rates. Our only choice was to do the calculations at that time. Fortunately, we had a very simplified product line at NML but we still only had a 40K computer with vacuum tubes—it was an IBM 705. So, we didn't really have much choice.

After that, we started putting students in on a rotation basis, and, for the most part, they disliked it for a couple different reasons. First, they felt they were getting away from their true actuarial function. Particularly at that time, computers were not considered to be something that actuaries should be involved with. Second, our biggest crunches would come at dividend scale time, which is when the students were studying for the exams. The pass rates of people who were assigned to programming functions were abysmally low. So we gradually phased that out. A couple actuaries fell in love with this whole thing. I, being one of them, basically had a career shift into the systems area.

We did some early experiments to fill the division. The most ill-founded one was at a time in the early 1970s where there was a belief that programming did not require specialized talent. Supposedly it took no brains and no talent. So we went to the clerical divisions and I was assigned to train them how to program. Eventually, they decided that we needed to hire better staff who didn't need as much training. We took control within our own division and began hiring math majors, which was extremely successful. Quite a few of these people have developed into outstanding technical people. We have continued that process for some time.

Now things have changed, and later Rob will give you an overview of what we're currently doing. You may see that things do come around a bit. We've evolved into a very professional IS staff. As I've said, it's become fairly large and very comparable to the people in the IS department.

From the Floor: When you said “professional” did you mean actuarial professionals?

Mr. Curran: No, I didn't. We're going to hear the word “professional” batted around a bit. In this particular case I'm talking about IS professionals. Actually, I'm talking even a little more generically than that. I'm talking about people who are professional in how they do their job, more than what their specific career is.

Value added: is there really any reason to do something like this? Do you really get anything out of it? For those of you who deal primarily with the IS department, picture going up to someone saying, “I want this done, and I want it done now” without being laughed at. I think that's the value added by having the systems group within the actuarial department. We report to the actuaries. We know where our bread is buttered, so we are responsive.

I'd like to give you a couple of examples of how this can work. The first example dates back to 1980. We decided to update to a 4% reserve interest rate. Of course, this was on a policy owner volunteer basis. We did a huge block of business going way back that we were going to allow to update to a 4% reserve interest rate. We had a combination of changes to make. First, we had the rate routines. It was a two-day change to the rate routines to incorporate the ability to update. We were able to work with the IS department to help them understand what we were doing. It went very smoothly. The actuaries really had control of the problem.

The second example occurred in 1986. We found that on a particular block of business we were paying a dividend lower than what we really wanted to be paying. This occurred at some of the younger ages, in the 20s. It was January, so the dividend resolution was already out, and we were into the dividend year. We wanted to change it, but, in order to change it, we had to go back to all policies that had already received dividends and do something to them. We formed a team consisting of one member of our systems division and one systems person from the IS department who was very good at project management. Within a matter of one to two months, we had all the dividends paid to the existing policies, and we were paying the dividends that we wanted to be paying. Once again, everything went smoothly. It shows how well things work when you have your MIS division within the actuarial department. You still have your counterparts within the IS department whom they can relate to, and you can get cooperation from IS. It also had another side benefit. I mentioned that we sent one of our people over to IS to help them out. They liked him so much that two years later he was over there permanently with a whole new career. It was a win/win situation for everybody.

We have the right people with the right training. We have people with math backgrounds and people who appreciate and understand what actuaries do. That makes all the difference in the world. They are very good at working with actuaries. The actuaries see people they can relate to, whom they can talk to without a glassy look on their face. I already talked about close links to IS; we have them, and it's good. We meet regularly with IS, who ask us to help them with their planning. Any time we have need for equipment or software, we only have to say that we have the need, and we get it—if it's within reason. We can have what we want, and that's very helpful.

From the Floor: Do you have a centralized actuarial function in you company?

Mr. Curran: Yes, we do. I think that makes it easier for us to justify having the division where we are, too.

From the Floor: Do you have your own mainframe system?

Mr. Curran: No, we don't. We use the same mainframe as the IS people.

From the Floor: Is your function separated from the rest of the company, the accounting and the transactions and all that? I don't quite understand, because, for example, most mainframe packages go for rate files.

Mr. Curran: We wrote our own system back in the 1950s, and we're still using it. It's just as up to date, and there are no problems with it. Is that right, Rob?

Mr. Rob Glazier: Whether or not there are any problems with it is debatable. We are in the middle of looking at purchasing some packages to handle certain blocks of business, but we still want to handle the creation of the rate routines in the actuarial department. There are other applications that need those rates beyond the administration system, so if we buy an administration system package that needs rates, we'll work with our IS group to integrate our rate routines into the administration system, and then they're still available for our own products.

Mr. Curran: It's the flexibility of the rate routines that we regard as so valuable, because we can do anything, and we can do it fast. The dividends can vary by any parameters you want, and you can change them instantly and respond quickly.

From the Floor: From my experience, when you price a product you don't actually have formulas to calculate the premiums. You do a profit analysis. At least, that's the way many companies do it.

Mr. Curran: Yes, we're European in how we do our gross premiums, which are really our net premiums loaded. We do the modeling and testing, but we still load net premiums.

When we did this update, however, we ended up with an odd situation. Our gross premiums couldn't change, so we had to net premiums on the old reserve basis, then use the new reserve bases for the valuation and cash value purposes.

The biggest issue we started with was IS distrust. You may find that surprising, but it was true. They had a project management system—one shoe fits all problems. We had this little area that did a few things, but they wanted us to spend more time doing project management. We wanted to spend our time actually doing projects. I have to admit that we were pretty arrogant about the whole thing. At one point the sales illustration system, which resided in the actuarial department back in 1983, was transferred to IS because of concerns about project management. That was a bit discouraging. Over time, however, we blended. Our ideas are now much closer on project management.

Another area of conflict involved IS standards. Some of the techniques we used to create the rate routines were techniques that IS didn't want to use. Again, I'm talking about back at the time when it was generally accepted that people who programmed didn't have to have strong technical skills, and IS didn't want the complexities of indirect addresses, pointers, and other things that we felt we needed. What we did to conquer this was to work with the really technical support people in IS. They ended up thinking that the actuarial systems people knew what we were doing and that they could learn from us. That has kind of stuck. One other thing is that the current head of the IS department is an actuary. I think that has also helped bring us much closer together.

Incidentally, on the IS standards issue, some of the things we did were just wrong. In the early days we didn't put any of our programs in production, and we didn't really have good control of it. We didn't even know how many programs we had. We were essentially not as organized as we should have been. Once we developed our organizational skills and started following reasonable standards, the IS distrust started to melt away. There was also some lack of professional status. When you're in the actuarial department and you talk about the professional staff, we're talking about FSAs. I don't mean to discredit or demean what's being done in the systems area, but that's how it is. It is an actuarial department, the professional staff of the actuarial department are FSAs. This has created minor problems, but not many. We have a great group of actuaries in our company, and their treatment of the people in the systems area is really great.

The other thing is that people in the IS field don't have a way of accrediting themselves. No real designations have been accepted. I think this is where the actuarial field has done a much better job than the IS field. We've lost sight of the department focus. We're support staff. The actuarial department was not created as an entity to create systems for us to write, and we have to accept that. We also have high-level direction coming from more than one place. We have high-level direction coming from the actuaries, and we have high level direction coming from the IS department. We have to know how to blend these together, and, in some aspects, actually keeping two bosses happy.

There is another issue concerning career path limitations. The career path of an actuary seems to be boundless and fast moving. Within our own area, we follow the same career path as the IS department. It is limiting, but I don't think it causes many problems. However, our department has been really good about providing opportunities for non-FSAs. We have three officers in our department who are not FSAs, and all of them came out of the actuarial systems division. I think the systems people look at that and see it positively.

From the Floor: How many people do you have on staff?

Mr. Curran: What is it now, Rob?

Mr. Glazier: Twenty-five.

Mr. Curran: It's grown from 4 in 1969 to 25 now.

From the Floor: How big is your IS department then?

Mr. Glazier: About 650.

Mr. Daniel M. Walsh: You said you had limitations on storing rates. Do you store dividend scales for historical purposes? Do you feel there’s a control problem without storing tables?

Mr. Curran: About ten years ago we did some experiments in this area. We started saving old versions of the rate routines that we could call up. I'll have Rob actually field the rest of it because that's not what we're doing now. We now have a better way.

Mr. Glazier: Like Larry said, we used to save old versions of the rate routines. The problem was that if the interfaces changed, you couldn't call up the old versions anymore. Since 1987 we have kept the old source code and updated it for the current interface. So from 1987 forward, we can call and get a 1997 dividend, a 1989 dividend, or anything up until now.

From the Floor: It's on the fly.

Mr. Glazier: It is still on the fly, but we don't table. I don't want to say that we don't table any rates, but we table very few.

From the Floor: Do you feel there’s a control problem in terms of making sure that it doesn't change over time?

Mr. Glazier: There's a great deal of difference. The issue is that we need rates, and NML needs rates on a wide variety of platforms. We don't just need rates on the mainframe system for administration purposes. The sales illustration system that is out in the field also needs rates. It's a PC-based system. We have some client-server-based-systems too. We have one set of rate routines and one set of source code. It's portable to all the platforms that we target.

Mr. Curran: I'll comment a bit on this, too. There are some concerns about that, and there certainly have been in the past. What we've done is to actually store the gross premium on the master record. Every time we go through anniversary processing, we go back to the rate routines and recalculate that premium. If there's a difference, then we know we have a problem.

In terms of dividends, however, we don't have any specific controls on that. Theoretically, over the course of the year the dividend calculation could go amok. But we don’t have any evidence this has happened. So, we're pretty confident that this is a way that does work well.

From the Floor: Do your dividends target your master record somewhere?

Mr. Curran: Yes, they do. Dividends are stored on the master record. But once the dividend is calculated during anniversary processing it is never recalculated.

Mr. Glazier: I'm the manager in the systems group that we've been talking about. NML’s home office has roughly 3,000 employees. We have the home office complex and a few regional real estate offices, but, other than that, all of NMLs 3,000 employees are located within the home office and are supported by an IS department, of roughly 650 people. Within NML we have our actuarial department which has about 100 people. They are supported by this actuarial systems division of about 25 people. There are two managers in the systems division, and I'm one of those two managers. My background is not actuarial. I was a computer science major in college and have a math minor. I started out as a programmer at a bank and came into the actuarial department as a programmer in 1986.

I'm going to cover the same general outline as Larry. I’ll talk a bit more about our current functions, some about our employees, and I’ll give my perspective on the value and some of the challenges that we face.

The functionality that we support is based on whether it meets an actuarial need or involves actuarial expertise. There are mainly two types of applications. First, we build rates and values routines. These are our responsibility because they involve actuarial expertise. Here we work very closely with the pricers and the people who specify the dividend scale to provide the necessary rates and values routines. The rate routines return numbers like premiums, dividends, cash values, and liability reserves. The values routines generate contract values that aren't rate based. Here we need actual policy values in order to calculate some of these other values needed by our administration system. So the values routines use actual policy input and return a value. We maintain a dividend routine to actually calculate the true policy dividend. This dividend routine calls the rate routines to get the dividend rate. It then applies the dividend rate to the policy values to actually get the true dividend. That's the difference between the rates and the value routines.

The second type of applications is where the primary clients are part of our department; for example, we maintain a valuation system, a studies system, and a reinsurance administration system. We are responsible for all of these. As far as the nature of the work, the only point that I want to make is that it is really IS-type work, as Larry indicated. We put all of our code into production, targeting multiple platforms, and so on. Since it gets very technical, we have more of a MIS or an IS-type person doing the work.

From the Floor: In your work are you actually making changes to the administration system itself, or is this a completely separate system?

Mr. Glazier: We are making changes to the administration system in that our routines are embedded by the IS staff into their systems, and at run time the administration system calls our routines as its doing its administration. If we change the rate routines, for example, they are automatically picked up by the administration system. This is what we call a “routinable function.” We change it, and the system automatically picks it up. Beyond that, we don't make any changes to NML’s administration system, and all the rest of the functionality I mentioned earlier is used just within the actuarial department.

From the Floor: Do you have universal life (UL) products or annuities on this same system?

Mr. Glazier: We don't have a UL product. We do have interest-sensitive products. I certainly think we can handle a UL-type product, but we don't currently have that on the system. We do have annuities, some fixed and some variable.

Mr. Curran: We have a separate account life product that has some UL elements that is handled on the system. It does have its complications. Although we don't really change the administration system, we may be called in to provide some help on it.

Mr. Glazier: I'll give you an example of the division of the work. We're going through product development right now. We sit down with the IS department, talk about the product, and talk about the calculations that are needed. When I say "we," I'm talking about actuarial systems people. We talk about the product based on the specs that we've received from the actuaries. Then we decide where we are going to draw the line between what the administration system is going to do and what we're going to do; we focus on what kind of routines we can build so they can integrate them into their system. I think they're much more comfortable with that because they think if it's actuarially technical, or if its calculations are intensive, the actuarial department can do it for us.

From the Floor: What programming language is used in the application?

Mr. Glazier: The rate routines traditionally have been done in PL/1, but when we had a need for multiple platforms we started doing development in C. Recently, we obtained a PL/1 compiler for the PC and we sort of retro-fitted all our PL/1 code so that the older series products were also available on the PC platform. We're just finishing that up now. It has not yet been delivered. So C is the language for the future or C++, depending on whether it's object oriented or not.

Larry covered most of the employee issues. I want to emphasize our focus when we hire; it's math, MIS, or computer science majors, not people who want to become actuaries. We want people who want to do programming. They're not coming into the division because they think it's a way into our actuarial program. If they want to become actuaries, they know they have to go through our actuarial student recruiting, which is a different path. Their main interest must be in systems development. However, we do emphasize to all the staff that they have to be willing to understand the actuarial business, our products, and so forth. They can't be people who are good technically with systems but not interested in the actuarial applications.

From the Floor: On the other hand, do you have some people who were originally interested in an actuarial career, but, because of their involvement in this department, they chose to change their hat?

Mr. Glazier: Yes. Larry was an example of somebody who did just that. Larry is an ASA, but decided to stop working as an actuary and become a systems person. So, yes, that does happen. I'm trying to think if we have anybody who has gone the other way. I remember a recent intern who came in as an intern in the programming area and then became an actuary, but not with our company. It certainly can happen. There are no walls or anything like that.

I mentioned that we try to emphasize to everybody that the skills they should have are both technical and business—to be able to understand our products and the way insurance is processed. That's the main point there.

As far as value, I think there's a stronger commitment by being in our department to the goals of the department. We have to be more sensitive to the needs of the customers, who are ultimately the actuaries. Because there's not a wall or a departmental boundary between us, I think we're more motivated. We like this arrangement. We're motivated to keep this area within the actuarial department. We don't want to have our own management become unhappy with us. It's possible that some of our functionality could go to the IS department if they weren't happy with the work we did. So, I think there's a great deal of motivation to be successful.

There's a strong relationship between the actuaries and the system personnel. We're right there together, physically. It's easy for the actuaries to drop in on the systems people and vice versa. I think, because of that, that a team relationship has developed, and we can be very informal with each other, very interactive. The actuaries can come to us and say that they want to do such and such with the dividend scale, can you try run this? Then the next day they come back and ask us to also try this, and would you mind changing what you did before? At NML, if you try to do that with our IS department, there's much more formality to it. They want specs written up and more lead time. This way we're just more flexible by having the actuaries and their systems support all in the same department.

Finally, I think we can act as advisors to actuarial management on technical issues that they're not as interested in but that they need an answer on. For example, picking software packages or asking what the IS department is doing about a given issue: What is this client server initiative? Does it affect us? Should we care? We can help in that regard. We also help with technical problems like when PCs are not working. If IS says it's going to take a day to get somebody here, we can run right over and fix it because the focus of the division is more technical. That's another bonus.

Finally, if you want to do this in your company, several challenges currently exist. First of all, systems people are in short supply in the market. Anybody who has a systems group is facing a recruiting challenge. What we can do that's a little different is to find some “gems” in math majors who would otherwise be overlooked by an IS department. They may not be really strong in systems as far as the course work that they've had, but they might be very successful math students who weren’t sure what they were going to do when they got out of college. So they went into math. They didn't want to become an actuary, for whatever reason, and weren't sure what they wanted to do with themselves. We've interviewed and hired people like that, and have had great success. That's where we can sort of solve the recruiting problem.

From the Floor: Do you get anybody out of the IS department?

Mr. Glazier: Yes, we do.

From the Floor: How do you handle a relationship that way?

Mr. Glazier: That's a good question. It's interesting that retaining top performers came up, because I was going to talk about that next. We have the same salary structure as IS, the same titles, and about the same promotion times. We want to make sure that we don't have competition between IS and us. We don't want people from our division to think it's better to go to IS because of some issue like salary, title, and so forth. And we don't want people from IS coming to our area because they hear that actuarial pays a little bit more, or promotes quicker, or whatever. So, we try to keep things in sync. But if there happens to be somebody in the IS department who thinks they'd like to work on an application that they've heard about that we have—one that sounds interesting to them—we certainly will do that and we have done that. We have also had people leave our area to work in IS. There's not much of it though. In the ten years that I've been in the actuarial systems area at NML, I think I've seen three people leave our area and go to IS. I know of two who have come to us from IS. So, it's not a really significant issue, but it's because we're so careful about the salary and the titles and so forth. If someone thought they could get a promotion out of it or advance more quickly, they might do it, and that's not why we want people coming to the actuarial department. We want them to come because they think our applications sound interesting.

From the Floor: Isn't it true that going into the actuarial systems division could enhance somebody’s career path, so there might be higher demand for IS people to go in that direction?

Mr. Glazier: Well, I think the way it could enhance your career is that it gives you experience with a different area of the company and some different issues. In general, I think if someone came from IS to the actuarial department they'd become more well rounded, but one limitation is that one can move up only so far in the actuarial systems area at NML. You're not going to become an actuary; you're certainly not going to become the department head. Where an actuary can go, like Larry said, is limitless. You can become the CEO. With a person in actuarial systems that is probably not going to happen. In IS, people work their way up and become vice presidents and so forth. If we have a very aggressive person, and they want a great career for themselves, they may eventually start to look outside the area. We try to solve this problem is by making other opportunities available for these people. There have been people who left systems and took other jobs at NML that gave them more advancement. Some systems background is a great augmentation to any other credentials they have. When we recruit we try to explain that they're not coming into an area where they're going to be promoted a couple of times and then they're stuck—because other possibilities exist.

Just briefly, managing scope is another important issue. Sometimes we are asked to do things that we don't think are quite right for the actuarial systems group. We generally go back to that scope statement that I mentioned at the beginning. Does it involve true actuarial expertise, or is it really needed primarily by people in our department? If not, we question whether we should be doing it. Just because it has a calculation in it doesn't make it actuarial, but some people think it does. Scope is a big issue for us. We try to be careful about that because we don't want to grow too big. We're at 25 now; that's about as big as I'm comfortable with. Maybe we could grow a little bigger but not too much, otherwise we'll get too unwieldy and more difficult to manage.

Larry covered my last item pretty well; namely the idea that we need to maintain a positive relationship with IS. We rely on them for some technical support, for budgeting for our software, for training our entry-level people, and for all the data that come into our area. If we need data not needed by anybody else, we have to convince them to hold it. In general, I think it makes good sense to have a strong relationship with IS.

From the Floor: This is working relatively well. As a result, are there any other areas that have their own systems areas?

Mr. Glazier: NML has a centralized actuarial department, so most of the actuarial functions are done there. Other groups in the company have created computing groups. If they were to describe themselves to you, I think they might describe themselves a little bit differently. Most of them are small; they don't necessarily have the same career path that we provide in the actuarial systems area. None is as large as ours. Then you get into the debate whether we should have a centralized IS function or a decentralized IS function. Some people could talk about that for a long time, but I'm not going to get into it.

Mr. Curran: I'd like to turn the podium over to Mark Evans, who can give you a slightly different perspective on how Providian is handling this type of work. We will have a short time afterwards for some questions.

Mr. Mark D. J. Evans: Providian is a very large corporation that was recently purchased by another, even larger, corporation, AEGON. I work in what's called the agency group, a part of Providian that focuses on individual life insurance sold through agents.

I'm going to go pretty quickly through, how the actuarial department is organized within the Providian agency group. We have three sections. First, there's the market development area, whose primary focus is pricing, but they also get into other related activities that you would typically associate with the pricing area. We also have a financial reporting area that basically does your valuation and similar functions. We have an actuarial department, which is where the chief actuary resides. It supports the other areas and also gets into actuarial-type issues that don't fall cleanly into either pricing- or valuation-type roles.

One of our missions, and one of the strong areas of our actuarial area, is software expertise. We're a little different in setup than NML in that we don't have a separate department within actuarial whose sole mission is to deliver computer expertise. We have a different approach, namely, computer expertise sprinkled throughout the actuarial organization, and this is built on the philosophy that all actuaries should be computer literate, including competency in a procedural language. Our focus is on using actuarial support professionals who use computer tools and skills to solve problems assigned to them. So they not only "own," so to speak, the responsibility of developing the software to solve the problem; they are the user at the same time. They are responsible for the actuarial aspects and the business aspects of the problem. One big advantage of this setup is that it wipes away the hand-off from the user to the programmer. You don't have those communication issues because you're only communicating with yourself. What happens more often than not is that the program can be written and tested by the user in less time than the user would spend just explaining the function to the programmer and assisting in the testing of the results.

From the Floor: By user, do you mean actuary?

Mr. Evans: Yes, I mean actuary, and I'll get to this a bit later. I mean support professionals working in the actuarial area.

Let me go ahead and define the employee. We have ASAs, students in the actuarial program taking exams but who have not yet attained ASA status, and some people whose academic background is in math, some with and some without an academic computer background. And then, I'm an FSA with heavy-duty computer involvement, both academically and based on experience.

From the Floor: How do you manage having students keep up with their actuarial curriculum if they are responsible for keeping current in computer-type issues at the same time?

Mr. Evans: That gets back to the philosophy of what do actuaries do? Actuaries work with formulas, large amounts of data, large amounts of information, and all different sorts of calculations. What do you use for those? Computers. Our philosophy is, if you're an actuary, you need to be highly competent in computers. Now, people are going to disagree with that, but that's the philosophy here. So, yes, it presents a challenge.

From the Floor: As far as implementation, though, is it many work hours, or do you have extra training? Do you bring somebody in for your assistance training for your actuarial students? Or what approach do you take to see that they get proper cross-training?

Mr. Evans: We provide time for that. I'll get to that a bit more later, but let me say right now that we expect them to learn certain things. Also, different students, depending on their interests, will develop different levels of expertise. We have self-training tutorials that they go through. We also have other people in the department who have been there a while, including some of the math majors I was talking about, who have very high levels of expertise. They have plenty of resources at hand to develop these skills. Obviously, when we're recruiting, we're more comfortable with those students who come with some computer background.

From the Floor: Is this structure of having FSAs and some students for all of the actuarial areas that you talked about? Or is that one FSA for each student?

Mr. Evans: I am the only FSA with a heavy computer background. But in each of the three departments I mentioned, we have, depending on the area within the actuarial department, six to nine people with very strong computer skills. In the market development area, we have a couple of people with very heavy duty mainframe programming abilities from a math background. In the valuation area, we have two very heavy duty classical programming-type people.

From the Floor: This philosophy that all actuaries should be competent and literate in computer programming can't necessarily encompass all the actuaries who use the process because you have actuaries who do not have that computer literacy.

Mr. Evans: Some of the actuaries would be at an in-between level. We've had some students, for example, who have become computer pros by doing heavy-duty computer programming. We've also had some who haven't showed much interest. It varies, it's a continuum.

From the Floor: You said that by this process you avoid the hand-off to IS people. Are you implying that these people will maintain these ongoing applications?

Mr. Evans: Yes, there are organizational and management implications of this that I'm going to address in a bit. Good point.

Next, I'll define the employee and talk about the necessary skills. As I mentioned in responding to the gentleman's question, the skill set does vary some by individual. Most of the people working in the actuarial areas have a direct mainframe log-on just like anyone in the IS department. I'll show you some of the languages people use. A Programming Language (APL) is a big one. We expect people to be familiar with IBM Job Control Language so they can develop batch jobs. We use a powerful sort package called Sync Sort because we process some extremely large data files in our work. We have a few people who use COBOL. We use a fair amount of Assembler, which sits right on top of machine language. And we use various IBM utilities. We use a language called Focus, which is an end-user report-writing tool. This is used primarily by some of our people who are not heavy into the computer area. We also use something called Quick Job, a proprietary language that I'll describe as a cross between COBOL and Assembler. We also make heavy use of PCs. Most of our spreadsheets are Excel spreadsheets. We have people who program in Visual Basic both on a stand-alone basis and as Excel macros. Last, we use Tillinghast Actuarial Software quite a bit.

Data sources are a big issue for us. In addition to expecting our people to understand these various computer tools, we expect them to be familiar with the various data sources. For example, we have many programs that read the policy master record—go into the administration system and pull information off the policy master file. We have millions of customers and a pretty large block of UL business whose data structures—the way those files are organized— are fairly complex. This is one instance where efficiency is still a big issue. So, in addition to knowing these various languages, we expect people to be familiar with the files and different data sources out there that they can use to get their information.

Next I'll talk about the value added to the department. Greater accountability is achieved when the work is done within the department. On our projects we get faster turnaround and better quality. One potential drawback is that you can get a less stable environment because traditional IS departments do have clear rules and procedures that are followed very well. In a smaller department and out in the user areas, you get situations where this lack of bureaucracy causes a problem over a period of time.

Some of the other issues around the value added to the department are, first, the relations with the IS organization itself. If they tell us, for example, that a particular project is going to take six months, and we know based on the level difficulty it should probably only take two weeks, we're in an excellent position to call their bluff. We can name that song in three weeks; when it's put in those terms they quickly come up with a better way to do it that's much quicker. Something else that has come in handy occurs when the IS department is working on some large production system. We're able to step in and help them de-bug it, get files from them, or test output. We write various programs in our area that can test the output of a production system under development, which makes things easier for them and things easier for us. You also get much better understanding and communication between the actuarial area and the IS area in such a situation.

From the Floor: On that first point you made about timing—six months versus two weeks. How successful have you been in getting them to back down on that? What technology do you use? Is it a higher management level?

Mr. Evans: No, nothing ever gets done up there. What you do is establish relationships within the IS area, which is a natural side effect of this anyway, because you're working with those people. We may ask what file we need, or they may call you up saying they lost their copy of this production file and ask if we made a copy of it last year end. Those kinds of informal things. There have been times when we've asked to see the production code. We’ll go through the code, mark the changes, and hand it back to them. They'll go ahead and put that in and test it out. That approach has worked pretty well. Usually with situations like that, the higher up you go, the worse it gets.

Our preferred candidate, outside of actuarial students, is a math major with some computer programming background. If we have someone with a strong math background and they're weak in the computer area, we'll train them. As long as they have the interest in the computer programming area, we'll make the training and learning opportunities available. We simply have not had good luck with individuals without a strong math background. Part of that is getting back to the philosophy that we don't walk up to someone and give them a specification. We explain the actuarial or business problem we're trying to solve. They're not going to be comfortable working this way if they don't have a strong math background.

We've had reasonably good luck at retaining valuable staff. In large part this is because we get people who are interested in programming but also enjoy a high mathematical content in their work. They're not going to get that in the traditional IS-type role. So those people come to us, enjoy the work, and we're able to retain them. There is, by the way, a fair amount of movement back and forth between our department and the IS department. Over the years we've developed a good deal of systems expertise, so they have an opportunity to work with people who already are experts, and that's attractive. If you like to program but you don't like the bureaucratic environment that's necessary in a traditional IS shop, we offer an interesting alternative.

The gentleman asked earlier about maintenance. If you're going to go with this approach, there are some things you have to think about. When you get into some of the more complicated tasks, APL code can be pretty intimidating for someone who isn't used to it. There was a session earlier about development in C++. Once again, there is a learning curve associated with this approach. Both these tools are very powerful. The flip side is if you develop applications that you're going to be using over a long period of time, the actuarial department must make a commitment to develop and maintain these levels of expertise. That's probably not going to work if you've based it on one or two people. So that is something to think about and, perhaps, a reason not to take the approach we've taken.

From the Floor: How long has this been the philosophy?

Mr. Evans: About 17 years.

From the Floor: As for the issue of maintaining your system, has that been an issue at all?

Mr. Evans: Within the Providian agency group that I referred to earlier, it has not been a big issue. We had one instance where we had an employee walk out without any notice. That hurt us, but it was an isolated event. In general, we really haven't had a problem with maintenance. I'm familiar with some other areas where they did not maintain that level of expertise—they just overlooked it, and the management wasn't committed to it or wasn't aware of it. I'm not sure what the problem was. At times you have individuals come in, develop really great software, and then leave. We would have some of Larry's clerks come in to maintain and run the software, and, yes, it was a problem.

From the Floor: How do you handle student rotations? How do you determine which language needs to be used? Don't you run into problems when you rotate a student who has no APL or C++ experience into an area?

Mr. Evans: The students rotate through the actuarial area. I'm not sure this has always been true. They all learn APL. All the students, of course, are going to learn Excel. One way we tackle that problem within some of the more long-term systems is by making sure that there's a permanent employee within the department who's not in the student rotation system, someone who has some familiarity with the system. We have some issues when students rotate out who have, for whatever reason, developed a long-term system. We make sure that there's the appropriate hand-off to someone else in the department. I'm not sure I fully answered your question.

From the Floor: So you maintain some sort of permanence there, as well, but you still have the learning curve issue as you move someone who doesn't know the language.

Mr. Evans: That's right, yes.

From the Floor: Do you have any problems with programming styles? People come in and want to rewrite the program the way they would have done it.

Mr. Evans: You have to guard against that. Unless the code is really spaghetti, you have to get people to accept it. If the code is not as elegant as it could be, before you go in there and rewrite it, you have to use some discipline and realize that you had better have a good reason. Yes, that is an issue. We've had some programs rewritten that would've been better off left alone.

From the Floor: We've had places where we've developed applications and administrative systems for new products using APL by having the student write it and try to maintain it. First of all, APL is often written with very few standards. It can be written very complexly but compactly. The problem occurred when we decided that we didn't want to maintain systems in the actuarial department any longer. It's hard to make the transition after you've gone down the road a ways, to hand it over to someone with different systems and to develop a whole new system to handle it. It's a fairly major decision to be made at that point in time.

Mr. Evans: Next I want to cover the things you want to look at to determine whether a project, a program, or a system belongs in the IS department or in the actuarial department. If it's hardware resource intensive or if it's going to crank away on the mainframe for hours and hours, IS will have to be involved because they're going to wonder, if nothing else, what you're doing with their machine. But also, at that point they're going to have specialized expertise they can bring to bear on your problem. If the system is going to be rather stable over time, you won't have many modifications. Those are all factors suggesting IS ownership. If it has a long expected shelf life IS typically has the bureaucratic infrastructure in place to support long shelf-life projects where they're not going to have the same personalities, or the same people associated with the project over its entire shelf life. If multiple departments are affected, IS is used to dealing with information coming in from multiple users and balancing the wants and needs of various departments to come up with a solution that makes everyone reasonably happy and reasonably unhappy at the same time. This is classical data processing-type logic. They are good at reading in a file, sorting a file, summarizing, file manipulation, transaction handling, and other things like that. Actuaries don't necessarily have a reason to get into much of this.

The last factor to consider is the presence of a “steady as she goes “philosophy—in other words, if you’re not in a hurry to change the direction of the ship, perhaps that suggests a smaller involvement of actuarial department in some of your software development needs.

Contrast these with characteristics that suggest actuarial systems management development of a particular project. First, we have new development and experimentation. If you're developing some sort of system or some sort of report where, you don't know exactly what it's going to look like when you're done, do it in the actuarial area. Prototyping fits into this category. If you don't know exactly what you're going to end up with, experiment with it, play around with it, and do that within the actuarial area. Don't get IS involved until you have what you want. Then if you want to move it into a production environment or a scheduled environment, or if you're generating reports that are going out to the field or throughout the company, then at that point, hand it off to them to use their IS expertise in getting this report distributed efficiently or put onto an imaged system, or whatever you want to do. If you're developing a system that's going to have frequent modifications and frequent variations, that will suggest that you keep it within the actuarial function. We all know what happens every time you put a request into the cue. If you're doing ad hoc things, one-time reports or special projects, once again, IS bureaucracy is not typically set up to handle something like that efficiently. You're better off doing it internally. If you are only going to use it for the actuarial department, or if you're in an accounting department and you're only going to use it for the accounting department, then your communication and all that is going to work out much better if you do it within the department.

The next type of project is one that involves complex formulae or heavily actuarial algorithms. This is familiar to anyone who has ever tried to explain a formula with more than three or four variables in it to a nonmathematical IS professional.

Once again, contrasting the “steady as you go” philosophy: if your company is always trying new things, different types of experiments, and other ideas like that, you're probably going to want to develop a strong actuarial systems function.

Mr. Alan Cutler: We did something unique recently. In a period of ten years, we developed our own mainframe administration system from scratch because we didn't feel the package systems gave us the support for all our products. That system is now in production. Part of the development involved another actuary and me full time in the development of this system, mainly the calculations. We also designed the valuation extract to feed the PC system and a full policy exhibit system. This gives us a very efficient file at the end of each month of all the records out there on the contracts: terminated and in force. I'm saying this because we do not have a separate actuarial programming function. We've moved it into IS. The maintenance involves their calling us, but in a way I envy you. We felt that there was no need to have that, but I think you all do so much more with your style of rates, dividends, and all that.

Mr. Glazier: This has been evolving. NML started out with IS supporting the valuation system, and they weren't properly staffed to handle it. We accepted the idea of the system coming over, and this has worked out well. It also gives NML a means to do cross-checking, because some inventory programs supported by IS in the administration system can be checked against our valuation in-force statistics: this provides some quality control. I prefer valuation being in actuarial; however, it is debatable. I could be comfortable with it being in IS under the right circumstances.

One of the things that I've noticed is that you are at zero contrast between Providian and NML in the type of employee that we have. Providian has actuaries involved. Basically we don't, but there is an exception with the pricing area. Here the actuaries maintain pricing programs in APL. Their results come to us in file format, and we use them to compare against the rate routines. It’s a wonderful means of checking.

I'm wondering if there's any feel out there as to, or any comments or questions about, these differences that seem rather strong between the two?

From the Floor: He is heavily involved in the systems area; would that suggest that actuaries could get heavily involved?

Stephen J. Strommen: I can respond to that. I had not been in the actuarial department; I was actually in another financial planning department.

Mr. Curran: I think his is probably the only other area I know that gets as heavily involved in using programming languages such as C++ and developing systems. The other areas of the company are using SAS report generators and other things like that. They may call themselves systems analysts, but that's kind of an exaggeration.

Mr. Donald B. Welch: I just wanted to let you know something about our organization. I've been there from the time the actuarial department contained the IS department to now, when the IS department is on its own. Over the years, we have tried many techniques and have evolved to the point where we are much the same as you. We don't have our own IS resources reporting directly to the actuarial department, but we do have resources that have evolved. More mathematically oriented programmers are dedicated to product development. They're the people who work on rates, values, and illustrations. So we've had much the same thing, but we don't have that direct reporting relationship, and it still works quite well. It has some bureaucratic details that have to be worked out, but it does work within the structure of the IS department. We do have that direct connection as opposed to having the relationship between the two departments.

Mr. Curran: So it's still very responsive.

Mr. Welsh: Yes, it’s very responsive because that's one of the commitments. Product is one of the high priorities, so it gets priority resources that way.