RECORD OF SOCIETY OF ACTUARIES
1991 VOL. 17 NO. 1

MODELING TECHNOLOGIES

Moderator:	MARK S. MAHANY
Panelists:	THOMAS NACE
	R. LARRY WARNOCK
Recorder:	MARK S. MAHANY

The panel will discuss the approaches and techniques used for pricing and modeling insurance products.

- Commercial software
- Internal systems
- Maintenance issues
- Staff training
- A/L systems

MR. MARK S. MAHANY: I am currently employed as a consulting actuary with Chalke Incorporated. My main area of expertise for the last three and a half years has been directing the development of marketing of PTS, Chalke Incorporated's insurance modeling software. The panel will not concentrate on providing you with a laundry list of relative advantages and disadvantages of various commercial systems.

When the panel met prior to the meeting, we noted that if we wanted to properly cover this subject, the panel discussion would probably last for a week or two. The topics and techniques to be discussed are so widespread that it became difficult to decide whether we were going to concentrate on one specific piece of the subject matter or provide a brief, general overview. We have decided on the latter approach.

For our panel, I have attempted to supply you with a cross section from the industry with respect to commercial systems and internally developed modeling systems. Tom Nace is currently the assistant vice president and corporate actuary at Penn Mutual Life. Tom is responsible for the modeling function in the corporate department, as well as statutory valuation.

Our other panel member is Larry Warnock. Larry is currently the manager of the life and health operation in Atlanta for Tillinghast. Larry was responsible for the coordination of Tillinghast's PROPHIT modeling system. Most recently, he has been performing appraisal work and value-added reporting for the Tillinghast consulting practice in Atlanta.

I would like to begin my piece of our presentation by reviewing different hardware platforms. The first platform I'd like to discuss is the mainframe. Mainframe use has diminished over the last few years. One important and obvious reason has been the increase in speed of PCs. Another has been the price differential. I believe the most important reason for the switch has been the development of state-of-the-art software that is currently being supported in a PC environment only. Over the last four or five years many internal (and external) modeling systems have been converted over to PCs.

A second option is a 386 or 486 PC environment. I think you'll find 286-based hardware does not provide a viable solution in today's marketplace. Since 486 machines have been available in the market, we've found that the relative marginal value of a 486 far exceeds the cost differential between a 386 and a 486. The speed difference between the two is substantial.

A third option is workstations. We see workstations as the next major expansion platform for commercial software. Workstations are approximately the size of a standard 386/486 PC, but are much heavier. You may build multiple "nodes" off the main processor, lending its functionality to a pure network environment. This allows file sharing and multitasking which I'll discuss a little later.

In addition, you should also spend time researching what I would refer to as peripheral hardware items. Many people operate on a network. This certainly maximizes your ability to completely integrate your pricing area, your corporate modeling area, maybe a GAAP reporting area, as well as your asset/liability management function. Chances are that all of the people involved in those areas are not going to be sitting next to each other. Typically, you will have one department on the third floor, while the other one will be on the 20th floor. Another advantage of a network is the ability to share data files, which will certainly eliminate a large degree of the error involved in transporting data files from one area to the next.

We believe mainframe linkage is extremely important. One of the tasks we have successfully attacked over the last year is providing our clients with the ability to "unlock" their mainframe. The obvious problem involved when moving those records into a PC environment is the task of "squashing" the seriatim listing into a workable format so that you may perform your modeling function.

The next area that I would like to review is commercial software that companies acquire to supplement their modeling needs. It is typically important that companies have some sort of spreadsheet capability. The spreadsheets usually supplement whatever modeling system you are presently on. It is very nice to be able to download your report data into a spreadsheet to make final touchups on your presentations to senior management.

A second item that continues to be on everybody's wish list is graphical output. This information is essential in providing senior management with a pictorial understanding of exactly what the numbers are saying.

I had previously mentioned multitasking. As the PC boom continues to grow, we think it is very important that your software be capable of running in a multitasking environment. Multitasking essentially allows you to process a number of tasks at the same time on one machine. So, while it may not be applicable in all cases, you should investigate your vendor's ability to support this climate in the next year or two.

The first question that people ask me about the construction of a model is, "How many cells are the correct amount?" I think it comes down to a very simple solution involving two parts. The first question is, "How fast is your software?" Or put another way, "How many cells can your software process per second?" Multiple cells per second is a little aggressive, but we hope to get to that point someday soon. The second part of the equation is, "How long can you afford to take?" This is the essential question that you must be able to answer. When you get right down to it, the fundamental question that you have to ask yourself is, "What am I trying to accomplish?" The best way to solve this problem is to continually push management to define the question they are asking and how they would like that question answered.

I think that the solution may involve an additional two-step process. We find it convenient to build two models, a large and a small one, with the smaller one built from the larger one. We continually find pressure to be extremely accurate, and this leads to a disproportionate number of cells in comparison to the task you are performing. And it takes you three days to run. The first thing that I would advise is to just make a "quick and dirty" run. This involves modeling your entire company with only 150 cells.

The second step would be to develop an overnight run of maybe 2,500 cells from the same data. During the day, you will be testing assumptions, growth rates for different years, different premium amounts, etc., with your "quick and dirty" run to get a better feel for the data. Results that you become comfortable with are entered into your overnight run.

In general, the best advice that I can give you is to concentrate on your most popular plans. Many times people spend a lot of time making sure that their model balances to extract reports for a block that represents 3% or 4% of their business. Make sure you spend a majority of your time on the plans that mean the most to your bottom line.

You may choose to eliminate lesser selling plans from your model. As a minimum, you should model the lesser selling plans with far fewer cells.

When building cells, one of the most important decisions you will make will be to decide on the breakdown of your assumptions reflected in the cells. Such items as male/female, age, average size, issue year, month of issue, underwriting category, premium pattern, etc., will need to be analyzed.

The most important tool currently being utilized in software is a tool used to "unlock" your mainframe to automate the modeling process. We call it a "model generator." Others are called "extract condensers." Their function is to take a large extract from your mainframe and condense it into a model office.

Once you've decided which cells you're going to model, you must build your "model office." It would be nice to have different populations from different lines of business. You should also integrate a corporate segment into your model. The corporate segment may function in a completely different manner than your universal life or single premium deferred annuity (SPDA) portfolio. This will force you to review your capital transfer policy.

We also build in something we call "macro parameters." This includes nonmarginal expenses (or overhead), taxes, including the deferred acquisition cost (DAC) tax, various calendar-year adjustments, trends, and other global assumptions that you do not vary at the cell level.

The separate corporate liabilities, corporate assets and corporate expenses were modeled as part of a corporate segment. We have modeled all of our different products in our portfolio. For asset/liability modeling, we build in the capability for segmented asset portfolios as well. Finally, we perform the consolidation among all segments. This includes tax adjustments, mandatory securities valuation reserve (MSVR), (IVR), etc. For reporting, we produce statutory income statements and balance sheets, GAAP income statements and balance sheets, cash flows, and a reconciliation of surplus. See Chart 1.

Once you have completed your "liability" model, we now must make some fundamental decisions on the asset side. Your asset model is comprised of four major areas: (1) interest-rate scenarios, (2) your asset portfolio, (3) a corporate behavior model, and (4) a policyholder behavior model.

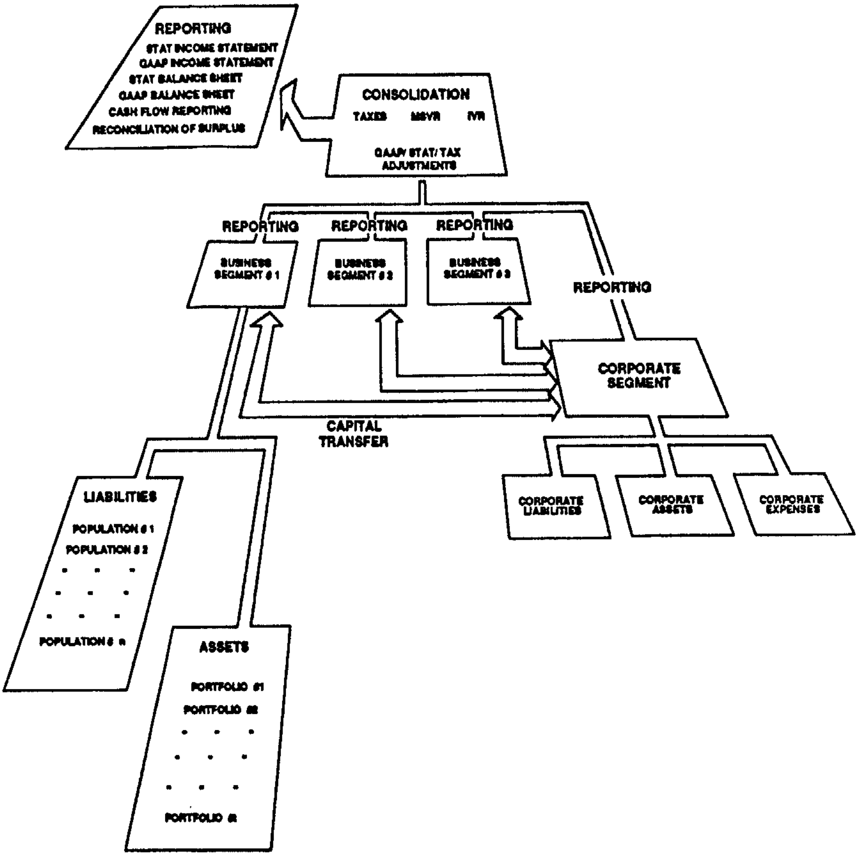
There are three options when entering interest-rate scenarios into your model. The first is deterministic, where you are just entering in multiple scenarios commonly made up of a short-term rate, an intermediate-term rate, and a long-term rate and projecting those over X number of years. Regulation 126 refers to seven specific scenarios and asks you to include a couple of inverted yield curves. The second option is using stochastically generated interest rates. These are commonly self-generated scenarios and will require you to test many more scenarios than the deterministic option. The final option, which we at Chalke Incorporated are internally moving to, is option pricing. These techniques have been commonplace on Wall Street for as long as I can remember. We are starting to incorporate option pricing techniques on both the asset and the liability side.

In order to have a successful corporate model capable of providing senior management with important information, you must be able to accurately model your asset portfolio. You should be able to download your current asset portfolio into your model, whether on a seriatim basis or in some sort of "modeling" capacity. You should also be able to model a majority of your current portfolio. You must be able to model calls, prepayments, defaults, sinking fund options, etc. on your standard bond and mortgage portfolio. You must also be able to model the newer offerings such as collateralized mortgage obligations (CMOs), interest-rate caps and floors, interest-rate swaps, etc.

The third major area is something we call the corporate behavior model. This is where we attempt to model internal corporate decision making with respect to interest crediting strategies, investment strategies and disinvestment strategies. For interest crediting strategies, you must be able to model new money strategies, portfolio strategies, investment generation strategies, and something based on an external index. You should also be able to portray your behavior with respect to such things as lagging rates or taking a weighted average over a certain number of months. Being able to model lags is essential to an accurate portrayal of your corporate behavior.

With respect to investment and disinvestment strategies, you should plan on spending as much time as possible with your investment department. And those of you that are already doing this, it's still not enough.

CHART 1
Corporate Modeling Flow Chart



The final area is the policyholder behavior model. You must be able to model how your policyholders react to certain market situations. If interest rates dramatically increase and your company keeps its credited rate fairly low compared to market, your policyholders are going to walk. You must be able to build those dynamic features into your model, without having to make adjustments by hand on a year-by-year basis. Other things that are affected would be persistency, partial withdrawals, policy loan activity, dormancy and new business assumptions. I believe you will see a proliferation of software applications in the coming years to help companies achieve better results in predicting policyholder activity.

I have already briefly reviewed some of the output. Certainly at a minimum, you should be able to develop statutory income statements and balance sheets, GAAP income statements and balance sheets, and a reconciliation of surplus. Any type of cash-flow reports are extremely helpful as well. Graphic capabilities are essential to senior management.

I will discuss what is commonly called a price behavior curve (Chart 2). Our software now has the ability to produce these for a certain segment of your business or your entire company. The curve is generated by using option pricing techniques to examine shifts in the interest rate environment. Unlike single-path valuation models, it allows you to accurately value embedded options in insurance products you sell or investments that you buy that are currently "out of the money." This tells me this particular company is in pretty good shape unless interest rates drop 200 basis points. A 250 basis point shift basically renders this company insolvent. We may recommend a 250 basis point interest-rate cap. In the coming years, you will see much more of these price behavior curves because of the wealth of information one picture conveys to senior management.

MR. R. LARRY WARNOCK: Like Mark's presentation, much of my presentation will be in the nature of an overview. Fortunately the overlaps in our two overviews have been accidently minimized. So I won't discuss too much of it all over again.

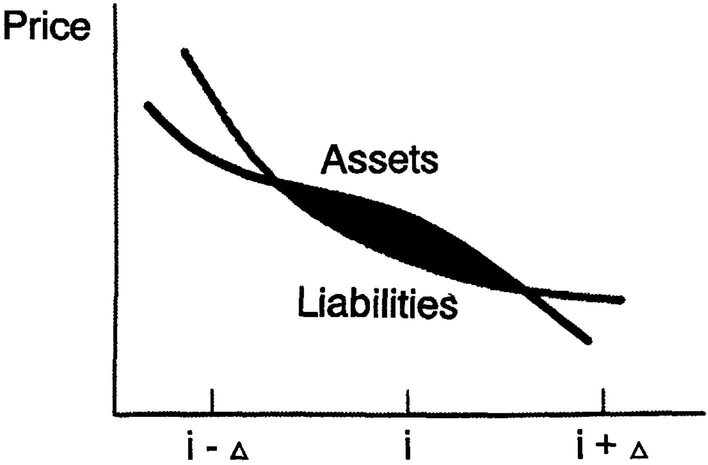
Okay, the first thing I want to do is back up in history a little bit. Having a few gray hairs, I can remember how things were done in the past. I'd like to contrast very briefly the past versus the new technology.

Historical uses. Thinking back to the mid-1970s, when I first got into the actuarial game, refined financial projections were just not routinely developed by very many insurance companies. There were some fairly good reasons for this. First, the process to produce those financial projections was fairly burdensome due to the limitations of technology and software. The PC had not been invented then, or at least was not nearly as powerful as it is now. And software packages were not widely available.

Model building often used manual processes. When I first started building models, we would go to a company and sit down in front of their hard copy valuation records and build the models by hand. In that environment, models with single average ages per plan were quite the rage. There were all kinds of justifications for this when you did it.

CHART 2

PRICE CURVE ASSET/LIABILITY



Actuarial appraisal work was fairly basic then and could be laborious. You tended to go off and develop factors of present values of future profits on a cell and duration basis, and then apply those factors to inventories of insurance in force. That is in contrast to the way we do it today, where everything is driven by some kind of a model financial projection, which communicates to management much more effectively.

Projections for planning and budgeting often used extrapolation techniques, as actual modeling was practiced by very few companies. Pricing projections had a single cell focus, often using asset-share-type calculations. The terminology "macro pricing" was not around at that time.

But perhaps most importantly, back in the mid-1970s, the industry was more stable, more predictable and products were simple. There just was not a perceived burning need to go out and implement ultra refined projection systems of the type that we have now.

Turning to technology, I guess the advances in personal computer capabilities during the last decade have been one of the major things that have made it possible for us to have such advances in the modeling approaches that we use now. Also, the availability of PC-based software packages from several credible vendors has allowed virtually all insurance companies to acquire in-house modeling capabilities at a reasonable cost. Finally, the migration from mainframe computers to PCs has been an important factor too, because it gives the actuarial department more control than they typically had when everything was done on mainframes.

I'd like to now just briefly go over a laundry list of current uses, also known as "Reasons for Doing Projections and Models." In the current environment that we see in 1991, many companies have acquired modeling and projection capabilities just during the last few years. I would guess that as recently as five years ago, it was a relatively small percentage of companies that had full in-house projection capabilities. While major consulting firms tended to have elaborate projection systems, they were mainframe-based, and not readily transferable to companies on a licensing basis. But today, I would say there are relatively few U.S. and Canadian companies of any significant size that don't have modeling and projection capabilities in-house. And driving the interest in these projection capabilities has been a long list of the things that you do with them, including product pricing and repricing, actuarial appraisals and financial planning, asset/liability management, and the newest regulatory thing, cash-flow testing, and even financial reporting using a value-added basis, or SFAS 97. Both tend to be driven by model projections where all the cell projections are aggregated which is different from financial reporting calculation techniques of the past that used cell-based factors.

Now what is the control cycle analysis all about? It's an analysis that begins with pricing and projections, followed by validation of results against emerging experience and continual fine tuning of models and assumptions. I think the control cycle is being done by more and more companies, even though they may call it by different names.

Just quickly, I will give you a list of some desirable system features. Before you build or buy any kind of a modeling or projection software, you need to set the objectives that you want to meet, and the desirable features of your system. That includes the software, the hardware, and the processes followed to get the project done. First of all, as I said, there are a number of reasons for doing financial projections. Ideally your system should serve all these needs to avoid duplication of effort and confusion and to get some economies of scale.

I took some of these quotes off this little calendar I had. I picked up the ones that seemed to be relevant to computer and modeling work. One is Sutton's second law, and it says, "The most useless computer tasks are the most fun to do." And I've come to believe that. So when you start doing modeling work on the computer, be sure that you set up a system that is efficient with resources. And it needs to be efficient with staff, computer hardware, and produce a reasonable product for a reasonable commitment of resources. And just remember, one of your goals is not to let your actuarial students have a great time.

Next, the system needs to be dynamic with respect to the key drivers, allowing ready testing of alternatives where a single deterministic assumption is not adequate.

The system needs to be understandable. I would submit that mysticism is not an advantage, and understandability enhances credibility of results. My next clever quote is, "Only someone who understands something absolutely can explain it so no one else can understand it." Many of you who have been involved with software probably believe that.

The next to the last desirable feature is that the system should be practical in application and result. The quote is, "The simple explanation always follows the complex solution." Finally, one day a light bulb goes off and you think, "Oh, so that's why that is. I should have understood that before doing all this complicated analysis."

Also, the system should provide a means of flexibility to handle those things that just will not be thought of in advance, or that just don't seem appropriate to the vendors of commercial packages to put into their generalized systems. So you need some way to achieve flexibility.

Now I'd like to talk about the key steps in building models and actually doing financial projections. This is a deceptively short list which might imply to a novice that it's pretty straightforward and easy to get a financial projection done. The first step is to determine the lines of business. You might refer to them as segments that make up the company, and construct the model or even a series of submodels for each one of those segments. And as Mark has discussed already, this can be expedited greatly by using a model generator or model builder system. Having selected that model, then you have a lot of data gathering to do. Essentially, you have to find the product specifications and assumptions that apply to each cell that you're going to include in your model. This data gathering process can be a fairly major task. And its size is directly proportional to the size that you make your model. So the more plans you put in, the more data you have to find, etc.

Next, you need to build all the data files that are needed to feed the projection software. Some of this data may be broken down by that data which varies by cell (i.e., cell-based data), that data which only varies within a model, and finally the in-force and new business records that you have to feed in. These too can be fairly major tasks.

The next thing you do is turn on the machine, run the submodel projections, and wait. Be sure that you scrutinize your results very carefully when they come out of the machine.

The last two things are to combine the submodels that you run into the lines of business or the segment totals, and develop those calculations that can only be done at the total company level. For example, things like projecting investment income on the free surplus and handling the existing tax loss carryforwards. It's very difficult to allocate those to a line of business and those are more logically done at the total company level.

Don't forget to validate the models. This should include a static validation and a dynamic validation. A static validation is like a snapshot to compare the model versus actual in-force data, in-force premiums, face amount, count and reserves. A dynamic validation is like a moving picture comparing income items that are projected versus what's actually being recorded in financial results.

Now, what is model construction all about? Model construction simply involves selecting a limited number of model cells to represent the actual in-force business of the company. In the PC environment, or any environment for that matter, seriatim projections are just not practical. I know that's an arguable point, and I seem to have that argument with someone every now and then, but it's my very strong belief that you just don't do seriatim projections. So models are absolutely essential.

In addition to selecting your model plans, part of modeling is also deciding how to handle plans that are not in the model, i.e., the nonmodel plans. Assigning the nonmodel plans to model plans or just using some kind of gross-up approach are the usual ways. And finally in-force files and new business files, or the population files as some of you may call them, must be created in the right model format in order to feed the projection software.

Now I'd like to discuss what a model cell is. It's important to understand what a model cell actually represents, because in the projection software, the computer is actually doing a separate projection for each model cell, and then aggregating the results. The more cells that are included in the liability model, the longer the machine is going to run. So controlling the size of the model has some very definite benefits, if you can control the size of that model and get it to validate all at the same time.

Generally a model cell is a combination of parameters. It includes plan, age, issue date, mode, size, and sometimes other features.

I might say one thing about data structure in the software. It's important to have a data structure that doesn't multiply the data storage requirements just to expand certain things, for example, issue month. If you were going to run a model that had one issue month per year of issue, that creates one cell. Now, if you decide you want to be real refined and have 12-issue months, you want to be sure that you don't create a data structure that multiplies your data storage requirements by 12.

Now, I'd like to discuss what a model builder does. I think a model builder is an essential piece of software for effective modeling. It may give you reports to assist the actuary in selecting model plans. I don't see a model builder as actually selecting those plans for you, but it can give you information that expedites your selecting and identifying to the software what those model plans are going to be.

The model builder can also take your input and group nonmodel plans with appropriate model plans. Having done those things, you can also have your model builder do a mechanized selection of average issue ages after the actuary has defined what the age groupings are. The main thing that a model builder can do for you, other than selecting the model, and this is a very important thing, is to do a mechanized creation of the in-force data files that have to feed into your software and to do it in the format required by the projection software.

I'd like to say a little about the static validation of the model, which is a bit different for traditional products than for interest-sensitive products. Traditional products are pretty straightforward, because the key in-force driver is usually face amount, and the schedules of premiums, reserves and cash-value factors per thousand are fixed. So development of the model in-force data that you're going to compare against actual in-force data is simply a matter of multiplying the in- force face amount by the appropriate schedules on the model start date. An observation here: for a traditional in-force model, anything that happened in the past is irrelevant. So if you have a block of business that's five years old, the first five years are really irrelevant for your static model validation.

For an interest-sensitive product, things are a little different. Normally what we do for an in-force interest-sensitive block is to take the starting fund value as an input item. When we were naive, we used to think that this solved all our problems, because starting with the current fund trues everything up so that we didn't have to worry about past history. But consider things like retroactive bonuses (for example, a refund of cost of insurance charges). If you have an in-force block of business, the refund may occur 10 or 15 years hence and may be affected by things that happened in the past. Given that, you need a way to get these past cost of insurance (COI) charges, or whatever the carryforward variable is, into the system; the process is often more involved than meets the eye.

Now, I will talk about investment income projections. The two methods that are generally used in projecting investment income are the extremes. The first method jumps directly to the portfolio-yield assumptions, making assumptions about the composite yield rates for each submodel that will apply in each year. The second method is a full blown model of the assets. Asset modeling, of course, is a fairly large topic in its own right. There is a middle-of-the-road approach that's fairly simple to implement where one makes assumptions about yields and maturities on old money and new money. The software then calculates the resulting portfolio rates. The advantage to this approach is that the result is responsive to changes in underlying new money assumptions. You might even consider this a rather simplistic type of asset model.

Let's discuss typical system structure. In referring to the liability model part of the system, we have our various cell based data called product data files. Essentially, product data files are the assumptions that are cell based. So you have many of them, because you have lots of cells. The model level assumptions don't vary by cell, but vary within a particular model as in your portfolio interest assumption. The in-force records feed the system. You have to know how much business you have in-force on the model starting date and how much new business you will issue. All of these data files will feed into a liability projection and might be supplemented by an asset model module that integrates back and forth with the liability model. Finally, you have your reports and extracts being produced.

I will talk just a little about the flow of calculations in the projection software of yesterday and today. I'll describe how the mainframe software worked that we invented about 1979 that was mainly for traditional products. Basically, we first ran profit studies on all the cells. Then the model system took the appropriate amounts of insurance in force for each cell and did some multiplying and adding on those stored profit test results. And that gave you a model. So the way projection systems used to work was to do all the calculations for your first cell for all the years and store them, then go to the next cells, store them, etc. until you're all done. Then you would just sum up all the cells. And that process made for a fairly straightforward system architecture.

It's more complicated when you have an asset model integrated with liability models. Particularly when things going on in the asset model or the economic environment may affect policyholder behavior, and therefore, your liability assumptions. So what you have to do here is to first take your first period and run through it for each cell that you have in the model. Then you sum all of these liability cells for the first period, and then move on to the asset model calculations for that particular period. Then you go to the next period and so forth. This means that the data for all the cells must be in memory all at one time, or your piece of software is picking it up off the disk somewhere. This change in the flow of calculations was a major obstacle in terms of the complications for system architecture.

The last thing I'm going to talk about is total company models. It is my observation that it's not a good idea to run one giant projection representing the entire company. At least not until you've debugged the components piece by piece. If you did that, and if there were errors, first of all you might not spot them as being obviously unreasonable. And even if the result is obviously unreasonable, you probably wouldn't know where to look. Reasonableness checks are better done if the projections are done in pieces and later combined.

There are usually a large number of items that are not easily allocated by segments and are more logically handled at the company level. I've mentioned these already, like the investment income on free surplus, tax-loss carryforwards, shareholder dividends, expense budgets, etc.

You can do the total company model as part of your base software. You can combine that all together. But another thing that I have found to be very handy is the Lotus spreadsheet. When you're putting together the total company model, all you are doing is multiplying and adding. So you don't have the same kind of complexities that you have in your base actuarial projections. There are all kinds of complications that you encounter from company to company. Therefore, it is very difficult to have a commercial package completely generalized to take care of all the things that might happen in a company. So if you find that the package won't quite do it, it's pretty easy to do a total company model that brings all the components together in a Lotus spreadsheet. In addition, if you do the total company model right, you can get virtually instantaneous sensitivity testing on some of the key variables, such as new business production and growth and expense levels. The typical way we might do this is to run our new business models on a one-million-of-premium-issued basis. The actual production amounts in each year would be one of the input variables to the spreadsheet. These would then be stacked up within the spreadsheet. We find this to be fairly effective, because you don't have to go back and run your basic new business models. Now we'll turn it over to Tom Nace.

MR. THOMAS NACE: Two years ago we purchased and installed modeling software at my company and integrated it into the major actuarial functions. Previous to purchasing this software, we had a home-grown system which did our product pricing. We also had a separate system which had been designed in-house to do corporate modeling. With the advent of Regulation 126 several years ago, we then designed, on an as-needed basis, a model to do the asset/liability projections. Each of these systems was independent of each other and each had its limitations which were quickly approaching.

With the emerging importance of the role of the valuation actuary and the need for more sophisticated cash-flow modeling, as well as our own management's desire to improve our corporate financial projection capabilities, we decided to replace our existing models with one system which would hopefully be able to satisfy most of our current and future modeling needs. This project took about 17 months from beginning to end, required five people full time, and approximately 10 people devoting at least half their time.

What I would like to do is present, from an insurance company perspective, the process we went through in installing new modeling software, highlighting the key issues which we faced, and hopefully in the process provide some helpful information to those companies who may be contemplating the purchase of modeling software in the near future.

We divided the project into two phases. The first phase involved purchasing or designing the software needed. The second phase was the implementation phase which involved model construction and validation.

Let's first address Phase I. Before you purchase or design your own software, it is important to lay out what your objectives are. Why are you buying the software? What benefits do you hope to realize through the new software?

At Penn Mutual, we had four objectives. First, we want to improve the overall accuracy of our model. Our second objective was timeliness. We wanted a model which would require minimal effort to update and use. Also included in this objective is actual run time for the system chosen. A third objective was to combine the modeling efforts currently taking place throughout the company. In our case this included (1) corporate financial projections, (2) product pricing, and (3) Regulation 126 cash-flow testing. In addition, since we produce an internal GAAP financial statement, we were also looking for a system which could generate GAAP reserve factors consistent with the recent SFAS 97. A fourth objective was to improve the overall documentation of our models and the modeling processes.

After you have set your objectives for the new software, you must then decide whether it is best for you to purchase the software from a vendor or to develop the software in-house. We studied each approach and outlined what we saw as the advantages and disadvantages of each.

Let's first look at the option where you develop the software in-house. The key advantage is that the model will be tailored to your company's specific needs. In addition, you will have more control over the project because you don't have to rely on an outside vendor for modifications, etc. A third advantage is, after the project is completed, the developers and the source code remain in-house. Finally, the new model will be familiar to users since they had a role in designing it.

The major disadvantage of developing the model in-house is that the time it takes to design and program the model may be very lengthy. As a result, it may turn out to be quite costly. It will also probably be a lengthy process to validate all the formulas and logic. A real practical disadvantage is that you may not have the technical expertise available to develop the software to either suit your company's needs or to match the same level of sophistication currently available in the marketplace.

A second option is to purchase software and use it as is. This approach has the potential for a much quicker implementation. When you purchase software from a vendor, you usually receive maintenance support from the vendor as well. In addition, you are eligible to receive system upgrades which the vendor makes available. Receiving upgrades for the new DAC tax, for example, means that you don't have to maintain an ongoing programming staff to handle such developments. A lesser advantage when you purchase software is the opportunity that exists to evaluate and trade ideas with other users of the same software.

Disadvantages of purchasing software and using it on an as is basis, include the high upfront costs that will be incurred. In addition, there may be annual maintenance fees that must be paid. Depending on the vendor, there will be a greater or lesser risk of insolvency. Another risk may be that the vendor could decide to stop marketing and maintaining software in order to concentrate on other aspects of his or her business. A fourth disadvantage is the unlikelihood that the software will meet all of your needs as purchased. Most software that you purchase, by necessity, is designed to handle generic situations that all companies will be faced with. Consequently, it will not handle company-specific product features, for example.

When you purchase software there is also a learning curve for the users. This is important to keep in mind when planning the implementation of the model, because it will have a big effect on your schedule.

Finally, when you purchase software you may, again depending upon the vendor, lose flexibility to custom design the software to meet your specific needs.

A third option as far as the software is concerned, and also the option which we selected, is a combination of the first two options -- that is, purchase the software which best meets your needs and then, depending upon the magnitude of your needs which have not been filled, either modify the software in-house or get a commitment from the vendor to modify the software so that it does meet the majority of your needs. At our company, the software modification issue was a major one, because at the time we were in the market for software, there was not any software available to our liking which would have handled pricing, cash-flow testing and the ability to do consolidated company financial projections. The one piece which tended to be missing was the consolidated financial projections. We worked with the vendor in designing this piece, and it has since been incorporated into the vendor's software and is available to all companies. The advantage of this third option is that you tend to maximize the advantages of both options one and two while minimizing the disadvantages.

Once you make the decision to purchase or design your own software, the next major decision you're faced with relates to hardware. Should the system be PC based or should it be on the mainframe, either your own mainframe or in a time-sharing arrangement.

The advantages of a PC-based system are increased flexibility and control. The PC systems tend to be more user oriented. Also the PC systems are a lot easier to modify, since the logic and formulas are accessible.

One advantage of a mainframe system is the quicker run time, although the actual turn-around time may not be quicker if it has to run overnight or off-site. Another advantage of the mainframe system is the ability to handle larger files normally without the same constraints you might have with a PC.

Having decided on the type of system you prefer from a hardware perspective, you must then determine the estimated cost of the project, and what, if any, cost constraints you have. Whether you design the software in-house or purchase from a vendor, whether it's a PC based or a mainframe system, the costs will vary. Included in your estimate should be (1) the cost of the software, (2) the cost of having the vendor incorporate any desired modifications, (3) the costs of additional or more technically advanced PCs, with the appropriate memory, back-up features, etc., and (4) the cost of any outside training or resources that may be needed to implement the software.

In conjunction with estimating the potential costs, it's also important to assess what resources you have at your disposal to assist in the implementation of the software and determine what additional resources, either internally or externally, you might need. Here, thought should be given not only to the number of resources, but also to the type or resources needed (e.g., actuarial, systems, accounting, investment).

Having determined costs and resources, it is important to get management's buy in on these issues and on the time commitment that will be involved. All of the above steps are suggested prior to the actual evaluation of the software in order to facilitate your evaluation by focusing on some of the key factors which distinguish the various software packages. Also in this vein, it is desirable to assemble ahead of time a checklist of information, level of detail, product features, etc. that you would ideally like to see handled by the software you intend to purchase, and then evaluate each of the software packages on each item.

Assuming you have purchased the software, Phase II of the project is the implementation phase. One of the first things that should be addressed before the actual implementation even begins is planning. In our case, because of the size of the endeavor we were undertaking (i.e., replacing our pricing, corporate projection and asset/liability models with one system), planning played a critical role.

It was helpful to first determine the needs of the various user areas and then set up a plan which would work towards developing a process to integrate the various areas. It is also desirable to define upfront, where possible, what interfaces with existing systems would be helpful in order to minimize the amount of manual coding that might otherwise be necessary. For example, you may want to have an interface which loads unit dividends from your dividend system into the traditional life model for participating policies.

The next step in the process is to validate the software for new business. Projections for a given pricing cell should be produced using the new software. Results can then be compared with existing pricing runs using the same assumptions. If you are replacing an existing corporate model, then comparing projected results for a given product is also desirable.

Any differences that do occur will fall into one of three categories. The difference could be due to a variance in methodology. Pricing systems might assume annual benefits and premiums, while the new software may distribute the cash flow evenly throughout the year on a monthly basis. There also may be differences because the purchased software may not handle company specific product features. The third reason for possible differences is that there may actually be an error in the software or existing pricing models.

Once you have identified the differences you should then evaluate them for materiality. Immaterial differences in methodology may not be worth changing. Other differences may actually be an improvement over what is currently produced. For those differences which are not material and do not represent an improvement over what is.already currently produced, you will probably want to make a modification to the software either through the vendor or internally.

Once all the differences have been identified and a plan to rectify the differences has been put in place, you can begin to build the new business cells for pricing and corporate modeling purposes. For each product, define the number and types of cells needed, keeping in mind the overall size of the model. The number and types of cells will vary by product. For the corporate segment of the model what we attempted to do was start with the pricing cells, which in our case were more detailed than we needed. We then reduced the number of cells to a more acceptable level and at the same time tried not to materially alter the overall profits by year.

In order to build the cells, you will need to determine assumptions for lapse, mortality, etc. If you plan on comparing product projections with an internal source, you should choose the assumptions to be consistent with the other source.

By next inputting or reading in all policy data, you can build the cells and populations. At this point you will be in a position to run the new business portion of the model and analyze the projected results.

The remainder of this discussion assumes that the pricing phase of the implementation is completed and relates entirely to the corporate model function. Building an in-force liability model is much more difficult than building a new business model because of the volume of business, the number of different products that may be in-force and the desire to match actual results. Because of the significance of the in-force business, having an accurate in-force model is key to the overall reliability of the model. As with the new business, there is the need to balance accuracy which implies a greater number of plan/age cells with run time, which is shorter with a fewer number of plan/age cells.

Some software packages contain a model generator to assist the user in modeling the in-force.

For blocks of in-force business which are relatively insignificant, you might want to use a more simplified approach.

As with the new business, once you have chosen the plan/age cells, you must obtain and feed your appropriate cell data and assumptions into the model. If you are reconciling to an existing model or system, the assumptions used should be the same as those used in that model.

Similar to the process for the liability model, we thought it important to first verify the calculations that were being performed within the asset model for such items as book value, market value and net investment income. This was done for each asset type handled by the software by comparing the model results against hand- calculated values using dummy assets. We also tested exactly how certain asset features like calls and defaults worked. Again, understanding how the model works is important in reconciling results with other systems and also in analyzing the results that you get from the model.

The fixed-income investments were downloaded from existing investment systems and were fed on a seriatim basis into the model. A verification of the beginning book value with comparable values from published financial statements was made. Adjustments for variations of the fixed-income assets which could not be handled by the software were necessary (e.g., interest-only mortgage with a balloon payment, etc.). Equities were modeled using one asset for each asset type. For example, all common stock was modeled as a single common stock asset.

There may be some other asset types represented in your in-force which the software does not specifically handle. A decision must be made by the user as to the materiality, whether a modification is necessary or whether its projection can be approximated by using a different asset type. The overall projected results from the asset in-force should then be analyzed for reasonableness.

Our asset projections were done separately by segmented account. Once we were reasonably happy with the asset projections, we performed an asset and liability projection, again on a segment-by-segment basis. Within each segment, expenses were derived on a direct basis. Federal income taxes were calculated on a marginal basis by applying a marginal tax rate. Through a stockholder dividend option in the software, we set assets equal to liabilities for each segment. Excess assets at the segment level or additional assets needed were taken care of by a stockholder dividend to or from the corporate or surplus segment which we created. Also in the corporate segment were (1) any overhead expenses not provided for in the business segments, (2) any surplus tax or other tax items not reflected in the business segments, and (3) any other peculiar items which were not projected anywhere else.

The next step was to combine all of the segment runs, including corporate, in order to produce a consolidated company projection. Finally, a comparison of the projections produced by the new model on a consolidated basis were compared with the projections which he had previously shared with our management using the old model. Then a reconciliation of the differences was performed. Documentation of the entire process and the results was key to concluding the implementation of the model.

To summarize, I have reviewed the various building blocks needed to produce consolidated company projections. The pricing cells, when combined, can be used to separately model the new business and in-force liabilities by segmented account. These liabilities will then be combined with the segmented assets to produce projections by segment. Running all of the segments together with the corporate or surplus segment through a consolidation routine will produce consolidated company projections.

Before I wrap up, I'd just like to discuss a couple of related issues which deserve special attention. If your company plans to do current-year projections throughout the year, reflecting year-to-date actual results, you may want to consider a separate model to do those projections, depending on the software you purchased. In our case, we do current-year projections at least quarterly throughout the year, so we developed a PC-based system in-house to track actual data and experience and also to project that data for the remainder of the year.

The corporate model that I've talked about may not be appropriate for this type of projection for two reasons. First, the assumptions that are used for corporate modeling are usually representative of actual experience over the long term, and will not match year-by-year fluctuations. Second, most of the software packages that we looked at were difficult to adjust on a item-by-item, year-by-year basis. If you use a separate system, however, it will be necessary to reconcile the projections for the current year which are produced by the corporate model.

The second issue which I touched on briefly, but for us was a major issue, is software modifications. If you plan on making a significant number of modifications, it is very important from a control standpoint that the modifications are well documented. This will be particularly important as other users begin using the software and when new releases from the vendor are installed. The work involved with refitting the modifications to new releases is also a good reason to think twice about the number of modifications that you may want to make.

Our projection and pricing model is PC based. Because of the number of users and uses of the software at our company, we decided to develop a local area network connecting all the PCs to a server. This facilitates the exchange of data and files among the users, and provides for consistency of information between areas. In effect, we have imitated a mainframe environment while maintaining the flexibility of PCs. A side effect of this configuration is that we had to designate a system administrator to manage software and hardware issues, as well as security and disaster recovery. In addition to the system administrator, because of the new model, we had to reexamine our staffing needs and job responsibilities.

Training of other users not involved with the implementation and reexamination of your processes and procedures is yet another step in the overall integration of the software into all of the actuarial functions for which it was intended.

MR. MAHANY: We now have some time for questions.

MR. CHARLES ROBERT DOLEZAL: In validating your model, what are people using as tolerances to how close you want to get to say matching your reserves?

MR. WARNOCK: Well I guess I've seen a lot of companies, a lot of consultants and a lot of in-house actuaries doing these things. And they tend to be all different. You will occasionally find the extreme actuary who, if he comes out with a model-to-actual ratio of 99.2%, is going to work on that for another few weeks to get it to come out 99.9%. But then you will find some who get 95% and they're quite happy. Mark mentioned the two different types of models. An example of the type where you want the liability model very refined, is where you're going to do financial projections or an appraisal where you run the model that you run one time. You probably want a very good validation ratio on that. But if you have the limited model which is intended for scenario testing, then you must have more tolerance there. I would think, in terms of a static model validation, once you move away from 95%, that's probably going to be significant.

MR. DOLEZAL: Also on your static model do you sometimes use more than one point in time to validate the model?

MR. WARNOCK: For the static validations, we typically don't do that because we feel pretty comfortable with them if we have them in one point in time. Now, to the extent that your new business model might be different than your in-force model in some of its parameters, you assume the same parameters applied to both the in-force and new business model. That may get out of whack as you go forward. So let's say you have a particular plan and you've set up the parameters on it. Maybe it all validates this year, but next year, after you've added one more year of issue, it may have some different averages that apply to it. That same model may not validate a year later. So you have to watch for things like that.

MR. MAHANY: It is important to know what is being asked of you. Try to keep the entire model in focus. You will find people trying to balance to 99.9% of their reserves and they are using a 12% new-money rate for 10 years. It really doesn't help you to be 100% accurate in one area and about 10% accurate in the other area. You must strive to be consistent with your assumptions.

MR. NACE: One other comment I'd like to add is that the degree of validation is subjective. It is up to the individual company to decide how accurate it wants to be versus how quickly it wants the system to run. You can certainly make the model more detailed in order to improve the validation with actual results, but you may also increase the amount of time it takes to run the model.

MR. KEITH E. GANT: I'd like to hear comments from any of you on what is the most appropriate period of computation. Should the internal calculations be done annually, quarterly, monthly? In gaining accuracy, is it really going to make any difference in the conclusions that come out of the process?

MR. WARNOCK: I would guess Mark and I would answer that the same way, since we both utilize monthly processing. My thought on monthly processing is no you don't necessarily gain additional accuracy. And in many cases it is possible to write approximate formulas that will give you very close to the same result without going through the monthly processing. On the other hand, the detailed formulas that support our software take up 2,000 pages. And when you start trying to write the approximations for this that apply to all the different variables that you've got in there, it can become fairly tedious. And the big advantage to the monthly processing is that the formulas are very straightforward. But you do get that extra baggage of more processing time.

MR. MAHANY: We actually went through the process that Larry was speaking about when we found that we were spending so much time validating the approximations that we were making for annual processing, we decided it was easier to move to monthly processing. It made everything much cleaner. You will be surprised to find that with most languages, if done in the correct manner, monthly processing will not take that much more time.

MR. NACE: I think in some cases monthly processing is actually the preferred method, or at least we view it as such. In particular, if you're talking about contracts like a GIC where you may have large amounts maturing in the future, to assume that they all mature on July 1 of a calendar year may result in a lot different results than assuming monthly maturities which can be skewed during the year. Consequently, there are some cases where I believe it's really beneficial to have monthly calculations.