1998 VALUATION ACTUARY
SYMPOSIUM PROCEEDINGS

SESSION 44OF

DATA INTEGRATION FOR VARIOUS CORPORATE PURPOSES

Cheryl A. Krueger, Moderator

Brian Cole

Roger W. Smith

DATA INTEGRATION FOR VARIOUS CORPORATE PURPOSES

MS. CHERYL A. KRUEGER: We're going to talk about data integration. I think that bodes well for the future of data integration. I will do a brief introduction, and then we'll go through some prepared presentations. At the end, we will give you an opportunity to ask questions as well as share your experiences in the data integration area with everybody else in the room.

I have been with Tillinghast for about seven or eight years, depending on when you start counting. We also have with us today Roger Smith, the president of Polysystems, and Brian Cole from ACORD. I'm sure you're all familiar with Polysystems. Roger has been with Polysystems for 17 years. Polysystems develops actuarial-related computer solutions for the insurance industry. Besides his responsibilities at Polysystems, Roger is the past chairman of the computer science section of the Society of Actuaries. He has been involved in various program and research committees for the Society. He's going to talk about some of the issues that need to be addressed as companies strive towards data integration.

Brian Cole is with ACORD. ACORD is a self-funded industry action group that's responsible for developing and managing the OLifE specification. The work with vendors and clients to help make the sales process (as opposed to the back-end process) more efficient. They work with specific vendors and companies to put data standards in place. Brian will provide an interesting perspective of what is possible when companies and vendors work together to provide a format for data integration. Brian is a "technical architect" at ACORD; his job is to help put together and review the standards that actually go into the OLifE standard. Brian has been with ACORD for about one year. Before that, he worked at Prudential and helped to develop the first system actually released using the OLifE data standard specification. So even though Brian has been with ACORD for about one year, he has had quite a bit of experience working with the OLifE standard.

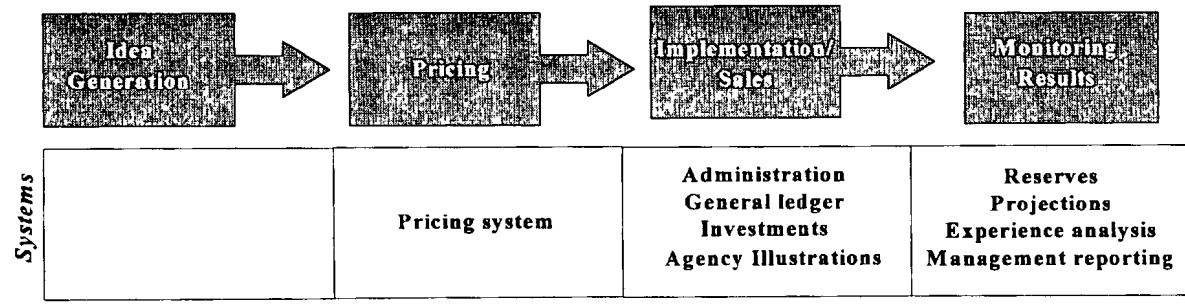
I'm going to talk about the benefits of data integration. How many people deal with data that's transferred between two or more systems? A pretty good number. How many of those people have

a streamlined, efficient way to make that data move seamlessly between those systems? A majority of the audience does that kind of work, and a couple of people say they've made some progress in that area. I'm not going to spend a whole lot of time discussing the benefits of data integration. I would like to describe the current environment that we're working in based on a product perspective. I'll then share with you the results of a survey that we did to show how automation can help to make your company work more effectively. Next, we'll talk about the opportunities that exist when data are integrated. Our other two speakers are going to talk about potential solutions for the data integration issue.

I'm going to start out with a product focus. You can look at the data integration problem in different ways, but products are really the driving force behind what we do. We start with idea generation somewhere in the marketing or product development area. Then we move to doing some pricing work. We then get into implementation and sales. Finally, we get into the area where a lot of us work, that being the monitoring of the results of what's happening to those products. Throughout that whole process, we use all types of different systems to do what is required by those different steps.

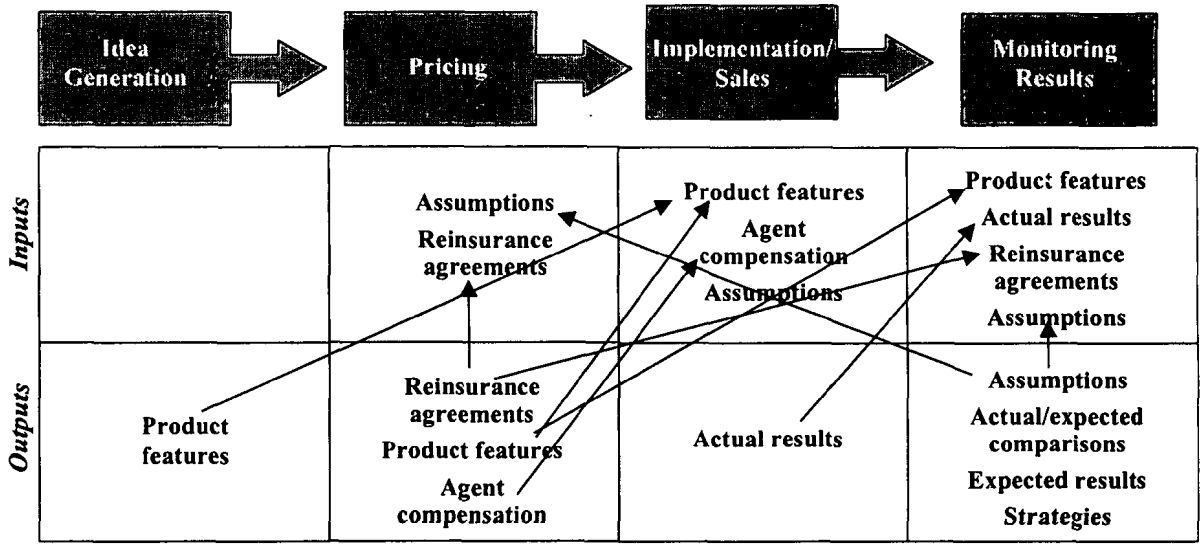
Our ideas turn into produce parameters and features that need to be input into a pricing system (Chart 1). Through pricing, we develop commission rates, premium rates, and so forth. These rates need to be entered into the following systems: administration, illustration, projection, reserve, and any other relevant system(s). The data sharing continues through the whole process.

CHART 1
Different Systems Used to Support Each Phase



Let's go through a few of the inputs used in this process. We start with product features as the major driver. The plan definition has to go into the pricing system, the administration system, the reserve system, the projection system, and the illustration system. When we're doing our pricing, we get our assumptions from data that we have in-house or from outside the company. That has to go into our pricing, projection, and our reserve systems (for GAAP reserves). We have to provide data to our reinsurers, and we have to set up our agent compensation system—you can see where I'm going with all these data. It's not a real pretty picture in terms of how much data needs to be shared between these systems (Chart 2).

CHART 2
Each Stage Provides Data to be Used by Other Stages



How did we, as an industry, get here? Many of the systems that we have were developed to do what they do best, but not necessarily by the same people. Vendors developed some systems. Some of the systems were developed in-house. As these systems were developed, there wasn't a whole lot of push from within to get multiple systems to talk to each other. In the past few years, it has become difficult for the everyday user of the data to retrieve this information from the mainframe

system. We got used to a culture that required a programmer to get access to the data we need. That's not necessarily true anymore, but historically the environment has been one where we were used to going through some trouble to get our data.

When vendors did what they did best, they also tried to tack on a few things, whereby they created some subsystems rather than coordinating with other vendors. Thus, if a software vendor had a really good administration system, rather than work with a valuation system vendor to move data between one and the other, he might have created something that would calculate reserves; possibly for some of the products, but maybe for all of them. Even if the vendor did an excellent job in creating this new information, it was hard to get the data in the format you wanted.

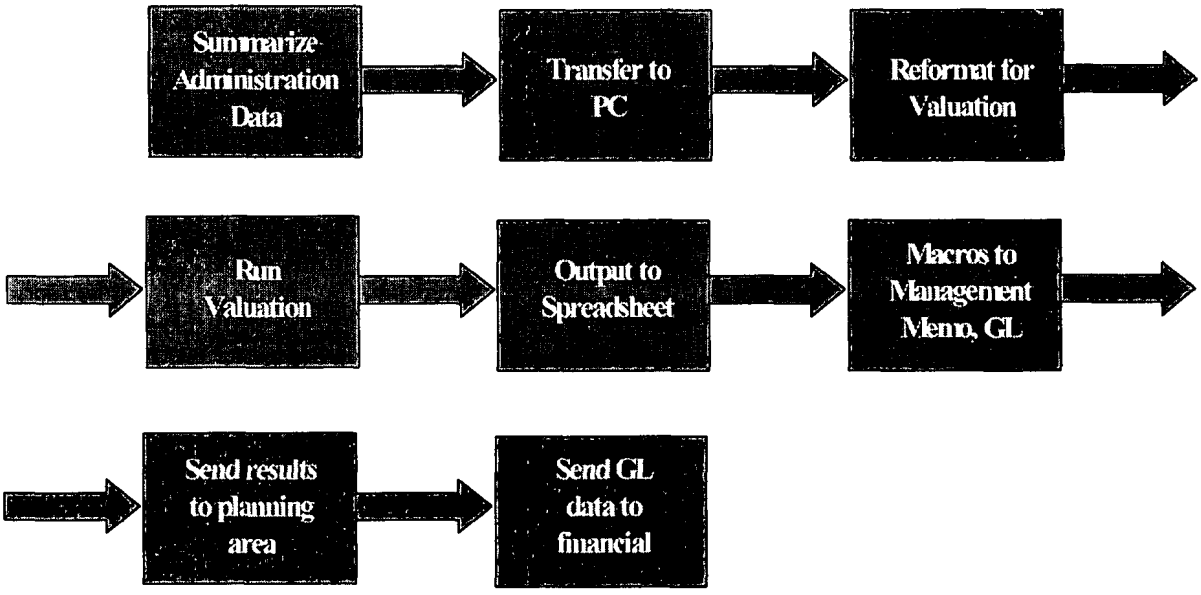
The demands that were placed on systems were adjusted to fit within desired time frames. People weren't demanding five days to close the books. We didn't have to monitor weekly lapses like we now do on some lines of business. Nobody demanded quarterly plan updates because it took an entire quarter to develop the plan.

What was the impact of this environment on our productivity? We surveyed some companies to see how long it took them from the time the actuarial area got the data they needed to close their monthly or quarterly books, to the time they passed the data on to someone else, to move it through all the different manipulations needed to get it from their administration system into their reserve system, and so forth. We had the companies grade themselves based on how highly automated they thought they were. We had them grade themselves in four different levels: 1) highly automated, 2) moderately highly automated, 3) moderately automated, and 4) not very well automated. Two-thirds of the companies graded themselves as highly automated or moderately highly automated. Those companies had an average of six days for their periodic processing compared to 11 days for the companies who said they weren't so well automated. Assuming that you close books monthly, you can automate this process to the point where you could add 60 days of productivity to your department. With the shorter time frame you're not managing data. You're actually doing analysis or can add some value to the company. That's why data integration is so important.

I'm going to go through a couple of examples of things that we've seen in projects where we've tried to increase the efficiency of some area of the actuarial function.

First, let's look at the monthly reserve calculation in Chart 3. The first thing to do is summarize the administration data in a format that goes into the valuation system. The raw data comes from other area of the company. You have to transfer it to a PC and reformat it for your valuation to a spreadsheet. Next, you run some macros to create your management memo and your general ledger entries. You send out a few pieces of paper to the planning area, and they go through their actual-to-expected analysis. Then they send some pieces of paper to the financial area so they can do their general ledger entries. We map out the processes that happen on a month-to-month basis, and look for areas where we could increase the level of automation.

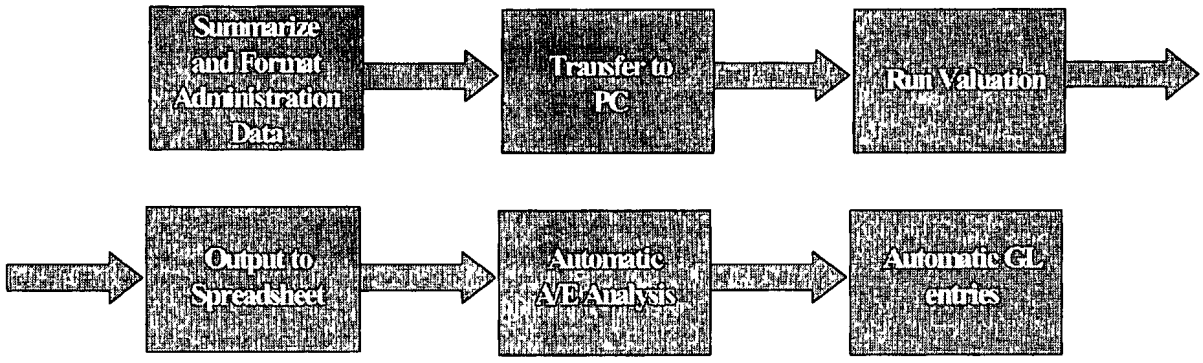
CHART 3
Opportunities—Monthly Reserve Calculations



What we came up with was a revised process where we first provided data in the correct format to the valuation system (Chart 4). We find that people don't systematically look for opportunities to cut down the amount of data transfer when they have to manually intervene with the system. After

the data are transferred to the PC, the valuation is run. We integrate the reserve output into an actual-to-expected analysis and to the general ledger automatically. We automatically reformat the data from the reserve system to be passed into another system and into another area of the company. For example, if we have to send the actual results over to the planning area, we automatically integrate the actual results in with the expected results.

CHART 4
Revised Monthly Reserve Calculations

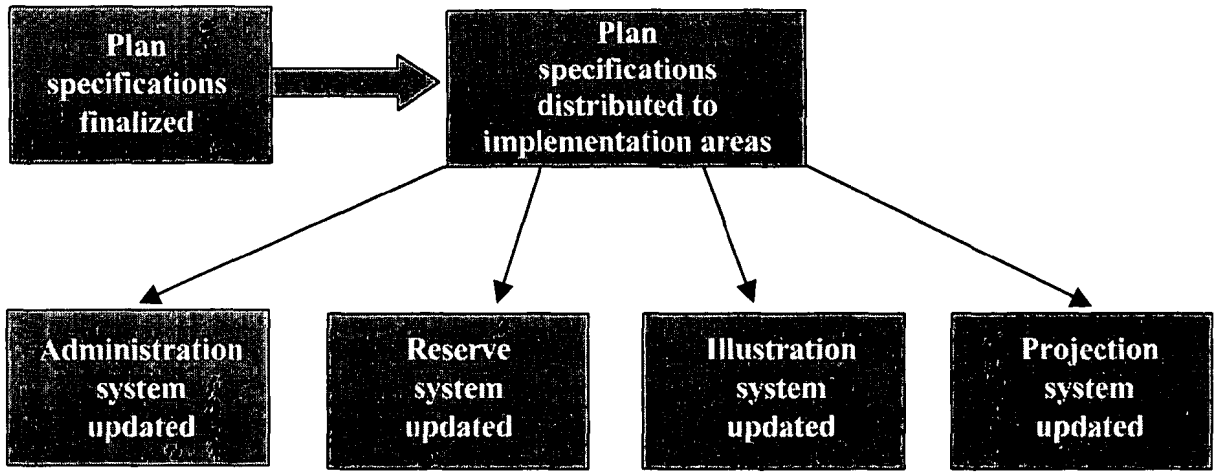


The second idea, which I haven't seen done yet, is to take the reserve output and automatically update the general ledger. Looking at this monthly valuation process, we can gain efficiency by integrating the outputs from one system with inputs to another system. Roger will talk more about that in this presentation.

The second example of data integration potential involves systems that share the same data (Chart 5). For example, when brining a new product on board, we finalize the plan specifications and distribute them. What we often see is a product book that describes what the product does. Everything is defined, down to the ones and zeros in the administration system turning different features on and off. This product book gets distributed to all the implementation areas. The administration system gets updated with the plan specifications that it needs to run. The reserve system gets updated by someone else with the same data in a different format to go into that system. The illustration system

needs another format to run their product illustrations, and finally the projection system gets updated. Sometimes there's some overlap with the pricing and projection system to make that a little bit easier.

CHART 5
Opportunities—Product Implementation



What if we could have data in the format to create the plan specifications and assumptions, and be able to share those between these systems (Chart 6). At this point, such sharing isn't done very much at all. Many of these systems are vendor systems, and we haven't seen a lot of push to have the vendors dedicate resources to put such a process together to update systems automatically. Brian's going to talk about what has been done on the sales side to try to make some of that happen with other types of data.

CHART 6
Reduce Implementation Time

How do we solve this data integration problem? One idea would be just to have one big system. I don't think this is a feasible solution. Another way to try to integrate data is to create a sharing structure within each company. The company has picked different vendor software to work with, and they have their own in-house software that works well for some things. A company could dedicate some resources to do data sharing, and actually have somebody go to the board and map out where they can transfer data efficiently to improve what they're doing. When we're looking within the company, where do we have our biggest holes? Where is the most time spent transferring data back and forth? Then we can find where we need to dedicate time to cleaning up that process. A data/work flow chart can open your eyes to some opportunities. One concern is that of working with vendor systems. You rely on the vendor to keep either their data input or their data output in the same format. You may have some difficulties when the vendor comes up with an upgrade to their system and hasn't considered the work-arounds that you've put in place for data sharing.

Another potential solution is data standards. This is already working in a lot of industries. I realize we have a rather complex data structure that we work with. We have a real challenge compared to some of those other industries. But there is potential for some types of data to be transferred in a standard format between different types of systems. Vendors and companies could work together to create an industry-wide effort.

MR. ROGER B. SMITH: I want to talk about something that's frequently called a "data warehouse" or a "financial data warehouse."

In the work that I've done on this topic, I have found that data warehouses are everything and nothing all at the same time. It very much depends on who you talk to and what the needs are. Frequently, different people around the company desperately want it, can talk with great enthusiasm and passion about how great it's going to be when it's all ready, but cannot describe adequately what it's going to be.

I define a data warehouse to be a back-end set of financial reporting and analytical tools that can analyze projected data and actual data. This would enable responsible financial managers to make

decisions affecting the performance of the company. We want to have a place where the data have to reside together. Today we find that actual information and projected information scattered in different data systems. Whomever is doing projections or financial modeling has some data on their computer. The valuation information is somewhere else. Actual reported information could be in a different location. Data integration means bringing the data together in one place so it can be used. I think of a data warehouse in terms of a set of reports and reporting capability that will deliver information that will help you make decisions. The ability to support decisions that lead to action is important because building the data warehouse is a very big, expensive effort. If it doesn't help you do anything, someone might question your wisdom in assembling it.

Who uses it? I can construct an example. Consider yourself in a global company. There are many financial managers that have to make decisions about the performance of the company. They might be anywhere. Users could be product managers halfway around the world or halfway around the country. All these people need to see the information and need to get some reports. There are two ways of doing it. The financial warehouse approach would allow those responsible managers to go to their terminals, request some reports, and view the data in their offices. The other techniques and the way it happens right now, is that they run down someone in the actuarial department to make a special report. You either do the report or think of a good excuse why you shouldn't have to do it.

Everybody should be working with the same data, particularly if it's project data. How many people would like to go into a key strategic meeting with the marketing folks when they've run their own models about profitability levels, and they've been totally free to pick all their own assumptions? You can see where that would degenerate into a worthless exercise.

No matter whose system you're using, how many people would characterize your projection models as so simple that anyone can run projections? Projecting today is very complex and very difficult. It is very difficult to teach, train, and empower other financial managers in how to run the models. Who's running cash-flow testing models or projection models today? Could your boss or your boss's boss run those same models? Would you turn those over? It's clearly neither desirable nor possible to have everybody running their own models.

Why use a data warehouse? There are many questions that need answers. If you're not in the position to answer them or enable the people raising the questions to answer them, then some questions will go unanswered. There just isn't time to go to find the people who have the answers. We need every competitive advantage we can find, and reliable, dependable data are needed to answer the questions.

What is special about the financial data warehouse? Insurance companies have only made limited use of such a tool. Other industries, e.g., the credit card business, are using these big warehouses with some success. The people that come from those industries know it can be done. The difficulty is that many of our insurance transactions and relationships are more complicated than your standard credit card relationship, so insurance applications are more difficult. Placing data into a warehouse will require more discipline. It will be more than actuaries are accustomed to in processes and assumption setting. It's going to have to become part of our lives.

What makes up a data warehouse? Obviously, financial data is going to reside in the data warehouse. A critical ingredient is a well-defined, robust reporting capability to answer questions from that data warehouse. The other aspects that are critical are the procedures for feeding, updating, and deleting data from time to time. It becomes a part of the data warehouse.

What kind of information would you see in the financial data warehouse? The work that we're doing involves both projected and actual reported information. You need several layers of projections. Almost everyone would likely have optimistic and pessimistic levels of projections that would be valuable for answering questions. For example, let's compare actuals to the most recent set of pessimistic to get a feel for how well we're performing relative to assumptions. So we'll have several versions of data present.

Perhaps we first projected our 1999 experience some time in 1996. We will continue to update those projections from time to time. Would anybody in your companies find value in comparing the last projection to the more updated one? Clearly, this is what we all do—compare the last report to the present one.

A large part will be data control structure. Since we'll have actual and different versions of projected results, we'll need some control structure so that the system can distinguish one set of data from another.

Some of the earlier efforts included the issue of data granularity. How much detail do you want to get into? Let's consider two extremes for granularity. By this I mean, at what levels can we receive subtotals? One level would be just grand totals for the whole operation. That would be a very small quantity of data. It wouldn't take much effort to implement, although it would not be extremely useful.

I've seen some companies that have started down the path that I would call policy level, where they want to keep all of the actual information and all versions of projected financial data at a policy level basis. Obviously, for some purposes, you need to have policy level information, but there's a need for the financial data warehouse operation to allow spontaneity. What you will find quickly is the time, effort, or computer power required to go through millions and millions of records really doesn't lend itself to the concept of spontaneity. If it takes three or four hours to get answers back, you'll stop asking questions. Somewhere between these extremes, you will find an optimal summary level and design the financial warehouse accordingly. Is there any right answer for everybody? Clearly not. Of course, there is no right answer. I'm not sure if anybody is using this.

What kinds of reports should the data warehouse prepare? Income statements and balance sheets are the kinds of things people will want to see. I think you'll need to design a set of analytical comparative reports in which you could compare the results from two different periods, or actuals-to-expecteds, or one estimate to another, or whatever might come up. You'll need some reports to show some "what-ifs," so you want to allow some what-if capability. This will be difficult because the more details and flexibility you add, the more difficult and time consuming it will be to teach someone how to use the warehouse. The reporting has got to be fairly easy to teach and learn. This is important because it will be used by many financial managers scattered around your company at one site, multiple sites, or around the world. It needs to be easy to learn how to run the reports.

We need to develop procedures to feed data. Consider a new kind of actuary—"the warehouse actuary." The procedures to feed the data will demand more discipline and documentation than what many of us are accustomed to. I'm envisioning something that many users will be able to use. Let's imagine we're defining optimistic and pessimistic levels of projections. The users of the information will need to understand the meanings of optimistic and pessimistic. Again, whenever assumptions change for whatever reason, there needs to be some documentation and some distribution of the reasons and the effects of the changes. I think everyone will have to develop some kind of a schedule for supplying projected data. If marketing managers or product managers are looking at updating reports they might be running, it will be important to know when new data might become available. Is it every month, is it every six months, or once a year? It needs to be documented and well publicized around the company.

How do we start building the tool? One approach might be to start with a set of reports, and then to work backwards. Once you have the reports and the capability that you want to generate, they will then tell you the kinds of controls, data, and granularity that you will build into this database. Again, start at the very back-end with these kinds of reports. Work backwards into the warehouse system, determine what the sources of those data are and when those data will be supplied.

I want to go through a couple of reports to stimulate our thinking. These reports were mainly designed so that I could show them on a table as opposed to some incredibly insightful report. I'm looking at a simple block of term business, with two years of projected information (Table 1). We see year one and two, data for block one and block two, and totals. Some of the data items that you will want to put into your data warehouse are: premium, investment income, expenses, and claims. In practice, you'll need more information, but this example illustrates the value of the reports.

TABLE 1
Report 1—Two Years of Projected Results

Block	1	1	2	2	Total	Total
Year	1	2	1	2	1	2
Premium	205,905	195,284	205,905	195,859	411,809	391,142
Investment Income	(9,800)	20,795	(9,800)	20,862	(19,600)	41,657
Total Revenue	196,105	216,079	196,105	216,721	392,209	432,800
Expenses	345,905	24,004	345,905	24,059	691,809	48,063
Claims	38,840	38,545	38,840	28,513	77,679	67,059
Income Reserve	0	125,795	0	126,231	0	252,026
Income	(188,640)	27,735	(188,640)	37,918	(377,279)	65,652

I've added some actual information next to the second year projected in this next report (Table 2). Let's imagine what we might want to do with this. Let's look at actual to expected. Note the differences. It's something that you're probably doing right now in your actual reporting.

TABLE 2
Report 2—One Year of Actual Plus One Year of Projected

Block	1	1	2	2	Total	Total
Year	1	2	1	2	1	2
Premium	205,905	195,284	205,905	195,859	411,809	391,142
Investment Income	(9,800)	20,795	(9,800)	20,862	(19,600)	41,657
Total Revenue	196,105	216,079	196,105	216,721	392,209	432,800
Expenses	345,905	24,004	345,905	24,059	691,809	48,063
Claims	45,500	38,545	42,000	28,513	87,500	67,059
Income Reserve	0	125,795	0	126,231	0	252,026
Income	(195,300)	27,735	(191,800)	37,918	(387,100)	65,652

I've introduced some differences in the claim levels for the first year (Table 3). There are also implicit differences, particularly in the lapse activity.

TABLE 3
Report 3—Comparison of One Year's Projected to That Year's Actual

Block Year	Total 1	Total 2	Difference
Premium	411,809	411,809	0
Investment Income	(19,600)	(19,600)	0
Total Revenue	392,209	392,209	0
Expenses	691,809	691,809	0
Claims	77,679	87,500	9,821
Income Reserve	0	0	0
Income	(377,279)	(387,100)	(9,821)

Now I have two years of actual information (Table 4). The second year now has changed for the two blocks of business, and many of the numbers have shifted.

TABLE 4
Report 4—Two Years of Actual Results

Block Year	1 1	1 2	2 1	2 2	Total 1	Total 2
Premium	205,905	184,921	205,905	185,465	411,809	370,386
Investment Income	(9,800)	19,692	(9,800)	19,755	(19,600)	39,447
Total Revenue	196,105	204,613	196,105	205,220	392,209	409,833
Expenses	345,905	22,730	345,905	22,783	691,809	45,513
Claims	45,500	36,500	42,000	27,000	87,500	63,500
Income Reserve	0	119,119	0	119,532	0	238,652
Income	(195,300)	26,263	(191,800)	35,906	(387,100)	62,168

This additional report might be helpful in that I've explained the difference between rates and volumes (Table 5). In the second year, I have some large differences.

If we looked at years one and two, original and projected, and one and two actual combined, we have some differences here. It is helpful to split differences because there's a different amount of business in-force, in the rate of premium, or in the rate of death.

TABLE 5
Report 5—Comparison of Two Year's Projected to Actual

Block Year	Projected 1+2	Actual 1+2	Difference	Rate Difference	Volume Difference
Premium	802,952	782,195	(20,756)	0	(20,756)
Investment Income	22,057	19,847	(2,211)	0	(2,211)
Total Revenue	825,009	802,042	(22,967)	0	(22,967)
Expenses	739,872	737,322	(2,551)	0	(2,551)
Claims	144,738	151,000	6,262	10,985	(4,723)
Income Reserve	252,026	238,652	(13,374)	0	(13,374)
Income	(311,627)	(324,932)	(13,305)	(10,985)	(2,320)

In the examples, we see some of the things that are demanded in a control mechanism. Imagine someone is going to sit down and specify a report. In these simple examples, we've seen that we'll need to specify:

- The block or blocks we want to report on
- Time periods
- Versions of data (actual, levels of projected)

Some of the reports involve comparison. If we're comparing two different versions, I need both at the same time, and I need to know which one is which.

What are some of the data items that these reports showed? These had premiums, investment income, expenses, and claims and reserves. Would you break these down at all? Of course, you'd break them down into many components. As an example, what might be necessary for premiums?

Do you customarily report premiums for the first year and renewal year? If you're doing that, then you also need all your projections and actuals broken down this way. What about line of business? Do you want universal life, ordinary, annuity, or health premium separated? These have design impacts that would go into the data warehouse. For UL, target versus excess premium is very important for some companies. These are all decision points on how we might want to stratify things.

I have talked about what the data warehouse is. Here's some indication of what it might not be. One of the things I've heard people say is if you keep data at a policy level, you can produce, on demand, mortality and lapse studies. However, this tends not to work out quite as well from the time standpoint needed. If you build this data warehouse, can you throw away your protection systems and your mortality measurement systems? No, you can't. You're still going to need those to drill down and study things. Here are some data items that you might want to carry by line of business, with the thought of mortality studies: smoker versus nonsmoker, expected deaths and actual deaths, and all on projected and actual bases.

This would not substitute for a full-scale mortality study that produces all the classical actuarial analyses, but it would give you some early, quick indication of differences in margins by smoker and nonsmoker—if that's relevant for you. Male/female is perhaps another split you might want to employ in some of the underlying data.

Cheryl asked me to talk about some of the time lines and degree of difficulty associated with implementation. The first step is the report capability definition. From a degree of difficulty, I would rate this as quite difficult, and maybe impossible. What makes it difficult is that ideas come up at the last minute. Actuaries are more creative and thoughtful when reviewing final specifications than when reviewing drafts. As you're working through a schedule to try to organize the data warehouse, sometimes you'll get ideas and thoughts that just complicate things at the very last minute. Or, you have the shotgun (random) approach to submitting requirements. It can be very difficult getting a good description of what reporting capability you want, and what it should be. I

would estimate that it takes somewhere between 6 and 12 months to get some of this done, including the reviews and exchanging notes and things.

Once the reporting capabilities are defined, it becomes more mechanical. The next step is defining the data needed. For this step, you need people that are very, very careful and diligent. The prior step required somebody to see the whole forest, to get that view from 30,000 feet, whereas this step definitely requires the tree watchers. It requires different sets of skills, and not everybody can do both tasks. You will find some special data feeding capabilities. One example I can think of is in setting actual amounts of incurred but not reported (IBNR) claim reserves. The IBNR is a function that very many actuaries would not want totally automated, where it would just show up in your general ledger. There is a need for experience and judgment when setting those particular values. Sometimes, even after you've seen the initial income statements, you need to apply that kind of judgment. On the other hand, collected premium can be totally automated and just flow right through. Whatever the data item is, I do think the warehouse actuary concept is important.

I might dispute something that Cheryl said about automatic feeds to general ledgers of reserves. I always feel more comfortable if someone can just look at them and bless them so there's a human control element in that process. My company creates valuation systems, so if I'm not comfortable having all those results just following through automatically, I don't think anybody should feel comfortable.

I estimate the time for data definition at one to three months. The next step is programming the report capability. This should be mostly mechanical. You'll need your structural query language (SQL) experts to write those programs, I plan on one to two months for each set of specifications, whether they be initial or revised specifications. Sometimes project planners arrive at this stage and realize that their design is not adequate for everything that they want to do. You have the potential for some throwaway work at this point. The degree of granularity may increase run times because the quantity of data is too large.

Data quality and control aspects are quite important in the process. Go back to procedures for feeding data and document those well so everyone understands what they are. It will eliminate surprises and perhaps some contention along the way.

MR. BRIAN COLE: I work for ACORD on the OLifE technical standard. I'm a technical architect. I've been with ACORD for a little over a year now dealing specifically with the OLifE standard.

OLifE originally stood for "OLE for Life Insurance." That name is a little limiting being that OLifE does not deal with just life insurance, and it does not deal with just OLE. OLE is a language that Microsoft created. Initially the specification was built by Microsoft, and ACORD took it over a couple of years ago. Since those concepts or ideals were limiting, we now think of OLifE as being "Objects for the Lifetime Management of the Customer." OLifE is a technical standard, and a standard is a means of sharing data across applications. In my presentation, I am going to cover two aspects of it: what it is now, and what it will be or is projected to be in the future.

Currently, OLifE is an integration of applications that reside on an agent's desktop in the sales force situation. You currently have a number of applications that exist out there on a given agent's desktop. You have contact management, financial planning packages, product illustrations, electronic applications, and data downloads. Each one of these applications has its own private data store where it stores data required by that application. What you end up having is redundant data existing on an agent's desktop. All of the same information on a client exists in multiple places. If you make any change to a particular individual's information, you have to go to each package separately and make that change. Data redundancy and data integrity problems exist. OLifE pulls all these data together into one common data store and exposes those data to all the applications existing on the desktop. It eliminates the redundancy of data, and it helps to ensure data integrity.

The data that is pulled together or shared is not all the data that exists, just the inner section of the data common across the applications. The specifications not only define the means of sharing that data, but what those data are. It is the common data in the overlap between the applications that exist

on the desktop. The applications still have unique data that they need, and those unique data can still be stored privately in private data stores. OLifE also provides a mechanism for enabling those data to be shared. Components in the OLifE specification enable an individual application to share its private data as well as the common data.

Another capability of the OLifE component is that it's the real time sharing of data that exists on the applications on the desktop. For example, you might have a given person's record, John Smith, and it's being viewed in two different applications simultaneously. In your contact management application, you have his record being viewed on your desktop. You're running a financial analysis on John Smith using a separate application that also exists on your desktop. Again you're seeing that same information in two separate applications. When you're working in one application, the contact management application, and you make a change to some of the data (a change in address), real time notification will be sent to the other application telling it that data have changed and it is no longer current. This happens automatically without the second application having to do anything. It will automatically be notified that data have changed, and then it can act upon it to reflect that current data.

Another aspect of the OLifE specification is business rules. Business rules are external to any one particular application. They're pulled out, and there's a mechanism in place to have business rules to be adhered to outside of any one individual application. For example, let's say you're in your contact management and you indicate that a person got married. You can have a business rule that says that if a person gets married, a sales opportunity exists. You can provide an automatic script to your agent that he would follow to take advantage of that situation.

Another example of a business rule that can be applied in this scenario is for document retention. John Smith contacts you and he says, "I hate you, don't call me anymore; I never want to speak to you again." You can delete John Smith from your database. A business rule then is put into place to delete John Smith. You then have to check to see if any application depends on information that is being stored on John Smith. I may have a sales illustration that I ran on him some time ago. Because of compliance reasons, I cannot delete John Smith from my application; I must archive his records and be able to recover that information.

A third example is compliance. John Smith receives a large inheritance, providing another sales opportunity. Business rules are designed to signal certain products to take advantage of this large inheritance. However, if you're not licensed to sell those products, a warning is issued that the proper licensing is not in place. The real message here is that the business rules are outside of any one particular application. They're a separate component that can be fired off independently of action taking place within any individual application residing on the desktop.

These examples represent applications in existence today. The specification is out there right now, has been tested, and is being utilized by a number of companies out in the field.

Moving beyond the desktop is the next phase with specification. This involves looking into areas such as distributed computing support at an agency level, where agents are plugging them back in. There is currently some support for this, but we're moving into better support for a true client-server implementation. With respect to legacy system support, some of the current applications are data downloads. We are also developing a standard format to apply for electronic submission of policies (application submission).

Internet uses. There are applications today that utilize an OLifE server on the Internet. Companies don't want to have different illustration systems used by agents as opposed to someone in the home office. Some companies have an illustration application on the Web site. The agent dials into that Web site to run the illustration, and the OLifE object enables data to be shared across a Web site and his own PC. This way, the same illustration engine is used in the home office, in the servicing office, and by the agent. Results are the same regardless of who's running it.

Service center support can use the same concept. Support is provided not only at the agent level but at the service center, and that data can be shared across the two. If someone calls into a service and makes a change, the agent should be aware of it and vice versa. How do you get the two to talk to one another or share that common data? That's what the specification is addressing.

All along, the specification has abided by this concept: don't reinvent the wheel. We've always tried to take advantage of existing technologies and build-upon them. Microsoft OLE, Com, D-Com, and Com-Plus are all existing technologies that the specification is built upon. It uses those tools that exist; it doesn't try to build new technology. I don't know how technical the audience is, but it's not limited to using OLE. You can also use Java and Common Object Request Broker Architecture (CORBA).

The specification also takes into account other existing standards. ANSI, NALBA, CLETIS, X-12, and IIA exist today, and the specification always makes sure that we're not crossing over boundaries or stepping on anyone else's toes. The OLifE standard is also broadened internationally. It's not only just used here in the U.S. Canada was involved right from the start, and we just completed a world tour this past summer where we've hit a number of countries, and a number of countries have, in turn, adopted the standard (e.g., Australia, South Africa, New Zealand, Japan). Australia is very active in using the standard.

So what is ACORD? ACORD stands for Agency Company Organization for Research and Development. We're a nonprofit organization that deals with insurance companies, agents and brokers, vendors, and individual agents. We initiate and manage industry-wide standards. We started on the property and casualty (P&C) side with report forms. If anyone has auto insurance, chances are you filled out an ACORD form, which was used for the auto insurance card that you carry. ACORD has more than 27 years of experience in developing insurance industry standards, and now supports over 1,000 member companies and 32,000 agents. The OLifE standard is funded by the insurance companies. We're self-funded, meaning the money that you pay to utilize and participate in the standard is used to make the standard run; it just covers the cost to manage the standard.

Insurance companies that participate in the standard set the direction for the standard. They tell us where they want the standard to go, and what data they want to see in the standard. We take that information and help make it happen. That is where the vendor community comes into play. We

go out and interact with the vendor community. We say, "This is what the insurance companies want. How are you going to provide this for them?" We have user group meetings and conduct annual subcommittee meetings where everyone meets, votes, and progresses the standard forward.

We provide participating members with testing tools, training, documentation, and sample applications and consulting.

Finally, we provide a certification process. When OLifE first came out a number of years ago, it was a buzzword. Vendors were claiming they were OLifE compliant, but it didn't mean anything because there was no way of substantiating whether they were or weren't compliant. Insurance companies then asked us to put a certification process in place to test applications to determine whether or not they're truly adhering to the specification. We can put a stamp on it that says, yes, they are OLifE compliant or they're not OLifE compliant. This way, insurance companies can make a better evaluation of products before they purchase them.

OLifE first came into being from an initiative that happened before 1994. It was called SLEEK, and I can't even remember what SLEEK stood for, but it was a partnership between Andersen Consulting, Microsoft, and a number of vendors. Initially it targeted the whole enterprise solution, and it had these great big ideas that would cost a huge amount of money and take two years to develop. Insurance companies were not interested, so it basically fell apart. Microsoft took it over exclusively, somewhere around 1995. They changed the name to OLifE, and went out and talked to all the insurance companies to get feedback. The insurance companies rejected it. Microsoft asked them what would they rather see, and what could be done differently to get this initiative going? Microsoft believed it was a worthwhile initiative. The insurance companies said that the focus was too broad, the projection time too long, and the money was too high.

Instead of focusing on the big picture and having this big enterprise solution, Microsoft targeted one aspect that they deemed as being most critical: the common data that existed in the applications that

their sales force agents were using. Client application data was most commonly used. Regardless of the application, and who's running it, you always have core data such as name, address, and phone number, so that's what they targeted first. They decided to start small and build gradually.

The core set of data was delivered in the 1.0 specification. Version 1.0 was delivered in 1996 by Microsoft. In fact, I worked for the company that delivered the first solution based on that specification (Prudential). We integrated three different applications: a contact management application, a sales illustration application, and a financial planning application. All targeted that core set of data; name, address, and phone number. It enabled those applications to share that data seamlessly. So regardless of what application you were running, that data was drawn from and written to a single data source. The applications also had real time data exchange. If you changed the data in one application, it was immediately reflected in another application. It was well received and successful in its deliverance.

After it was released and proven, Microsoft decided they were not in the business of maintaining standards. They develop new standards, but are not looking to maintain them. So ACORD, which does manage standards, took it over.

We're now into Version 2.0, which expanded the scope of what 1.0 delivered. In 1.0, there was no policy information; it was just pure client information. The 2.0 version of the specification expanded into policy information. Included are all types of insurance policies; life policies, annuities, mutual funds, health, and disability policies. Moving beyond that, we're looking for a complete enterprise integration. We're looking to expand beyond the desktop into the back office so that we may facilitate an enterprise solution.

There are about 50 participating vendors currently utilizing the OLifE standard in their applications that they are selling. For the very latest list, you can always go out to our Web site, www.acord.org. Participating OLifE insurance companies include U.S. companies, Canadian companies, Australian companies, and a South African company. The specifications can be downloaded free from the Web site. In addition, there's information on all participating members, vendors, and companies.

MR. STEPHEN L. WHITE: A question for Roger about your data warehouse examples in which you were combining actual and projected experience. In your concept of how that was working, was the data warehouse itself making any projections, or were all those projections feeding in as data items just like any other type of data?

MR. SMITH: In my vision of how that would work, projected information would still come from a present tool. I believe that there are some limited abilities for the data warehouse to calculate some information, but projected information would come from whatever projection model you're currently using—not an additional system.

MS. MARY ANN BROESCH: I thought both of the topics were very interesting and really relevant right now with the problem of trying to gather data. Brian, would the standard that you describe be able to be used to support the types of reports and the data that Roger talked about? How could you integrate the two concepts of being able to take data and put it out into reports in an automatic way? Can it go beyond just the type of data like name, address and so forth?

MR. COLE: Yes. Actually the two pieces actually work well together. What Roger is talking about is the physical place in which the data are stored. The OLifE specification deals with how those data are shared, not where they are stored. Once the data are stored, we provide a means of how to get to those data, and how to make those data accessible to any application that exists in a standard way. This way you can build these applications to be plug and play. As long as they adhere to the specification, regardless of how the data are stored, applications can get to it in a standard fashion. If you have a data warehousing implementation, applications can be built to get to that data without having to know that it is actually being stored in a data warehouse. What Roger is dealing with is the physical storage of data and how they get stored. There would have to be work done so far as agreeing to what that data are.

MR. SMITH: Just to add a couple of thoughts. In defining what the data structures will look like, I envision a number of data supply functions that will translate information, combining various things. In the data warehouse, I think you will have far fewer data items than I would expect to see

in a projection system. You'll be combining some data items from your projection system into your data warehouse. You will also have a need to include new business projections. It could be that for a new, innovative product design, you have a completely different projection tool. Maybe you had to write your own program because the product structure is so different than what you're pricing. You will still want to move that information into the data warehouse. The product features might be completely different and not supported by commercial projection systems, but premiums, reserves and benefits are still the same across those lines. You'll have a translation process. Defining the data you want in the data warehouse will be a very critical step, and it should be as clear and as clean as possible.

MR. JOSEPH LEONARD TUPPER, III: I'm a consulting actuary. I am concerned about any new approach. I think that data warehousing offers a lot of promise, but after five or ten years, it is the legacy system. Going forward, we have the problem that things may change. If we had data warehousing 20 years ago, for example, a lot of companies would not yet have distinguished perhaps between smoker and nonsmoker. You can pick any kind of change that affects the structure of your company over time. Lines of business change over time, and sometimes products will move around from one to another. Sometimes you'll decide to start universal life or you'll bring in variable annuities. You may subdivide a line of business as you realize you have something that ought to be two separate profit centers. In a data warehouse, if you're accumulating this data over time, which is ideally what it would be useful for, how do you keep track of these changes? How can you plan for future changes, in a sense, without knowing what they may be?

MR. SMITH: The physical engine will change as well. I would envision one of the big commercial database products (think of Oracle or Sybase) or similar products as being the repository for that. You mentioned legacy systems, and as an industry, we have a lot of legacy systems around. Some of the difficulties exist because there are a lot of very unique, creative data storage techniques designed to take advantage of the computing technology at the time. Many of the systems still being used were designed to take advantage of the computing technology at the time. Many of the systems still being used were designed with a mid-1970s computer technology. It is important to build all

of your tools in a public standard kind of way. Use something that involves common applications and common data format. In this way, it's much more likely that you'll be able to carry that forward and make adjustments to it.

MR. CRAIG W. REYNOLDS: This is a question for Cheryl. I noticed that Tillinghast was a signee on the OLifE standard. Can you just comment on what that means for some of the Tillinghast systems, in particular, TAS? To what extent has it or will it comply with the standard? What will that mean in terms of other applications being able to access TAS data and make use of it?

MS. KRUEGER: That's a work in progress. We are a member of OLifE at this point; we don't have any OLifE compliant systems.

MR. JAMES A. KAISER: I noticed that you had mentioned CORBA, which is, I believe, being formed by another industry standards setting group, Object Management Group. To what extent are your objects going to be compatible with those objects that might facilitate extending the objects into the rest of the enterprise? Will it be compatible as it crosses different industries and as insurance starts dealing more with mutual funds, banks, and other organizations?

MR. COLE: We are a member of CORBA, and we participate in that effort. In doing that, we're ensuring that there is no conflict between the specifications. It has already been recognized that the banking and insurance industries are moving closer together. It fact, we have a couple of financial institutions that are already members and are using the OLifE standard because it's not limited to life insurance. Mutual funds are modeled already, and the bulk of the data are actually client data. Clients are clients whether you're doing financial analysis for them or selling them insurance. So, yes, we participate in the CORBA initiative. We're making sure that there are no conflicts, and ideally there would be a merger or a build with one standard and the other standard. As mentioned, we always make sure that we take into account existing standards in our development work.

There is also an initiative right now called J-Life, which is a Java implementation of the life standard. It's not limited to just OLE and Java. You can have a CORBA and Java work together so you can have a Java implementation of OLifE.

MS. SUSAN T. DEAKINS: I have a question for Roger. You showed an example giving financial information to a variety of people. We've thought about doing that for a while, but you're giving information to different people with different levels of expertise. If you're giving them actuarial information in a more simple form, how do you make sure they don't reach the wrong conclusions by playing with some of that stuff themselves?

MR. SMITH: There's probably no way to completely prevent that. As I mentioned, the warehouse actuary would supply known data calculated using the published assumptions. In doing this, you would have comfort that the data are calculated correctly and accurately. Then we get into the kinds of reports that are available. Let's go back to my days at an insurance company. Somebody in our parent company was tracking the ratio of incurred GAAP commissions to death claims, and one month, wanted to know why that bounced out of line. I thought, why in the world are you doing that? I would say that correct calculations would reflect the reporting capabilities. You define reports that lead them towards the correct conclusion.

FROM THE FLOOR: So a lot of the work would be in educating them?

MR. SMITH: You need to document those reports so that they'll know, for example, to answer this kind of question using report No. 7.

MR. VINCENT J. GRANIERI: In my experience, one of the laws of the universe is that if you keep the same data in two different places in an organization, they'll never balance. Given that, what does that say for the data controls and the data integrity issues that arise? If a data warehouse were in place, what kinds of things would it be able to replace? We said it wouldn't replace projection system. One of the things that might make sense is a greatly streamlined general ledger. If this data

warehouse really works, we could report off of the data warehouse. Therefore, there would not be the need for as detailed a general ledger. We could let the ledger do what it can do, and data warehouse do what it can do, and not have redundancy of systems all with similar capabilities. This, of course, leads people who are in different disciplines towards the solution that they're most comfortable with. Could you comment on those issues as they've arisen in your thinking?

MR. SMITH: One of the key components here is that one piece of data is only in one physical place. You can have a logical data warehouse that's physically separated on different machines and even in different locations. However, any given piece of data is only in one location, and the warehouse knows how to get it. In terms of the functions that could be replaced, I would say that it's not so much replacing functions, but automating functions. These are financial analytical steps that people do now, but they are downloading things into spreadsheets, doing special comparisons, going through various reports, and going around and asking people questions who may or may not know the answers. This financial analysis is taking place, although it's a very manual process. It's very likely several people are doing the same things and answering the same questions, or trying to. I see it more as an automation of functions that are already happening. It might be some expansion of some capabilities because not all people have this information available. I agree with you in terms of general ledgers. I've seen people try to get this functionality in a general ledger but the results are usually very poor.