# RECORD, Volume 27, No. 1*

Dallas Spring Meeting
May 30–June 1, 2001

## Session 64 IF
## Data Quality

Track: Computer Science

Moderator: FRANK E. KNORR
Panelists: CRAIG J. BLUMENFELD

*Summary: The quality of data is paramount to the success of today's data-driven enterprises. The insurance industry is no exception. Insurers use a variety of resources and techniques to validate data. Further, insurers are faced with short development cycles and partial product specifications. Speakers discuss how to develop alternative or improved processing methodologies to reduce errors, current error tracking tolls, and metrics. Participants discuss balancing reasonable testing given time constraints.*

Mr. FRANK E. KNORR: I'm with American United Life Reinsurance Management Services, the long-term-care division. My work encompasses collecting data from over 40 different ceding companies. Occasionally, we get asked to bid on a block of long-term-care business for potential acquisitions, so I deal with quite a bit of data in that regard. I've also been involved with Computer Science Section since its inception.

Craig Blumenfeld is our other presenter. He's with Arkidata Corporation in Downers Grove, IL, which specializes in information integration software. Craig has leveraged his nine years of actuarial experience into a non-traditional role, as an information engineer at Arkidata. He is using his actuarial expertise to ensure data quality for benefit-related systems.

First, I'll give a little bit of background on the topic of data quality and how it relates to our work. Then Craig will talk about integrating data from different sources. I will go over a few examples before we open it up to the floor to discuss the items that you feel are important to data quality.

I guess part of the background is just thinking of us in our roles as actuaries. We sometimes say things like "DAC is recoverable," or "the premium rate for a 35-year-old is $53.00." Statements like this should be treated as more than just casual conversations. We're speaking as actuaries. The idea is that people rely on



**Note:** The chart(s) referred to in the text can be found at the end of the manuscript.

statements that actuaries make. Those people have to understand, first of all, what we're saying, and we have to understand what those people are going to be using the data for.

These people trust that the things that we say are correct, that is, that they're supported by data and also that we follow certain actuarial principles and standards. The Actuarial Standard of Practice (ASOP) 23 is the one relating to data quality.

Just very briefly, ASOP 23 says that we need to check the data that supports our statement, even if someone else checks the accuracy and completeness of the data. A lot of times we rely on other people for that—other departments even. But even if that reliance is done, we still have to check for reasonableness and consistency. The ASOP uses words like relevant, appropriate, current, independently verified, accurate, complete, and material when they're describing the data.

What is data? I like to think of data in two forms. One is numbers that can be added up, and these are things like premiums, premium rates, claim dollars, and reserves. The other is categories. That is, how do we arrange the numbers? Categories are things like age, sex, plans, state, policy number, things like that. It seems like every actuarial study has to have things by age and sex. Mortality tables, age and sex morbidity tables—those categories are critical for our field. Some data items can fall into both numbers and categories, for example, a face amount of an insurance policy. Sometimes we want to add up all the face amounts to get the total volume. Other times we want to take a look at premiums or experience in face amount categories.

ASOP 23 states that assumptions are not data. I feel this can be misleading, because once those assumptions are keyed into a data file, they become data. I think that the intent of ASOP is that it doesn't encompass checking the reasonableness of assumptions. Checking reasonableness of those assumptions is covered by other standards of practice. But once those are checked and they're keyed into a data file, then it becomes data and it should be covered by ASOP 23.

MR. CRAIG J. BLUMENFELD: Before I do my presentation I want to point out that the intro to this Data Quality Session talks about "from the insurance perspective." Certainly what I'm talking about applies to insurance, and I'll try to cross-reference as much as possible. I got my start in data quality more from a pension administration database. A lot of what I'm talking about is from that angle, but as I said, I will try to relate it to the insurance industry, because it definitely applies to everything from pensions, healthcare, and insurance. I do work for Arkidata Corporation, and I think an interesting story is how the company got started. I thought I'd share that with you, because it'll really give you an idea of what it is that we do and what I mean by data quality. I think a lot of what I've heard (data quality, people checking, etc.) certainly what Frank is mentioning is very important. But making sure that the data that's going into your assumptions is accurate is of the utmost importance.

We take a look at it from a different angle at the company. First of all, why is it called Arkidata? The president of the company, his name is Arcady Maydanchik. He

got his ASA in two sittings. He's one of those child prodigies that we all hate. But at any rate, he was hired by the Federal Reserve to investigate money laundering. And in his process of trying to see whether or not there are any money laundering processes going on, he found that the data quality was horrible, and he was trying to analyze trends. What he found was not that there was any money laundering, but actually that there were errors in the databases that were being used to investigate it. This is how Arkidata was born. So, what we do is analyze data for trends, and upon doing this analysis, we're able to pinpoint which systems have not only created these errors, but are continually creating them. So we offer data quality, not only from a past perspective, but also on an ongoing basis, which is very important.

But I'll get into that a little more in my presentation. First of all, the problem. The problem that we investigate with data quality is not your typical keypunch errors. Certainly those are important errors, but what we find is that errors are generated systematically. Systems, like those in an insurance company, are a perfect example. You have an operational system, but then you also have your management system. Managers use these systems to make decisions—whether it's creating assumptions or deciding which policies are profitable. These systems aren't optimized to do both. And we never will have that. What you need to perfect this is one system that can do everything, and this is something that has been attempted for the last couple of decades. Enterprise Application Integration (EAI) is one of the biggest things, and I think the new buzzword is XML. That's like the language that's going to optimize all these systems to interact with each other. The problem with that is XML is merely a vocabulary that allows systems to talk to each other, but it still doesn't optimize each of the systems to do everything you can possibly want it to. And because of this, as these systems interact with each other, as you're pulling data from disparate systems, you're going to find inconsistencies in these databases.

These systems do lack the mechanism to ensure that the information that you are pulling from them is accurate. And I think, if I could leave you with one thought, these are independent systems that you are pulling the data from. And because they are independent, unless you have access to all of these systems, you'll never be able to ensure your data quality. I will review that thought with you at the end of my presentation to make sure that it's clear. Because right now, that's going to seem foggy, but hopefully you'll understand it when I'm done.

At any rate, the goal of EAI is to integrate information. That's the goal that a lot of system architects are using to do exactly what it is that I'm talking about. The idea is to create one system that everyone can access, whether you're financial or pure data analyst. For any possible reason that you need data, the goal of the EAI is to integrate these systems, so you could just go to one system as opposed to going to the disparate systems.

Our goal is to create this ultimate system, although it's an ever-changing goal as new systems are created, and architects are getting more creative. But that is something that we're striving for as system architects. In doing that, it allows you to integrate the information and make a more efficient operation.

What causes these errors? As I said, right now you're dealing with Legacy systems. You're now dealing with Internet systems. You're dealing with a myriad of systems that you are required to gather your data from, whether it's insurance, pension, health care, etc. And these are what we find that a majority of our errors are created from. A typical example is an insurance company. Let's say you decide to go to a new administrative system. This would fall under the category of system conversions. And certainly a lot of IT folks are spending their time and energy in coming up with these programs to take something from a Legacy system into an statutory accounting practices (SAP) system. This is great, but unfortunately, there are errors that are created during this process. There are unforeseen program bugs. And as much as you try to integrate these systems, you will create errors. Certainly one of them is downloading the programming bugs that I mentioned, from a creation standpoint, as well as an ongoing standpoint. I was fortunate enough to work for an insurance company and one error is they'll write a program to fix that error, but unbeknownst to you, they inadvertently created another that might not be caught for another six months.

Certainly there is manual entry, but there are the benefit plan changes. To find a benefit change, is it just an architectural change? These are the myriad of errors that we see. I hope that I've broadened your idea of what a data quality issue may be, other than data entry. A data entry example is someone incorrectly keying in someone's premium, name or address.

Now, from the defined benefit perspective, and my case study goes into something more along the lines of defined benefits, the typical errors that you will find are date of hire, whether they're eligible or exempt, what elections they've decided to choose for their benefits, and on a pay frequency, the hours worked. An insurance company would have very similar errors, so you would find the correct premiums, for instance, a variable premium. Hopefully you can think of similar errors that you might find in insurance, or even healthcare for that matter. Again, these are not errors that have been generated because someone punched them in wrong. These are errors that were created as time has progressed, people have converted these systems; they've tampered with them. Even now as they lower them into SAP or PeopleSoft, these are creating errors. And these are the errors that Arcady found when he was working in the Federal Reserve project. This is how he stumbled upon this concept.

How does this affect all of us? It certainly affects the administrative overhead. In my case study, I think I have a couple of examples where there was a pension administration. The plan sponsor was making 82 edits per week. And after a data quality initiative, they reduced that to 8.2 edits per week. Certainly you can see a direct correlation in overhead, just from an employee cost. If you have poor data quality, and your managers are making decisions based on this, that leaves a lot of room for error and potential mismanagement.

The employee relations issues have a big effect on defined benefit world, as the Internet has become more prevalent. You find that employees are able to go on Internet sites and see what type of benefits they have and for some reason, if

these don't match what their actual benefits turn out to be, you have those types of issues. They lose faith in their plan sponsors and their companies. The companies, in turn, lose faith in the actuary administering the benefits. There's a whole myriad of issues that are related, from just having poor data quality and miscalculating one's benefit, to misrepresenting their information.

Then you have the potential increase for premium and administrative fees to vendors. It seems as if there is a disconnect between the plan sponsor and the actuary, because the actuary wants the plan sponsor to clean up the data, and the plan sponsor wants the vendor to clean up the data. And it doesn't seem to be the key method that's necessary to make data quality as important as it should be.

Certainly, there's the increased risk to fiduciaries, if the benefits calculate incorrectly. There are a couple of examples of lawsuits, and certainly everyone's heard about problems with respect to being cash bound and the loss of credibility. If you don't have any faith in your systems, down from the management to the employee, data quality is important.

Here are some examples of the effects of poor data quality from the litigation perspective. The first example is the Metro National City Pensioners who were overpaid $2.3 million from '87 to '95. And while some pensioners were underpaid $2.6 million as a result of incorrect pension calculations, and two 20-year-old calculation errors for Los Angeles County pensioners resulted in $1.2 billion in unforeseen liabilities, we'll probably force officials to spend an additional $25 million a year to make up for insufficient contributions to the fund. By the way, if you are really interested in data quality, I recommend this book, which is the source of these quotes. "Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits" by Larry P. English. Mr. English is very reputable in the data quality area. He runs a company called Information Impact, and his Web site is full of information. This is an excellent book, especially from a systems architecture standpoint and making sure that your systems are running efficiently. If you're very interested in this, I recommend this book highly.

Now the case study. There was a major airline with a population of 60,000 actives. They had gone to benefits outsourcing and because of that, they realized the need to streamline their systems. They sent their data to the benefits outsourcing for pension administration. However, when it came to a benefit calculation, they were forced to go back manually to the paperwork. They couldn’t rely on their systems. They decided that they're going to do benefits outsourcing to leverage that effort. They also decided to streamline their systems. And why should they have to go through that manual effort of determining benefit calculations at retirement? They decided to integrate their systems. They used Arkidata to perform this, and we provided the single database that was ultimately used for the vendor to administer the benefits. Now they are able to reduce some of the overhead that they had, but more importantly, they've ensured that their data is accurate.

Just give you a little breakdown on the case study: this is the distribution of the errors that we found. The most prevalent errors were eligibility level codes, more or less. Just to let you know the basis of the case study, I was working as an

information engineer, investigating data quality and I was just curious as to how this affected the actuarial world. I decided to do an investigation. I did a before-and-after analysis on evaluations with the data to see whether or not it would affect a valuation. Now, obviously, I had an idea of what might happen. And indeed it did happen that the errors average out. You're going to have some over-estimations, and you're going to have some under-estimations. In this case, I found there were a lot of eligibility errors. For this particular airline, the eligibility determined whether or not the employees worked at a pension able position. They might be working for 10 years for a company, accruing vested service. However, what we found in those years, was that they might not have been credited the service. And that's why that pulls us over 50% of errors. That was a very significant error.

But the results of my study were that there were no material biases pulled from ASOP 3. I found that there were no material effects. Essentially they netted each other out. As I said, there was some over-estimation of benefit services, and there were some under-estimation as well. And they tend to average out.

However, what I did find from a benefit calculation perspective, out of an active population of roughly 60,000 actives, if I recall correctly,15,000 or more were affected from a benefits calculation perspective. You can imagine the problems that this company would run into if, in order to streamline, it chose a benefit outsourcing company and blindly sent it the data that it was currently using to administer pension benefits. It's litigation waiting to happen.

I want to share with you the methodology that we use. There are many methodologies out there. Oracle has a methodology that it uses that is very similar to this. But it doesn't have the technology that we have to implement it in quite the way we do. Our process is more of an efficient one. We tend to do data quality projects in about six months. If you were to follow the Oracle manual of how to do a data quality project and ensure that your databases are accurate, it tends to range in close to a year or so.

Not getting too much into marketing, this is a methodology that, if you're not close to this, I guarantee you're doing it wrong. I know that there are a lot of actuarial vendors out there, they have data scrubbing software and, again, this might be able to be applied to life insurance companies. They might have a data scrubbing technology and, this is where I'm going to drive the point home. They have something that can analyze whether an active is followed by an inactive status? And then is it followed by an active again? You can verify someone's pay. You can do that. But the problem is that if you're a vendor analyzing the data that the plan sponsor is sending you, you're wasting your time. That's a bold statement, and some people might disagree with this. But all you're doing is catching some data entry errors of some sort. You might see a couple of anomalies. But the problem is, unless you're analyzing independent data sources, basically the source of the data, whether it's Legacy system backups or payroll backups from 10 years ago, you're merely verifying the proliferation of errors. As I said, through system conversions and system upgrades, these errors are being complicated through each conversion or each program that's being written for these databases. Typically, the vendor will try to analyze what a plan sponsor is sending them. And

as I said, the problem with that is you're merely looking at the proliferation of errors that have been sent to you.

The only way that you can have data quality is when you have buy-in from the organization you are working with, to try and ensure data quality, because you need their data. You need independent sources. Otherwise, you will not be able to verify any of the anomalies that you may think may be there. Another interesting thing that I should point out is, people say, "Can you do a data analysis for us?" And you really can't do a data analysis to see how good your data is because this is stepping through a myriad of doors. And after stepping through one door, a whole new set of doors opens up before your eyes. It's only after correcting errors that you might stumble upon another set of errors that might not have been prevalent had you not corrected the first error in the first place.

What you're doing is analyzing the data. You're looking for trends—and again, these are not data-entry errors. These are systematic errors. We're not talking about five errors. We're talking about thousands of errors that tend to show up from your analysis. And that's how you're able to pinpoint where the errors are created and what systems created them and at what time they were created. If there's one thing that I want you to walk out of the room with today, it's this: the only way that you will ensure a data quality project is if you have independent data sources in which to verify your database from.

Now I'm going to share with you the methodology that we use to implement this. This is the methodology we use and we've had a tremendous amount of success with it, but by no means is this the only methodology that you can use. As I said, there are many others out there.

Essentially we look at data, and at the sources. As I said, they're independent data systems. We start to become familiar with the data, and at the end of the project, no one knows these databases better than the analysts that are working with the project. Then we consolidate the data and clean it, and then we convert it to its ultimate destination. The analogy that I like to use to paint this picture is, it's a giant funnel. You have your data sources at the top, and you have the data targets at the bottom. You have this funnel with a myriad of filters that you're creating. Simple tests: is this employee inactive, is his first status an active status, and does he have any pay prior to this employment history?

You're just verifying the integrity of the data. You're taking a rather large data problem and decomposing it into smaller problems, very simple business rules that you write.

We collect the data from the data sources. Essentially, we pick out which population we like to use. Our subjects are typically the active population. Then we'll build the source data model. That's just saying we understand what resources we have at our disposal. We might have a pension administration system. We'll have a payroll system. We'll have some backup valuation systems, again, just independent data sources. And we try to understand what each of these systems is used for, as well as what each of the database columns are intended to be used

for. That's what we verify. Then we perform our population analysis, verifying that everyone that we are deeming to be an active vested retiree is indeed an active vested retiree. That might include cross-checking all the various sources to make sure that they are verified and for those populations.

From that, we begin deriving the business rules. This is something that's very important, where I'm leveraging my actuarial experience into the data quality career. The premise that we go on is that if you just leave this data quality job up to an IT expert, there's room for error there. I mean who better understands an actuarial database than the actuary? It makes the most sense that a subject matter expert is the one deriving these business rules that ultimately ensure the integrity of the data.

From here we define data audits, and data audits again are these miniature filters that I'm talking about, the business rules. The decision tree is very important. The decision tree is where you might write a business rule, and I'll give you an example. Let's say that you have verified that someone is an active employee, and prior to that, he was an inactive employee. This is obviously going to affect his credited service. The decision tree must know that as you build this hierarchy of business rules, ensuring the data quality, that there might be a business rule that you come up with, down the tree, that might affect some of the data that you had tested in an earlier business rule. It's important to have this decision tree be a continual process so that any time you change or correct data, it must always be rerun through this decision tree, to ensure that you're not introducing any more errors. Not only are you correcting something historically, but you want to ensure that any changes you make are correct from an ongoing basis.

In Phase 2, as we're defining the data audits and trees, we're defining the correction rules and we're verifying the correction samples. This is an ongoing process. We're constantly fixing data, making sure it's correct. This is not a one-time type event. This methodology is meant to be an iterative methodology, because otherwise you are running the risk of introducing errors, which is contradicting the whole purpose of the data quality initiative in the first place.

Finally, in Phase 3, it's the data conversion. At this point, we've looked at our data sources. We've become comfortable with them. We've fed them through our filtration process, and now we want to deliver the final data to the target database. This should not be taken lightly. Many people feel that as soon as you clean the data, you're done. But if you take this last phase lightly, what ultimately happens is you get the same problem that created these errors to begin with. That is, if you don't convert the data correctly, you're going to introduce errors and you're back to the same problem.

In Phase 3, after you have cleansed and consolidated the data, you then build the target model. You define the maps from your databases that you currently are working with to the target databases that are ultimately used to administer the pension, life insurance policies, and healthcare.

And we typically do preliminary conversions, meaning we run through a myriad of tests, and we find that there are errors that are created. We claim to be experts in data quality, and we have this tool. Basically, the premise of our company is built around the technology that we have developed. And even with that, we find that errors are created. It's very important that this is an iterative process. They're continually testing and verifying what it is that you will send to be the final database. After you fine-tune and identify the residual errors, you run the final conversion and then implement ongoing information integration.

What I've been talking about has been from a historical database point. We've looked; we've analyzed errors that have been created in the past. You've created a myriad of business rules to institute that your database is accurate. Why not use that from an ongoing basis, to validate that your database continues to be accurate?

To illustrate what I was talking about, here's an example of what we have today. This is a challenge that anyone involved in EAI is trying to attack. This is systems architecture. We live in a world today where even with XML, it's a great vocabulary, but you still have unique systems that are being implemented. You still have to extract from one system, convert it to the next. Granted, XML provides you the level playing field of being able to convert on this same vocabulary. But these are still independent systems that you're required to grab your data from. Ultimately, what EAI is trying to accomplish is shown in Chart 1. We want to create an intelligent information hub. This is important because when companies are working with their databases, making changes, making systems upgrades, you always have this hub that information will run through. It's a hierarchy of business rules that will ensure that your data is accurate. As you're making changes to one system, having this in place will guarantee that you're not bleeding elsewhere. And that's why it's important to try to achieve this picture. And that's what EAI is all about.

MR. KNORR: When I go through the data analysis, the data cleansing, the data conversion process, I like to start out by considering where the data came from. That is, where do we get the data that supports the statements that we make as actuaries? Data, like rates and factors, a lot of times, come from ourselves. We actually create the data. Other things are brought in from outside sources, or even various independent systems within our own company, like Craig explained. All these independent systems are coming together and being translated and brought into one common internal system. And then finally, using internal data to compile, summarize, apply factors, and sort, that would be from one internal data file.

If we look at the data that we create, ASOP 23 says it does not recommend that an actuary audits the data. I guess I would differ in cases where we actually create the data. I would say that we have an obligation to make sure that that data is correct, and if that means auditing the data, then we should be an integral part of that process.

One thing that we can do is check the logic that creates that data. There is an expression "measure twice and cut once." In my process, I like to measure three times, cut once, measure again a couple of times, cut again, until I get it right. A lot

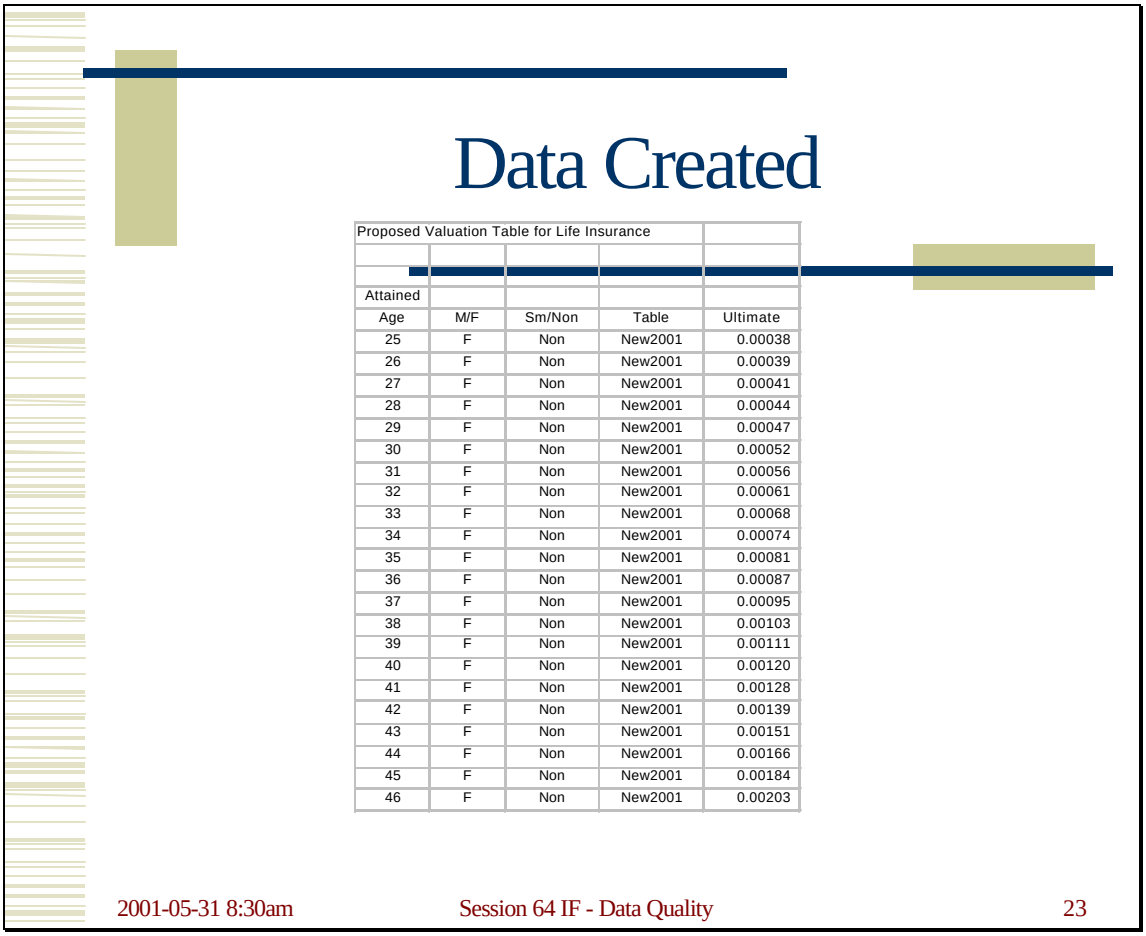
of times the data that we create has errors in it that are not obvious, but the programming logic has some obvious flaws in it. It might be easier just to check the logic, and actually do some manual calculations of a few of the cells.

My normal mode of operation is to check the logic fairly quickly, and then check the output for inconsistencies. That is, inconsistencies with prior versions or between one category and another. Then does the data make sense? Graphing data can help a great deal.

Also we like to make sure that when we're creating the data, nothing gets dropped and we don't end up with data that we didn't plan on having. That is, we don't want too much or too little data.

Now I'm going to switch to a data file in Microsoft Excel. The numbers are the *qx*s, these are mortality rates that I downloaded from the SOA Web site. This is a proposed life insurance mortality table for valuation. The numbers are the *qx*s, the categories are attained age, gender, smoking status, and the table name. In this data file, I also have prior versions or prior mortality tables. Also I introduced some errors into the data just to demonstrate some of the points of checking the data for errors.

Table 1



Data Created

Proposed Valuation Table for Life Insurance

| Attained Age | M/F | Sm/Non | Table | Ultimate |
|---|---|---|---|---|
| 25 | F | Non | New2001 | 0.00038 |
| 26 | F | Non | New2001 | 0.00039 |
| 27 | F | Non | New2001 | 0.00041 |
| 28 | F | Non | New2001 | 0.00044 |
| 29 | F | Non | New2001 | 0.00047 |
| 30 | F | Non | New2001 | 0.00052 |
| 31 | F | Non | New2001 | 0.00056 |
| 32 | F | Non | New2001 | 0.00061 |
| 33 | F | Non | New2001 | 0.00068 |
| 34 | F | Non | New2001 | 0.00074 |
| 35 | F | Non | New2001 | 0.00081 |
| 36 | F | Non | New2001 | 0.00087 |
| 37 | F | Non | New2001 | 0.00095 |
| 38 | F | Non | New2001 | 0.00103 |
| 39 | F | Non | New2001 | 0.00111 |
| 40 | F | Non | New2001 | 0.00120 |
| 41 | F | Non | New2001 | 0.00128 |
| 42 | F | Non | New2001 | 0.00139 |
| 43 | F | Non | New2001 | 0.00151 |
| 44 | F | Non | New2001 | 0.00166 |
| 45 | F | Non | New2001 | 0.00184 |
| 46 | F | Non | New2001 | 0.00203 |

2001-05-31 8:30am Session 64 IF - Data Quality 23

In this file, we don't need to page down to see all of the data elements here. I have a pivot table set up here, so that if I wanted to see the male mortality rates instead of female, I can just click on the M and that changes the numbers. It's fairly easy to change from one variable to another. Let me go back to females, non-smokers. But even with the ease that the pivot table has of changing the data so that you can look at things, you can go cross-eyed just looking at all the numbers. In cases like this, I like to put the numbers into graphic form. In a graph of the mortality tables, the female *qx*s above age 105, the curve wouldn't be exactly smooth. This is not one of the errors that I introduced either. That anomaly, if you want to call it that, does not seem to appear in the male data. The male data seems to be much smoother.

Also, below age 70, the curve would be very smooth. On the other hand, the numbers are so small that it's hard to tell, visually, whether there is any anomaly in there. In the case of mortality tables, or things that look like they're exponential curves, I like to put them in a log scale. Log scale is available in most graphics packages. Once we show the log scale, we may see that there is a point that looks off, and that's one of the errors I did introduce at age 56. In fact, if I go back to the basic data and correct that error, the log curve would look smooth.

I wanted to show the slope of the curve. Even though the curve itself looked very smooth, the slope of the curve seems to bounce around quite a bit. And you can see within the male smokers another error that I introduced at age 105. There is one error, but it introduced what looks like an anomaly at 105 and 106. That's because of the way slope defined here: the ratio of the *qx* at one age divided by the *qx* at the prior age. So, error in the slope would be produced when the error in the *qx* shows up in the numerator and in the denominator.

We can also compare this data to the basic 1980 CSO, and also to the 1990-95 mortality tables. This shows that the general curve of the mortality rates is fairly smooth and consistent. Not only are we comparing them visually, but we're also taking the ratio of one table to another. In the graph, the portion of the curve that's over 100% shows that the new table is greater than the 1990-95 table at those ages. The new table has mortality rates that are less than the 1980 CSO basic data.

Likewise, we can show the difference between male and female. We can do this by first, looking at the log scale of the data and then also taking the female-to-male ratio. I'm surprised at the mortality rates for females at some young ages are greater than the mortality rates for males at those ages. And these are also not errors that I've introduced. These are the actual data that I downloaded off of the Web site. If I were analyzing this, I would question that and find out if there is any questionable data in the female rates.

Finally, comparing smokers and non-smokers, you can see at age 106, one of the errors that I introduced. Again, we show the log scale and the ratio of non-smokers to smokers. This table seems to imply that once a smoker reaches age 105, he has the same mortality as non-smokers, and I guess that's true for the very few that make it up to 105.

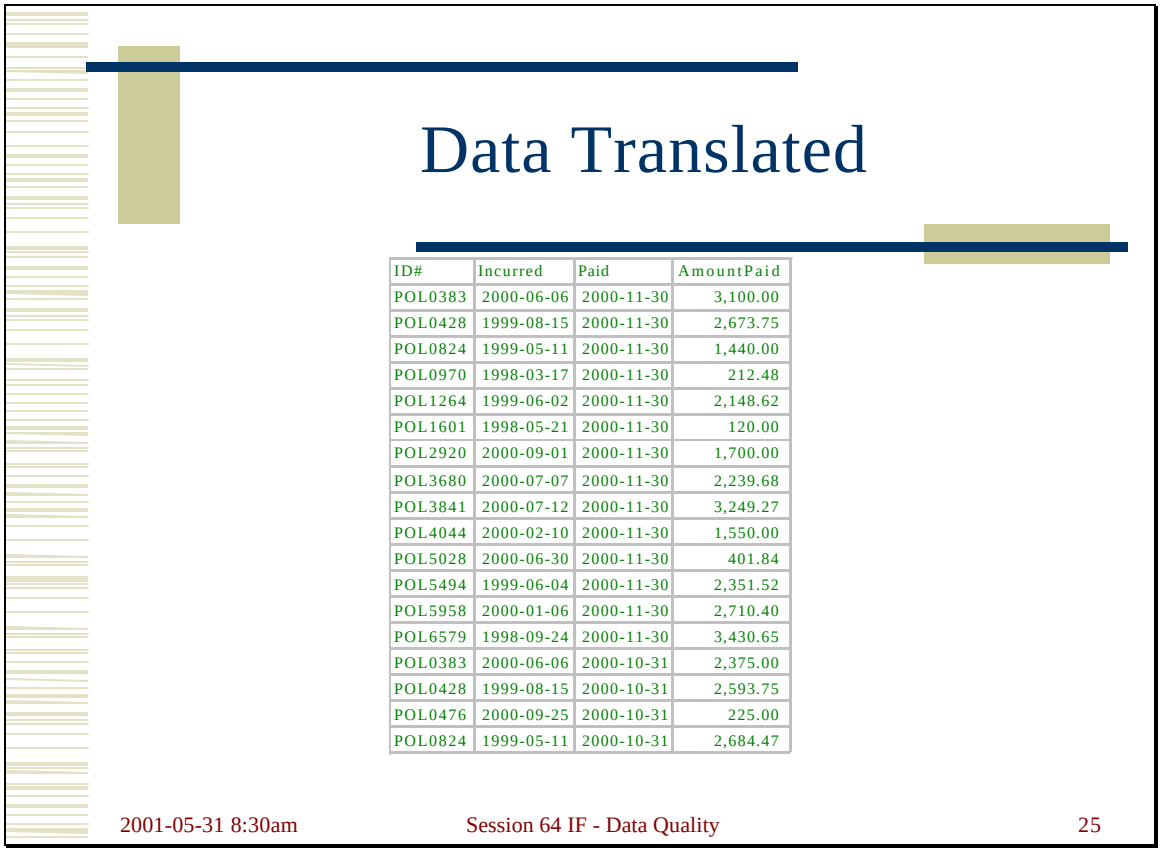
Now for data that we don't create, data that we gather from other sources, other sources being other companies or other systems within your own company, where the data needs to be reformatted, derived, converted, merged, whatever. A lot of things can go wrong when you merge that data, translate it or calculate things. Checking the logic by using a sample to check a little bit more thoroughly is a good idea. You want something that you can actually bring into a spreadsheet. It always works for me, because it's something that I feel I can get a handle on, and change and look at a lot closer. When I ask for samples, I like to identify the sample by a certain digit in a policy number, or some kind of identifier that appears in all of the data files that I'm bringing together. And typically policy number is a common element. For example, if you take a second to the last digit in the policy number and if that's a 7, then I want that as part of my sample. And that way, I'm pretty much assured of a 10% sample since there can be any one of 10 different digits in that second from last position.

Also, when you do that, then you know that policy will have matching data from the other files, assuming that you're matching by policy number. You can get a 1% sample or even a 0.5% sample, using a similar method.

When you're bringing data in from different systems, you want to make sure that the numbers that you had before the translation or the conversion match the numbers that you have after you merge the files. Program edits are always a good idea to test for things like that. Dates are important fields like you'll see in the next example.

This is a data file that comes from the claims system. And this is our basic data that we want to work with. Here the number field that we're working with is the amount paid to the insured for a claim. The categories are policy identification code, incurred date of the claim, and the payment date of the claim. And we can go to the bottom here and see that the data is pretty limited. We can also check the number of records by putting together a quick pivot table. The pivot table shows that there are 263 records. The total amount paid is $456,000 and if we wanted to, we could break that $456,000 by incurred date or paid date or policy ID. But I want to show this information by other categories. I want to know how much we paid by age, by plan, even by diagnosis. What kind of ailments did the person have when we paid the claim?

Table 2



Data Translated

| ID# | Incurred | Paid | AmountPaid |
|---|---|---|---|
| POL0383 | 2000-06-06 | 2000-11-30 | 3,100.00 |
| POL0428 | 1999-08-15 | 2000-11-30 | 2,673.75 |
| POL0824 | 1999-05-11 | 2000-11-30 | 1,440.00 |
| POL0970 | 1998-03-17 | 2000-11-30 | 212.48 |
| POL1264 | 1999-06-02 | 2000-11-30 | 2,148.62 |
| POL1601 | 1998-05-21 | 2000-11-30 | 120.00 |
| POL2920 | 2000-09-01 | 2000-11-30 | 1,700.00 |
| POL3680 | 2000-07-07 | 2000-11-30 | 2,239.68 |
| POL3841 | 2000-07-12 | 2000-11-30 | 3,249.27 |
| POL4044 | 2000-02-10 | 2000-11-30 | 1,550.00 |
| POL5028 | 2000-06-30 | 2000-11-30 | 401.84 |
| POL5494 | 1999-06-04 | 2000-11-30 | 2,351.52 |
| POL5958 | 2000-01-06 | 2000-11-30 | 2,710.40 |
| POL6579 | 1998-09-24 | 2000-11-30 | 3,430.65 |
| POL0383 | 2000-06-06 | 2000-10-31 | 2,375.00 |
| POL0428 | 1999-08-15 | 2000-10-31 | 2,593.75 |
| POL0476 | 2000-09-25 | 2000-10-31 | 225.00 |
| POL0824 | 1999-05-11 | 2000-10-31 | 2,684.47 |

2001-05-31 8:30am Session 64 IF - Data Quality 25

I've asked the IT department to merge some files for me. I asked them to get things like line of business, name, date of birth, state, issue date, and plan out of the policy file. And I also asked for date of service and diagnostic code from a different claim file, which identifies more information about the claim. They got all of this data and they matched it to the data they already had. In doing so, someone, probably me, used the date of birth and the incurred date of the claim to identify

what the age of the person was when they incurred the claim. I used the incurred date and the issue date to identify what the policy duration was for that claim. I used the plan code to identify what the elimination period and benefit period are, so those are all things that I've derived with a calculation or a look up table.

Now I have this new file that has what I had before, plus a few extra fields. I add up these numbers and I've gained a couple of records. I now have 265 records, where I had 263 records before. Where I had $456,000 of amount paid, I now have $456,789. If I show this information by policy ID code, there are extra records to the policy that showed up. As it turned out, when we were merging the data, the extra data that showed service date and diagnostic codes for one of the payments, there were two service dates. In the merge, the IT people didn't know that when you have two records going to one, that you split the amount paid between the two. They just assume that you use the same amount paid for both records. And so we doubled up on a couple of records in this file. That's one of the errors that I would find.

Other things that I would also look for are dates. There is a natural sequence of events. People are usually born before they're issued a policy. They're usually issued a policy before they incur a claim. And then after they incur a claim, there are services that they incur, so the service dates come after the incurred dates. Payment dates come after service dates. I would test certain things like taking the incurred date and the issue date. For example, a cell may show that there were policies issued in 1997, and the claim was incurred in 1996. Another way to identify the error is to look at the policy duration for that claim, and you see we have a negative one in there for a policy duration. That's a clear sign that something must have gone wrong. Another thing to watch out for in this case is the first valid policy duration is zero. In a lot of actuarial studies, the first policy duration is one. In that case, when the first one is 1, then zero would be suspect data.

Also, things like negative ages should be looked at closely. In some cases, a negative age indicates a Y2K problem. It's really a problem when only the last two digits of the year are stored.

The last thing I'd like to discuss here is the look up tables. We have things like benefit period, which we've derived from the plan code. The plan code identifies what the benefit period is. We have some N/As in there, where we couldn't find a plan code, A1O. That's a typographical error. Someone miscoded A10 as A1O. The other place where the N/A shows up is C109. This is not a typographical error. There is a plan code C109, but it's so new, that someone hasn't gotten around to updating the translation table, the look up table. Those are things that can go wrong when you're merging data.

When we're creating data files, control reports can be used to check the output of the reports. Or again, checking the logic is always a good idea. If there are any hard-coded numbers in there, or hard-coded dates, those should be eliminated as much as we can. In my work I see a lot of monthly reports come to us, and a lot of times the data in a certain cell doesn't change from one month to the next. And it ought to change from one month to the next. A lot of times it's even the date at

the top of the report. If it is the report for July, it shouldn't have June at the top of the report. A lot of those things are just that someone who was keying in the data, updating the spreadsheet, got distracted by something more interesting than keying in data.

Another thing that might go wrong from one month to the next is if factors are only loaded for the first 10 years of a policy, and policies start going beyond the 10 years. Then it may not be obvious in the first month or two that certain numbers have dropped. Eventually, those factors that are applied to things will either kick out as an error or show up as zero, and many times the numbers should be far away from zero.

This is just another example of a report that can be checked internally. This is the Long-Term Care Experience Reporting Form A, and in this form, you can check numbers in one column to something from last year's report. You can check the reasonableness of the loss ratios. Those should be steadily increasing. The rows represent how old the block of business is, so if you know that the business is a closed block of business and has been closed for the last five years, you should not see any numbers in the first few rows. Those are the kinds of things that you can check for in reports.

Table 3

FOR THE YEAR 1998 OF THE XXXXX LIFE INSURANCE COMPANY

LONG TERM CARE EXPERIENCE REPORTING FORM – A
NATIONWIDE EXPERIENCE for experience in calendar year 1999
CUMULATIVE CLAIM EXPERIENCE
WAIC Group Code XXXXX
WAIC Company Code XXXXX

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 |
|---|---|---|---|---|---|---|---|---|---|
| Policy Form | First Year Issued | Calendar Duration | Earned Form by Duration | Incurred & Paid | Reserve for Incurred by Unpaid | Total Incurred Claims | Change in Policy Reserves (ALA) over the Experience Period Loss | Anticipated Calendar Duration Percentage | Number of Insured Lives |
| abc - 123 | 1989 | 0 | 2,341,802 | 44,842 | 515,823 | 560,665 | 3,765,480 | 20.2% | 6,754 |
| | | 1 | 5,560,683 | 378,969 | 806,403 | 1,185,372 | 2,669,753 | 22.0% | 12,229 |
| | | 2 | 5,353,074 | 796,639 | 1,719,467 | 2,516,106 | 3,014,314 | 29.4% | 4,980 |
| | | 3 | 7,959,895 | 739,348 | 1,285,841 | 2,025,189 | 4,394,520 | 38.1% | 8,052 |
| | | 4 | 10,250,529 | 1,681,251 | 1,929,653 | 3,610,904 | 5,393,431 | 44.4% | 17,948 |
| | | 5 to 9 | 45,383,252 | 9,023,552 | 7,101,165 | 16,124,717 | 17,104,502 | 64.0% | 46,667 |
| | | 10+ | 276,226 | 56,688 | 19,059 | 75,747 | 89,842 | 106.2% | 515 |
| | | | | | | | | | |
| | | Total | 77,125,461 | 12,721,289 | 13,377,411 | 26,098,700 | 36,431,842 | 52.1% | 97,145 |
| Policy Form - Calendar Year (a) Actual Loss Percentage (Col. A7 / Col. A4)_____ 33.8% | | | | | | | | | |
| (b) Anticipated Loss Percentage (see Inst. Form A Item 4)_____ 52.1% | | | | | | | | | |
| (c) Actual to Expected Loss Percentage ( (a) / (b)) _____ 64.9% | | | | | | | | | |

I'm not sure how true this is, but I heard about one company that made a decision to get out of a certain line of business or to sell a company or something like that, based on certain reports. These reports were put together by some data processing area. The reports only left room for a certain number of digits in the profit line. Any high-order digits were truncated. If they had $12 million worth of profit, the one was left off and they only showed $2 million worth of profit. Based on that information, they said that it wasn't profitable enough so they sold the company and later discovered the error.

MR. WILLIAM A. WOOD III: I have a couple of tips for Excel and a question of my own. I think first difference graphs would be really useful to demonstrate disparities in the data, particularly if you're looking at a mortality table. An error in monotonicity would jump out there obviously. Progression from one age to the next meant the mortality rate went down and you didn't expect it to do so. Something I use a lot and you might find handy, is conditional formatting in Excel. I basically know that a number should be something, and I say that if this number is not, that's something that it should check to. It puts it in big bold red font. This is

something that is staring at me saying, "Error," or "Do Not Use," and that's right across the top line of the front page of the report.
My question, and maybe open this to the crowd, is about relating spreadsheets. Unfortunately, a lot of times you have two different spreadsheets that refer to each other. I hate it. It's always a nightmare and if you insert a line in one, you're a dead man. I found one tool that's very useful and I can't exactly remember the name. You'll see it in the Excel newsgroups. Everyone should really be familiar with Internet newsgroups, to get problems solved with software. It has about 26 utilities, add-ins to Excel. I think you can get it, including its macro source for about $60. It has a linked table manager, where it will give you a nice report in a spreadsheet, even all of your linked tables. Unless there are a lot of problems with them, again, from inserting. Does anyone have any useful advice?

MR KNORR: I have spreadsheets linking to other spreadsheets, and I am always in a dilemma as to whether I should bring that original data into spreadsheet. Therefore I'll have duplicate copies of something. If something is corrected on the original sheet, or if someone else has access to it, and introduces errors, then that data is not the same as what I have in the sheet that I would have linked from. I always have a dilemma with that and many times when I open up spreadsheets, it asks me if I want to update the link, which is nice to have, except a lot of times I don't know why I would ever want to link from that spreadsheet. It throws up a flag that there is a mystery cell somewhere that I must have copied and it's not easy finding that cell and eliminating it, because I don't want to link to something outside the spreadsheet.

The comment about conditional formatting is something I stumbled upon too, and that's very powerful. I enjoy using it. It can identify the highest number or the numbers that are out of line, those kinds of things, fairly easily.

MR. DONALD L. GLICK: I tend to download a lot of stuff from our mainframe and the mainframe is very tolerant when it comes to dates. We had dates like January 31, and guess what? A month later, it's February 31! And they're just happy as can be with that. I'm not. Anybody have a generalized solution to screwy dates like that?

MR. BLUMENFELD: I think my perspective is a little different on that, in that I'm looking at this data quality issue much more from the systems end. If that's happening, there's something wrong. Dates shouldn't be changing policy status. My solution is that the system's architecture should be preventing that from happening. I have a very business rule-driven methodology, so any time you're getting information, the premise is that the information, before it gets to you, should be assured somehow. I'm more of a behind-the-scenes type person, dealing with systems architecture, and everything I've been talking about. We have technology that we use to ensure that.

Certainly, I would love to hear any other perspectives that the rest of the group has. From my angle, something like that should never happen because there should be something in place to ensure that such errors do not occur in spreadsheets and anything of that nature. Wouldn't it be great if we lived in a world where you could

just get your data and not have to question the accuracy of it? That's the goal. That's what's happening now. That's what's happening with data quality. People don't want to have to go through spreadsheets, adding columns and verifying. It would be nice if you could just get it and you're off doing what the data was intended to be used for. And so, I've taken the perspective that it's something that you should do and it's absolutely crucial and prudent to do so. I want to deliver the information, so that you're so confident that it's accurate, that you'll never have to question it. From my standpoint, that's the idea of the intelligent information integration hub. It's in the middle, sitting in all those systems, so that when you get it, nothing like that will ever happen. But certainly, that's an idealistic utopia. I guess I'd almost pose it to everyone here in the room, and say if there are any solutions to something like that, I guess inexpensive solutions is probably more appropriate. Because what I'm talking about is systems architecture.

MR. KNORR: Yes, I've run into similar things and, first of all, you need a rule. For example, if you're born on February 29, when do you celebrate your birthday? On the 28th most years? Or March 1? If your rule is that January 31 is the last day of the month, and then a month later is the last day of February, then you need to program something in there that converts that to February 28, unless it's a Leap Year, and to follow that rule. I understand your problem about downloading it into a cell, because Excel sees something like that, and says, "This is not a valid date. Goodbye". What you have to do in that situation, is: (a) identify all the ones that are not valid dates, (b) separate them between year, month, and day, (c) figure out what month you're talking about, (d) figure out what the last day of that month is. It's a headache when you're downloading information into a system that doesn't have the rule programmed into it, like Excel, because it's just going to be an error message. You already know that it's an error.

FROM THE FLOOR: The bottom line is to trust no one. Regarding the issue of dates and everything, and I think Craig has brought this up, is you have to validate everything. Interpretation of the rules, is the biggest issue, because one participant may interpret a rule one way and another one may interpret it another way. Then it gets back to the source of information and within insurance company's environments. You have your data stored in different warehouse environments, be it an administrative system or claims system, and it's the integration. The biggest problem I think everybody has in contributing and with data quality is the fact that they're integrating all that into one format. That takes the information from a different source and then interprets the rules as far as what goes on. I guess the whole thing with data quality, is that it's a general statement, you just have to validate everything and after that point, just follow along whatever business rules you've established. It's not easy and the more you deal with Legacy systems, the tougher the problem is because a lot of times Legacy systems go on and people don't check the data quality in those as much as they do in developing new systems.

MR. KNORR: I have a feeling when you're working with data, you not only have to know the system, how the system treats certain data, how the data gets arranged, and how it gets into the system, but also the business. You have to

understand the business and know, for example, that payment dates should come after service dates.

MR. BLUMENFELD: Whether it's data warehousing projects or data quality issues, there are companies out there that have software that can help. Different companies do different things—such as name and address cleaning. You know that's something that we don't do, but there are some very successful companies out there, and just to give you an idea of the costs: software ranges anywhere from $5,000 and then there's a name and address data quality company, Navasoftware. I think their software runs $200,000. If you ever do decide to do a project that requires data quality, you can find them; they're on the Internet. They're evolving as data quality becomes more important.

Chart 1