RECORD Volume 30, No. 2*

Spring Meeting, San Antonio, TX
June 14–15, 2004

Session 57 L
Business Intelligence for Actuaries

Track: Computer Science, Futurism

Moderator: Michael K. Rigby

Lecturer: Michael Levine

Summary: This session covers context and general background on business intelligence (BI) and online analytical processing (OLAP), how to build a successful application and how to integrate with other applications.

MR. MICHAEL K. RIGBY: Mike Levine is an actuary who works for G7 Systems, Scarsdale, N.Y., as a consultant. I worked with Mike at MetLife for many years in a number of different roles, and Mike has had an interest in technology and systems as long as I've known him. One of the first systems that I worked on with him was a system to determine the value of a producer by analyzing his book of business and transactions. Needless to say, that involved tremendous amounts of data and the need for analytical tools. With that start, looking at data warehousing and analytical tools, Mike moved on to look at different areas of the company: compensation, compliance, sales reporting and expense analysis. In each of those areas, he brought new insights, techniques and tools to those difficult problems.

At an earlier session, they talked of IT versus actuaries. Both Mike and I see the real value in professionals who are comfortable in both worlds — both the actuarial and the IT worlds. As a consultant now for almost two years, Mike's work is concentrated in the business intelligence (BI) area, working with the techniques and tools that he'll be describing for you. I know you'll find this interesting.

MR. MICHAEL LEVINE: I've broken this subject into four topics. First, I'll provide some context and background on BI, including some demonstrations. Next will be a basic discussion of data modeling. Third, I'll review the technology and tools currently on the marketplace. Last will be a discussion of implementation strategy in terms of how actuaries would work with IT in this area.

Here's a list of tools: Excel Pivot Tables, Microsoft Access, Cognos, MicroStrategy and Actuate. A couple of buzz words are data modeling and data warehousing, and there's a reference to SQL, the main coding language for extracting data from relational databases. I'm not going to go through the whole list.

There are basic problems that all of these subject areas have. First, aggregate data can mask the underlying business patterns and drivers. It's the overall mortality or overall expenses, and you can't see where it's good and where it's not so good. Detailed data can be difficult to manage, organize and visualize, and it can also be difficult or impossible to reconcile aggregate data to detailed data, especially when you did not create the aggregate data. When different users and different processes take different approaches to these problems, the results can be what we call multiple versions of the truth. In my experience these problems reoccur in many applications and can be frustrating, especially in these days of Sarbanes-Oxley. You want better capabilities to address these problems. That's what the BI implementation is intended to do.

In the IT world, different approaches and techniques have evolved. Business intelligence, as an actual term in the IT world, has come to refer to a proven established set of approaches and technologies that can help a company to understand what's going on in the business. This is what I intend to explore in this talk.

I like to think that BI provides a window to the data. In other words, you can get lots of useful reports and analyses all derived from a consistent set of data where you don't have to do a lot of manual work to get the report of the analysis every month each time the data changes., You can download it to Excel, and you can operate against the data and not spend your time setting up the data and setting up the report. Most of you were familiar with Pivot Tables. I liken this to having a Pivot Table with access to millions of rows but with the ease and facility of a Pivot Table.

For those of you familiar with SQL, the basic way many of these tools operate is they transform the results of an SQL query into a report or an analysis. They generate SQL automatically. They take care of the access to the data.

I realize that actuaries have been confronting and solving problems involving large volumes of data for many years even without any new techniques and technologies. This, in my opinion, has led to a situation where actuaries have not been taking advantage of some of these newer technologies. It's my perception that a marketing department is more likely to use some of these tools and techniques than the actuarial department. My goal is to help you to become more familiar with what is out there and to gain confidence in working with it. I also suspect that in many of your organizations, you have these tools and techniques in-house. You may not be using them, and this will give you ways to get more comfortable with working with them.

Another general term I'd like to define is online analytical processing, or OLAP. This is a term whose meaning has changed over the years. In the early days of large commercial databases, the term "online transaction processing," or OLTP, developed, and that basically refers to things like an airline reservation system, an ATM or the amazon.com system. Each airline reservation, each bank withdrawal, or each book order is a transaction. Databases and technologies evolved to handle many transactions, where each transaction is limited to small amounts of data. For example, Amazon can find information they've stored about you in their database quickly.

As that type of technology was evolving, it gave rise to the opposite concept, which was that users needed access to large volumes of data to create aggregates or to run algorithms against them, usually for analytical purposes. We do that, for example, in assembling transaction detail into a ledger line item. OLAP originally came to mean the opposite of OLTP, and this would involve different ways of setting up the database and different ways of setting up the technology.

The term has taken on a new and more specific meaning, and I think the best way to show that is with a picture (see Chart 1). OLAP has come to refer to technology that supports multidimensional structures that allow the user to look at data across many business dimensions or hierarchies. OLAP structures are sometimes referred to as cubes, and I think that's easy to see in Chart 1. The real physical cube has only three dimensions, at least on earth.

However a real business problem typically has more dimensions. I've represented five dimensions from a ledger example: products such as life, annuity or health; organization, which would be the company's organizational structure, departments or cost centers; account, which would be line items in the ledger, expenses or reserves; a time dimension; and what I'm calling a scenario dimension. As an example, you could think of scenario dimensions as GAAP versus statutory. Other dimensions are also going to be applicable in a ledger.

OLAP functionality refers to the ability to slice and dice along different dimensions, for example, to show account by time, as on a 10(k) report; income statement items by time; account by product for a single point in time, such as on financial reports, that show financials by line of business (individual life or individual annuity) or by account; or by product and organization for a single account. Showing expenses by product and organization would be something that you would do when you're performing an expense allocation analysis. That's definitionally what OLAP is. I'll be getting into more demonstrations soon.

As these technologies have evolved the different examples are sometimes called styles of BI. One, for example, is the use of dashboards: executive dashboards with automatic links to important data, e-mail notifications and the like. Again, there's a massive amount of consistent data, and this is a way of getting at it.

Another example is what's called enterprise reporting. For example, in an organization there may be 10,000 cost centers, including rollups of cost centers distributed across the entire U.S. or even worldwide, where you automatically provide the monthly expense versus budget reports to all cost center managers over the Web with security to all levels of the hierarchy. That's an example of enterprise reporting.

Interactive reporting is setting up reports with some customizable aspects. I'll be giving an example involving claims. There are basic reports, but the users can select which columns to look at or can filter the report in various ways.

There's the real OLAP slice and dice against a logical cube. I'll be giving an example of that involving expenses versus expense allowables, sometimes called expense revenue. Another example is the OLAP slice and dice, but with the ability to drill down to individual transaction details. If you see an aberration in expenses, you theoretically could, with the tool, drill down to the actual expense item and see what seems to be amiss.

In the more advanced forms you can apply advanced formulas to sets of data. This is something the credit card companies do a lot. What are the characteristics of a good borrower, and how do we predict who's going to default on a credit card? In the life business, the marketing context would be who are the good customers, who keeps their policies in force, and how can we predict that? That's in the area of data mining, and predictive modeling, which is what the industry buzzword is.

I'm going to switch gears and go through some demonstrations. The reason for this is not so much to demonstrate the content. I don't want you to look at examples for content, but I want you to get a sense of the look and feel. What I've been saying so far may seem a little abstract, but I want you to get some flavor for the types of applications.

I'm going to start out with a dashboard and a stable reporting example. What a company might set up is the same thing you have in your normal corporate Intranet. The first place users might go would be to their dashboard. This example is demonstrating dashboards and using some traffic signals to show who's a good sales manager and who's not performing. It has speedometers, or some metric related to assets. A typical dashboard would allow drilling in this case into one of the sales managers and getting some more information about the sales manager and that sales manager's performance.

Another example is an expense summary. The demonstration came out of a hypothetical manufacturing company, but expenses are expenses give or take. I think that you would relate that well to the insurance industry with commissions and other forms of compensation. There is some ability to drill down into expense line items. All of these are basically preformatted reports, but what BI technology is doing is organizing them in a coherent way with logical drill paths that would make sense to the right level of user. In all of these examples, the reports are not just dummy reports. They are going against real data that are set up in Access databases on my laptop. It would be the same technology in a larger installation, just a larger database.

Next I would like to look at some finance-type reports, for example, a balance sheet report. Your ledger is going to have balance sheet data, and maybe a number of you probably have relatively new PeopleSoft ledgers or SAP ledgers. Or you may have an older homegrown ledger or some earlier version of a ledger. One way these tools tend to get used is as the front end to the ledgers because the tool vendors have gotten sophisticated in report distribution, whereas the companies that make ledger systems are more worried about the transaction aspect of the ledger. They have reporting tools, but they're not necessarily sophisticated.

This example with the balance sheet could show total company or different business units within the company. The "click for account" view would show assets as positive and liabilities as negative, so it adds up to zero at the bottom. It might be the type of thing that the right level of analysts would want to confirm. This example uses inventories because it's a manufacturing example, but I think you could relate to the idea that reserves are similar to inventories as balance sheet items that you'd want to know more about. I clicked on the inventory line of the balance sheet and was able to bring up some more detail on inventory.

We go to the income statement. From what I've seen, the vendors are trying to market all of these things in terms of their usefulness with respect to Sarbanes-Oxley. I don't know how many of you attended the Sarbanes-Oxley discussion, but the idea was that even if your data are disparate, by bringing them together in an automated way, you can verify that certain figures are consistent with other figures. Those types of checks could be built into the software. Because everything is automated and hands-off in a full-production implementation, you've supported your analytics with your IT control. That's a general ledger example.

Those are examples of reporting. Here's a prompted report. I have to navigate a bit to get to it. This may not be an actuarial example except in certain industries or in the reinsurance area, where it's important to drill down to individual claims and look at patterns of activity by claim. Here are some data set up for claims. The analysts can bring in company and coverage, maybe the date the claim was received, diagnosis, metrics such as gross reserve and net reserve. They want to group on company and coverage, and they want to summarize on these two reserve values. I won't worry about sorting and filtering. They can create a report on the fly.

Again, don't worry about the content if you're not working in the claims area or if this seems relatively simple. I think you can relate it to more advanced analyses that would be of interest to you. Another type of feature is table of contents, so you could drill to a page on the table of contents.

Another thing I could demonstrate is you can also run these things to Excel. One frustration with these products five years ago was that it was hard to get from the report to Excel, and you always want to be able to get it back to Excel.

I also have an example of software that's a little bit more slice and dice-oriented. This is a product that I think is like that Pivot Table with access to millions of rows of data. This example involves looking at expense revenues. Is everybody familiar with that concept or with expense allowables against expenses? I've already set up a report that shows that a hierarchy of products — traditional life broken up into two categories: universal life (UL) and annuities.

Let's define another metric. I have the expenses and the revenues, and now I'd like to define something that I'll call profit, so there's a little editor. I'd like to bring in revenue and subtract expense. I'll validate that, save it and call it profit. I'll go back to the report, go into design mode, find the metric I set up, bring it down here and rerun the report. Many more things can be done along these lines, but you can set up these reports and these formats. You're going against large quantities of data, and it's not that different from Pivot Tables. The advantage is the database is updated. Any report that you've set automatically can be updated. This type of thing can also be sent to Excel. I don't want to spend more time on drilling and demonstrating the functionality of this tool. We could come back to it if we want. I wanted to give you a flavor for what these tools are like.

Let's go on to some of these ctuarial examples that have always occurred to me. Experience studies are a natural; claims analysis, like I showed, can get much more sophisticated; anything in terms of slicing and dicing the general ledger is multidimensional; and reserve reporting and analysis can be done.

You may have generated your reserves by using PolySystems, homegrown systems or feeds from your subsidiary. You bring it all together in a database, and then you can use these types of tools to analyze patterns and trends over time, compare statutory to GAAP by coverage and look at all the different dimensions related to reserves or reserve-like items such as deferred acquisition cost (DAC). You can do sources of earnings analysis, which I demonstrated with the expense and the expense revenues, and mortality versus mortality costs. Different companies have different important ways of measuring their businesses, but I think many of them lend themselves to these types of things.

If you're an actuary and are not purely in the finance area — you could be in the sales area or generally supporting the chief financial officer (CFO) — where your job might involve supporting sales reporting or audit and compliance. There are many uses of these tools.

Chart 2 shows a generic BI architecture. It's showing data from different sources. The data from the different sources are going through an extract and transform process and are getting into more optimized forms for these tools to use. Do you remember what I said about OLTP? Many of these might be OLTP. They're set up for daily ledger updates. They're not optimized for use by a tool. The tool looks at the data, and then the users have that window into the data.

I'm going to discuss how you set up the data, data modeling and data management. I'll talk about BI tools in general and how to go about selecting a tool or working with IT to select a tool, and then I'll talk about integrating the whole thing.

I'm going to discuss data modeling. There's a lot of data out there. You're going to start working with these tools. You're not going to look at all the data in the entire enterprise all at once. Even more important than the tool is getting a handle on how you want the data to look and how you think the data should be organized. The first thing you have to do is to focus your business need. It's not enough to say you can do better analysis with better data. You have to define what your required capabilities are and how you would use those capabilities to optimize your business. You want to narrow down to a subject area. Examples of subject areas are the examples I've been giving.

What are you trying to accomplish with this first BI effort? You want to improve your way of analyzing claims instead of using old mainframe reports that IT programmed. Maybe you have a lot of extracts that you have to manually assemble to do mortality studies. Maybe you like using Pivot Tables for slice and dice, but the data are getting too big for that. Perhaps you want to look at expenses. You have to narrow it down to a subject area. Don't try to solve everything at once. You want to anticipate that you'll be working with IT. I know from that discussion yesterday that relationships with IT are not always optimal, but this is an area where it's important to work with that department. I'm going to give you some strategies for making that happen.

What you can do in advance of working with IT is think through and define the logical structure of the data. This gets to a definition. The logical data model is a map that defines the data elements in the subject matter area and explains the relationships among them. A test of the model is whether your desired output — your reports, your views or your drill downs — can be created if you have the data as defined in your model. I'm not asking you to worry yet about how to create a physical database or how to populate it. The first thing to do is to think through the data.

Chart 3 is a highly simplified and schematic example of a data model. This is for claims, a somewhat mundane area, but I think everyone can relate to it. Think of each box as a logical table. Think of each element listed for a table as a logical column on the table. The little tables surrounding the big table are what we call dimension tables. I've highlighted company dimension, examiner dimension (the claims examiner), diagnosis dimension, coverage dimension and time dimension. Each dimension table has a unique identifier, another attribute such as the name. The company dimension table is a list of all the companies or legal entities in this example in your organization. There's a unique identifier, there's a name, and there may be dozens of other attributes, such as the state of incorporation or the year it was established. This is not a big table. It's listing all the different companies.

Similarly for the claims examiner, there's a unique identifier, the last name, the first name, the department or cost center he or she is a part of, and maybe other attributes such as the types of claims he or she is specialized in.

The big table in the middle is called the fact table, and the columns on the fact table include the unique identifiers of the dimension, such as the company ID, the product ID and the coverage ID. This big fact table also includes columns for metrics. Metrics may be maximum claim amount (whatever the maximum is under the coverage), the year to date or the cumulative claim amount, any reinsurance amount and reserves. If you're in this area, you'll have your company's definition of the important metrics.

The fact table has keys, which point to the dimensions, and it has metrics. Intuitively, on this particular fact table, each row of the table is a claim. This is a large table. This has every claim, all these important attributes or dimensions and all these important metrics.

Chart 4 is a logical model for expenses. Dimension of the company is the same. Department is the same. Time, product and account are similar to that cube I showed earlier. Each row of the fact table is an intersection of the different combinations of dimensions.

One thing that's different about this example is that the fact table is not designed to go down to the expense transaction level, and this fact table is more of an aggregation. That's not a mandatory difference. That's a difference in terms of this model. If your analytics don't need to go down to the expense transaction detail, you wouldn't model for that. If you want to be able to analyze down to the expense transaction level, your fact table would reflect that it's an expense transaction. In either case, most of these dimensions are the same: the company, the department, the product and the account. This is another example of logical modeling.
The third one is a mortality studies example (see Chart 5). I'm not going to spend any time on this. I think you're getting the picture. Identify the key dimensions or attributes associated with this subject matter, create unique identifiers for them and then define what your fact table is going to look like in terms of how it relates to the different dimensions and what the important metrics are. The common term for these are star schemas because they look like stars.

Having shown it by example, I'm going to review a few components to reinforce what I've said. The important concepts are granularity, dimensions, metrics, hierarchy, sorting, filtering and derived metrics. Granularity refers to this: When you're thinking through your subject matter, you want to think of what the grain of the model is, or in other words, what constitutes a row in the fact table? In the claims example, it was a claim. We want to go down and look at every claim. In the expense example, it was more aggregated. It was every combination of product, department, company, account and month. It could be that for your application or for your need, you need something more granular. That would be the example of going down to the expense transaction. This is something you want to think through in advance. You don't want to go to IT and not know what the answer is.

Dimensions are sometimes called attributes. They're sometimes called characteristics. It depends on whose book you read and whose software you use. They describe the granular entity such as company or coverage. They're usually nonnumeric, and the claims examples are the ones listed here.

Metrics are sometimes called measures. They're sometimes called key figures. They are numerical quantifications associated with the granular entity. They could be claim amount, reserve amount, revenue expense, revenue amount or revenue-based amount. They're numbers.

Set your model up with a fact table where each row is an intersection of the attributes and contains metrics. Identify your dimension tables. Dimensions are usually organized into hierarchies, so here's an example of that. The examiner is part of a department. The department or cost center may have an extensive hierarchy on top of it. When you're thinking this through, you want to be able to explain to IT, "I don't need to see every examiner, but I need to see every department," or "I want to be able to see both the lowest level department or cost center and any roll up of that." You have to identify what the important hierarchies are. Products have hierarchies. Accounts have hierarchies.

Putting it in these terms, most reporting and analysis simply involves taking different views of the available dimensions and metrics. What BI is about is managing your data and defining your dimensions in metrics so that it is easy to get all these reports and analyses. As you think through your model, will it support your analytical requirements?

There are some other key concepts. Keys are unique identifiers on the dimension tables. You want to be able to uniquely define those. You want to think through what columns you would like to be able to sort on or filter on. You'd like to think about derived metrics, or metrics that aren't inherent in the data but that you want to mathematically be able to express, such as the ratio of expenses to revenues.

There are some things to keep in mind with this dimensional modeling, and this is also why as an actuary you want to be involved in this process and not leave it all to IT. Don't let dimensions get confused with each other. More than once I’ve seen the individual life product get confused with the individual life department (product as in line of business and department as in whoever in the company is responsible for that). These are different. Just because the name's the same doesn't mean it's the same. Again, that's an area where an IT person who is not as familiar with the business can get confused.

Your company may already have an enterprise data model. Your IT department has probably done some work on this. Don't be afraid to leverage it, but don't be afraid to challenge it, either. You can do this initial work without a sophisticated tool. You can set up each table as a grid in Excel, or Access is good for this.

The next step for data management is to identify the likely data sources. You know this already. Where do you currently get your data, what's the structure of the data, and what are the inadequacies? If you want to build a good, consistent solution, this is your chance to start asking for better data and identifying the problems with the data.

You're going to want to think through what the likely data sources are for the reference structure of the organization. In other words, what's the official department hierarchy? What's the official account hierarchy? It's often the ledger, but if it's something like sales reporting, there may be a sales office hierarchy that's different from the cost center hierarchy. There are different reference structures. You want to think about where the model would require actuarially generated data, such as reserves or reserve factors, where you are an input or somebody in the actuarial organization is providing data as opposed to having an external source. Get a sense of how you would transform the data to a logical view.

I'm going to give a simple example that I've seen as a problem more than once. We're going to get the annuity data from these two administrative systems. We're going to get life data from these two administrative systems. We have the corporate ledger. Each of these systems has a different key or an identifier for the issuing company. You need to standardize the issuing company. You also have to make it rigorous. I've also seen some of these sales systems where the issuing company is not rigorous. It's often oversimplified. You know the data better than IT does. You're probably familiar with these issues from having worked in this area before. You want to start putting structure on it. You're the expert. If you don't take this opportunity, it won't happen.

I think except for showing you the data models, I'm not saying anything new. I think if you've been working and solving problems involving large quantities of data, you're familiar with concepts such as dimensions and metrics. The one thing I can say for myself is I started trying to solve problems and thinking things through this way, and I've used the term dimension or metric and didn't even realize 10 years ago that it was a technical IT term. I thought I was figuring it out myself. In one sense it's reassuring to realize that what I've gone through is standard IT terminology. This is the way IT wants you to think about it, and because as actuaries you're probably pretty good at thinking about it this way, it's a natural thing. What you want to start doing is putting it more into the structure so that IT can use your logical data model to start creating a physical database that covers data modeling.

There is a good book by a gentleman named Ralph Kimball. I saw the first edition and thought it was pretty good. The second edition is excellent. It makes all of this transparent. It gives you insight. It gives you a sense of empowerment so that you'll be able to tackle the problem and analyze it and model it.

FROM THE FLOOR: I notice some of your fact tables contain more than one fact. Is that standard?

MR. LEVINE: More than one measure?

FROM THE FLOOR: Yes, more than one measure, where some models will use only one fact.

MR. LEVINE: I would not say that there's a rule against putting multiple facts or multiple metrics in a record. I think that's a normal thing to do. In the expense revenues case, for example, there's the revenue, there's maybe the revenue base, and then maybe there's the factor that converts the base into the revenue. I think it's a normal thing.

FROM THE FLOOR: I've seen tables, though, where there's a field that tells you what the fact is. Therefore, you would have only the one metric.

MR. LEVINE: That could be done, and that gets into the realm of a physical implementation. For thinking through the data, I think it's important to document, understand and define all the metrics that make sense at that level of granularity.

I'm going to switch to technologies and tools because they're good concepts. What does it take to make it work? To discuss data management technologies, I'm going to start by explaining a strategic BI architecture. You're not going to go out and build this or get IT to build it right away, but it will help to relate to how these technologies and tools work together, and then I'll explain some approaches that are less strategic but that might be good as a first step.

This strategic architecture may appear daunting and more than needed, but it's useful to review it. Your company may already have this in place, but you don't know about it. That would be a good situation, and you could see where your application would fit in. Chart 6 is an example.

The main components, as I showed before, are data sources and a process to clean and transform those data sources. In a strategic architecture you're going to put all of that into one integrated data warehouse.

The data warehouse is organized into smaller data marts that are optimized for different applications. The BI tool — such as MicroStrategy or Cognos, — will typically have a client component, or something that can go onto a desktop, and a server component, so the desktop component is literally like Excel at your fingertips. The desktop component tends to be used by power users and developers. The server component is for all those people in the 10,000 cost centers who are going to get at the data through the Internet.

It's getting too technical to give you exactly what all these server components are, but I'd be willing to come back to it. In most modern tools there's a web interface that users can use to access the components. There's no longer a need to put clients on everybody's desk. Basic data sources could be anything. The cleaning and transforming processes are sometimes called extract, transform and load (ETL), which is jargon that you'll start hearing from IT if you get involved in this area.

The useful thing about an enterprise data warehouse if your company has one and it's working, which is unfortunately not always the case, is that it is the place to integrate data from multiple applications. All data go through a transformation, which standardizes the ID for the issuing company for all usages. If it's a good warehouse, additional subject matters can always be added to it. This is not a talk on data warehousing, It's good if it's in place, but it doesn't have to be there. Data marts are the physical or logical subsets of the data organized and optimized for the application and the analytical tool.

For example, with the experience studies, you might want policy level data but not all the customer data that marketing might want, such as the address of the person, or you might want cell data. I'm going to jump to an example that I did with the expenses versus the expense revenues set up in an Access database.

Although it seems more complicated than the logical models, I had the setup in the physical Access database, which you can think of as a mart. The mart is going to take the logical model and convert it to a physical data model. There's a fact table for expenses and a fact table for revenues. There's a product key, for example, and it traces out the whole product hierarchy. This physical implementation of a mart looks a lot like the logical model, and that's why you're designing the logical model. The physical marts can be set up similarly, although not necessarily identically. There may be other technical reasons not to make it exactly the same, but it should generally conform to the logical model.

The heart of it is the BI software itself. Typically there's a specialized vendor product, and I've listed the vendors that you may have heard of: MicroStrategy, Cognos, Actuate, Hyperion/Brio, Business Objects and Crystal. These are some of the vendors in this space. Is it generally the case with this group that you have some of these in your company? Is it generally the case that you're a user or you're not even aware how it's being used but that you know it's there? You have some familiarity. The product could be a component of SAP BW. Maybe somebody's used that. That's built into SAP or is something you could buy with SAP. That's where you have the client and the server.

I've run through a lot regarding the Web interface. I showed you strategic architecture. I've rattled off some things. Even though it's a computer science discussion, it is starting to sound nice. There is a lot here.. Perhaps you want to build something simple. You want to study a few things. You can't afford to have this big thing created. How do you deal with that?

The first point I'd like to make is maybe there's more in place than you realize. If your company has this in place for nonactuarial problems, it has it in place. What's the worst case? You're coming late for the party. It's a good thing in a way because many of the issues may have been addressed already. If not, you still may not need a big data warehouse. Let's take one problem such as mortality studies or reserve analysis and build a mart for that one problem. You may not need Web distribution. You want to put a strong analytical tool on a few people's desktops. In fact, that would have been the main paradigm five or 10 years ago when the Web interface was not that mature or sophisticated. Something like this might make sense.

You know you have your data in some flat file or in some less than optimal warehouse, you understand that data, and you don't want to build another warehouse. You don't want to build another database. Some tools, not all tools, can connect directly to any data in any form. It's going to be more limited, but I want to make the point that if you want to get some of the analytics done, you don't necessarily have to commit to the whole architecture. It's best if it's there and IT can support it, but you're not necessarily committing to it. There are other ways to do it.

If your company has PeopleSoft, PeopleSoft sells a component called EPM, or enterprise performance management. It's a warehouse, and you can put anything into it. It doesn't have to be PeopleSoft data. That platform can be used to support the data.

Regarding usage, if you're in a situation where your company does not have any of these tools or your company has a number of these tools and maybe the part of IT that supports you likes one of them, and some other part of IT likes another one, you're in a situation where you can make a decision. What are the considerations because not all tools are the same? These are things you want to think about.

What type of BI are you talking about? Are you talking about slice and dice, Excel Pivot Table-type stuff, or dashboards and reports that don't have to be that interactive but that have to get to a lot of people? How much data are you talking about? Are you talking about aggregated expenses based on ledger of balances? Are you talking about expense detail? How many users? Five people all in the same office in the actuarial department or are you working on a project that has potentially more users, such as underwriting offices or even sales offices? In terms of architectural considerations, what is in place already? What is IT already supporting? What version? Is it using a current version of a tool? Is IT happy with the tool? Is what you're intending to do foundational or stand-alone.

Has somebody given you the mission?“I'm glad you went to that lecture and learned something about BI. We need it." What's your assignment? Is it budget, scope, roles or skill sets? Different companies have different philosophies as to how deeply they want actuaries to get involved with technology. That's a cultural issue as I've seen it, or maybe it's a local issue to you.

One thing you don't have to worry about is what database to use. I've been at companies where people are arguing about whether it should be DB2 or Oracle. I don't think it matters. That's an area where you can defer to IT. You don't want IT to be making this tool decision on its own if the decision has not already been made, however. If there are different parts of IT, you'll want to engage different areas and not just your local IT support area. Get them to bring in the other people because you know the answers to the previous set of questions. What's the nature of the application?

If you're still at the point of having to decide, the vendor tools are not all the same. This is an example of tradeoffs with these vendor tools:

	Product A	Product B	Product C
Core Competency	OLAP Slice and Dice	OLAP Slice and Dice	Enterprise Reporting
Data Management	OLAP is directly against the relational database. Performance requires a database that follows a particular model.	Loads data into proprietary "cubes." OLAP is against these cubes.	Can access data from any source in any format.
Advantages	Unlimited "logical" cubes and drill down.	Cubes can quickly be developed from existing sources.	Get any report to anyone from any source, including interactive reports.
Disadvantages	Must have built the database that meets the standards	Hard to drill beyond the "physical" cube.	Not true robust OLAP.

The first column is the tool and the remaining columns represent different vendors. Different vendors have come about approaching BI in different ways. The A and B products approach was to start with the slice and dice. They wanted to build a better pivot table. The product C approach is more of the enterprise reporting variety. The dashboards get everything to everybody, and there's a Web interface. They also handle data differently. The product A approach doesn't create a physical subset of the data.

In the product A approach you put the data in a database as I've put it in Access for these examples. The product A report developer and project developer will map your logical model directly to a relational database. When it's doing the queries, it's pulling from that underlying relational database.

The product B approach maps your requirements to the database, but then it has an interim step where it loads all the data into a proprietary bit of software, with a proprietary data structure, that's a real physical cube. The analysis is going against that proprietary structure; it's not going against the underlying database. You can refresh that structure from the underlying database, but that's a stand-alone separate structure.

In the product C approach, it can access data from any source in any format. These people who were working on enterprise reporting were willing to go to flat files, and you don't have to have any interim place to leave the data. You can go there directly.

The advantage of the product A approach is that you're going against the whole database. There are unlimited logical cubes. If your database goes down to the expense transaction, your tool on your desktop can go down to the expense transaction. You can set it all up that way. The disadvantage is you have to build a database. You have to build the database that's like that Access database I explained to you. It has to look like the logical model. It can't deviate too much.

The Product B approach is you can go to almost any source to put the data into the cubes. Once the data is there, it's much harder to drill to another cube. If you build a cube with aggregated expenses or aggregated mortality information, you want to drill down to that actual expense. It's a separate structure. It's not just part of that standard OLAP cube that you first built. The advantage of that is those cubes can come from any data source, and it's quicker to develop if you don't want to go and restructure your data.

The product C approach is to get any report from any source, but it's not the true OLAP. If you know what your metrics and dimensions are, you can create reports that are consistent with those, and that's why you thought it through, but it doesn't have the same slice and dice feel. Depending on what's important to you, this is an area where product selection matters.

Having said all this, the industry stabilized and matured in a way, and now what's happening is all of these vendors are trying to get into each other's areas of expertise. Cognos and MicroStrategy now have enterprise reporting solutions, and Actuate has cubes. My perception is they're still best at their core competencies. It's an area where it's less important, but that's also where the version you use matters.

If your company is on Actuate 5, you're not going to have certain features versus the current 7 or the soon-to-be-released version 8. Similarly with MicroStrategy if you're on version 5 versus 7you may not have the same features. That's where the version of the software matters. There may be a decision. Do you upgrade your current software, or do you bring in a different product? That's my last point in terms of convergence. The IT department will have people who are up on all of this stuff. My purpose is to reduce the jargon and give you the ability to communicate with them.

Regarding implementation strategy, every company and every environment is different. Every relationship with IT is different, as are scope, subject matter, budget and existing capabilities. Everything is a little different so it's hard to be specific about implementation strategies, but I'll give you some ideas of what I'd like to see going in.

Every actuary should be able to expect that their IT departmentwill do these types of things because they are the responsibility of the IT department in any significant organization. They should have a protocol for project initiation. Development should follow a structured, scheduled, monitorable process. Role and responsibilities on any project of any type should be clearly defined between business and IT. There should be risk management protocols in place. For example, if the project is taking too long, the issue is elevated. Important applications when they're completed are put into production, meaning they run automatically under the IT department’s control, and you don't have to worry about moving data around. Application development is segregated from production, meaning that if you're working on developing some cubes, you're not going against the same underlying database that the general ledger is operating with.

Now you want to build a BI application. You have this good framework with IT. It has good tools and technology in place. What else do you want? You want IT to appreciate the importance of iterative development. IT may have good tools in place. They can have a good framework but still say, "You want to build this? It's going to take a year, and these are the exact phases."

I don't know whether you saw Susan Lee's discussion earlier at this meeting about different IT methodologies that related to the issue that some methodologies require a step-by-step process, but that's not what you want with a BI application. You want an iterative methodology so you can quickly get a subset of the data, start building a report and determine whether the report looks as though it makes sense. You want a risk management strategy— that elevates issues involving both financial risks and technology risks.

Here's a good way to allocate responsibilities to get the thing built. You should leave it to IT to deal with infrastructure and environment. You should take that first step on defining a logical model. Once you've thought that through, you want to work with IT people who are going to have titles such as data administrator and are going to have tools that are specialized for building the model, but you don't want them to do that without you. Let IT develop a physical database. Let IT worry about the ETL, subject to your specifications like that issuing company example. They'll need a lot of specs such as tool selection and joint tool implementation to build into the environment. That's their job. If you or your actuaries are comfortable with it, I recommend that actuaries do the report development like that example with the slice and dice, these other reports, or the expense reports. If they're not comfortable, IT can do it.

I've seen both situations. In one company with accountants and not actuaries, a couple of accountants right out of school were instructed, "You're going to develop these reports against our new ledger." They worked with IT and got pretty good at it. If you're comfortable with that, it's a good way to get things started, but you don't want to be responsible in the long-term for the maintenance. You have to follow their rules and protocols. I think that's a good way to get things done rapidly.

Leave Web development to IT. There should be shared project management. If it's not an ideal situation in terms of your relationship with IT, recognize you don't want to take too much on your own. You can end up with mission-critical things that don't have mission-critical support. My general advice would be to do as much as you can at low cost and low risk so that you can ultimately engage IT with much of the problem already solved.

Work out the logical data model. Use Access if you're familiar with that following this type of an approach. Develop prototype reports based on the data. Use Excel Pivot Tables. Say to IT, "We have this all thought out. We have a subset of the data. We can do all these things. This is exactly what we need done." It's not the most profound advice, but I've found it effective since you have to do your homework in advance. Don't go to IT with a general vague wish that you need something because it looked cool. Show that you've done a lot of work already. Start building your confidence so that you can do it. This also helps you to find any mistakes you may have made. Look for an opportunity, such as we saw in that PeopleSoft example.

I'm working with a non-insurance client right now, who is using the PeopleSoft EPM. Some other users in the organization needed a report. It was easy to throw three tables into that PeopleSoft EPM and attach the same tool that the main ledger work is based on, and they had their reports. Look for these kinds of opportunities.

Another good reason to put things into Access is it's a good way to have a tools assessment process. This can be useful even when you have a good relationship with IT. If you can set up 60,000 rows in Access, you give it to MicroStrategy and ask, "What can you do with this?" Or give it to Cognos: "What can you do with this?" They'll do a lot of work without getting paid, and you get more bang for the buck when you do pay them because they'll have made it work. That's how their marketing people do demos. That's how I'm using these Access examples as a standard tool. Let the vendor make the case for you.

I hope I've given you some exposure to this area and given you some things to think about. Here are some resources.

- Kimball & Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, Second Edition, John Wiley & Sons, 2002
- Vendor Web sites
  - www.actuate.com
  - www.businessobjects.com
  - www.cognos.com
  - www.hyperion.com
  - www.microstrategy.com
  - www.sas.com

Every one of these Web sites has extraordinary free PDFs and white papers. You can learn a lot about this subject area — logical modeling and data mining — by going to these Web sites. You can go to the MicroStrategy Web site and download its full software for a 30-day free evaluation. This is a competitive, aggressive industry, and the companies offer a lot. I don't know what your company's rules are in terms of doing that and loading it onto your computer, but you certainly could do it on your home computer, although they might ask you who you work for. There's a wealth of information from these sources. They don't quite explain how they're different from each other, but there's a wealth of information.

FROM THE FLOOR: One question I did have is, what do you think are the actuarial areas most amenable with respect to the use of these types of tools?

MR. LEVINE: Pretty much any problem I would have had in actuarial would have been an appropriate candidate. Where I'd like to see more aggressive use of these tools would be in mortality and experience studies because there are so many dimensions, and dimensions keep getting added in a sense, such as underwriting factors. There are large volumes of data, so that's one example.

For enterprise reporting, it's my personal view, and I'm not employed by any of these companies, that ledger reporting, 10(k), 10(q), statutory reporting and accountability reporting such as expenses and budgets all could have much better reporting presentation by using one of these tools. It's not that hard to attach to the underlying ledger. In my opinion, PeopleSoft is a good ledger, but I don't think Envision, for example, is comparable to some of these other tools.

A point I should make is that this is not actuarial software. This is not reserving software or cash-flow testing software. As I see it, the output of those tools goes into the database so that you can integrate those data if relevant with other data or if you want to do something like embedded value or value-added metrics and want to do it by distribution channel. That's what we got started on. Value-added reporting is not just another regulatory requirement to do in Europe. The real value is to do value-added by channel or channel and product. All of these variations of things you want to do will start to result in large volumes of data. It can be exhausting to assemble the data. It can be nice to have a tool that makes it easy to look at the data.

MS. FRANCES O. JONES: How do you solve the problem of keeping historical data? You have a data warehouse, and they're refreshing it once a month, but for actuarial purposes you need historical data from the start of the policy.

MR. LEVINE: In that fact table, you put observed date as a dimension. There are two situations. One situation is you're taking a policy snapshot every month. That date of snapshot is another dimension. Every logical row is the policy on this observed date. That's how ledger things would work — expenses for this observed month. Date becomes another dimension.

If you want to capture every transaction on a policy, it's the transaction date. There may be operational or physical reasons that IT would suggest, "If you're not going to use this that much, we'll store it in a different physical database." My advice to you logically is to make sure you put that in the model.

Also, the company's organizational dimension hierarchy changes over time. You know that happens, so that itbecomes not just the hierarchy, but the hierarchy on the observed date.

MR. JOHN W. ROBINSON: In my department I tend to use a combination of downloads with ODBC Microsoft Access. We got a call from IT, who wanted to form a data warehouse. I said that at least for my line of business, I don't need one because I'm connected to all these things. Is ODBC an underappreciated resource or am I perhaps putting too much stock into it?

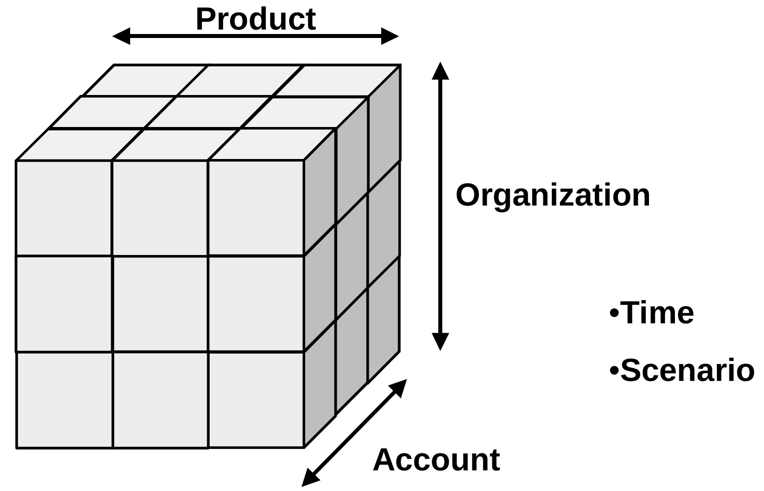
MR. LEVINE: There are technical reasons why what's called a native connection to DB2 is preferable to ODBC, but I don't think that's the question that you're asking. You're gathering data from different sources. You're managing it by putting it into an Access database, and you have it structured in a certain way and, hopefully, in a way that it is useful to you. You're probably writing queries against the Access database reports using Access' report capability. For a departmental solution you're doing exactly the right thing.

The reasons to put this into a more IT-controlled environment are more along the following lines: What if you get sick one day? Where are you saving it? Who's doing the backup and recovery? What about the next version? What if the next release of Access comes out and doesn't upgrade properly? These are mission-critical things. I have to imagine that what you're doing is important to the company. You want them under IT control because it's a controlled environment and because of Sarbanes-Oxley.

If you're running into issues where Access seems to have slowed down or you have too much data in there, it's a reason to get IT involved. If IT wants to be involved, I would ask what your concern is? Is it going to be too expensive or too difficult to work with them? Is it a small application, and it's not cost-effective? I can't say that what you're doing is wrong without knowing more about the size, the scope and the mission criticality of it. There's nothing technically wrong with what you're doing at the level that you're doing it. All IT would do is do it with stronger tools in a production environment. It's not logically different.

Chart 1

MultiDimensional "Cube"



Product

Organization

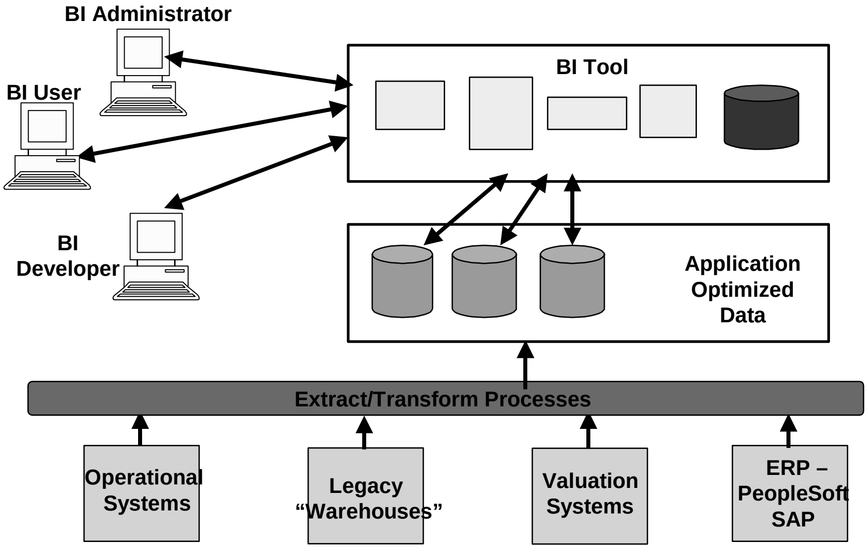
Account

- •Time
- •Scenario

9

Chart 2

Generic BI Architecture



BI Administrator

BI User

BI Developer

BI Tool

Application Optimized Data

Extract/Transform Processes

Operational Systems

Legacy "Warehouses"

Valuation Systems

ERP – PeopleSoft SAP

13

Chart 3

Claims Logical Model

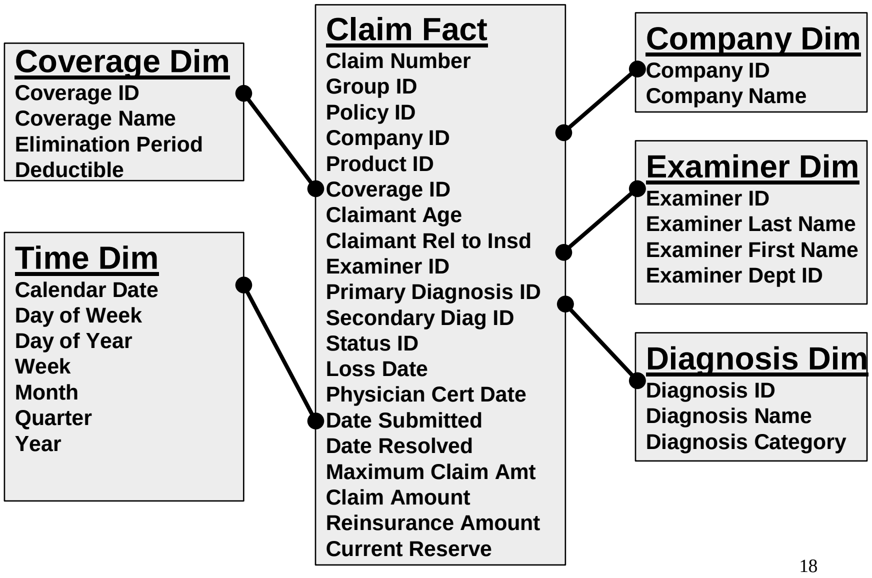


Chart 4

Expense Study Logical Model

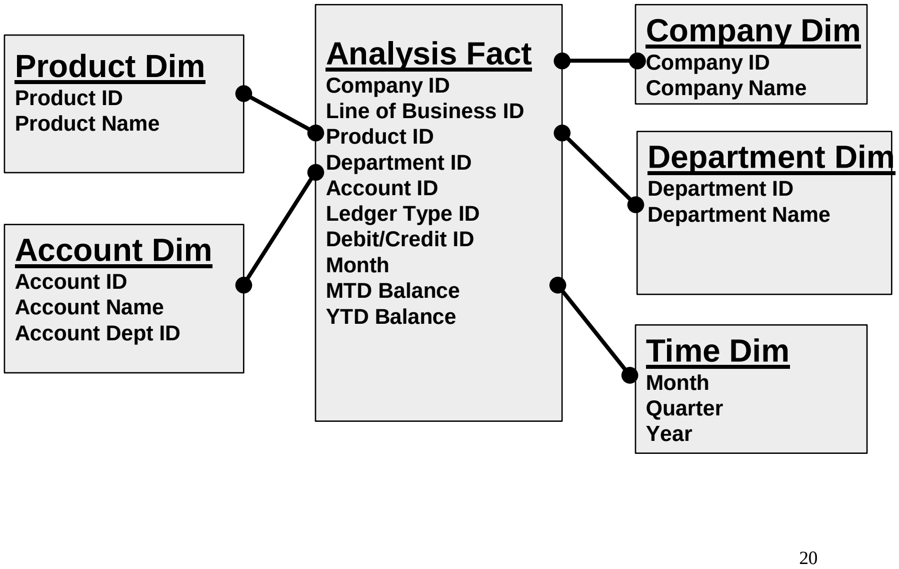
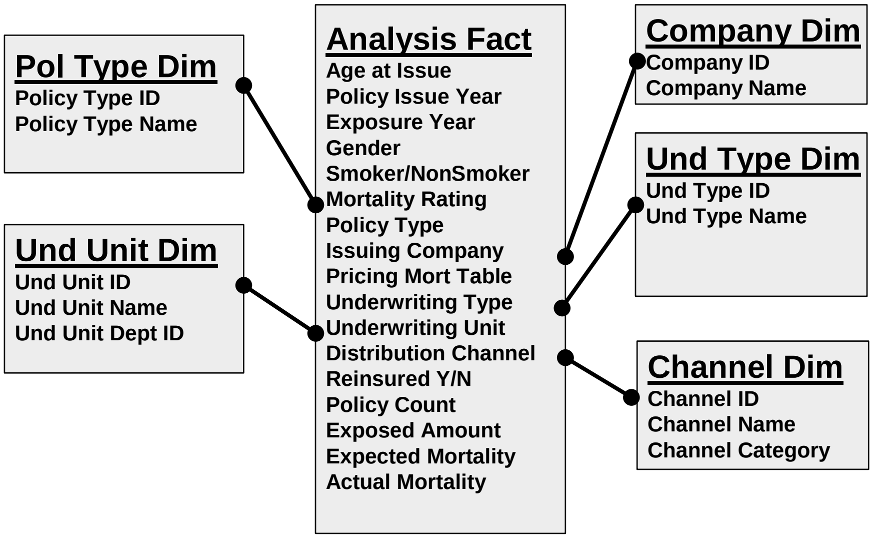


Chart 5

Mortality Study Logical Model



Pol Type Dim
Policy Type ID
Policy Type Name

Und Unit Dim
Und Unit ID
Und Unit Name
Und Unit Dept ID

Analysis Fact
Age at Issue
Policy Issue Year
Exposure Year
Gender
Smoker/NonSmoker
Mortality Rating
Policy Type
Issuing Company
Pricing Mort Table
Underwriting Type
Underwriting Unit
Distribution Channel
Reinsured Y/N
Policy Count
Exposed Amount
Expected Mortality
Actual Mortality

Company Dim
Company ID
Company Name

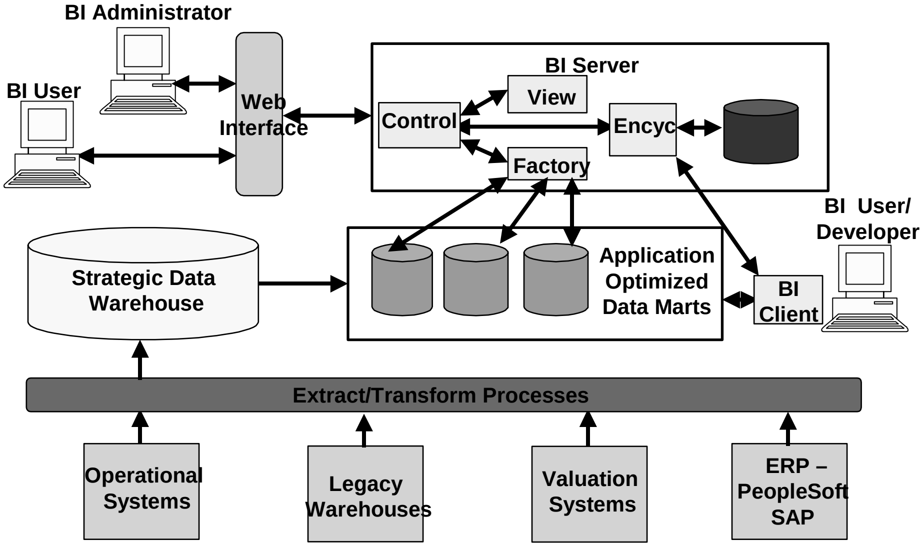
Und Type Dim
Und Type ID
Und Type Name

Channel Dim
Channel ID
Channel Name
Channel Category

21

Chart 6

Strategic Architecture



38