CHAPTER 7

Elements of Business Intelligence Solutions

In an earlier chapter, we discussed the numerous steps required for most data warehouse projects. No matter how the data was captured initially or what storage format was used, few data sources are constructed for use in BI. The typical enterprise has multiple data sources and archived information that may be required to put together the information needed for analysis. The steps typically look like those shown in Figure 7-1.

The diagram implies a cycle. Some illustrations used by vendors today show a path that leads from the first process to final delivery. The implication is that you do this dance one time and you're finished. This is seldom, if ever, a one-way series of events. Things change. New data is added. New systems are purchased (e.g., ERP, CRM). Mergers and acquisitions happen. Most of all, the business needs change. We have to account for all of these in order to provide an effective BI strategy.

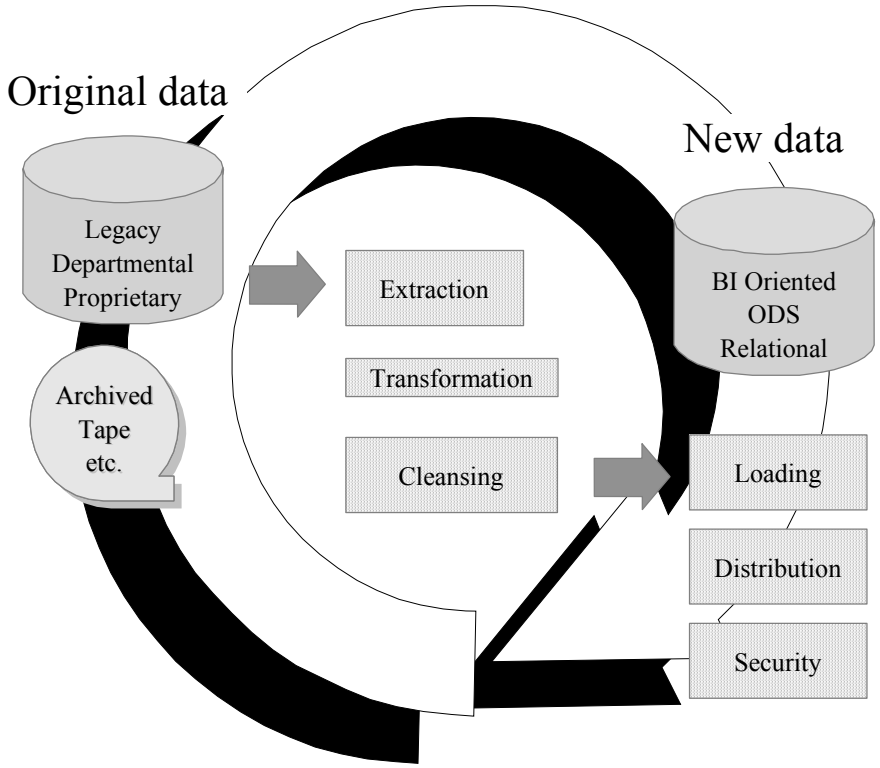

Figure 7-1 Data warehouse processes.

Data Warehouse versus Data Marts

There seem to be two camps of theory and thought about how best to approach the creation of new data structures to deliver BI:

- One school of thought is the "top-down" approach, involving the creation of a data warehouse or operational data store (ODS) for the enterprise. These will require massive amounts of time, effort, and money to deliver.
- The other school of thought is the "bottom-up" approach, whereby you create data marts and roll the marts up into a cohesive data warehouse later.

The top-down approach is nearly impossible to deliver in many cases due to the sheer amount of time it would take to implement it. Enterprises that were early adapters of relational database systems have the best chance to complete such ambitious efforts due to their knowledge of data and the many tools used to build, load, move, tune, and copy

information. You have to provide a database that is malleable, or you simply cannot be fleet of foot to handle change.

The bottom-up approach tends to deliver solutions faster due to the reduced scale of data and effort required. The problem normally associated with this approach is that the marts seldom get to the enterprise roll-up phase, and you often wind up with multiple marts with redundant information.

The most important thing to decide regardless of either approach is what the impact will be on the business and what the goals are when creating any new data source. If you decide at the onset that you will create a value as the "parent" for all permutations of the data, then the planned roll-out will allow you the maximum flexibility.

For example, suppose that a value such as SALES for an item is to be used in several areas within the business. One interest group may be concerned with the sales for any product, regardless of whether the item was kept or returned. Another group may be concerned with returns of the item for quality purposes or may want to have accurate information on how many of the item went out the door and did not come back.

In such a scenario, we will have different "adjustments" made for gross sales, net sales, returns, etc. If the core value of *sales* is used in multiple areas, it is a candidate for a warehouse. If, however, many areas use gross sales, net sales, and returns (or whatever) in their reporting and BI applications, then all these values should be maintained in a central store.

If we don't spend time understanding how a seemingly common value is used in the enterprise, we wind up with BI anarchy. We also wind up with multiple, disparate reports and results because returns may be calculated at a different time than the latest sales figures were updated, and we have to account for matching dates and more.

I have been in many meetings in which there were multiple, disparate values stored or derived for the same data. The vigorous discussions among users and IT as to who is holding the real value and why are awkward for the participants and painful to witness. There are often multiple derivations of the same value strictly due to timing or some differences in business processes.

As a former marine biologist, this *modus operandi* reminds me of how a jellyfish is constructed. Jellyfish are not single organisms with a body *per se.* Rather, they are colonies of cells that specialize and somehow work together. Many enterprises operate their BI environment that way. Somehow, they arrive at the same place and survive, but they have little or no direct communication or coordination among all the parts. Most jelly-

fish wind up on the shore drying out, because they simply cannot swim strongly enough to remain at sea.

A Rule for Storing Data Values

Use only one way to store data values or derive the data value. The enterprise is best served if derived values are stored as part of the server. Even simple calculations may be incorrectly performed, and dependence on a tool creating the result versus the server is risky. The greater the reliance on the users creating "the math," the greater the risk of obtaining incorrect results.

The goal of any data warehouse or data mart project should be to create a *single version of the truth for data* within the enterprise. If you create any value by using two different methods and approaches, you're setting up a formula for disaster.

Setting Up Information for BI Processing

The majority of this book is oriented to the analytics tools used in BI. This is primarily because that area brings the greatest value to any BI solution. Until access and analysis of the data is performed, there is *no value* to all the agonizing work performed to get the data into a shape suitable for analysis.

However, you need to spend some time on the building processes to educate any end users to the incredibly complex world of data transformations, cleansing, etc. This is an area that requires incredible amounts of time, energy, and knowledge about data and systems. Let's use a simple example to begin our discourse.

Data Extraction, Transformation, and Cleansing

The ABC enterprise has its current data held in some VSAM files and other sources. Company officials would like to begin constructing a customer-centric warehouse that includes the last five years of history by customer. Much of the detailed data has been archived to tape because it would take enormous amounts of disk space and they seldom use it. Currently, they haven't the disk capacity to store it.

Their data has some "problems." Customer information in the original systems was stored as a series of encoded numbers and text. Many of their applications use a look-up table (another file) to match the customer numbers with the appropriate names and other relevant information. Much of their historical data has inaccuracies, including numerous customers that they no longer service.

The company has been discussing a merger with another company, which has a different set of customer data on different databases and platforms.

They installed a relational database to hold some of the more current customer information and they joined the customer data to the look-up tables to provide a view of the customer in a format that their users like today.

They could just load the large volumes of historical data to their RDB (relational database) and perform the same joins, but because they have data integrity issues, they would populate the bad with the good. They need to address the missing or long-defunct customers that have no bearing on the data that they currently use.

They resolve discrepancies with "on-the-fly" fixes to the new RDB based on business knowledge from their Customer Relations and Sales departments. They have not integrated some of the newer, advanced data cleansing technologies, so they are performing asynchronous data scrubbing.

If they clean out the defunct customers, they have less information and smaller volumes of data. However, they also want to use buying patterns, product sales, and market analysis such that some of the defunct information might be of value to them. They face these issues in such a scenario:

- The sheer size of the data
- Multiple forms and formats of the same data
- Different users of parts of the data, making a universal delete or adjustment difficult
- New data constantly being added so they can't "stop the presses"
- Many manual adjustments

If this sounds like your environment, then rest assured that you are not a pioneer in this space. Things that seem trivial to the non-technician can be immensely time-consuming and complex for those who support you. What is required to find a record on a tape? The tape itself has to be physically mounted. You just have to hope that you have the correct tape volume because there is no fast-path to identifying where the record actually resides. The records on the tape are written sequentially and must be processed the same way: Read the first record; no, that's not it; read the second one; no, that's not it either; read the 250,000th; no, that's not it! Now, perform an action. You go to the next request and perhaps find that the record you need has already been passed, so you have to rewind and start over. This is just basic data processing.

What if you cannot accurately match customer records from different sources for any number of reasons? You may have switched systems and stored the data in different ways. It is very common to see online systems in which the data has been stored with terse, encoded strings representing business information that you pull out programmatically. You might have stored your customer data as a string in which the state, county, district, blood type, whatever is represented in a format such as A993029987534111DEFG. In positions 6 and 7, you store the state key. The value 29 represents the state of California (or is it the state of confusion?). Positions 16 and 17 keep departmental information. In our hypothetical case, 1D is the Catalog Sales department.

This storage technique is often referred to as using "intelligent keys" to store lengthy information in a more condensed format. What if you no longer had Catalog Sales, or the state code for California was changed to 2765 because of a new system you brought in? What if you decided to lump all former catalog sales into a new, generic bucket?

On top of all this confusion, we add the normal processing errors from data entry or other input mechanisms. I may have MR. J SMITH, MR. JOHN SMITH, MR. J. N. SMITH, MR. JONH SMIHT…and they are all the same person. He may have moved three times in the past five years and have changed jobs twice. Maybe it's a woman who has opted to change her last name or has decided to hyphenate both her last names to retain some family identity.

How do you combine the following into a single, coherent household for use in a CRM solution?

Mr. J SMITH A993029987534111DEFG
Mr. JOHN SMITH A993012987534111DEFG
Mrs. LILIAN JONES-SMITH A993029987534111KLFG
Mr. JOHN SMIHT A993011987534112NEFG

The answer is…very carefully! There are significant anomalies at hand here. The long strings beginning with A99 all have differences within them reflecting changes in location, etc. There are different permutations of the names, as well as a misspelling and a hyphenation to boot! Where do we even begin?

It is obvious that we need to automate as much of this data investigation and transforming work as possible. There will always be anomalies and other data integrity issues. We will always face the possibility of needing to merge new or different data sources into our current architecture. Most pressing of all are emerging new business reasons that no one anticipated but that force us to produce different analyses or business information.

Rather than take a longhand approach to these issues, we may decide to look at ETL (extract, transform, load) tools. The purpose of such offerings is to automate the steps and to offer sophisticated functions that allow us to somehow come up with a "JONES" household based on the gaggle of records shown earlier.

ETL tools allow us to construct documented and replicable processes/steps to be used over and over. They also allow us to add new information sources or modify the steps so we might have a faster turn-around in processing changes. Along with all these processes, we have to deal with the metadata associated with the information before, during, and after the steps have been completed.

Metadata capture and utilization is critical to the orderly flow and understanding of what has been and what has now become new information. No matter what vendors and tools you examine today, you can't "plop in and use" any of them. It is simply too complex an environment because of the issues surrounding data that we have just described.

Extraction technologies abound, and many that are quite good. If we look at our example of the state of California represented by two different values, many extraction and transformation tools provide a simple way to move data from one source to the other and change its value on the fly. We could take the values of 29 or 2765 and create a California state value for those that match.

When it comes to data storage and assigning values, there are numerous schools of thought. States may be a bad example, but it is one to which many can relate. Because we have shorthand notation for states that is universally accepted (e.g., CA for California), why not store the CA as a value in a state column? Change the data to match the usage whenever possible, and see if you can curtail dragging along old baggage such as cryptic and "user-hostile" values.

Data cleansing may be the toughest component of all and the one that takes the greatest amount of time. On occasion, there are situations in which the data is simply beyond hope. It may not be impossible to cleanse all data properly, but there are impossible situations in which you simply cannot reconcile the values. Let me provide an example.

I encountered one situation as a consultant in which a customer had used a large number of temporary hires to enter data. The business also involved providing a large number of part-time employees to other firms. The temps entering data had to put in a valid social security number for the part-timers being utilized. There was no check-and-balance system, so many temps used their own SSN as part of the input data. Reconciling billing later proved to be nearly impossible in some circumstances.

When I attempted to build a file of valid SSNs matched to the part-time hires, it was nearly impossible to match anything that made sense. Overall data entry for this application was so poorly implemented that field after field had information that had been omitted, wrongly keyed in, or was misspelled to the point of embarrassment. It was literally impossible to ensure integrity for any set of records. Users with some PC tools and little IT skill had originally built the system. There were no consistent values within fields that any data-matching software could grab onto.

The only intelligent approach I could offer was to propose a solution whereby, from the first day of usage of the new system, we could verify and validate accuracy that would be as close to 100 percent foolproof as possible. Retrofitting old information in an automated fashion simply was not within the realm of possibility. The client was not overjoyed at this, but they attempted several alternatives and verification processes only to reach the same conclusion.

Data must be handled as a valuable and critical resource, not just some bits and bytes that get written on disks to be used and tossed later. Customer information is extremely valuable and expensive to accumulate. Business processes change, and along with them the data required to provide meaningful analysis changes. You have to consider how data will be used in the future or how it might be used in the business.

I worked with one customer who had built a data warehouse that was partially implemented and partially loaded. The end users had been given access to the semi-complete solution and had their appetite whetted just enough to begin to play with the data in its immature form. To compensate for some of the shortfall, they had to extract data back *out* of the warehouse. This was used to match to another set of extracted records created by a 4GL with which they had some experience.

The partially completed warehouse became a deathtrap. The changes and embellishments that the IT staff planned were going to dramatically affect the form and format of the existing partial solution. The users were in open rebellion because they had invested considerable time and energy using this hodgepodge solution to compensate for the shortfall. It also turned out that the user's enrichments being done outside the warehouse were not planned for the new version. The parties were at an impasse, and no one was even remotely happy.

Rules about Handling Data

Do not develop partially complete data solutions without a plan to replace them. Nothing temporary should be sacred.

If you have to create stopgap processes for data to get the job done, you *must* be prepared to throw them away when the more permanent solution becomes available.

If you are part of the enhancement process and not within the warehouse group, you *must* be sure that the enriched information and how it is derived are part of the new plan.

If you do not make long-range plans for data and its uses and embellishments, you will always be placed in a "tail-wags-dog" scenario. The more processing you rely upon performed by the server database, the less you will have to rely on the tools that access them.

The Data Side of BI

I am a "server nut." My preference is to create the values that users will access within the server whenever possible. For example, suppose we have columns with a table that can be added or subtracted. Columnar math (e.g., COL1 - COL2) is a level of query function that most users can learn. However, if we have the ability to load the results of COL1 - COL2 as a column, the degree of interaction is substantially easier as suggested earlier.

As shown in Figure 7-2, there is a wide range of choice in the decision regarding where the actual analytics math takes place. SQL can perform some very complex mathematical functions. There are some things that it simply does not support.

Even if SQL supports the operation, should you always permit the users to launch a query and wait for the results? I believe the answer to be "no." If you have performed a decent end-user survey and matched the users' requirements to their skills, and you have created the underpinning data for them, you should know how much data should be created and how much should be allowed for calculations.

The greater the reliance on the tools to perform the math, the greater the burden is on the following:

- End-user skills
- The analytics tools versus the database
- The network
- Security outside the database
- The tools vendor to provide new functions and calculations

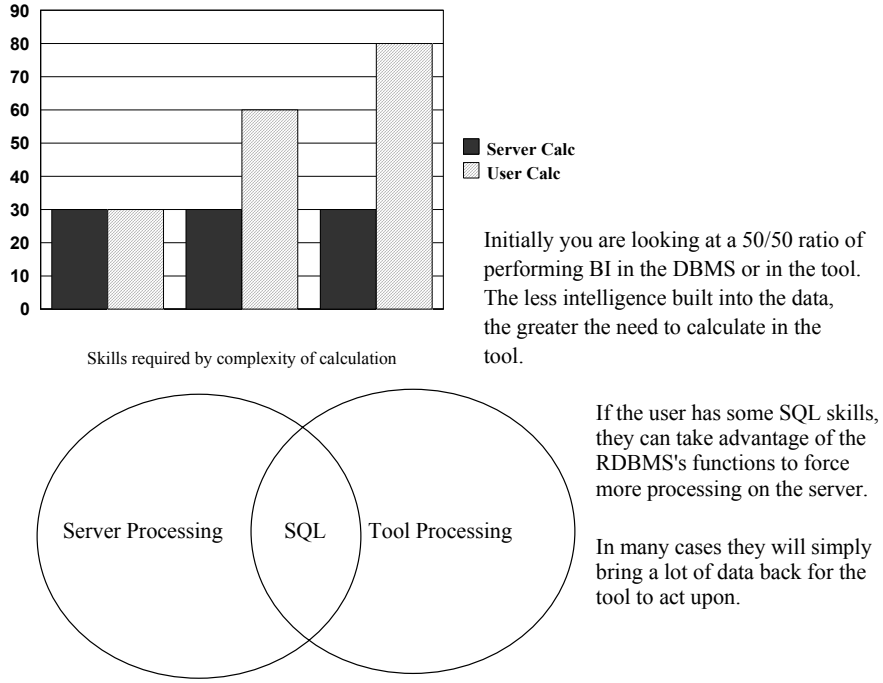

Figure 7-2 Server versus tool calculations.

One goal to set is to maximize the delivery of data at the highest level of aggregation and calculation. In Figure 7-2, the bar chart suggests that calculations provided at the server level will require a consistent level of skill. If one relies on the tools to perform a majority of the analysis, then the skills necessary to create accurate output accelerate.

If you have placed the majority of the values in the server, the users will be able to select columns and produce reports and output consistently after they learn how to navigate the tool at a reasonable level.

As the complexity of calculations increases, the skill level goes up substantially because the burden is now placed on the tool and the end user to perform the math. After the analyses have been pushed out beyond the central point of control, coordination of changes and results becomes difficult at best.

It is not unusual to be held hostage by a vendor when the investment in tools far outweighs the benefit, but the customer is entrenched beyond the point where a simple migration plan will work. There is also the risk of producing multiple generations of the same reports and output. Some of these reports are accurate, and some are not.

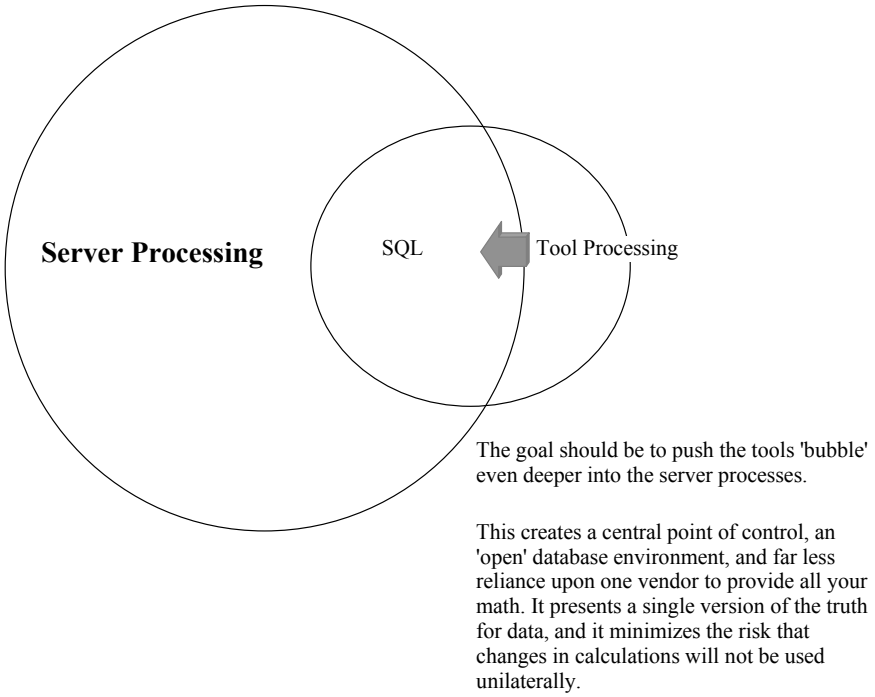

Figure 7-3 Push the analytics processes to the server.

I submit that a server-centric approach (as depicted in Figure 7-3) is the most cost-effective and open environment (yes, I know, I have said this several times now!). I will discuss OLAP solutions in greater detail later, but the concept of centralized processing for calculations is at the heart of OLAP. The greater the *if-then-else* logic required for values to be used throughout the enterprise, the greater the benefit to performing them once and making them available to all tools.

If you were to take inventory of all the values that end users wanted access to and whether they are provided or required calculations to create them, you would have a tremendous amount of "usage metadata" for the BI you were about to perform. You would also understand what analyses need to occur and where.

If you take the server-centric approach for storing your BI values and less dependence upon the tools, the ETL functions take on greater importance. From the perspective of the end user, the point of processing is irrelevant. However, those people evaluating these tools (typically IT) will have to be far more conscientious in considering the mathematical capabilities.

One of the most significant outcomes from thorough user processing surveys is the exposure of any user assumption that may not be true. The hype behind many of the analysis tools often leads the user to believe that many difficult processes will be a snap. They believe that they can almost fall on their keyboards and out pops a complex result. If only it could be so easy. So what tools should you look at and why?

The Analytics Tools

Let's begin with a checklist that came in from a customer with the criteria that they felt were pertinent to selecting a query tool. This real world example (shown in Table 7-1) arrived during the writing of this book.

Table 7-1 Minimal Requirements for Query Products

Query User	P	Comments
Reliable connection to the XYZ platform	1	
Easy to understand/learn SQL user interface	1	
Complete set of SQL functions	1	
Series security/authority levels respected (payroll concerns)	1	
Ability to limit query time by user, group, etc.	1	
Ability to limit rows by user, group, etc.	1	
Editing of SQL statement	2	
Paste SQL statement from previous query tool	2	
Complete set of sorting, grouping functions	2	
Joins—Inner, left outer, exception	2	
Prompt conditions	2	
Draw prompted conditions from Excel table	2	
Output to a variety methods; screen, report, PC file, DB2 table, HTML, XML, etc.	2	
Ability to import special calendar	2	
Ability to check status of a query, cancel a query, etc.	2	
Collection/table metadata availability	2	
Recognizes CYYMMDD date format	2	
Good online help, tutorial, examples, etc.	2	
Is there an add-on to Excel that allows queries to be run/ refreshed through Excel?	2	

Table 7-1 Minimal Requirements for Query Products (continued)

Query User	P	Comments
Connection testing outside of user interface	2	
Input from a variety of sources (individually) DB2, Excel, dbf	3	
Batch over interactive retrieval	3	
Canned, non-modifiable queries	3	
Reporting functionality		
• Can you report totals of groupings only, eliminating the detail lines?	3	
• Can you compute percentages off of totals? How?	3	
Complete scheduling function • Server or client level scheduling?	4	
If Web/browser based:		
Fast?		
Limited functionality?		
Web server requirements?		
Load balancing?		
Bandwidth requirements?		
Manageability issues:		
Robust methods for controlling user IDs, permissions, views, tables, etc.		
Robust methods for managing system resources, servers, CPUs, disk, etc.		
Metadata about connections, ports, usage, etc.		
Ease of upgrade path		
Support		

As you scan the list in Table 7-1, how many of these options are related to what a user does to produce analytics results and deliver BI information? The average RFP will be very IT-oriented in nature. It is easy to ask for global elements such as speed, security, backup, etc. The many aspects of BI user functions are more difficult to address. What does *fast* mean in the item labeled "If Web/Browser based"? I haven't seen a single vendor add a line that suggests, "Our web technology is quite easy to use, but it is slower than a glacier in returning the results."

In most cases, IT just wants to users to stop complaining and get off their backs. The normal BI exercise is to post some data in a relational database and help the users select a tool to attach to their data.

There is seldom a provision for failure. What if we cannot produce the results? Is this due to inadequacies in the data or the tool we have selected? What is the plan to back out and take another tact if we have a problem?

End-User Assumptions about Tools

If you have ever tried to create a formula in a spreadsheet, it may be something you can relate to in learning a BI tool. Chances are good that the first attempts at creating a result with a formula in a sheet took a bit of effort. By the time you entered a value in the data entry cell and the others actually changed instead of giving an error message, you may have been exhausted. It was time to declare victory and go home! In many cases, the math used in a formula is not verified for accuracy. Because the formula works, the results must be correct!

In a previous chapter, we covered user segmentation and skills areas. It is extremely important to make absolutely *no* assumptions about any tool and the user's abilities to learn and drive it. Can any user be taught to drag and drop some columns onto a template and create a new report? It's amazing, but the answer is "no"—if we mean *any* user, literally.

What should we, therefore, look at in different tools and why? Are there features that have greater weight and importance than others? What should we expect a tool such as a query and report writer to be able to deliver? Has there ever been a tool that a large majority of users have taken to? Yes, the spreadsheet.

The Spreadsheet's Role in BI

Let's begin with the success of spreadsheets to deliver BI. Because most of the world's BI analysis is performed on spreadsheets to this day, is there something "magical" about them? The answer is "yes." There are many *magical* things about them:

- Spreadsheets allow the total free-form entry of data.
- They are very much WYSIWYG (What You See Is What You Get) unless they have been highly modified and the normal view has been changed and customized.

- The math (formulae, etc.) is relatively easy to understand.
- The user can play with the sheet and his data in privacy, and no one will know what he is doing or what he is screwing up!
- The nature of the spreadsheet lends itself to BI due to the free-form nature of how the user can address cells and ranges as opposed to rows and columns.

Let's look at an example of interacting with data from a spreadsheet versus a traditional query and report writer. Figure 7-4 illustrates some of the subtle differences between them.

"The sheet" knows no structural processing boundaries and restrictions such as rows and columns. You can add, subtract, and perform any combination of calculations on ranges, cells, and anything that can be held in the sheet. The "elasticity" of a spreadsheet is often assumed to be available in more traditional row/column reporting tools. The biggest problem with spreadsheets is getting data into them. One of the key requirements in any query tool evaluation and checklist will include:

Item nn**:** Interoperation with Excel (add-in, etc.)

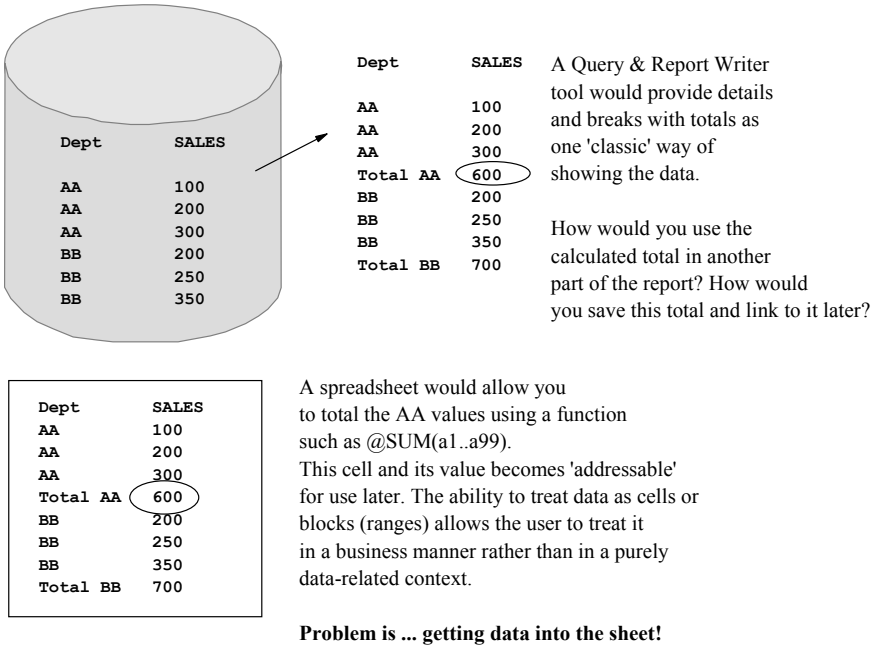

Figure 7-4 The spreadsheet environment.

The users want to know that they can take whatever they have been able to create and somehow get it back into their old, familiar friend—the spreadsheet. You can take a cell from this sheet, a range from that sheet, a calculated value from another sheet, and so on. You literally build the solution/result based on how you think or how processes flow. To mimic such processes in traditional query and reporting tools is very difficult, if not impossible.

The Three Major Categories of BI Analytics Tools

Here again are the three tools categories:

1. Traditional query and reporting
2. OLAP
3. Data mining

The first two pretty much provide answers on questions we feel we need to ask or for values we have determined we need to produce. Users may say, "I may not know how to create a report for profitability by product for all segments, but I know I need it. I may not like what I see when I get it, but it is what I asked for."

Data mining is a very different beast. There are numerous algorithms available in different tools, and the most intriguing are those that perform *data discovery* types of operations. Here, users say, "I need a tool that simply makes me think outside of the box. I need something to show me what I *ought* to look at!"

If you recall the tools pyramid, the largest segment of the user population was supported by query and reporting. One way to identify an OLAP solution and its applicability over traditional query solutions is if you see a pattern of queries that are similar but slightly different. Each level of detail is a slight variation of the previous. You notice that users are attempting to mimic drill-down analysis by launching query after query. Such patterns may be better served with an OLAP solution.

The same goes for data mining. If your users are creating and executing query after query in the hope of identifying some anomaly or aberration in the data, mining may be able to point out such results in a fraction of the time. Remember that the quest is to use BI intelligently and to change the business, not just get your tools investment back by the sheer volume of the queries that you execute!

Query and Reporting Tools

There have been so many tools and so many vendors in this space that it's difficult to know where to begin. These common features and functions are considered minimally essential from any vendor:

- Supports the leading databases and most common file formats
- Features an easy-to-learn and easy-to-use GUI
- Generates standard SQL
- Handles large queries and result sets
- Uses client/server (two-tier, three-tier, or more topology) for full client
- Has thinnest client possible
- Offers 24x7 operation with fall-back, backup, and recovery options
- Provides tracking, tuning, performance, and governing
- Offers output to a multitude of file formats (e.g., PDF, Excel, HTML, etc.)

I'll provide a more comprehensive checklist later. For now, suffice it to say that the demands on any query and reporting tool are extreme. The ability of the tool to place results on the glass (the screen) or other output device is the ultimate determining factor of its success or failure. How do you determine ease of use? What constitutes a "fast" query?

Do you have one or more report writers in-house right now? What is your satisfaction level with them? If you are unhappy and considering a change, there are several things to consider prior to going through the challenging process of tools evaluation:

- What are the specific problems or shortfalls with the tool(s)?
- Are issues primarily data related, or are there true inadequacies within the tool?
- What are you going to do different this time if you select a new tool?
- What long-term, strategic features and functions are required to increase satisfaction and not repeat past difficulties?

Here's an approach that I have seen work well: Take a set of the data in its new and final form, and set up a proof of concept (POC) with the end users to see if their analytics can be delivered from the data in a manner that they understand. Are they confident that they can attain the results they require?

The worst approach most often taken is using existing reports as the basis for testing and verification of results. People who use this technique will justify it by saying,

"Well, we have to be able to replicate what we have today before we move on!" What is this supposed to accomplish? We take our known data—reports we can already produce—and replicate them. If our data is in the format we require and we are already producing effective reports, what is the point?

Another difficult "landmine" area is when users insist that a report or chart format *must* look a specific way. I mentioned a real-life example of this earlier. It often takes eons to get some small, subtle effect produced in a report or chart after the initial (and quite readable) output was produced in no time at all. This links back to the bad idea of letting users form preconceive notions about what their analytic tool should allow them to do.

Think of what your creation might be if you wanted to develop your own report writer and go to market. What things would immediately come to mind that are necessary to make your product appealing to others? If you decided to do so, you would certainly have plenty of company from the many who have gone before you.

As I stated earlier, this is not a reference for products or vendors, but I have used some things to evaluate report writers and other tools that may be helpful in your decision-making processes. Concepts like ease of use and flexibility are very subjective. Any BI software vendor surviving in today's marketplace has to be doing something right, but which vendor has the proper solution for you? Ask these questions when choosing a vendor:

- What is the overall design strategy for the product(s)? Why was it created, and what was the target audience?
- How does the solution meet the needs of our user segmentation, and why?
- What databases and files does the product or suite access, and how?
- What data access technologies (e.g., ODBC, native access, etc.) does it support, and why did the vendor select them?
- We want to see where results are stored, how they are buffered, and how the product uses memory and temporary storage. Would you please diagram how the product(s) access data in our supported databases?
- What platforms are supported, and why? What are the current versions and plans for each product per platform? Why are you making these enhancements?
- What languages are used in the product?
- Walk me through the steps and processes from installation to user hands-on. What are the "tough" parts, and why?
- Who are your competitors, and how do they stack up? Can you list the top ten reasons why we should consider your solution over others?

- How do you scale? How do you handle backup, recovery, tracing, performance measurements, and other system-like elements?
- Would you walk us through your support structure and describe how we get support for our IT users, non-technical users, and heavy technical users?

The catch here is that, unless there are significant roadblocks to selecting the vendor's solution, we should not rule one out due to some IT-based issue. Another catch that arises too often is that many prospective customers do not wish to involve their end users in the seamy details of technology, so they are kept out of the process.

End users will want to know how all this ties together and how their data is going to be handled. They want to know how lengthy the process is and whether there are any stumbling blocks to processing their queries efficiently. One thing users should be intimately involved in is the process of which data elements are stored and how users will expect to use them.

Time and Date Elements in Reporting

One thing all businesses look at in an infinite number of ways is time. Comparisons will need to be made based on YTD, Last Year versus This Year, special periods such as promotions or limited time offers…you name them. There will be many requirements for segments such as a rolling average of the last 12 weeks, and this is based on the current date. Sometimes, you'll need to specify a series of dates/times.

There will be aggregations such as YR, MO, QTR, and others. Most query/reporting tools handle time/date options fairly well. If your users seldom access data at the granular level of detail, then might it not be better to store aggregated data and reduce the average result set? Oops! We're getting back to data-related issues.

The vendor should be able to walk you through how aggregations, dates, times, and other collective measurements are met. If there are better ways to store the data that would optimize the use of their solution, then the vendors should be forthcoming with it. If vendors claim that "We can do it all," you should meet that claim with healthy skepticism.

A Vendor Evaluation Concept

Here's my advice on evaluating vendors:

1. Set a plan in place to perform the most thorough and fair evaluation possible. To do this, set an evaluation process in place that provides the vendor with a well-defined and understandable methodology.

2. Have your database designs, user processing requirements, and other pertinent information ready in a document that you present to the vendor in advance. If your business is important to any vendor, you should expect them to provide senior, knowledgeable technicians and sales individuals to present and demonstrate their wares.

3. Set up the evaluation/presentation to encompass IT, end users, and combined areas in such a manner that all parties clearly understand that their input and requirements are part of an enterprise decision. If you take a team approach, then you'll uncover any unpleasant surprises or areas where you might work together to make your selection more productive.

Too many BI solutions are selected based on a hasty RFP and multi-vendor "beauty contests" or some biased decision from an individual or department with influence. This is also how many businesses wind up with a plethora of tools that perform similar functions but easily could have been provided by one.

In Table 7-2, I have included an expanded version of the query and reporting checklist that might be a bit more useful for your evaluation. One thing that must be considered in your selection criteria is how well the tool fits your end-user population and the ability to deliver the proper BI information.

Let me refer to one BI vendor's product architecture, without naming names. This vendor offers a full client, a thin client with some limitations compared to the full client, and the ability to switch from thin client to full client on the fly for users who occasionally need more. These three configurations cover the gamut of users who would actually create or interact with reports and queries at various skill levels.

This vendor also offers some significant "push" capabilities that enable output to be directed to a user or series of users based on setting triggers, events, etc. The recipients being pushed to may be technically savvy (heavy) users or those who receive a report over the net and have only a browser installed and no option to perform any work at all.

It is as important to evaluate the ability to deliver information as it is to create it. If a tool allows you to create one report for multiple users or customers and burst it into sep-

arate reports, that is a better option than having to create a separate and unique report for each user and then keep them all in sync when things change.

Table 7-2 A Proposed Reporting Tool Checklist (Recommended Requirements for a Reporting Solution)

General Section Topic General functional areas for IT and end-user information	Priority	**Comments and Descriptions** Put in as much information as deemed necessary to reply to the topic. If extensive comments are needed, please number and footnote.
Company information—market position, etc.		
Consultant reviews (Gartner, Meta, IDC, etc.).		
Countries where you are doing direct business?		
Ten references with at least five in our industry.		
Largest customer and number of users.		
Number of years in business, market position.		
Business Partner Channel? If yes, top five you recommend.		
Public or privately owned?		
Collateral available, URL, etc.?		
System information		
Databases supported?		
OLAP sources supported?		
Topology—two-tier, three-tier, etc. Explain or provide configurations.		
Platforms supported—server components?		
List server components and functions.		
List prices of server components.		
Platforms supported —end-user components?		
List user components and functions.		
List prices of end-user components.		
Number of versions/releases—current version(s)?		
Library management: How are objects tracked?		
Metadata strategy to server, user objects?		
Security architecture: database, user objects, etc.?		
Migration between platforms? How?		

Table 7-2 A Proposed Reporting Tool Checklist (Recommended Requirements for a Reporting Solution) (continued)

Maintenance and support options?		
Local support? How do we engage them, if yes?		
Load balancing?		
Scalability: How is this handled?		
Backup and recovery strategy?		
Tuning and performance is done by?		
Open API supported? Languages?		
National language support: Which ones?		
Web servers supported?		
Connectivity to system(s) options?		
End User Functions		
Describe the EUI (end-user interface).		
Wizards or guides supported?		
Complete set of SQL functions?		
Ability to display SQL from GUI?		
Ability to edit generated SQL?		
Ability to undo edits: How many “undos” supported?		
Governing functions—ability to limit rows by user, group, etc. Administration or user defined or both?		
Cancel query options: How are runaway queries handled?		
SQL level supported—restrictions if any?		
How do users manage objects? What information is provided on usage?		
Describe how data is returned to the tool and where the processing takes place for full client? For web-based client?		
If internal or proprietary data is created (e.g. micro-cube), where does it reside? How is security handled?		
Exceptions, flags, triggers supported? Describe.		
Imbed graphics (logos, etc.) in reports?		
Full range of fonts supported?		
User can sample NN rows for testing? How?		
What result set sizes do you recommend for decent response times? 100K rows? More? Why?		

Table 7-2 A Proposed Reporting Tool Checklist (Recommended Requirements for a Reporting Solution) (continued)

Currency conversions: How are they handled?		
Version support for queries? Reports? How?		
Import of queries from other tools: Which ones?		
Migration from other tools to the new one?		
Complete set of sorting, grouping functions?		
Joins—full, inner, left outer, exception, etc.?		
Accessing data from multiple sources: How or recommended solutions?		
Join relational or other data with OLAP data? Which ones? How?		
Prompts for user's input: How many? For which options?		
Describe Excel support.		
Output options; screen, report, PC file, DB2 table, HTML, XML, etc.?		
Users can share or restrict objects? How?		
Users can concurrently write back or update which data sources? Describe.		
Ability to import calendar from which tools?		
Ability to check status of a query, cancel a query, etc.?		
Describe how users view source data information (schema, star, masked from the users, etc.).		
Date formats: Which are supported?		
Time dimensions supported (year, month, week, custom, etc.)?		
Good online help, tutorial, examples, etc.?		
Supports star-schema data format? How? Other structures supported?		
Connection testing outside of user interface?		
Input data options—DB2, Oracle, Excel, dbf, Access, etc.?		
Ability to create one report and burst for multiple users? If yes, how is security handled?		
Batch and interactive processing?		
Canned, non-modifiable queries?		
Ability to control the user's options and interface?		

Table 7-2 A Proposed Reporting Tool Checklist (Recommended Requirements for a Reporting Solution) (continued)

Personal portal option?		
Describe reporting aggregation options: Totals Percentage of total Averaging Sum, Max, Min, Mean Others . . .		
Calculate columns in addition to existing data values?		
Rotate/pivot views of reports? Charts?		
User products are "aggregate aware"? Describe, if yes.		
Customized reporting options (WYSIWYG)?		
Formulae format (Excel-like, etc.) to extend calculations?		
What advanced analytics are supported?		
Complete scheduling function: Server or client level scheduling?		
Use of "triggers" to kickoff a process or report? Describe.		
Country languages supported and how (NLS, etc.)?		
Multiple reports on same screen? How?		
Describe charting options: pie, bar, etc.		
Report and chart on same screen at same time?		
Executive information system options such as dashboards, alerts, etc? Describe, if yes.		
"Lite" OLAP functions natively supported? Describe, drill-down, slice, etc.		
If OLAP-like functions available and report/chart on same screen, do they drill in sync with each other?		
Estimated time to learn for users: casual users, ad hoc, heavy analysis. Describe.		
Describe the typical (normal) processes for a user getting to a skill level: casual, moderate, heavy.		
Training and education options available: on-line, CBT, classroom, etc.? List courses and costs if applicable.		
Full and web (browser) clients supported? If yes, describe differences.		
Can full and thin client versions be combined? If so, how?		

OLAP Tools

The term *Online Analytical Processing (OLAP)* seems far too technical and obtuse for the power and business information that it delivers. One attribute of many OLAP solutions is their use of proprietary, multi-dimensional databases to store their results. What is the big deal about OLAP?

There are many permutations of OLAP-like products: MOLAP, ROLAP, DOLAP, etc. I don't like to make judgments about which technology is better, because that is subjective and has little to do with your business requirements. I do tend to favor a pure multi-dimensional (MOLAP) solution overall for many reasons, including superior performance.

If you adhere to the philosophy that much of BI is about "the math," OLAP should emerge as one of the most powerful solution components of your enterprise's strategy. Storing data in multi-dimensional formats typically provides phenomenal processing speed. But what about the calculations involved in creating each value for the OLAP intersection points? Let's consider a simple three-dimensional model like the one shown in Figure 7-5.

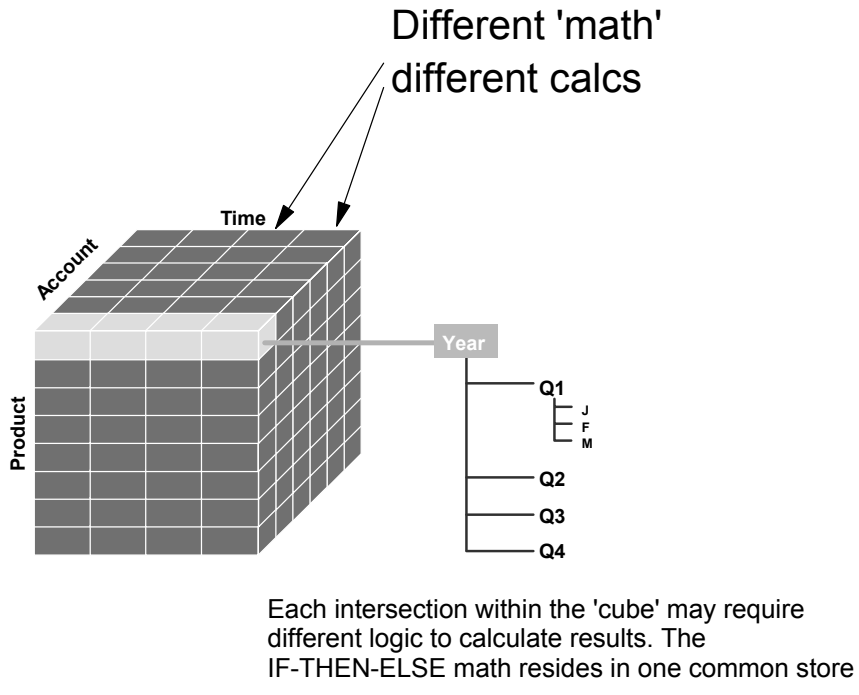

Figure 7-5 The "math" contained in OLAP.

OLAP solutions primarily store numbers and very little text. They are built for speed. Many users wish to view their data from different levels within the organization, by different time dimensions, and by other metrics. The idea is to get in, ask a flurry of questions, and get out.

Movement within a cube involves drilling up or down from one level of a hierarchy to another or looking at a slice of a cube from different metrics—for example, looking a single product from a multitude of angles.

The names of the various dimensions and measures are typically translated into business terms to which end users relate, and not to the original source data names. However, if the underpinning warehouse or mart is set up to support OLAP, renaming may not be required.

If the user needs to go to a deeper level or higher detail than the OLAP data supports, most OLAP solutions allow you to set up a drill-through capability (as illustrated in Figure 7-6). Drill-through entails using the intersection points for the level of data being viewed to build the appropriate query (SELECT, WHERE, etc.) and accessing the underpinning source data at a level normally not loaded into an OLAP "cube."

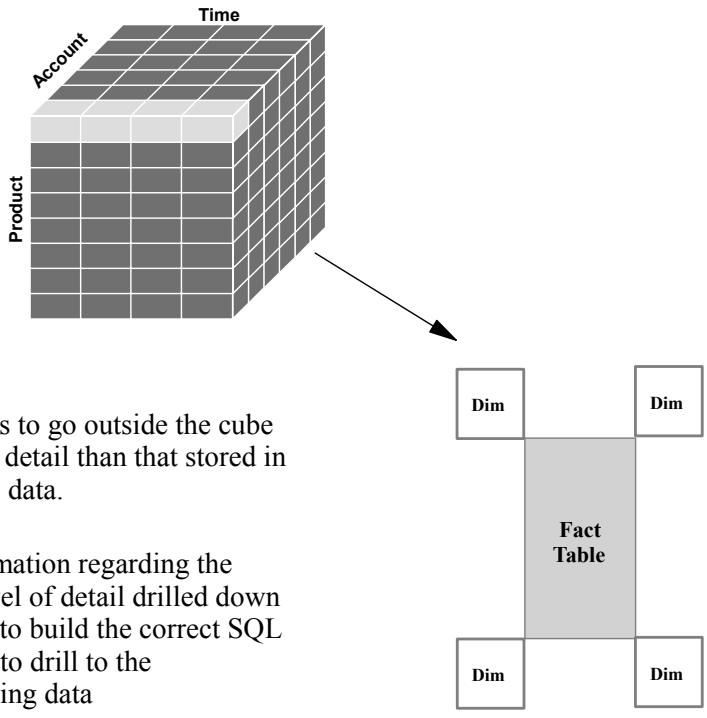

Figure 7-6 OLAP drill-through.

Let's look at one of the most—if not *the* most—significant elements of what OLAP delivers to the users and the enterprise. Not only do you need a single version of the truth for data (e.g., a warehouse or mart), you absolutely need to deliver a single version of the truth for the analytics themselves. This goes back to "the math."

If we have to calculate a series of values that may vary based on other aggregations or other values (IF-THEN-ELSE), we have the potential for several errors in the traditional spreadsheet/query/reporting environments.

- Every permutation/combination may require a different query with different SQL statements.
- We may have to create a separate object (report, etc.) for different combinations.
- If we update the objects and their data, we have to coordinate among all the users.
- If we update or change "the math" in the objects, we have to ensure that all copies and versions are also updated.

It is the last bullet that places us at greatest risk. For example, suppose we calculate a value based on the percentage change from time period 1 to time period 2. If the percentage change is above some threshold, we have to apply a different factor or rate. If the change comes from a table, it should be picked up automatically. If the rate is entered as part of a report's calculations, we have a problem. What if the actual formula changes and we have disseminated dozens of reports with the old formula?

Thus, OLAP applications place all your calculations in one basket, so to speak. Creating an outline of the business analysis that has all calculations in it does this. If a formula changes, it is changed in one place at one time.

How serious can it be to have one's math a little "off"? One of my customers was fighting a hostile takeover years ago. In the midst of the horrific fight, they rolled out a spreadsheet that was incredibly inaccurate. The sheet was targeted for use by their buyers (this was a retail shop) and resulted in massive over-purchases in several key areas. The customer was absorbed in the takeover. I don't know how much of an effect this over-purchase situation helped to seal their fate, but I know it had significant impact according to my insiders. An OLAP solution possibly could have stopped this by allowing the customer to either change the math immediately or to take the incorrect OLAP results offline.

Every BI tool seems to tout some "OLAP-ness" about it. Attributes of OLAP solutions include the drill-up and drill-down and "slice-n-dice" functions. This does *not* make a product an OLAP tool. Here are some attributes of "true" OLAP solutions:

- Multi-dimensional storage built for speed
- Condensation of a multitude of queries into a single source of results
- User-driven processes and building functions not technician-dependent
- Single point of mathematical calculation
- Open to a variety of front-end tools
- Extensive mathematical operations with the ability to use aggregates from other dimensions and other calculation-based usage
- Based on business rules, not database-dependent
- Drill-through capabilities to a wide variety of data sources

One thing that OLAP solutions often provide is highly enriched data. The combination of all the calculated values and the massive number of IF-THEN-ELSE combinations may only exist in the OLAP data. Many of the correct values may *only* exist in the OLAP data and nowhere else.

Having everything in a centralized data store is not a bad thing. It is a better approach than all the hit-and-miss attempts to create the truth in multiple objects in multiple locations. We are so keen on doing this with our data, but we are a bit more cavalier when it comes to the results we produce.

All Your Queries in One Basket...One Way to Look at OLAP

Think of the number of queries and their permutations that it would take to create all the combinations of queries, their math, groupings, and exceptions that you would store in OLAP. How many queries would take to mimic drill-down scenarios? What are the chances that your users will get them all correct? What is the effort required when changes are required?

What is the cost to the enterprise if you perform *inaccurate BI calculations*? It is probably the one good thing about not penetrating the upper echelon with BI information. When you provide a true decision-maker with inaccurate information, you can have a huge impact on the business.

We see surges of interest and efforts in providing executive information systems or executive "dashboards." Everyone wants to provide powerful yet simple-to-use applica-

tions for executives. The problem has historically been that the functionality of the tools and their potential are seldom attained. We'll discuss Complex Enterprise Analytics in a bit. Let's get back to OLAP and its enriched data.

Given any data source, what are the chances that you will effectively ask all the questions that need to be asked and answered? What is the chance that you would be able to "think outside the box" and come up with creative and never-before-asked questions?

Data mining is designed to deliver straightforward answers to complex questions. It also is a means to uncover information that would have taken eons to find using standard query and comparisons.

Data Mining Tools

The most important element of any mining tool is its ability to directly access your data warehouse or marts, assuming that you have these in place. Mathematically, the mining tool should be able to substantiate its algorithms to the statistical "literati" within your organization. Business-wise, it should be able to deliver information that is *new news* to you.

Data mining is *not* a generic tool for the masses. One mistake that enterprises often make is to ask for a demo of some mining technology, thinking that it is some clever little tool that a number of individuals will employ. They use their own understanding, or lack thereof, as the paradigm for acceptance or rejection of the tool and its applicability for their business.

One of the most common and, unfortunately, high return areas for mining is in fraud analysis. As a recipient of a mining run, I may not understand the data that well and I may not understand the algorithms, but I am very concerned to uncover a source of $20 million of fraud being perpetrated.

So what approach should be taken for evaluating data-mining technologies? As I mentioned in the first sentence of this section, make sure that the mining tool can access the entire range of data that you intend to analyze. Many mining tools are extremely compelling and have lots of promise.

However, if the tool under scrutiny requires an extraction of your data because it cannot handle a large volume of data, you are always dealing with subsets of data and samples that may not represent the entire spectrum of what needs to be analyzed.

The Data Mining Process

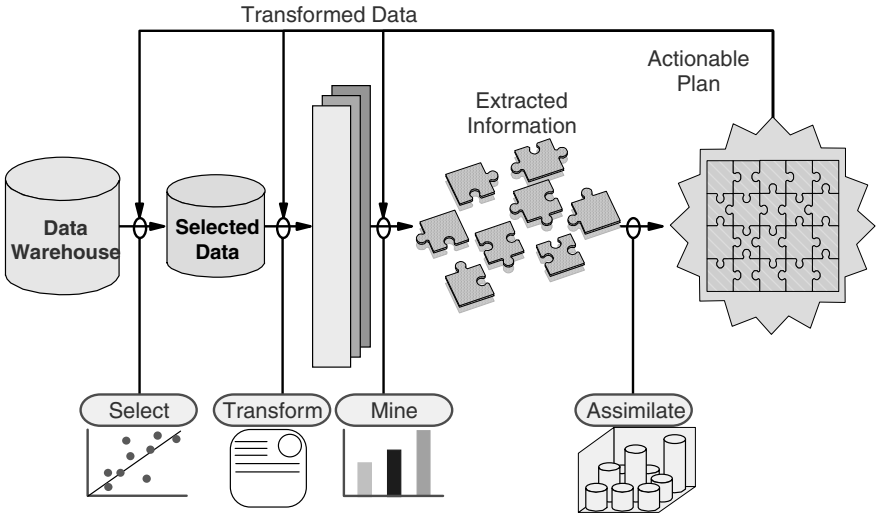

Figure 7-7 Data mining processes.

In Figure 7-7, I depict one approach to data mining. The most important step in the process is the selection of data. *Note that this does not indicate extraction, but selection*. If you have a requirement to subset your data for mining, it should be done via SQL, not with a file extraction.

Some mining solutions place limitations on the size of the dataset on which they can work. Technology limitations on mining due to file size will only worsen with time. I seriously doubt that the volume of data that you need to access will decrease over time. The volumes of data that we all deal with have increased at phenomenal rates, and this will not change.

Sampling techniques are certainly acceptable in many statistical methods, and there is nothing wrong with applying them to certain mining algorithms. However, it should not be acceptable to be forced to sample or sub-set data because that is the only way the tool can function.

Another key element to evaluating a mining solution is the ability to store the results in a format (e.g., an RDBMS) that can be used by other tools. Can you merge the results into your existing data warehouse or data mart environment to enrich your sources of information?

Some technologies exist that allow you to mine OLAP data as well. Remember that many OLAP sources contain data that exists only within the OLAP database. What if you had a 20GB "cube" built? Do you think that anyone has ever looked at all the values in the data? Could there be some hidden trends, clusters, or patterns that might be significant to the business?

Some of the techniques you will encounter in mining solutions include:

- **Clustering:** Neural and demographic
- **Market Basket Analysis:** Single trip and over time
- **Classification:** Neural and demographic
- **Prediction:** Back Prop Neural and Radial Basis Functions
- **Time Series Analysis:** trends over time

Clustering algorithms and results are one of the most commonly used and significant elements of data mining. The way we normally segment populations is to use attributes and rules that we believe to be true. We issue queries based on what we know, and when we find something slight askew, we may "go fish." A mining tool often uncovers and aggregates groups of which we were totally unaware.

If you do any homework to look at others who have utilized data mining tools, you'll discover that a single successful mining application pays for the cost of software and effort many times over. Significant results are typically "the norm." The problem is that the evaluation or exploration of mining technologies seldom deals with the business results we may expect. More often than not, the evaluation centers on how much the reviewers feel comfortable with the tool.

Indeed, most evaluations are done based on what the users see in the tool. They may ask, "Can I possibly use and understand that?" Mining technologies simply are not used in the same manner as query or OLAP tools. You are trying to discover things that you don't know, not validate things you suspect or know to be true. This information may be priceless and should far outweigh your desire to be able to mine within your comfort zone.

I had an IBM Data Mining Specialist walk me through some of the significant applications in which he had been involved. I have had a reasonable amount of statistical training, but the math behind the solutions we discussed was certainly out of my league. However, it did not matter because the results and aberrations, clusters, and patterns certainly made sense.

A Challenge: "Mine" the Current Success Stories

Take some time to assign an individual to exploring mining technology use in your industry. It should become apparent without massive hours of research that many have used data mining technologies to deliver significant new results.

You may be able to arrange a demo or discussion with other mining technology users outside your industry (non-competitive scenarios). If you are involved in heavy customer measurement scenarios, then any industry will do. If you are in an industry that may involve significant exposure to fraud, others may be able to show you how mining made a huge difference.

If you cannot find others who have walked the mining walk before, you may be in a position to gain a significant advantage by employing it first. One thing you will find is that mining solutions are typically heavy on ROI, and deployment to the masses is not required.

There is not a single enterprise on the planet that cannot use data mining in some significant capacity. The key is to look for the business application, not the technology. I have seen data mining used in the gaming industry to uncover groups of individuals who potentially might spend (maybe the correct term is "blow") more money at the tables, but needed incentives to travel.

Advanced Analytics—Executive Information Systems (EIS)

Every executive on the planet wants to have access to data that helps him make decisions, spot trends, and react to situations in a rapid manner. Ideally, he would like to be able to spot trends or predict events before they happen. Some executives wish to be notified only when and if some event occurs so that they do not have to wallow through reams of data to discover areas of concern.

Many executives would like to comfortably play "what-if?" scenarios with data and see if some changes might offer better results. There seems to be an attraction by many to the concept that engaging in massive hours of analysis is a good thing.

A major source of Business Intelligence information published some charts a few years ago that showed a significant amount of time is spent collecting data and performing analysis when an enterprise does not have strategic analysis applications. I am not able to include their entire work here, but their conclusions are worth discussing.

According to their findings, an enterprise that implements strategic BI applications will spend significantly less time performing analysis and far more in the decision-making process. As a result, the ability to formulate a plan and act on it improve significantly. You must avoid the "paralysis of analysis" syndrome. At the executive levels, you need to provide "think time," and it must be based on accurate, pertinent information.

EIS solutions have been around for many years and have been created using all BI technologies that have ever existed. Some solutions have openly stated that they are an EIS, period. If information delivery to executives can make such a difference, why aren't they just proliferated everywhere? There are several reasons for the shaky growth of such solutions:

- The technologies are simply too difficult to implement.
- The perception that executives can "drive" the products proves false.
- The time to create the proper output from data access to delivery takes too long; it's too hard to get to the data.
- Real-time updates were promised as part of the solution, and the cycle to update and remain current is unrealistic, so little or no "think time" improvements are offered.
- There was little change in the business after delivery, and the executives rejected the solution over time as being of little value.

There is nothing invalid in wanting to implement an EIS. One of the inherent problems often seen is that the BI tool selected doesn't provide a "packaged" EIS, and the executive's results and views must be created. E-gad! That is a horrendous amount of work. In order to create sophisticated, complex results, which are the typical executive request, one must be a tech-heavy user. Most of the competent users are not familiar with the information requirements at the executive level, and it can take a long time to deliver the results.

The executives, on the other hand, do not wish to appear as lummoxes and often simply do not know what to ask for that is within the realm of possibility for the sanctioned tools. It can seem like a game of "Guess what I have behind my back?" on the part of the executives and the developers. Neither knows what the other one really needs or is capable of doing.

Today, there is a growing trend within many vendors to provide toolkits, which include canned and customizable analytics with metrics such as KPIs (key performance indicators), dashboards, and more. They have made significant investments in learning what

key executives look at and why. As a result, you can deliver key business metrics in a shorter time than ever with less guesswork.

One reason for these enhancements is that the brass ring of an EIS's value has never gone away. The lure of providing top management with invaluable information cements a vendor in place for a long time. Another reason is that the vendor transcends above the normal BI "noise" and frantic activity to the rarified air of being a strategic application provider, not a tactical and therefore replaceable, solution.

Within customers, there is also a growing awareness that this BI "stuff" is a bit harder to implement than many imagined. There are limitations within any tools that you may acquire. However, if you are looking at creating your own advanced analytics algorithms and formulae for executive use, you have a huge uphill climb ahead of you. Is it not more productive to deliver results based on the measurements and criteria that their peers and modern business schools have validated and have been built in to a product suite?

One approach that some have taken is to create a data mart with executive content in it and utilize their existing tools to deliver key information. However, this assumes the executives will actually take the time to scan all the results and look for pertinent information. The flaw in this plan is that either the executive develops some product skills or you have to pare away many functions to give the interface a usable set of functions.

A better approach would be to establish key measurements and thresholds that would get the attention or should be addressed and deliver information based on these "triggers." Many executives know the top *nn* things that they wish to see and what constitutes a significant piece of good news or bad news. Deliver what they absolutely *must* see today, and refine the application and values later.

Years ago, I helped a customer develop a system that insiders dubbed the "electronic pessimist." What we developed was a set of metrics that was shown only if key measurements went awry or something was horribly amiss. It was predicated on the need to see things in "doom-and-gloom" mode and otherwise ignore them.

The driving force behind this application set was the need to reduce the volume of data and information that the execs were expected to view. This customer was an early adapter of BI technology (*nee* decision support at that time), and they had been doing BI before BI was invented! They knew that their ongoing efforts yielded a gem from time to time, but they wanted to deliver a better set of results to "mahogany row" … the executive level.

Suppose you were a sales executive (perhaps you are!) and a system delivered the following information: "You are currently 30 percent behind your current forecast for this year and 25 percent behind the previous year. The outlook for next month indicates that you will fall to 39 percent behind your current plan." Would it matter to you if it came as a chart, a report, or an e-mail in pure text? This type of news may be better if delivered by carrier pigeon, so no one else could view it!

BI Tools: It Isn't Always What You Buy, But How You Use It

You have a cornucopia of solutions available to you from a multitude of vendors. You also have a responsibility to your corporation to use information responsibly. This includes the creation processes as well.

If you have a suite of power tools in-house and source data that weakly supports them, you have set up an inordinately complex analytics environment with more dependence on the tools than is necessary.

If your approach to BI is to put data into some reasonable format and hand users some tools with the expectation that they will create wondrous results from complex calculations, you are setting a course for limited ROI at best.

The fewer calculations required by the clients, the greater the chance of producing accurate results. The greater the investment in the server side for creating and storing the required analytics data, the greater the productivity of the users. More users with less technical skill will be able to interact with the tools and the data.

In the advanced analytics space, you are faced with a "make versus buy" scenario. Implementation of an EIS is not trivial, and there should be no assumptions made prior to engaging in any efforts to deliver one. You can get away with throwing some BI output on the corporate wall to see if it will stick for those at lower levels in the enterprise. For delivery to those at the top, this is not a good idea.

If you have decided to go forward with a BI initiative, what return on your investment might you expect? Are there ways to measure and quantify BI solutions? How do you measure costs?