CHAPTER 3

The History of Business Intelligence

If you have been involved in data processing for any length of time, this probably all sounds familiar to you. I won't include here my version of a time line chart that tracks the chronology of some product or technology over the years. I do wish to include some of the major milestones of the predecessors of BI solutions and point out their good and bad points. Not everything from the past has been a misstep.

Some of the "eras" we'll cover here may include elements of the environment you have set up for BI today. You may be actively using what might be called a *tool of the past*. This does not imply that you are behind the times or that others have a better plan than you do. BI, like anything else, is what you make of it. If you decide that the best you can do is to provide the end users with an orderly approach to accurate data, then so be it. If, on the other hand, you provide only that degree of commitment because you are not convinced that there is value to BI solutions, then don't be surprised if others pass you in the marketplace.

Those who forget the past are doomed to repeat it. If you have not dabbled in BI yet, learn from the mistakes or experiments of others and perform a quantum leap. One element of many BI failures is the constant search for the perfect tool. Many tools have

been rejected as being too hard to use and continue to be so for the life of their product cycle by some evaluators somewhere.

IT often interprets this as a call to find an easier-to-use tool than the latest selection. How can a drag-and-drop tool be that hard to use? In most cases, what the users should have said was "We can't get the results we need from this tool. We can't figure out how to solve our mathematical problems."

The reality is that many users cannot tie *any* tool onto their existing data and perform the necessary calculations. That is a very different problem than ease of use. I submit that any scenario regarding ease of use should be examined from both the tool perspective and the source data. It may be that a data transformation will be more effective than replacing a tool. So where did all these tools come from and why?

The Early End-User Computing Era

The complex and obtuse world of IT and its many acronyms and seemingly bizarre technologies has been too much for most normal humans to deal with. Years ago, end users had to wait for systems, wait for programming changes, and wait for reports—users mostly sat around waiting for things to emerge from central programming and computer sites. "If I could just get to my data with something I could use!" echoed in every installation around the globe. There was no way to access and use a computer from outside the IT organization.

In the query, analysis, and reporting space, there are insidious assumptions made that both the end users and IT staff often erroneously make:

- The reports created using traditional languages such as COBOL is simple to replicate.
- The formatting and layout of existing reports is far easier to replicate and enhance than the options available in arcane languages.
- Very little processing logic is used in the older objects; therefore, a "modern" tool makes it easier to replace what we have and to create new objects.

The early tools used for query and reporting were all sold as "do-it-yourself" solutions. In the mid-1970s, several vendors began offering tools that allowed a non-programmer to delve in the world of data access and analysis. Nearly every vendor's product set included internal, proprietary data formats.

One of the compelling reasons for this was to give an end user the ability to create his own data and to place data into a form that was optimized for the tool. Another reason was that the era of relational databases, such as DB2, had not yet established itself for common usage and implementation of end-user data, so vendors were forced to offer their own data solutions.

There are obvious difficulties with such data sources:

- They were closed and proprietary; they worked only with that vendor's tool.
- Extractions of sets of source data were normally required.
- These extractions were then out of sync with the customer's original source data.
- Most could not contain the volume of data needed.
- IT assistance was always required to pull information from the original source.
- Significant investment into these technologies could isolate and trap key data used within a tool that might later fall behind the technology curve.

There were significant numbers of customers quietly dabbling in these tools. Well, maybe *quiet* is not a good adjective. Many were *noisily* trying to make these technical miracles work for them. The majority of the systems and data being accessed were mainframe-based because that was where the majority of the data resided. The tools themselves tended to provide very powerful capabilities if you could learn to use them.

Many of these tools were command-line driven, and the interfaces provided were seldom something to write home about. However, they did offer some hope to the non-technical user, and many departmental specialists emerged who were capable of navigating the technical issues and difficulties in using these powerful but primitive tools.

One positive aspect for those learning and using these tools was the need to understand how data is stored and accessed. Departmental specialists also would learn how to handle the processing of data and the steps required performing calculations. For example, if data were not sorted in the proper order, specialists would often obtain strange results when calculating subtotals or producing totals by breaks. Sometimes, they would find that sorting took extremely lengthy processing because the source data had not been stored in physical record sequence. They simply had to learn some of the issues that the IT staff dealt with every day.

One negative aspect quickly was uncovered: There were massive anomalies and inaccuracies in the data. There could be missing values, partial records, misspelled informa-

tion, inaccurate data, and more. In other words, the users learned the many negative aspects of working with data that their peers in data processing had to deal with.

Most customers of such tools ended up with a small set of users who could and would deal with all these issues in order to establish some independence in producing information from corporate data. The skills required to perform this artistry kept the user success rate low.

The Information Center Era

In the early 1980s, the *Information Center* concept was born. The idea of end users doing their own thing had been slowly catching on. The missing piece was to have some semblance of order behind their selection of tools and the skills required for using them. As I mentioned earlier, the number of users capable of performing data magic with little assistance was quite small. The idea was that you could go to a central site and get assistance from those whose job it was to navigate the corporate IT waters and shorten your learning curve.

The Information Center was traditionally set up as a central support organization designed to provide a set of services for end users and to act as a liaison between the non-technical users and IT. It was a center of competency and sanity that provided invaluable assistance for users to learn the proper skills in the tools supported in the organization. The IC, as it was typically called, was able to identify where the data resided, how to get to it, and what tools to recommend and to provide training in the tools and ongoing support.

Many ICs became PC competency centers as the personal computer emerged as the new frontier for processing. But the emergence of spreadsheets in the marketplace led to the demise of the Information Center. After the end users got their hands on a tool that they could drive independently, the demise for the ICs was inevitable. I still lament them, because I do not believe they have ever been effectively replaced.

No other tools or functions replaced the majority of the value that the Information Centers provided. *One major loss was the centralization of knowledge regarding the analyses employed in many areas of the enterprise*. Many disconnections between the end user and IT had been bridged for a while with the ICs. Now many users were taking old, reliable reports and keying them into spreadsheets.

Charge-Back Systems

IT costs somehow came under fire more than ever around this same time. Whether the end users were succeeding or failing/flailing, their impact on production systems (cycles, DASD, and other costs) was beginning to be noticed.

In order to share the pain and expense of the new analysis age, many corporations began to charge the end users for their associated systems load. Some of this movement was intended to make sure that some care was taken when performing interactive computing work on systems already burdened with production processes that could not be allowed to suffer.

Users were charged for processing, for user IDs and maintenance, for DASD space taken by all their work and more. The implementation of charge-back methodologies unilaterally made the end users pause to mull over the value of their interactive computing, and often they ceased altogether.

If you haven't assigned a significant business value to such processing to begin with, why would you continue to pay for services with little or no reward? Why would you attempt to bring new users and usage along if the cost would far exceed any perceived benefits? Many charge-back systems were implemented strictly to drive the users away and discourage their continued efforts in the interactive computing area. I have sat in meetings with IT staff and heard these sentiments echoed often. This end-user computing stuff was simply a diversion from the real business at hand. It was not considered mainstream or valuable.

Personal Computers

Needless to say, anyone in the early 1980s who expressed the opinion that PCs were merely toys and not to be taken seriously feels a little foolish today. At first, PCs seemed like quaint little versions of more powerful systems with some simple functions but little analysis or processing power. Then came the announcement of Lotus 1-2-3. The spreadsheet revolutionized the ability of individuals to perform their own analysis and computing.

As I mentioned earlier, after PCs were given some viable processes and software for businesses, the world as we knew it changed. The dilemma and pain associated with getting data to a machine seemed a small price to pay when you could use a personal machine to analyze and crunch numbers to your heart's content.

Users weren't making mistakes in public, so to speak. The end user could go behind closed doors and work in private. Any exposure of their lack of technical skills was purely accidental because they no longer had to work in the IT arena. They finally had a tool available to handle their math, and even though they couldn't get to the data, they had a degree of independence heretofore unavailable. One small problem remained: They still had a very difficult time getting to the data.

The Client/Server Wave

In the late 1980s came the near lemming-like run to embrace client/server systems. The basic tenets behind this revolution were:

- Mainframes were expensive and passé.
- Data should reside on smaller, less expensive boxes.
- The logic and calculations took place on the server database and the end-user tools.
- Distributed processing would be the norm.

Several elements of client/server systems proved to be less than ideal for those espousing their virtues. The one that offered the most damaging results was cost to implement. These solutions often ended up costing far more than the systems they were selected to replace. Because the "firepower" of mainframes had to be replaced, numerous servers were needed to equal the host. We began to hear the term *server farm* applied to these clusters of servers. Where many believed there would be a smaller, condensed machine, there were burgeoning populations of processors.

One of the other very serious aspects of implementing client/server systems was the need to provide ongoing operations of existing systems. Existing online systems, their data, the batch processes, and the entire IT infrastructure had to be maintained or replicated if replacement was the strategy. If you were to begin working on system replacements, why not add all the enhancements and changes so often discussed but not delivered? Heck, why not reengineer the entire IT system?

Business Process Reengineering became the term *du jour* of the industry. Many pundits and pontificators made tons of money on the lecture circuit talking about how to perform these massive system transformations. It's a bit like having a tiger by the tail. Or in today's terms, it's a bit like being Steve Irwin on *The Crocodile Hunter.* "Croiky! It's a big one, and I'm not sure 'ow I'm gonna let her go!"

Many organizations had an eclectic mixture of mainframes, distributed systems, fixed-function terminals, several databases, and personal computers. Processing was fragmented across multiple systems, and there was data duplication everywhere. Getting the data into the new server in a form that was useful and timely could be a nightmare.

Along with these client/server solutions came numerous analysis tools. The overwhelming majority of these tools were based on using SQL (Structured Query Language) as a base for asking data-related questions. Because the majority of the data required for analysis was in non-relational format, the tedious and costly job of extracting it from the host and transferring it to the new servers for loading was a full-time job.

Several relational database vendors emerged in this era, and one very good aspect emerged. All implementations of SQL were not the same. The need to establish an open standard among all the vendor offerings of RDBMSs became paramount in the industry. As a result, the customers received some very important BI-related benefits to this cooperation among the vendors:

- The analytics tools supported multiple vendor DBMS offerings with a common language.
- The RDBMS vendors pushed each other to excel in enhancements within the various vendor offerings and to make these enhancements part of the open standards.
- Skills in relational technologies (SQL skills and others) were reasonably transportable from one system to the next.
- Some common ground emerged by which to evaluate databases and tools. Query scenarios were established for benchmarks that permit an intelligent comparison of providers' wares.

Client/server as a panacea for replacement of mainframe systems has dropped as a driving factor, although many organizations have continued to examine how to restructure their systems to run on smaller and less costly platforms.

I suspect that if the processing power of today's hardware had been available in the late 1980s, we would have seen far greater success in host replacement. However, there is still all that darn data to contend with. It doesn't seem to shrink with time, and the number of sources just increases all the time.

One of the ancillary benefits in this urge to purge is that the costs associated with mainframes, especially hardware, have been dramatically reduced. One of the negative aspects is that some vendors have taken advantage of the mainframe customers and

have jacked up the price of software and maintenance for products that will not be replaced by new vendors because investment in mainframe software technology has dramatically dropped off. The mainframes of today are being repositioned and more creatively utilized as large, fast, reliable data servers in many environments.

The Information Warehouse Concept

In the late 1980s and early 1990s, one intriguing but mercifully short-lived fad was to implement information warehousing (IW). Instead of transforming the existing data into new, useful information, the idea was to leave it where it was and access it from anywhere with any tool.

Elaborate technologies emerged as many sought to define complex data relationships in order to access it by software and hardware "plumbing and wiring." Users could get to the data *in situ* and perform analysis. There were many negative aspects related to such an approach, including the following:

- Any anomalies or errors in the data were brought back as-is, and the users had to deal with them.
- Many BI applications require data from multiple, disparate sources that need to be matched and joined; thus, the complexity and sheer volumes of data were extreme.
- Validating and qualifying the results for accuracy was problematic. Most implementers were so relieved to get data back that they didn't care if the output was accurate and had no way to validate it anyway.
- Lack of performance was a huge problem.

The one very positive aspect of the IW approach was that everyone realized there was a very strong requirement for *metadata*. Because there were so many different and disparate sources and definitions, there had to be a way to define and understand not only the original data but also any new definitions and terms being applied.

The nightmarish aspect other than data-related issues was that many customers were convinced that they could snap any tool onto the IW infrastructure and pummel the data into submission. We were faced with a situation in which different users with different tools from different locations could all access the same data repository that may be replete with errors and anomalies. The users could all make random acts of analysis violence against a common pool of data.

The most serious flaw in this approach was the ease with which users could access data and perform their own analyses without an approved and agreed to definition of what

analytics were being used. Who would be the “traffic cop” for the analysis definitions and the math being used?

In such an environment, all extensions (new analyses) to existing data would probably exist only in the end-user tools. How would one produce a single version of the truth for analyses? You are saying, “Here he goes with ‘the math’ again!” And you are correct. Here, we have little choice but to perform all the analysis within the end-users tools themselves. This set up would force the widespread implementation of extremely “fat” clients (see Figure 3-1).

If you contrast this approach with the data warehouse approach that we discussed in Chapter 2, we see that there are some common elements that carried over into today’s approaches to BI, including:

- Definition of all source data and associated metadata
- A central repository for users to access data
- Concern that the end users must work from a common set of “math” for analysis.
- The current form of the data may not be amenable to BI analysis; thus, access in place may not be a very wise approach.

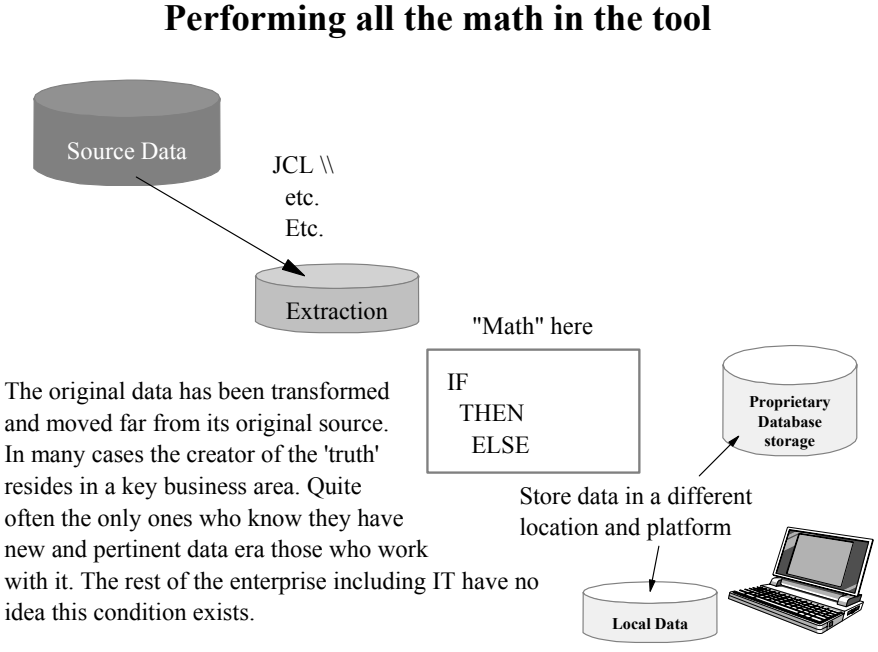

Figure 3-1 Performing all the math in the tool.

So what is the optimal form of data for BI? I suggest that the data warehouse or data mart approach with a star schema topology is the best format for BI. Because we are going to take existing data and redefine it, why don't we also add the changes and embellishments (the math) as well?

The argument about data and how much detail or aggregation to provide becomes an ongoing warehouse debate. Different users have differing requirements; thus, some detailed reports sorted properly with major breaks (by department, by division, etc.) may be all I need. If the only users you talk to have an analysis profile like mine, your assumptions for enterprise-wide processing will be highly skewed. The one-eyed man is indeed king in the land of the blind.

The Data Warehouse Era of BI

I stated early on that I would inject my biases from time to time, and this is one of my more adamant positions. In Chapter 2, I mentioned the RFP/RFI factor. Recently, I had a lengthy conversation with a customer and the IBM Team about how an enterprise-wide BI infrastructure might look. The customer had three major layers to its theoretical plan. At the bottom was a cluster of all the data sources regardless of what platforms they resided on and in what format they were stored. The middle layer was a logical data layer comprised of views of data and/or OLAP data that had been created from the sources. At the top were the end-user tools for BI access of the logical layer. However, they were not amenable to building any new data stores.

I kept asking, "Who are the users, and what do they need to accomplish?" This part of the plan had not been sorted out yet, despite the fact the customer was deeply into the planning phases of the project. The customer was very reluctant to create "clones" of the existing data, thought building a data warehouse would take too long (possibly true), but had no idea what the users wanted to do. How do you convince someone that a structure such as the one the customer described may not work for the company because the architecture seemed to be so logical?

Many warehouse schemas are restructured into new, accurate collections of tables normalized into some form and made available for analysis. The difficulty many users have with this is their ability (or inability) to perform the calculations. I know what you're thinking: "There he goes with the 'math thing' again!" My position has always been to perform the entire math or as much as you can at the server. The end users will have it far easier, and the population of those actually interacting with the data on their own will be far greater.

The data warehouse or data mart is far more than just a reorganization of data. It also is much more than a "cleaner" version of existing data. It is an opportunity for you to deliver creative and new information that is oriented toward the analyses used in the business. If new values and calculations are used within the enterprise, there will never be a better time to add them.

The entire gamut of data-related functions (extract, cleanse, etc.) has become a set of standard and expected processes that are associated with data warehousing. Most individuals working with such projects can cite the steps by rote. This is goodness for the customer, because the many vendors wishing to offer data-related solutions understand that they must provide these functions or interoperate with the most popular providers of ETL tools.

In a later chapter, we will discuss the analytics tools but let me interject a thought here. All tools are not alike, and even similar tools will have their quirks. Query and reporting tools in particular can all begin to blur as you evaluate them.

One of the dilemmas with tools is that you are constantly trying to match the features and functions with the source data to try to exhibit meaningful information to key end users. What is the proper output for a vice president of marketing that would most accurately reflect the data he needs to see? Should you present the analysis results in a pie chart? Would it be best to produce a bar chart? What is best to portray any results? Are you collecting and calculating any information that is going to change the business?

Can't someone deliver a suite of analytics that are predefined and germane to users holding specific positions within the corporation? Are there solutions out there that provide both "canned" and changeable options?

Advanced Analytics: Delivering Information to "Mahogany Row"

You might hear the terms KPI (key performance indicator) and dashboards applied to BI solutions. Most executives want to be in on the BI action as well. However, the rarified air level of data that they deal with is seldom produced in most BI environments. The primary reasons for this are the math they require and the coalescing of data from the multitude of sources needed.

There have been many Executive Information Systems developed over the years. Like so many of the early technologies (4GLs, etc.), these systems were often closed, propri-

etary, and isolated from the traditional IT processes. The concepts were sound, but implementation was often very difficult.

Some vendors today are trying to deliver solutions that are amalgams of canned BI analytics and a toolkit allowing the customer to modify or even define its own metrics at the executive level. Think in terms of a needle gauge that shows three separate areas: red for bad, yellow for okay, and green for good.

Today's executive may receive some reports or charts from a tool that allows them to play some "What if?" scenarios. Their existing tool set may allow them to learn how to perform some changes in scenarios where the goal is to turn the red figures into yellow and preferably green.

Using our needle gauge example, they would simply grab the needle or use a slider bar to drag the red figure displayed to the green area. The associated numbers required to attain the green status would also change with the display.

Many BI solutions are simply too low on the ROI scale or delivered at too raw an implementation to provide executive information effectively. The key to delivering executive-level information is to determine the key metrics required and to deliver them regardless of whether the processes fit the current corporate data warehouse structure.

Executives are not going to play around on the BI solutions delivered for extended periods of time. If they do, then they have become entrapped and enamored with technologies and have lost sight of the prize. Many BI solutions today include "triggers" and thresholds that notify the end users when something pertinent has happened; if nothing important has occurred, no actions are taken and nothing is sent.

BI Milestones

We've come quite a way in the quarter century and more of BI-like activities. However, in many ways, we haven't traveled far at all. What appears to be lacking is the embracing of BI as a key part of all corporate strategies. What have we learned so far?

- Early user-friendly languages emerged to offer a bridge between end users and the hostile IT environment establishing the concept of end-user computing.
- Centralized centers of competency were created to provide a means for end users to become productive quickly. The need to set corporate standards for analysis tools was one of the most significant benefits from these centers.

- With the era of client/server systems came the understanding that keeping data *in situ* may not be conducive to analysis; thus, reengineering of data into BI-friendly forms and formats was ideal. The most commonly accepted form of database was a relational store that supported SQL. The need to establish and adhere to standards for all vendors' SQL became a mantra.
- The Information Warehouse proved that accessing data in place is not always desirable, but capturing the metadata about existing information makes perfect sense. Before we transform current information, we need to know all we can about its current contents and form.
- Data Warehousing projects brought all the pertinent steps together for taking existing information sources and creating new, analysis-based data. It also proved that the tasks related to data transformation could be incredibly long and costly. The argument as to whether a warehouse or a mart is more appropriate continues. The most significant aspect of warehousing or "marting" is the realization that the back ends will probably remain and processes to transform and create new data stores must be automated. These are not one-time events.
- We are entering an era where packaged BI solutions are desired. One driving force behind these is the need to deliver sophisticated metrics and analyses to top management.

As I have stated several times, we seldom hear about the delivery of BI for the enterprise. One reason for this is the sheer mass of energy required to organize all elements of the business into a united front. The impact of BI solutions can be phenomenal, but at what price? Now let's look into the impact of BI at various levels within the organization.