Data Warehouse Design Redux:

A Data Driven Approach

May 2011

A White Paper by

Dr. Barry Devlin, 9sight Consulting
barry@9sight.com

The data warehouse has now been with us for a quarter of a century. Its architecture and infrastructure have stood largely stable over that period. A range of methodologies for designing and building data warehouses and data marts has evolved over the years. And yet, time after time, in one project after another, one question is repeatedly asked: "why is it so difficult to accurately and reliably estimate the size and duration of data warehouse development projects?"

This paper first explores the issues that plague data warehouse development projects and the most common trades-off made by vendors and developers—choosing between speed of delivery and consistency of information delivered. The conclusion is simple. This trade-off is increasingly unproductive. Advances in business needs and technological functions demand delivery of data warehouses and marts with both speed and consistency. And reliable estimates of project size and duration.

One compelling solution to these issues emerges from taking a new look at the process of designing and building data warehouses and marts from a very specific viewpoint—data and the specific skills needed to understand it. From this surfaces the concept of data driven design and a number of key recommendations on how data warehouse design and population activities can be best structured for maximum accuracy and reliability in estimating project scope and schedule.

Contents

Sponsored by:
WhereScape®

Sir Ashley Burns III gazed sadly across the antique oak desk he'd inherited half a century earlier from his grandfather, the founder and first CEO of Fire and Life Insurance Protection (FLIP) Ltd. Penny Wise, CIO since the previous Thursday and only her second meeting in Sir Ashley's paneled corner office, pursed her lips, waiting for her boss to break the silence.

"This data warehouse project must be older than my flippin' desk." Burns paused ominously before turning towards Bill Hacker, long-time Head of BI. "Blast it, man! When are we going to see some results? Or even have an idea of how much the bloody thing will cost to build? Pardon my French, Penny, but my finance department is excelling in comparison, with a damn spreadsheet!"

Hacker's face reddened as he drew breath to reply, but Wise cut in quickly; she'd already experienced his lengthy explanations. It might be prudent, she thought, to prevent Burns living up to his name. "I have a plan," she extemporized, "Let me and Jim discuss it and get back to you."

"By the end of the week," Burns smoldered. "And only if the price is right, Penny!"

—o—O—o—

Back in her still impersonal office—she'd only been three days in the job—Penny closed her eyes and sank into thought. Her previous role as a senior partner in a large consulting and services firm had taught her two key problems that almost always derailed data warehouse projects. First, slow-moving, grandiose DW projects that seldom lived up to users' expectations and often ran out of steam, and money, before being mothballed and replaced by "spreadsheet nirvana". This was surely the history and the imminent danger here at FLIP. She allowed herself a wry smile—if only the Head of BI would live up to his name! Well, she could do nothing about the history, but she could certainly light a fire under Hacker! The second, and fundamental, nemesis of DW projects she knew from experience was sizing. Sizing a DW project was like trying to estimate the length of an archeological dig—you never knew how many layers of dirt you had to sift through before you reached the real artifacts that would vindicate your perseverance.

The layers here at FLIP were many and complex. Three generations of Burns' had led the company from card indexes and manila folders of client and claims data, through COBOL programs and tape storage on an early IBM S/360, and on to a modern relational database, client-server, (allegedly) integrated claims management system. Not to mention the numerous diversions on that path into databases and file systems that were still in limited use, but no one understood any more. And an acquisition a few years back of a Yorkshire-based property and casualty insurance firm with the quirkiest systems he'd ever seen, according to Hacker.

The FLIP DW project had been running for nearly eighteen months now, and was clearly bogged down. The users had been through so many requirements gathering session that their heads were spinning. Penny was suddenly convinced that the BI department was using these sessions as cover for a deeper problem—an inability to get to grips with the dirty, disparate and disconnected data in the bowels of FLIP's computer systems. She reached for the phone: "Bill, can you come in here? And bring the mobile whiteboard … and that huge source systems map you love so much."

Penny's conviction was short-lived. Within an hour, half the BI team was in her office, all trying to explain source systems, data field meaning, and linkages, until her head was spinning. Yes, Humpty Dumpty-like the data was all over the place, but nobody in the team could seemingly put the pieces back together—and they had years of experience between them.

"Everybody, STOP!" she finally almost screamed. "This data warehouse project is D–E–A–D! Let's regroup and re-scope. We seem to have made most progress with life clients and premiums; let's get something delivered. We're going to focus on a comprehensive data mart in that area…"

It was Bill who finally broke the silence. "Emmm, we already have *three* of those, the last of which was the first delivery of this project, twelve months ago. Jim, here, spends most of his time trying to keep up with user demands for new analyses and data. He's leaving in a fortnight… we're desperately trying to train up a replacement."

Penny sat down heavily. This was all she needed; and on her third day in the job.

“Excuse me, ma’am,” Ivy Lee Grant was a summer intern at FLIP, all the way from Brown University. “What y’all are missing is this one central point: every one of those source systems is working just fine, else the business wouldn’t run. Right? So, the data must be fine and dandy, too… for what it was meant to do. And so, what we need to do is go back to that data...”

Bill Hacker and the other long-time BI team members began to roll their eyes. But, as the grandfather clock in the elegant foyer below chimed noon, Penny realized that Ivy actually had the answer or, at least, the question that could lead to the answer. And she further realized that the history and tradition of the company that had so attracted her in the first place, and had led her to relocate from New Zealand to the ancient City of London, could be one big pile of dog-doo—a favorite phrase of Sir Ashley’s—that she’d have to clean up even before she could decorate her office.

She leapt up from her desk. She and Ivy would show these Brits a thing or two. It was time for action—they had only four days to find a good pooper-scooper…

Why are Data Warehouse projects so difficult?

> “Problems cannot be solved at the same level of thinking with which we created them.’”
> *Albert Einstein*

FLIP’s problems are, unfortunately, far too common. Since the earliest days of data warehousing[1], project managers in BI departments and systems integrators have struggled to satisfy business users’ demands for timely and effective decision support systems. These difficulties arise primarily from the fundamental realities of (1) users’ information needs, (2) the sources from which such information is obtained, and (3) the modeled design-point of target BI databases. While these are old problems that have been described many times before, the sad truth is that, 25 years after the invention of data warehousing, we have failed to satisfactorily resolve them. It is, therefore, of value to restate the problems, and next to frame them in the context of today’s very different business needs and technology environment. And we can ask: what have we learned and what, if anything, is fundamentally different now that might enable us to resolve these concerns?

Business users—you’ve got to love ’em

Sad to say, many data warehouse project managers believe that life would be far easier if only users would make up their minds what they need and stick with it for a reasonable period of time. This belief is perfectly true, but also perfectly unreasonable. Business users may not know what they need exactly, but they need it quickly and they want to be able to change their minds about their needs tomorrow. And to them it makes perfect sense.

> *“After more than 35 iterations of the business requirements, we knew that we needed a new approach”*
>
> Richard Ridge
> IT Manager, First Data

Dan Power[2] traces the roots of modern, computer-based decision support back to the mid-1960s and discerns two related disciplines intertwined—one delivering reports to managers of production processes and the other providing an interactive exploratory environment where analysts can explore and play with information in search of new insights, solutions to problems and so on. These two aspects of decision support came together in the data warehouse architecture of the mid-1980s with the recognition that both required the same information resource in total and that in business usage, exploration and reporting are two sides of the same coin.

As a result, the original data warehouse architecture, shown in Figure 1, proposed a single, logical store of all information required by decision makers, cleansed and reconciled from

[1] Devlin, B. A. and Murphy, P. T., *“An architecture for a business and information system,”* IBM Systems Journal, Vol 27, No 1, Page 60 (1988) http://bit.ly/EBIS1988 and
Devlin, B., *“Data warehouse—From Architecture to Implementation,”* Addison-Wesley, (1997)

[2] Power, D.J. *“A Brief History of Decision Support Systems”*, version 4.0, March 10, 2007, http://DSSResources.com/history/dsshistory.html

its disparate sources. The majority of individual users seldom need or even care about the overall structure or consistency of the entire information store from which they explore or report. However, the need of the enterprise as a whole to have a self-consistent view of the status of the business poses a dilemma of considerable proportions for the project managers and developers tasked with delivering such an all-encompassing system. Such a system is typically too large, complex and heavily interdependent to build in the timeframe demanded by its immediate users.

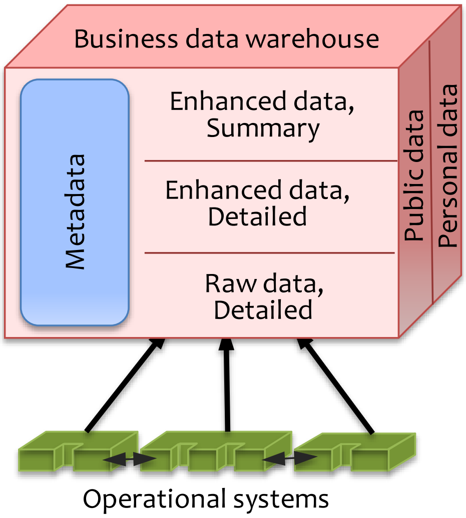


Figure 1:

The single logical information store proposed by the first data warehouse architecture (Adapted from Devlin & Murphy)

Over the years, developers have adopted two distinct approaches to resolving this dilemma. The first was to deny that the problem needed to be solved! This led to the introduction of the data mart concept. Data marts re-scoped the information need to that required by a particular group of users for some well-defined set of business needs. As a consequence, results could be delivered to users faster and more predictably. (And it was no coincidence that vendors could sell and deliver data mart solutions more quickly and profitably than data warehouses!) In addition, limiting the size and information scope of the solution favored predefined reporting with relatively limited *slice-and-dice* querying over true exploration solutions. Sourced directly from operational systems, these ***independent data marts*** satisfied the widespread business needs of the time for management oversight and control of well-defined production processes such as the emerging ERP (enterprise resource planning) systems. Unfortunately, the proliferation of data marts also drove increasing inconsistency of information across the business and raised users' expectations that delivering information for decision support was quicker and easier than it actually was.

The second development approach was to accept the necessity to build an enterprise data warehouse (EDW) and devise a staged implementation methodology to do so. Such an approach could deliver ***dependent data marts*** to business users relatively quickly while developing in parallel the EDW, which fed (or would feed) the data marts and assured overall information consistency. The resultant architecture is shown in Figure 2. While mitigating some of the problems of delivery speed, this approach does little to ease project managers' dilemma in accurately estimating likely project size. That problem stems largely from the IT world, as we shall see in a moment.

Like the data mart approach, staged data warehouse implementation has had its share of success and failure. Typically favored by large enterprises with experience of strategy and planning in IT, the staged approach can deliver an enterprise infrastructure capable of supplying consistent, quality information to satisfy decision-support needs across large swathes of the business. By using dependent data marts, it can deliver consistency. The inclusion of temporary, independent data marts fed directly from operational sources (so-called ***warehouse bypasses***) enables early delivery of business function and information. However, the later migration of these temporary solutions to be fed from the EDW may be resisted by the business, which tends to prioritize delivery of new information needs over the consistency benefits and IT cost savings that such migration offers.

When IT shoots itself in the foot (or worse)

As any data warehouse development manager will confirm from bitter experience, the biggest technical challenge they face is in understanding the source systems for the warehouse, extracting the data from them and building a consistent set of information from the combined sources. Put bluntly, operational systems are often poorly designed and inadequately documented—at least in the context of getting bulk data out of them. The older ones, especially, often store data in highly encoded and dense formats to increase speed and decrease storage demands on older hardware. Furthermore, the designers and developers of these systems are increasingly retired or gone, leaving a knowledge vacuum about database designs and processing approaches. While modern, commercial ERP systems have eased these problems to some extent—their data storage is more integrated and consistent, less complex and better documented—very few businesses of any size have totally eliminated legacy systems or have managed to limit themselves to one, single, uncustomized ERP system.

Irrespective of whether the approach adopted is based on data marts or EDWs (despite widespread and insistent advice to the contrary, most medium to large enterprises have developed and maintain more than one EDW), one of the first and most demanding tasks of a BI project is to understand and document to some level the sources of the data required and to define the extract, transform and load (ETL) processes to populate the mart or EDW with the data needed. The phrase *data archeology* has been coined to describe this process, and it makes a lot of sense—the structure and meaning of data is often buried in multiple and unrelated layers of design and coding that have to be excavated and painstakingly interpreted to ensure that the extracted data is clean, consistent and correctly represents true business reality.

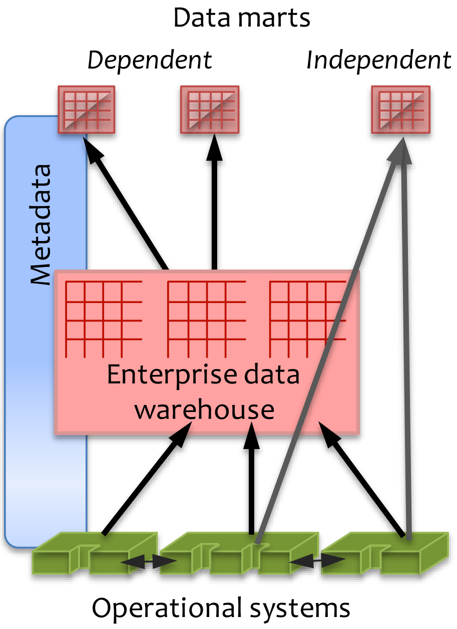


Figure 2:

A typical BI environment with EDW and dependent and independent data marts

Beware attractive models, on or off the catwalk

Whether building an EDW or a data mart, best practice decrees that the database should be modeled according to one of a number of schemas. The choice for an EDW is usually a relatively normalized enterprise data model. Data marts are often designed according to a dimensional or star schema approach. Various combinations and enhancements of these modeling approaches have also been developed. The primary aim of any model is to describe the information needs of the users—either as a whole across the enterprise or within a well-defined set in a function—in a clear, unambiguous and understandable way that is independent of any considerations of physical database design or population. Such models are called logical models.

In the early days of data warehousing, defining and documenting these logical models was a lengthy, expensive and time-consuming undertaking. Today, the fact that most enterprises have already built a number of EDWs or data marts, together with the emergence of common industry models, has lessened the effort and time associated with this phase of the development. However, significant work remains at the next lower level of modeling, the physical level, where the final layout of the data in the physical database is defined. The problem is sometimes characterized as ***impedance matching*** between the source and target data / database designs; it a less engineering-oriented world, it might be more aptly described as geek meets glamour model!

In general, operational systems' data stores, from VSAM files to object stores, relational databases and everything in between are optimized for specific operational needs. These include performance—specifically speed of access and update for individual records, reliability—assured capture of all changes, and data consistency within the application—guaranteeing that transactions are completely recorded, irrespective of the physical distribution or temporal span of their effect. Consider, for example, a banking system recording financial transactions against accounts. ATM transactions must be based on wholly reliable current account balances and result in instant balance updates in order to avoid the possibility of issuing cash that a customer doesn't have. Transfers of funds between different banks must guarantee that money leaving one account arrives accurately and reliably in an account in another bank halfway around the world. (Although, I still await a scientifically-viable explanation as to why electrons moving at close to the speed of light often take three days to cross even a city from one bank's data processing center to another's.)

These performance, reliability and consistency demands often lead to data stores that are highly complex in their design and impose significant temporal and procedural constraints on their use. Extracting data from them to a purely modeled environment is an extremely complex task. The final outcome is always a compromise. The physical database design of the EDW or data mart is only an approximation of the pure model dictated by business analysis needs. While this may be undesirable from a user viewpoint, the impact can usually be mitigated relatively easily with view or index creation. The more significant impact is on the development process of the data warehouse environment, where designers try to serially optimize extraction from the sources, performance of transformation and speed of loading data into databases that are as closely aligned as possible to the ideal data usage model at the informational EDW or mart level.

Moving Beyond the Speed vs. Consistency Dilemma

The problems just discussed have long been with data warehouse developers. As a result, developers have found themselves faced with a binary choice: deliver "fast" or make it "consistent".

Deliver "fast"

Business users, especially those in customer-facing functions such as sales or marketing, are not renowned for their patience. An earlier approximate answer is often more positively received than a later one with higher accuracy.

Fast delivery usually entails narrowing the scope of the data included in the project to that understood and/or required by a limited set of users from one business function. These users are often well satisfied with the project results, as are other groups subsequently served in a similar manner. The problem is that consistency of data delivered across these multiple projects not guaranteed. Results are delivered in essentially independent silos, and one of the key aims of data warehousing—management information integrated across the company—is missed. The proud owners of the various data marts arrive in front of the CEO with inconsistent answers to key questions.

Make it "consistent"

IT shops that focus on the objective of cross-enterprise consistency of management information are usually driven mainly by the finance function or, less frequently, the CEO. These business users are certainly not impressed by long project schedules or, worse, overruns (particularly of budget, as is all too common in this case). However, they are aware of the value a common base of information for regulatory reporting, long-term planning, or simply eliminating circular and expensive "who's right" discussions in the boardroom. In larger organizations, particularly, the CIO is often supportive of these needs, knowing well the fundamental inconsistencies that exist in the enterprise's databases and the problems for IT that ensue when they are revealed at board level.

Early, spectacular failures in delivering large EDW projects aimed at creating a single, consistent store of enterprise data before delivering reporting or analysis solutions to the business have resulted in a blended, process approach to EDW development and delivery today. The best-practice process consists of a well-planned and staged delivery of slices of the EDW in combination with dependent data marts fed from the evolving EDW. This approach generally delivers specific business solutions more slowly than the "fast" approach, but enables at least an approach to more consistent management decision making across the enterprise.

"I want it all, and I want it now"[3]

Today's business is increasingly unwilling and, to a large extent, unable to live with the compromises inherent in this speed vs. consistency dilemma. It's time to seriously pose the question: is it possible—given the advances in technology, all we've learned in methodologies, the evolution of business needs, and more—to envisage how we could devise an overall approach and specific tooling that could enable us to have it all and have it now?

There have been a number of developments that ease the task of data warehouse designers. First, there has been a considerable improvement in the structure and cleanliness of operational sources since the 1980s. As mentioned previously, large-scale deployment of ERP systems has significantly reduced the number and variety of old, custom-built applications and the dependency on the rapidly dwindling numbers of developers who understand them. While understanding and extracting data from these newer ERP systems is far from trivial, they are generally better structured and documented than the older applications they replaced. In addition, the vastly improved performance of computers and the exponential increases in memory and storage sizes, has reduced the need for

[3] Queen, 1989

compact and complex data structures with multiple-use fields and extensive encoding of data that was done previously. Furthermore, an ongoing focus on data quality has gradually improved the cleanliness and accuracy of data stored in operational systems.

Second, over time, data mart and EDW models have become less rigorous in structure and more flexible in implementation. In part, this has been driven by the experience of implementers who have had to deal with the design compromises the hard way—manually and with little formal support. In addition, common and largely standardized models for different industries and purposes as well as for a variety of database structures have become the starting point for many DW implementations. Improved hardware and database performance have also enabled some relaxation of the constraints that query and load performance put on adherence to a particular preferred model.

Third, as ETL tools have improved and their use has become more widespread, a more extensive and reliable set of metadata describing source and target systems has been gathered, as well as the transformations required to transfer data between them.

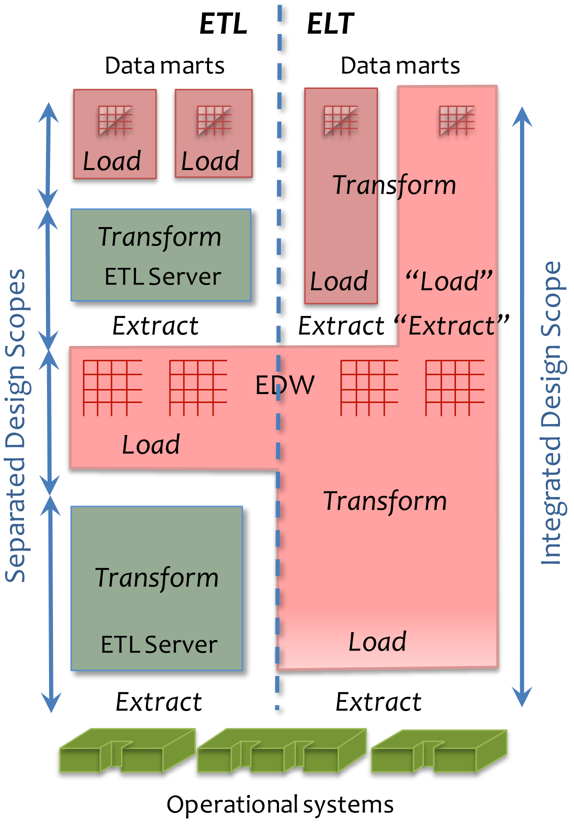
Fourth, and perhaps most interestingly, is the emergence of extract, load and transform (ELT) approaches to populating data warehouses which have recently been gaining traction in the Data Integration market[4].

Integrating the design scope

ELT tools use the power and functionality of the target database to perform the required transformations, rather than a separate, intermediate server as typically used in ETL. ELT is benefiting strongly from the increasing power of analytic databases and RDBMSs in general and the increased use of in-database processing. As shown in Figure 3, the significance of ELT as opposed to ETL is that it leads directly to a more integrated design process with all processing of inbound data taking place in one environment—the target database—and all metadata produced being stored and manipulated in the same environment. On the left, as shown, the ETL process involves five separate servers—the operational, EDW and data mart systems and two ETL servers—to move data from operational systems to data marts (four when data marts are collocated with the data warehouse). ELT, on the right, involves two to three. For simplicity, staging tables, index builds and implementation issues such as parallel processing are omitted from the diagram.

Figure 3:

ETL vs. ELT



The design-time issue is not the number of servers *per se*; it is rather the implications of the separate servers with different design tools and methodologies, different metadata designs and stores, and often different teams responsible for the process on the different servers. In the ETL approach, most design and development centers on the functionality available in the ETL Server, particularly transform and the extract approaches from the sources. While load is also covered, many design aspects of the load phase are closely linked to database design issues and there is usually an iterative approach between load / transform capabilities of the ETL tool and structural and performance constraints of the DBMS. This leads to two interlinked, but partially disconnected design scopes. Figure 3 shows this happening in the EDW and data mart population design phases. This disconnect is materialized in separate metadata stores in the ETL Server and DBMS with very different characteristics. Organizationally, larger DW shops may have separate DBMS and ETL design teams leading to further fragmentation in the ETL design process.

[4] Russom, P., "Next Generation Data Integration," TDWI Best Practices Report, 2nd Quarter 2011

In the case of ELT, shown on the right of Figure 3, the load and transform steps are collocated on the DBMS server, and database functionality is used for processing, with all metadata stored in the database itself. This leads to a much more highly integrated population process that emphasizes the natural design linkage between transform / load in ETL and database design in the target DBMS, and facilitates an organizational structure where ETL and database designers work as a single team. As Figure 3 also shows, if data marts reside on the same server as the EDW, it is even possible to completely avoid offloading / reloading of data between the EDW and data mart layers.

WhereScape RED, an integrated development environment for data warehouse creation and maintenance, is a classical example of an ELT-oriented approach to data warehouse and data mart design and population. It clearly illustrates two key benefits of and integrated design scope with a common metadata repository:

1. **Rapid prototyping:** multiple, short design and delivery cycles are possible, based on the integrated nature of the environment

2. **Retrofitting:** the common metadata between population and DBMS design provides an ideal base for retrofitting existing and poorly documented designs to a new, standardized and well-defined model, and can support later migration of the older systems to a newer model.

These are significant advances in the design and development process for data warehouses and marts. Design is simplified. Delivery is faster. And organizational simplicity is enabled. However, a significant challenge remains unresolved: how can the likely size of the project be reliably estimated ahead of time? How can a project manager gain some dependable insight into the cost and schedule of the project?

Data driven design—the final frontier

"… to boldly go where no man has gone before."
Star Trek

Although it may appear blatantly obvious, there is one common thread that links all aspects of designing and populating data warehouses and data marts; that thread is data. It is the structure and content of source system data that determines the extent of the challenge in extracting data from these systems. It is the structure and content of target databases in the EDW or data marts that drives complexity and subsequent performance issues in loading data into the targets. And finally, it is the relationships between the structures and contents of the sources and targets that establish the transformation rules and their complexity. So, let's examine the process of designing a data warehouse (either EDW or data mart) through the lens of data.

Table 1 lists the steps involved in this process. Although the steps are numbered for convenience of reference, the order is only an approximation. Some steps can proceed in parallel. For example, steps 1-2 are usually performed in parallel with either steps 3 or 4, although often by disparate teams. In addition, there usually is a high level of iteration among the steps, with a major redesign iteration occurring at step 9.

Examining the expertise needed in each step, it is immediately clear that the initial seven steps depend entirely on data-related skills. It is only at step 8 that design and programming skills related to the chosen ETL / ELT environment and tooling need to be engaged. This observation points to a significant transition point in the process and affords the opportunity to break out these earlier steps and optimize the tooling and support for a particular set of users.

Who are these users? They are clearly those with specific types of data expertise. In many cases, they come from a blended business / IT background. Perhaps they are business users who learned how to extract data into spreadsheets and play with it there; and discovered that they had a particular interest in the data itself and its sources. Perhaps they are business analysts who reside in IT as part of the data warehouse design team or competency center. They may be data or database administrators. Who they are certainly not are programmers or ETL / ELT experts.

This particular division of the process is seldom evident in today's data warehouse design and build projects. Although these distinct data-related skills do exist, their input is limited as a result of two very distinct organizational structures widely found in data warehouse projects.

In the first case, we find these people rolled into a project team largely dominated by programmers whose first instinct is to start creating the ETL / ELT transformation functions they believe will be required, aided and abetted by increasingly sophisticated and user-friendly graphical user interfaces. The outcome is a serious under-playing of the initial exploration and design phases so vital to a high-level assessment of the potential trade-offs and issues in the population process.

In the second case, they exist in isolated pockets of the business or IT organizations and used as "external experts" by the main project team. Again, the process is driven from the ETL / ELT graphical user interface by a team largely composed of programmers. The outcome here is an even more serious under-playing of the initial exploration and design phases.

No.	Step	Skills, especially data	
1	Identify and locate relevant source data systems (existing or being updated)	**Source data** expertise	"Data driven design" / "pre-ETL" steps
2	Explore, understand and document source data structure, content and limitations	**Source data** expertise	
3	Evaluate industry or reference data model in the context of enterprise (EDW) or specific business (data mart) needs	**DW data** modeler	
4	Define logical target data model based on enterprise (EDW) or specific business (data mart) needs	**DW data** modeler	
5	Explore possible mapping of source data to logical target data model	**DW data** modeler and **source data** expertise	
6	Understand and document an existing data warehouse or mart structure and content (upgrade an existing system)	**DW data** modeler and **database** administrator	
7	Design physical target data model given source data constraints	**DW data** modeler and **database** administrator	
8	Design data mapping (extract, transformation and load) functionality	**Source data** expertise, **database** admin., and ETL / ELT design & programming	"Development" steps
9	Iterate physical target data model and mapping (go to step 6 or 7)	**DW data** modeler and **database** admin., and ETL / ELT design & programming	
10	Implement ETL / ELT function and test	ETL / ELT design & programming	

Table 1:

Steps and skills in the data warehouse design and population process

In both cases, the contribution of these data experts is deeply embedded in the main path of the overall design process, the sizing of which, as mentioned earlier, proves almost impossible to estimate. However, breaking out the first seven steps of the process and focusing up front on ***data driven design*** in the ***pre-ETL*** steps provides three particular advantages:

1. These data-specific skills can be utilized to their full potential and with appropriate tooling in an independent part of the overall population design and development process
2. Data-specific skills from different areas can be properly aligned and integrated. For example, the source data skills needed for steps 1-2 and the data modeling skills in steps 3-4 can be brought together to avoid theoretical modeling without constraints or context
3. (And most important), by applying these skills before engaging the larger team and beginning the more extensive in-depth design and implementation work of the development stage (steps 8-10 above), the resulting overall project can be more accurately sized at an early stage and the developers are building what the users actually need in terms of information

WhereScape 3D

In May 2011, WhereScape introduced their new product WhereScape 3D (data driven design) that focuses exclusively on the data driven design or pre-ETL phase of data warehouse population design described above. Its purpose, briefly, is to assist project managers to scope, size, cost and de-risk data warehouse, data mart and BI projects before they begin. It supports them in answering the following questions, and similar ones, that until now were particularly difficult, if not impossible, to answer:

- What are the characteristics of our source data? How difficult will it be to use it?
- Can we build a data warehouse or data mart using this design and our source data, in any reasonable amount of time?
- Will the design model we're using actually work?
- Can we effectively populate the purchased reference model from our source data?

As an initial release, WhereScape 3D focuses on some of the most common use cases in the process of designing a data warehouse or data mart. The initial cases cover (1) documenting an existing source data system, (2) documenting an existing data warehouse and (3) producing a data warehouse specification.

Documenting an existing source data system

The first use case can be seen as a fundamental task required in the design phase of a new data warehouse or mart, and aligns to steps 1-2 of the process above. It consists of a detailed, systematic profiling of all relevant existing source systems, examining and sampling both metadata and real data, in order to understand and document:

- The structure of the source data, including identifying data columns or fields, determining their business meanings and technical details such as data types, and identifying declared relationships between fields

- The content of the source data, including data ranges, special or unusual values such as nulls or dates like 31/12/9999, dirty or clearly erroneous values and so on, and identifying likely relationships between fields based on content similarities

The output is a concise, structured document aimed specifically at data warehouse and mart designers enabling them to understand the usability of the source data for a particular business need and to begin to estimate the level of effort that will be required to extract clean, consistent and complete data from the source. This forms the basis for understanding what sorts of data warehouses and marts can be built with confidence from the available source data. A further output is a formal XML definition of the source data for use by other tools as needed.

Documenting an existing data warehouse

This use case focuses on a key task in any data warehouse / mart upgrade—what is the actual structure and content of the database, step 6 above. While it may be assumed that existing data warehouses and marts are well-documented (we *have* been preaching this for many years!), the reality often fails to match the expectation. Common problems observed include: (1) a well-defined logical model, but a poor description of trade-offs made in the physical implementation, (2) a well-defined initial physical design, but a failure to document extensions or upgrades over time, and (3) the ultimate in poor design practice—an entirely undocumented design for the warehouse / mart.

As with the previous use case, the output is a concise, structured document aimed specifically at data warehouse and mart designers, in this case to understand the existing structure and content as a basis for estimating the effort required to upgrade it. In general, the focus here is more on the structural characteristics of the data, because a primary design point for data warehouses, in particular, and data marts is that the data content is cleansed and made consistent and suitable for access by business users, directly in the case of data marts or indirectly for EDWs.

Producing a data warehouse specification

While the previous two use cases support fundamental tasks in the design process, this third use case brings together a design process workflow, in this case, a specific application of steps 1,2,4,5, and 7 in Table 1. The first use case above, "documenting an existing source data system", clearly comprises the initial phase of this process. The next step involves the design of a data warehouse or mart schema based closely on the information derived from the exploration and documentation of the source system.

The output of this use case is a detailed set of specifications for the data warehouse and the mappings from the source system data, produced in a highly automated fashion by WhereScape 3D.

Further use cases

The above set of use cases illustrates the fundamentals and a full-task example of the concept and implementation of data driven design. There are clearly other interesting use cases, including:

- Report driven development: "I need these reports from my source systems(s)"
- Pre-built implementation—analyze sources for a pre-built model: "Is it feasible to implement?"
- Buy a model and map to source systems: "Produce a Mappings Document"
- Logical model vs. implemented model: "Produce a Gap Analysis Document"
- Monitor source system changes over time: "Produce a 'What has Changed' Document"
- Align stories to models: "Show me how a user story relates to a data model"

It is envisaged that these and further use cases will be delivered by WhereScape and third parties.

Recommendations and Conclusion

But, wait! Whatever became of Penny Wise and Ivy Lee Grant? Have they solved the data warehouse project predicament of FLIP Ltd? Has Sir Ashley Burns III seen his business needs met, or is he on the verge of another apoplectic outburst?

Well, the good news is that Penny and Ivy discovered an early version of this white paper! And they immediately began to put into practice a number of key recommendations that reshaped and re-energized the data warehouse project:

1. **Focus first on design-level work:** The first priority of any data warehouse population project is to understand the issues raised by the source data systems, both in terms of cleansing and in their implications for physical design of the target database
2. **Empower the people doing data driven design:** Providing tools focused on data driven design and creating an organizational structure ensures that ETL implementation follows from the data driven design work and re-establishes control of population projects
3. **Get a *reliable* delivery plan in place:** Once a high-level understanding of the limitations of the source data and their implications has been gained, a reliable estimate of the likely size of the overall implementation effort can be made
4. **Break the delivery into small chunks:** Unless this is a very simple upgrade or small data mart project, implementing the full population process for the entire data warehouse from multiple sources will take longer than most business users are willing to endure. So, use an agile development approach to define and deliver small wins incrementally
5. **Put an overall process in place:** Keep in mind that data warehouses and marts undergo continual change and refurbishment as user needs evolve. Standardized and, where possible, automated documentation of all design decisions is mandatory and should be an integral component of the overall data warehouse infrastructure

Focusing closely on the data aspects of data warehouse design and population leads to an intrinsically more balanced approach to project scoping and sizing. Understanding and documenting the source data is a mandatory first step. Then, and only then, can decisions be made about the physical database design required. However, these two steps are intimately connected and their union can only be truly understood and cemented by skilled data practitioners. Tooling to support data driven design, such as that provided by WhereScape 3D, separate from and prior to the design and development of ETL processes is key to enabling these skilled data practitioners to fulfill their role.

With careful application of data driven design principles and attention to the organizational structure of the delivery team, data warehouse design and population projects can now be placed on a much firmer basis of realistic and reliable scoping and sizing.

Dr. Barry Devlin is a founder of the data warehousing industry and among the foremost worldwide authorities on business intelligence and the emerging field of business insight. He is a widely respected consultant, lecturer and author of the seminal book, "Data Warehouse – from Architecture to Implementation". With a Ph.D. in physical chemistry, Irish-born Barry has almost 30 years of experience in the IT industry, mostly with IBM, as an architect, consultant, manager and software evangelist. He continues to define and discover novel solutions to real business needs in the area of the fully integrated business—informational, operational and collaborative—providing an holistic experience of the business through IT. He is founder and principal of 9sight Consulting (www.9sight.com), specializing in the human, organizational and IT implications and design of deep business insight solutions, working with leading analysts and vendors in BI and beyond.

About WhereScape

WhereScape enables companies to get value from their data warehouses faster. Its flagship product, WhereScape RED, helps builds data warehouses faster. Its latest product, WhereScape 3D helps to build the right data warehouse.

WhereScape RED is the only comprehensive Integrated Development Environment (IDE) for data warehousing that supports the entire data warehouse management life cycle, integrating source system exploration, schema design, metadata management, warehouse scheduling and enhancement into a single, simple integrated design.

WhereScape 3D is the only comprehensive data driven design tool that incorporates data into the design process leading to better designs, less surprises and lower risk.

More than 400 customers worldwide are using WhereScape software products on a variety of platforms. Projects utilizing WhereScape products typically come in under budget, ahead of schedule, with improved performance, greater transparency and built on more solid foundations over the systems they replace. WhereScape has offices in Portland Oregon, Auckland New Zealand, and Wokingham UK.

WhereScape USA, Inc.
2100 NW 133rd Place, Suite 76
Portland, OR 97229, USA

www.wherescape.com

Brand and product names mentioned in this paper may be the trademarks or registered trademarks of their respective owners.