

**A New Approach to
Data Integration and Management
for Business Intelligence:**

*Realigning Data Warehouse Best Practices Based on Technological
Advances and Lessons Learned*



Hired Brains, Inc.

By Neil Raden
Hired Brains, Inc.
Version 2, December 2003

A New Approach to Data Integration and Management for Business Intelligence:

Realigning Data Warehouse Best Practices Based on Technological Advances and Lessons Learned

Data warehouses are expensive and complicated, requiring the careful integration of numerous major components. The innovations data warehouses were meant to foster are often stalled by the degree of time, cost and policy issues it requires to sustain and enhance them. In addition, the leverage that these investments yield is too low and the data warehouse informs too few people and processes. These two classes of problems, architectural and practical, and their solutions, are the subjects of this paper.

Now is the time to rethink the data warehousing concept. Data warehousing needs to be recalibrated to line up with lessons learned and the changes in technology over the past decade-and-a-half since it emerged. This paper presents a conceptual foundation for an improved approach that is based on model-driven architectures. The result is to de-emphasize the physical aspects of data warehousing and turn the iterative design and innovation over to business people and processes.

The concept of the data warehouse/business intelligence environment, initially developed in the 1980's and commercialized to this day, predates many stunning innovations we now take for granted. For example:

- The onslaught of Moore's Law, still raging, and the collapse in CPU, memory and storage prices
- Universal connectivity (the Internet), which we take for granted today, but just ten years ago was a major impediment to enterprise systems
- Universal access to data everywhere (the Web), which embedded in our consciousness the idea that information can be shared anywhere, anytime
- Y2k and the renewal and/or replacement of legacy systems
- Enterprise systems, particularly ERP, displacing myriad legacy technologies and bringing the reality of a consistent reference model to back-office processing
- E-business and the rise (and fall) of the dotcom's, but persistence and ubiquity of internet-enabled business
- Vast improvement in relational database technology, especially for scalability and query optimization for data warehouses
- Grid computing, network-attached storage, web services, EAI and EII and host of other technology innovations that changed the landscape of corporate computing

The immediate conclusion can be drawn is that many architectural innovations should have occurred in data warehousing, but in fact, the basic approach is largely unchanged from its initial formulation.

Shortcomings: Architectural

In most data warehouse methodologies, the actual databases that provision data for BI are the data marts, which contain only a fraction of the data in the Enterprise Data Warehouse. This approach was taken because, fifteen years ago, the tools and resources needed to query a large data warehouse were prohibitively expensive for most organizations. Relational databases today can optimize queries against huge tables efficiently. The use of data marts, or, more accurately, subsets of the data warehouse, is perfectly reasonable in many cases, but using data marts in this fashion, to segment the data and to essentially create new stovepipes, is not a supportable approach with today's tools. In the past, this approach made sense, but it doesn't take into account what technology can do today.

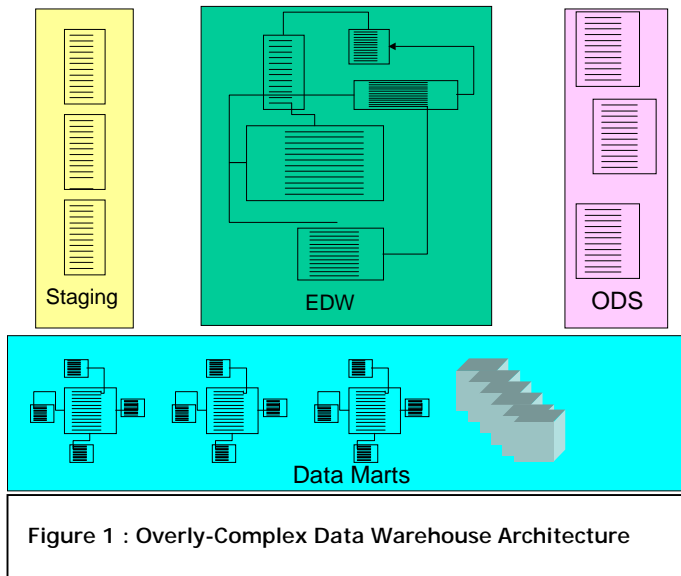
The reason is that technologists own data warehouse design. It is largely an exercise of hardware integration, relational database design, data modeling (in the relational paradigm) and software to extract and transform data. In addition, because the process is complex, one or more methodologies and frameworks are used, such as the Zachman Framework, a comprehensive roadmap for enterprise systems architecture. These frameworks require that the designers compose a solid architecture first, and all subsequent development is in conformity with it. In essence, the model follows the architecture. This is overwhelming the prevailing method in data warehousing today.

Although these kinds of general frameworks are not designed specifically for data warehousing, they are applied because of the enterprise scope. They are highly formal and require a great deal of effort and lead-time before useful products are generated. When applied to the analytical space, its results are far from compelling. There are success stories using these approaches, but the discipline, rigor and a corporate culture are usually the driving factor, not adherence to the framework. On the other hand, there are countless examples of organizations that took the highly structured, formalized approach and failed to produce anything of enduring value over the long run.

The data-centric designs of data warehousing were conceived at a time when the Business Intelligence (BI) problem was thought to be a problem of data management - finding it, cleaning it and storing it. Nothing in early literature addressed the needs of knowledge workers other than a vague promise to solve the "connectivity" problem - getting at data. Data warehouse "architecture" simply assumed that provisioning data was sufficient. It was entirely focused on the back-end of data warehousing, which led to many of the disconnects we see today. For example, the primary data warehouse is off limits to most analytical processes. Stovepipe-like data marts are created when the original concept of a data warehouse was to integrate data across the stovepipes. Overly-complex designs were created, full of latency from too many layers and burdened with ongoing maintenance costs that exceed those of most operational systems.



In the case of data warehouses, the practitioners might draw up a paper design of the architecture, based on framework *principles*, and arrive at a structure that makes sense from a data architecture perspective, but the complexity interferes with the smooth operation for which it was designed. The problem with these kinds of frameworks is that things tend to work fairly well at first, but as conditions change, either gradually or radically, the approach proves to have the fluidity of poured concrete - only fluid until set. These convoluted architectures are then patched with layer upon layer of new structure, creating an overall design that is too brittle and inflexible to be useful in a BI environment (see Figure 1). Each new artifact adds latency to each query and each new patch adds latency to the delivery of needed features. The combination of artifacts creates an ever-



increasingly-complex structure that impedes or even prevents enhancement or modification. Ultimately, the expense, delay and dysfunction overwhelm the process. The environments built become too unwieldy to respond to changing business requirements or, for that matter, even the original ones they were meant to support in the first place.

Implicit in the basic concept of these overly-complex architectures is that the single repository of all relevant data guarantees the "single version of the truth," a concept that is widely accepted but rarely achieved. The "single version of the truth" is

currently a multi-part process that does not reside entirely in the data warehouse but instead, is spread, and sometimes fractured, around the organization. In actual practice, the components of the "truth" are arrived at through aggregations, calculations and models that are developed outside the boundaries of what we call the data warehouse. Analytical tools added-on after the data warehouse models are built and populated provide that functionality, each with their own schema and metadata. The classic data warehouse design and methodology do not extend that far, making the "truth" a difficult process to control.

Data warehousing raised the issue of metadata, to its credit, but failed to codify its use or meaning. It is widely agreed that metadata is essential, but a typical data warehouse environment will have a half dozen or more metadata schemas, one for each piece in the chain. The schemas are incompatible and, in most cases, not even actively participating in the development, operation or maintenance of its own area, taking up space as a passive place-keeper. There may be metadata in place, but it is not delivering the value it should because there isn't a solid understanding, from end-to-end, of how the data and metadata fit together to form useful applications. The chasm between data warehouse methodologies and how data warehouse information is used is currently too wide for a unified approach. Unifying the metadata for the entire data warehousing/BI realm is a necessity, but one that can't be resolved with the current architecture-first practices. A model-driven approach is needed.

A *model-driven architecture* has proven to be highly effective in delivering useful results with good ROI in a short period of time. In particular, a model-driven architecture can excel at providing high levels of reusability, flexibility and maintainability. The concept behind model-driven architectures is to build declarative models, in the time and place where they are most needed. These models are combined over time to form not only more comprehensive models by the union of their functionality, but to enable their emergent properties to appear as the models are joined.

While the data warehouse was a worthy concept in the era of expensive servers, weak data integration tools and under-powered BI tools on the front end, the march of technology is putting pressure on the industry to retool and reevaluate best practices. Specifically, these architectures are under performing in the following areas:

- Reusability of components: many organizations use more than one BI tool, and analytical development cannot be leveraged in another because there is no unified analytical model
- Limited data abstraction/metadata not performing: the processes access data stores directly to the physical structures, such as tables and columns and views, making it impossible to rapidly modify physical and even logical models. Each process creates its own metadata
- Failure to leverage low-cost resources: Without a mechanism to virtualize data stores, data is collected in single repositories
- Brittle integrated environments lead to slow innovation: The concatenation of ETL, database, BI tools and other pieces creates a structure that impedes modification because the impact of one change cannot be predicted precisely, leading to long development and testing cycles
- High maintenance costs (software and personnel): Data warehouse environments are inordinately expensive to maintain, especially for ETL, DBA and data modeling professionals. In addition, the inability of the application to provide the functionality that businesses need leads to a significant amount of "Shadow IT," the cost of non-IT professionals doing quasi-systems and programming work.

In addition to architectural drawbacks, from a purely practical perspective, current data warehousing practices have also lagged behind the technology curve and failed to deliver the functionality and benefits they should.

Shortcomings: Practical

Because first-generation BI tools were essentially desktop tools and limited to only very small volumes of data, it was assumed that BI was largely the province of summarized data, but it turned out that this was a classic case of the tail wagging the dog. Most business people are interested in answers to questions that appear, on the surface, fairly mundane, but that can only be satisfied by taking a slice of very UN-summarized data. These questions can actually be quite complex queries against terabytes of data. If left to their own thought processes and not limited by conventional wisdom (from their IT counterparts), more of these questions would be asked, but the limitations of most BI tools have had the effect of constraining people's inquiries. For example, the question, "How many people bought satellite radio in our cars this year, " is easily answered with a classic, summarized BI approach. But if this question were posed by a real product manager, it might be phrased more like, "Of the purchased satellite radio options, rank them by the top five other high-end options purchased with them."

There are an infinite number of perfectly normal, easy-to-phrase questions that business people are likely to ask that are exceedingly difficult for pre-aggregated, pre-calculated, pre-subsetted data structures. It would take an army of geniuses to guess what all the combinations should be, a resource that is obviously not at hand. What's the alternative? The only answer is to break down the walls of the data warehouse, collapse its stovepipes and free up the information for use in the way it was originally intended.

The propagation of BI in organizations has been very slow. Current estimates are that BI penetration into organizations is less than 10%, likely even less than 5%. Unclogging the logjam in BI requires getting the message out that data warehousing practices need to be revisited. The lack of data depth/richness combined with complexity and high support costs limit the potential of BI deployments. Above all else, relevancy and simplicity are the keys. Few people in organizations are able to integrate data warehouse/BI tools into the work they routinely do, a major contributing factor to the cost and prevalence of Shadow IT. Only a radical approach to informing people and processes will change the profile of usage to any measurable degree. Shifting the costs and efforts from a traditional data warehouse to a truly manageable information integration intelligence architecture will accelerate ROI through:

- Reduced duplication of effort
- Reduced need for specialists, power users and data-czars
- Improved change management
- Consistency of models, definitions and data
- Sharing of information and technique
- Moving beyond low-hanging fruit to tackle the difficult problems
- Reduced ramp-up time for employees in new positions
- Solving end-to-end problems
- Integrating analytics with operational systems

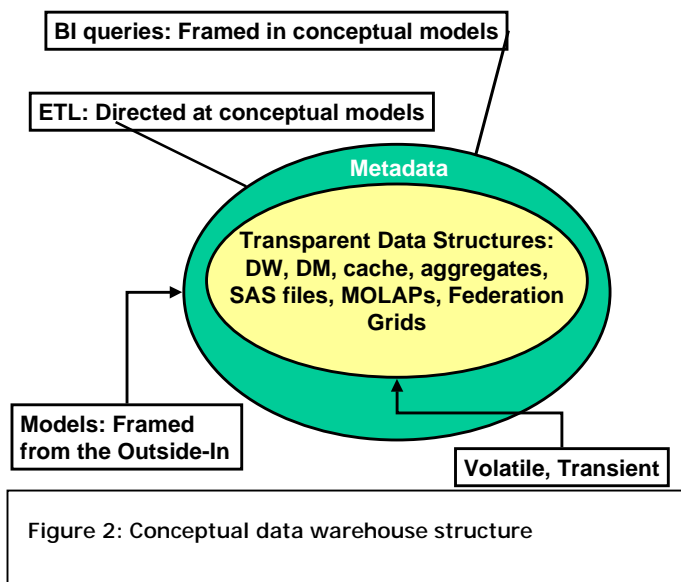
Each of these topics warrants a more thorough description and suggestions for delivering them, but in every case, the solution starts with a fundamental change in the way data warehouses are developed, using a model-driven approach.

A model-driven approach to the entire process of data integration and management, including metadata, changes everything, especially the way stakeholders view and use the utility. From the model objects a businessperson manipulates to create a report all the way back to the system of record of the data needed for the analysis, there has to be a single, coherent model that ties it all together. Second, there has to be set of services that interact with the model and work with it as a single, uniform data source. This "virtualization" layer insures that no person or process refers directly to a physical object, allowing the BI environment to be constantly tuned and optimized without an impact on processes in production.

The model-driven approach to data integration and management can't be delivered with the current ETL/data warehousing/BI vendor incumbents without drastic enhancement to their portfolios. Some new entrants to the market have thought this problem through and are now delivering products to implement this approach.

A New Architecture

Given the state-of-the-art, it's time to really rethink the whole proposition, which is a controversial concept. In Figure 2, a conceptual diagram shows that the structures of the data warehouse lose focus because the models are embedded in the metadata, not the physical



structures. The actual designs of relational databases, MOLAPS or any other structure is a secondary consideration and completely invisible to the people and processes who interact with the environment. Some of the data structures may not even be there at all, they are merely the projection of an ETL process that maps, but does not move data from it's original location. The "metamodels" are created and modified by the people and processes that use them, with the structures dynamically adapting. This is a radical change from the data warehousing concept, where all modeling was controlled strictly by technologists and how they interpreted "business

requirements," a difficult process that has proven to be less than effective in today's business climate.

Some characteristics of this environment are:

- Optimizing the use of resources across the enterprise (and beyond) by managing the materialization and federation (and everything in between) of data and making intelligent decisions of what to materialize and what to define logically, what is persistent and managing the entire process
- Virtualization - what is actually materialized in the data integration and management environment is not exposed to anything other than the data integration and management engine. All bi-directional data flow must happen through the virtualization layer, not to physical objects like data warehouses, data marts, multidimensional databases, flat files, SAS datasets and a host of other structures that are "read" or "written" to in ETL or BI applications. This preserves the value of those structures as the whole operation is migrated to the new architecture, though over time, they may change or disappear.
- BI software that aids people and processes in constructing queries, analyses and all other forms of information retrieval must no longer query physical objects, such as relational tables or their own structures (cubes, flat files, etc.). This enables all tools to utilize and contribute to the unified analytical model.
- ETL software that is tightly integrated into the architecture and operates in the same manner as the BI software, mapping to the abstraction layer, not to physical objects. This does not preclude ETL services from a third party, but it is more likely that the integration will be completely transparent when it is a part of an overall system sharing a single architecture and metadata.

The combination of these features and qualities makes it possible to vastly simplify the entire data warehousing domain. In Figure 3, a simplified diagram of this architecture is presented. At the bottom, the data sources such as databases and other disk-based files are depicted, but also some non-traditional sources are as well. Keep in mind that the one-way flow of data in traditional data warehouses is another assumption that is losing currency; these sources might just as likely be targets. Message queues, in particular, those that are carrying transactions and sub-transactions between operational systems are emerging as a vital source of near-real-time data. Web Services are also carrying vital information that is a rich source for analytical systems and the contents need to be examined and acted upon in analytical systems. In the case of message queues and web services, interoperation of analytical and operational systems depends on the ability for these enhanced "data warehouses" to make use of these information sources.

What is different in Figure 3 from a traditional data warehouse environment is that, instead of showing the data moving through the ETL process into a data warehouse, the result is merely the projection of the data onto a plane. This projection represents all of the information that is mapped through the operations of the system, whether it is actually materialized or not. Populating, or materializing, this data is entirely under the control of the managing software system. This is the essential difference of this approach, a complete virtualization of the analytical model with each of the software components working entirely through the metamodel.

The implication is that data that does not have to be passed through the data warehouse sieve but can remain in place provided it doesn't interfere with the performance of the host systems. Because the data warehouse concept was to gather as much data as possible, very little effort has gone into evaluating, from a cost/benefit perspective, whether it is worth the effort of gathering and maintaining certain data. Does anyone know what data is being used and what isn't? Accepting the inconvenience of retrieving from source in the infrequent occasions when it is needed may be a better alternative. This is just another consequence of the approach -- staging

data becomes a matter of efficiency and optimization, not driven by a pre-determined architecture and outdated concepts like "single version of the truth."

This environment is precisely the kind of environment enabled by newly emerging tools. All of the players communicate through a semantic layer, a virtualization of the data, and data is mapped from source systems to the single model, but not necessarily materialized to a data warehouse or any other kind of intermediate persistent storage. The software should provide all of the components, in a single architecture, to deliver a fully functioning architecture like the one described in Figure 3.

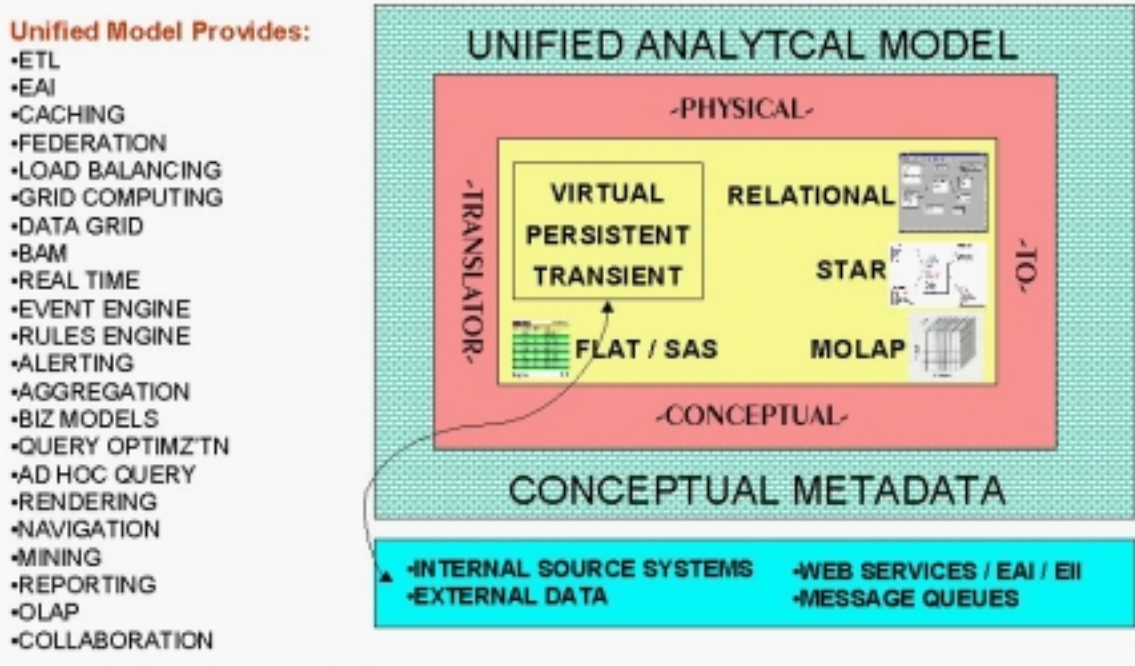


Figure 3: Unified Architecture

Data warehouses will not lose their relevance anytime soon, however. For the time being, they are the only trusted source of integrated, historical data and they dwell in their own computing environments, not affecting the operation of other systems. *Our point here is that best practices have to evolve with changing circumstances and its time for a critical look at some of our cherished ideas about data warehousing.*

Conclusion

Traditional data warehousing is not based on a model-driven architecture, and cannot provide the breath of functionality and depth of semantic richness needed by today's analytical requirements. The current standard creates a separate, partitioned universe of analytical data and operations, too rigid for rapid innovation and incapable of providing the blended analytical/operational processes that are only now emerging. All of the components in the data warehousing technology stack, ETL, databases, OLAP/Reporting/Query tools, are capable of providing this new environment, but an integrated tool that is designed to perform in this manner is clearly the best approach.

The model-driven architecture can provide three key improvements in data integration and management initiatives:

1. Propagation of BI that can leverage the investment made in data warehouses and data integration and management, by enabling more relevant functionality through continuous improvement and enhancement rather than the stepwise methods of traditional data warehousing
2. Higher-value BI resulting from greater data depth and breadth through advanced architecture, permitting people and processes access to the totality of data, models and understanding from a single point of view
3. Integration of BI with business processes through common understanding and metadata and fewer layers of integration

The unmet need for injecting analytical information into work and business processes cannot be satisfied with current information integration and management tools and methodologies. New tools are on the way to deliver a unified server to move from rigid, costly and under-performing data warehouses to a model-driven architecture that promises to vastly improve the delivery of analytics.