Beyond Business Intelligence

Barry Devlin, Ph.D., 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*. He is founder and principal of 9sight Consulting (www.9sight.com).
barry@9sight.com

Barry Devlin

Abstract

It has been almost 25 years since the original data warehouse was conceived. Although the term business intelligence (BI) has since been introduced, little has changed from the original architecture. Meanwhile, business needs have expanded dramatically and technology has advanced far beyond what was ever envisioned in the 1980s. These business and technology changes are driving a broader and more inclusive view of what the business needs from IT; not just in BI but across the entire spectrum—from transaction processing to social networking. If BI is to be at the center of this revolution, we practitioners must raise our heads above the battlements and propose a new, inclusive architecture for the future.

Business integrated insight (BI^2) is that architecture. This article focuses on the information component of BI^2—the business information resource. I introduce a data topography and a new modeling approach that can support data warehouse implementers to look beyond the traditional hard information content of BI and consider new ways of addressing such diverse areas as operational BI and (so-called) unstructured content. This is an opportunity to take the next step beyond BI to provide complete business insight.

The Evolution of an Architecture

The first article describing a data warehouse architecture was published in 1988 in the *IBM Systems Journal* (Devlin and Murphy, 1988), based on work in IBM Europe over the previous three years. At almost 25 years old, data warehousing might thus be considered venerable. It has also been successful; almost all of that original architecture is clearly visible in today's approaches.

The structure and main components of that first warehouse architecture are shown in Figure 1, inverted to match later bottom-to-top flows but otherwise unmodified. Despite changes in nomenclature, all but one of the major components of the modern data

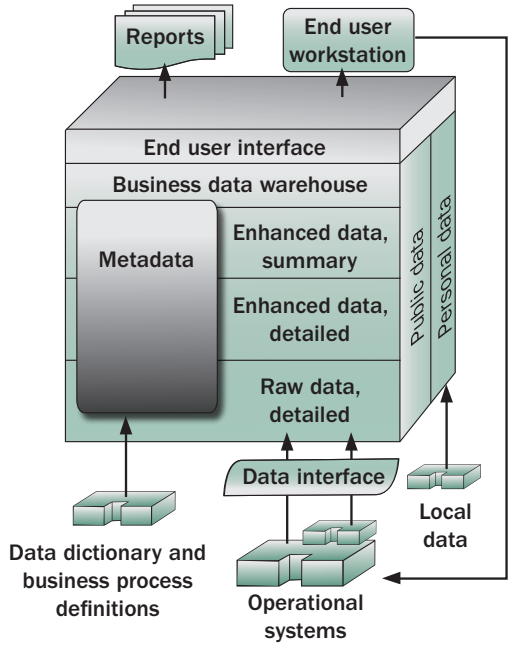

Figure 1. Data warehouse architecture, 1988

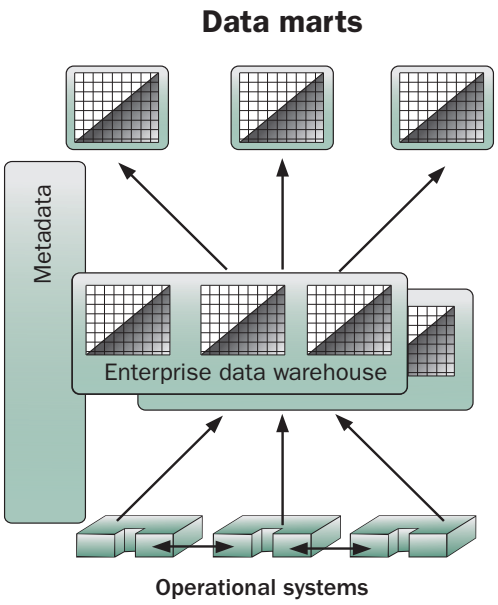

Figure 2. The layered data warehouse architecture (Devlin, 1997)

warehouse architecture appear. The data interface clearly corresponds to ETL. The business data directory was later labeled metadata. The absence of data marts is more apparent than real. The business data warehouse explicitly described data at different levels of granularity, derivation, and usage—all the characteristics that later defined data marts. The only missing component, seen only recently in data warehouses, is enterprise information integration (EII) or federated access.

Figure 1 is a logical architecture. It shows two distinct types of data—operational and informational—and recognizes the fundamental differences between them. Operational data was the ultimate source of all data in the warehouse, but was beyond the scope of the warehouse: fragmented, often unreliable, and in need of cleansing and conditioning before being loaded. The warehouse data, on the other hand, was cleansed, consistent, and enterprisewide. This dual view of data informed how decision support was viewed by both business and IT since its invention in the 1960s (Power, 2007).

A key mutation occurred in the architecture in the early 1990s. This mutation, shown in Figure 2, split the singular business data warehouse (and all informational data) into two horizontal layers—the enterprise data warehouse (EDW) and the data marts—and also vertically split the data mart layer into separate stovepipes of data for different informational needs. The realignment was driven largely by the need for better query performance in relational databases. The highly normalized tables in the EDW usually required extensive and expensive joins of such tables to answer user queries. Another driver was "slice-and-dice" analysis, which is most easily supported using dimensional models and even specialized data stores.

This redrawing of the original, logical architecture picture has had significant consequences for subsequent thinking about data warehousing. First was a level of mental confusion about whether the architecture picture was supposed to be logical or physical. Such a basic architectural misunderstanding divides the community

into factions debating the "right" architecture—recall the Inmon versus Kimball battles of the 1990s.

Second, and more important, is the disconnect from a key requirement of the original architecture: that decision-support information must be consistent and integrated across the whole enterprise. When viewed as a physical picture, Figure 2 can encourage fragmentation of the information vertically (based on data granularity or structure) and horizontally (for different organizational/user needs or divisions). The implication is that data should be provided to users through separate data stores, optimized for specific query types, performance needs, etc. Vendors of data mart tools thus promoted quick solutions to specific data and analysis needs, paying lip service—at best—to the EDW. In truth, most general-purpose databases struggled to provide the performance required across all types of queries. The EDW is often little more than a shunting yard for data on its way to data marts or a basic repository for predefined reporting.

The third, and more subtle, consequence is that thinking about logical and physical data models and storage has also split into two camps. Enterprise architecture focuses on data consistency and integrity, often assuming that the model may never be physically instantiated. On the other hand are solution developers who focus on application performance at the expense of creating yet more copies of data. The result is dysfunctional IT organizations where corporate and departmental factions promote diametrically opposed principles to the detriment of the business as a whole.

Of course, Figure 2 is not the end of the architecture evolution. Today's pictures show even more data storage components. Metadata is split off into a separate layer or pillar. The EDW is complemented by stores such as master data management (MDM) and the operational data store (ODS). Data marts have multiplied into various types based on usage, function, and data type. The connectivity of EII has been added in recent years. In truth, these modern pictures have become more like graphical inventories of physical components than true logical architectures; they have begun to look like the spaghetti diagrams beloved by BI vendors to show the current mess in decision support that will be cured by data warehousing.

This brief review of the evolution of data warehousing poses three questions:

- After 25 years of changing business needs, do we need a new architecture to meet the current and foreseen business demands?
- What would a new logical data architecture look like?
- What new modeling and implementation approaches are needed to move to the new architecture?

What Business Needs from IT in the 21st Century

The concepts of operational BI and unstructured content analytics point to the most significant changes in what business expects of IT over the past decade. The former reflects a huge increase in speed and agility required by modern business; the latter points to a fundamental shift in focus by decision makers and a significant expansion in the scope of their attention.

Speed has become one of the key drivers of business success today. Decisions or processes that 20 years ago took days or longer must now be completed in hours or even minutes. The data required for such activities must now be up to the minute rather than days or weeks old. Increasing speed may require eliminating people from decision making, which drives automation of previously manual work and echoes the prior automation of "blue collar" work. As a result, the focus of data warehousing has largely shifted from consistency to speed of delivery. In truth, of course, delivering inconsistent data more quickly is actually worse in the long term than delivering it slowly, but this obvious consideration is often conveniently ignored.

As the term "operational BI" implies, decision making is being driven into the operational environment by this trend. Participants from IT in operational BI seminars repeatedly ask: How is this different from what goes on in the operational systems? The answer is: not a lot. This response has profound implications for data warehouse

architecture, disrupting the division that has existed between operational and informational data since the 1960s. If BI architects can no longer distinguish between operational and informational activities, how will users do so?

Agility—how easily business systems cope with and respond to internal and external change—is a major driver of evolution in the operational environment. Current thinking favors service-oriented architecture (SOA) as a means of allowing rapid and easy modification of workflows and exchange of business-level services as business dictates. Such rapid change in the operational environment creates problems for data loading using traditional ETL tools with more lengthy development cycles. On the plus side, the message-oriented interfaces between SOA services can provide the means to load data continuously into the warehouse.

Furthermore, the operational-informational boundary becomes even more blurred as SOA becomes pervasive, especially as it is envisaged that business users may directly modify business processes. Users simply do not distinguish between operational and informational functions. They require any and all services to operate seamlessly in a business workflow. In this environment, the old warehousing belief that operational data is inconsistent while warehouse data is reliable simply cannot be maintained. Operational data will have to be cleansed and made consistent at the source, and as this occurs, one rationale for the EDW—as the dependable source of consistent data—disappears.

Turning to the growing interest in and importance of unstructured data, we encounter further fundamental challenges to our old thinking about decision making and how to support it. We are constantly reminded of the near-exponential growth in these data volumes and the consequent storage and performance problems. However, this is really not the issue.

The real problem lies in the oxymoron "unstructured data." All data is structured—by definition. "Structured" data, as it's known, is designed to be internally consistent and immediately useful to IT systems that record and analyze largely numerical and categorized information. Such hard information is modeled and usually stored in tabular or relational form. "Unstructured" information, in reality, has some other structure less amenable to numerical use or categorization. This soft information often contains or is related to hard information. For example, a business order can exist as: (1) a message on a voicemail system; (2) a scanned, handwritten note; (3) an e-mail message; (4) an XML document; and (5) a row in a relational database. As we proceed along this list, the information becomes harder, that is, more usable by a computer. On the other hand, we may lose some value inherent in the softer information: the tone of voice in the voicemail message may alert a person to the urgency of the order or some dissatisfaction of the buyer.

Business decision makers, especially at senior levels, have always used soft information, often from beyond the enterprise, in their work. Such information was gleaned from the press and other sources, gathered in conversations with peers and competitors, and grafted together in face-to-face interactions between team members. Today, these less-structured decision-making processes are electronically supported and computerized. The basic content is digitized, stored, and used online. Conversations occur via e-mail and instant messaging. Conferences are remote and Web-based.

For data warehousing, as a result, the implications extend far beyond the volumes of unstructured data that must be stored. These volumes would pose major problems—the viability of copying so much data into the data warehouse and management of potentially multiple copies—if we accepted the current architecture. However, of deeper significance is the question of how soft information and associated processes can be meaningfully and usefully integrated with existing hard information and processes.

At its core, this is an architectural question. How can existing modeling and design approaches for hard information extend to soft information? Assuming they can, how can soft information, with its loose and fluid structure, be mined on the fly for the metadata inherent in its content? Although these questions are not new, there is little consensus so far about how this will be

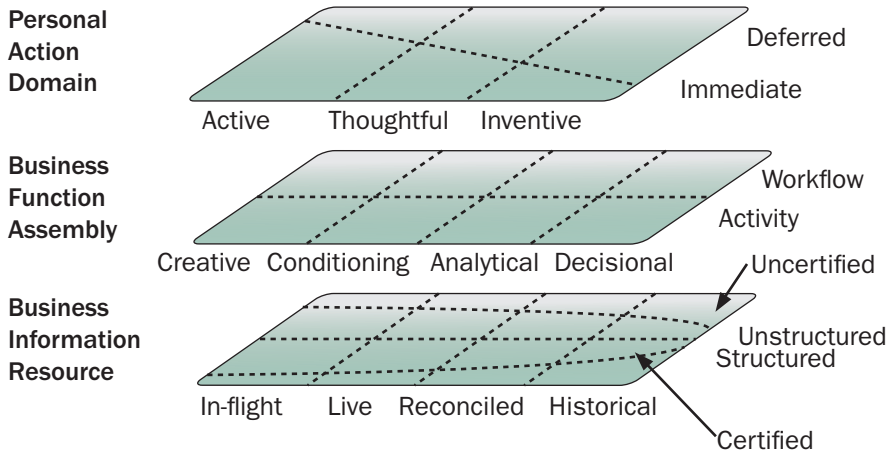

Figure 3. The business integrated insight architecture

done. As was the case for enterprise data modeling, which matured in tandem with the data warehouse architecture, methods of dealing with soft information will surface as a new architecture for life beyond BI is defined.

In the case of operational BI and SOA, the direction is clear and the path is emerging: The barrier between operational and informational data is collapsing, and improvements in database technology suggest that we can begin to envisage something of a common store. For the structured/unstructured divide, the direction is only now emerging and the path is yet unclear. However, the direction echoes that for operational/informational stores—the barriers we have erected between these data types no longer serve the business. We need to tear down the walls.

Business Integrated Insight and the Business Information Resource

Business integrated insight (BI^2), a new architecture that shows how to break down the walls, is described elsewhere (Devlin, 2009). As Figure 3 shows, this is again a layered architecture, but one where the layers are information, process, and people, and all information resides in a single layer.

As seen in the business directions described earlier, a single, consistent, and integrated set of all information used by the organization—from minute-to-minute operations to strategic decision making—is needed. At its most comprehensive, this comprises every disparate business data store on every computer in the organization, all relevant information on business partners' computers, and all relevant information on the Internet! It includes in-flight transaction data, operational databases, data warehouses and data marts, spreadsheets, e-mail repositories, and content stores of all shapes and sizes inside the business and on the Web.

This article focuses on the business information resource (BIR), the information layer in BI^2, to provide an expanded and improved view of that component of Figure 3. The BIR provides a single, logical view of the entire information foundation of the business that aims to significantly reduce the physical tendency to separate and then duplicate data in multiple stores. This BIR is a unified information space with a conceptual structure that allows for reasoned decisions about where to draw boundaries of significant business interest or practical implementation viability. As business changes or technology evolves, the BIR allows boundaries to change in response without reinventing the logical architecture or defining new physical components to simply store alternative representations of the same information.

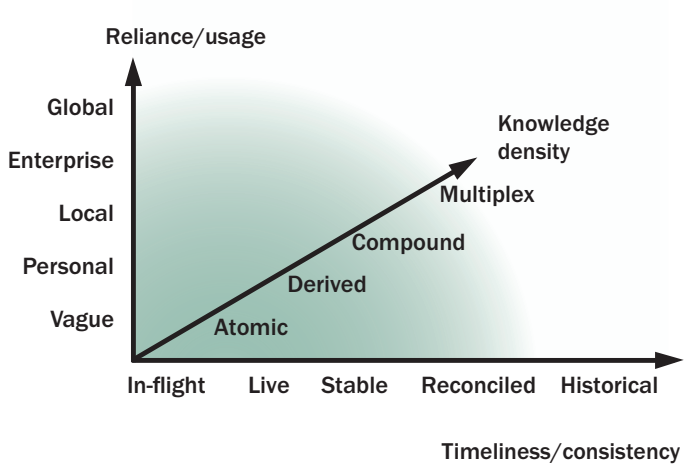

Figure 4. The axes of the business information resource

The structure of the BIR is based on data topography, with a set of three continuously variable axes characterizing the data space. Data topography refers to the type and use of data in a general sense—easy to recognize but often difficult to define. This corresponds to physical topography, where most people can easily recognize a hill or a mountain when they see one, but formal definitions of the difference between them seldom make much sense. Similarly, most business or IT professionals can distinguish between hard and soft information as discussed earlier, but creating definitions of the two and drawing a boundary between them can be problematic.

The three axes of data topography, as shown in Figure 4, provide business and IT with a common language to understand information needs and technological possibilities and constraints. Placing data elements or sets along the axes of the data space defines their business usage and directs us to the appropriate technology.

The Timeliness/Consistency Axis

The timeliness/consistency (TC) axis defines the time period over which data validly exists and its level of consistency with logically related data. These two factors reside on the same axis because there is a distinct, and often difficult, inverse technical relationship between them. From left to right, timeliness moves from data that is ephemeral to eternal; consistency moves from standalone to consistent, integrated data. When data is very timely (i.e., close to real time), ensuring consistency between related data items can be challenging. As timeliness is relaxed, consistency is more easily ensured. Satisfying a business need for high consistency in near-real-time data can be technically challenging and ultimately very expensive.

Along this axis, *in-flight* data consists of messages on the wire or the enterprise service bus; data is valid only at the instant it passes by. This data-in-motion might be processed, used, and discarded. However, it is normally recorded somewhere, at which stage it becomes *live*. Live data has a limited period of validity and is subject to continuous change. It also is not necessarily completely consistent with other live data. That is the characteristic of *stable* and *reconciled* data, which are stable over the medium term. In addition to its stability, reconciled data is also internally consistent in meaning and timing. *Historical* data is where the period of validity and consistency is, in principle, forever.

The TC axis broadly mirrors the lifecycle of data from creation through use to disposal or archival. Within its lifecycle, data traverses the TC axis from left to right, although some individual data items may traverse only part of the axis or may be transformed en route. A financial transaction, for example, starts life in-flight and exists unchanged right across the axis to the historical view. On the other hand, customer information usually appears first in live data, often in inconsistent subsets that are transformed into a single set of reconciled data and further expanded with validity time frame data in the historical stage.

It is vital to note that this axis (like the others) is a continuum. The words in-flight, live, and so on denote broad phases in the continuous progression of timeliness from shorter to longer periods of validity and consistency from less- to more-easily achieved. They are *not* discrete categories of data. Nor are there five data layers between

which data must be copied and transformed. They represent broad, descriptive levels of data timeliness and consistency against which business needs and technical implementation can be judged. Placing data at the left end of the axis emphasizes the need for timeliness; at the right end, consistency is more important.

It should be clear that the TC axis is the primary one along which data warehousing has traditionally operated. The current architecture splits data along this axis into discrete layers, assigning separate physical storage to each layer and distributing responsibility for the layers across the organization. Reuniting these layers, at first logically and perhaps eventually physically, is a key aim of BI^2.

The Knowledge Density Axis

The knowledge density (KD) axis shows the amount of knowledge contained in a single data instance and reflects the ease with which meaning can be discerned in information. In principle, this measure could be numerical. For example, a single data item, such as Order Item Quantity, contains a single piece of information, while another data item, such as a Sales Contract, contains multiple pieces of information. In practice, however, counting and agreeing on information elements in more complex data items is difficult and, as with the TC axis, the KD axis is more easily described in terms of general, loosely bounded classes.

At the lowest density level is atomic data, containing a single piece of information (or fact) per data item. Atomic data is extensively modeled and is most often structured according to the relational model. It is the most basic and simple form of data, and the most amenable to traditional (numerical) computer processing. The modeling process generates the separate descriptions of the data (the metadata) without which the actual data is meaningless. At the next level of density is derived data, which typically consists of multiple occurrences of atomic data that have been manipulated in some way. Such data may be derived or summarized from atomic data; the latter process may result in data loss. Derived data is usually largely modeled, and the metadata is also separate from the data itself.

Compound data is the third broad class on the KD axis and refers to XML and similar data structures, where the descriptive metadata has been included (at least in part) with the data and where the combined data and metadata is stored in more complex or hierarchical structures. These structures may be modeled, but their inherent flexibility allows for less rigorous implementation. Although well suited to SOA and Web services approaches, such looseness can impact internal consistency and cause problems when combining with atomic or derived data.

The final class is *multiplex* data, which includes documents, general content, image, video, and all sorts of binary large object (BLOB) data. In such data, much of the metadata about the meaning of the content is often implicit in the content itself. For example, in an e-mail message, the "To:" and "From:" fields clearly identify recipient and sender, but we need to apply judgment to the content of the fields and even the message itself to decide whether the sender is a person or an automated process.

This axis allows us to deal with the concepts of hard and soft information mentioned earlier. The KD axis also relates to the much-abused terms "structured," "semi-structured," and "unstructured." Placing information on this axis is increasingly important in modern business as more soft information is used. Given that such data makes up 80 percent or more of all stored data, it makes sense that much useful information can be found here, for example, by text mining and automated modeling tools. Just as we have traditionally transformed and moved information along the TC axis in data warehousing, we now face decisions about whether and how to transform and move data along the KD axis. In this case, the direction of movement is likely to be from multiplex to compound, with further refinement into atomic or derived. The challenge is to do so with minimal copying.

The Reliance / Usage Axis

The final axis, reliance/usage (RU), has been largely ignored in traditional data warehousing, which confines itself to centrally managed and allegedly dependable data. However, the widespread use of personal data, such as spreadsheets, has always been problematic for data management (Eckerson and Sherman, 2008). Similarly,

data increasingly arrives from external sources: from trusted business partners all the way to the "world wild west" of the Internet. All this unmanaged and undependable information plays an increasingly important role in running a business. It is becoming clear that centrally managed and certified information is only a fraction of the information resource of any business.

The RU axis, therefore, classifies information according to how much faith can be placed in it and the uses to which it can be put. *Global* and *enterprise* information is strongly managed, either at an enterprise level or more widely by government, industry, or other regulatory bodies. It adheres to a well-defined and controlled information model, is highly consistent, and may be subject to audit. By definition, reconciled and historical information fall into these classes. *Local* information is also strongly managed, but only within a departmental or similar scope. Internal operational systems, with their long history of management and auditability, usually contain local or enterprise-class data. Information produced and managed by a single individual is *personal* and can be relied upon and used only within a very limited scope. A collaborative effort by a group of individuals produces information of higher reliability and wider usage and thus has a higher position on the RU axis.

Vague information is the most unreliable and poorly controlled. Internet information is vague, requiring validation and verification before use. Information from other external sources, such as business partners, has varying levels of reliability and usage.

The placement of information on this axis and the definition of rules and methods for handling different levels of reliance and usage are topics that are still in their infancy, but they will become increasingly important as the volumes and value of less closely managed and controlled data grow.

A Note about Metadata

The tendency of prior data warehouse architectures to carve up business information is also evident in their positioning of metadata as a separate layer or pillar. Such separation was always somewhat arbitrary and is no longer reasonable. We have probably all encountered debates about whether timestamps, for example, are business data or metadata.

This new architecture places metadata firmly and fully in the business information resource for three key reasons. First, as discussed earlier, metadata is actually embedded in the compound and multiplex information classes by definition. Second, metadata is highly valuable and useful to the business. This is obvious for business metadata, but even so-called technical metadata is often used by power users and business analysts as they search for innovative ways to combine and use existing data. Third, as SOA exposes business services to users, their metadata will become increasingly important in creating workflows. Integrating metadata into the BIR simply makes life easier for business and IT alike. Metadata, when extracted from business information, resides in the compound data class.

Introducing Data Space Modeling and Implementation

The data topography and data space described above recognize and describe a fact of life for the vast majority of modern business processes: Any particular business process (or, in many cases, a specific task) requires information that is distributed over the data space. A call center, for example, uses live, stable, and historical data along the TC axis; atomic, derived, and multiplex data along the KD axis; and local and enterprise data on the RU axis, as shown in Figure 5.

Although this data space illustration provides a valuable visual representation of the data needs of the process and their inherent complexity, a more formal method of describing the data relationships is required to support practical implementation: *data space modeling*. Its aim is to create a data model beyond the traditional scope of hard information. Data space modeling includes soft information and describes the data relationships that exist within and across all data elements used by a process or task, irrespective of where they reside in the data space. To do this, I introduce a new modeling construct, the *information nugget*, and propose that a new, dynamic approach to modeling is needed, especially for soft information. It should be noted that much work remains to bring data space modeling to fruition.

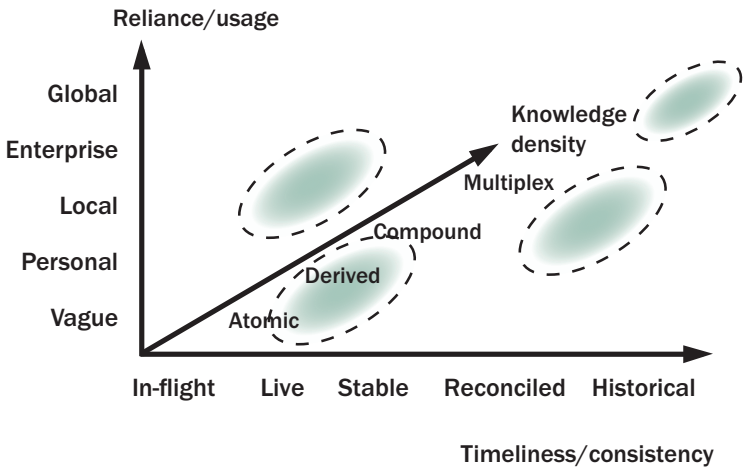

Figure 5. Sample data space mapping for the call center process

The Information Nugget

An information nugget is the smallest set of related data (wherever it resides in or is distributed through the data space) that is of value to a business user in a particular context. It is the information equivalent of an SOA service, also defined in terms of the smallest piece of business function *from a user viewpoint*. An information nugget can thus be as small as a single record when dealing with an individual transaction or as large as an array of data sets used by a business process at a particular time. As with SOA services, information nuggets may be composed of smaller nuggets or be part of many larger nuggets. They are thus granular, reusable, modular, composable, and interoperable. They often span traditional information types.

As modeled, an information nugget exists only once in the BIR, although it may be widely dispersed along the three axes. At a physical level, it ideally maps to a single data instantiation, although the usual technology performance and access constraints may require some duplication. However, the purpose of this new modeling concept is to ensure that information, as seen by business users, is uniquely and directly related to its use, while minimizing the level of physical data redundancy. When implemented, the information nugget leads to rational decisions about when and how data should be duplicated and to what extent federation/EII approaches can be used.

Modeling Soft Information

Traditional information modeling approaches focus on (and, indeed, define and create) hard information. It is a relatively small step from such traditional modeling to envision how the relationships between multiple sets of hard information used in a particular task can be represented through simple extensions of existing models to describe information nuggets. The real problem arises with soft information, particularly that represented by the multiplex data class on the KD axis. Such data elements are most often modeled simply as text or object entities at the highest level, with no recognition that more fundamental data elements exist within these high-level entities.

Returning to the call center example, consider the customer complaint information that is vital to interactions between agents and customers. When such information arrives in the form of an e-mail or voicemail message from the customer, we can be sure that within the content exists real, valuable, detailed information including product name, type of defect, failure conditions, where purchased, name of customer, etc. In order to relate such information to other data of interest, we must model the complaint information (multiplex data) at a lower level, internal to the usual text or object class.

Such modeling must recognize and handle two characteristics of soft information. First is the level of uncertainty about the information content and our ability to recognize the data items and values contained therein. For example, "the clutch failed when climbing a hill," and "I lost the clutch going up the St. Gotthard Pass," contain the same information about the conditions of a clutch failure, but may be difficult to recognize immediately. Second, because soft information may contain lower-level information elements in different instances of the same text/object entity, each instance must be individually modeled on the fly as it arrives in the store.

Automated text mining and semantic and structural analysis are key components in soft information modeling given the volumes and variety of information involved. Such tools essentially extract the tacit metadata from multiplex data and store it in a usable form. This enables multiplex data to be used in combination with the simpler atomic, reconciled, and derived classes on the KD axis. By storing this metadata in the BIR and using it as pointers to the actual multiplex data, we can avoid the need to transform, extract, and copy vast quantities of soft information into traditional warehouse data stores. We may also decide to extract certain key elements for performance or referential integrity needs. The important point is that we need to automatically model soft information at a lower level of detail to enable such decisions and to use this information class fully.

Conclusions

This article posed three questions: (1) Do we need a new architecture for data warehousing after 25 years of evolution of business needs and technology? (2) If so, what would such an architecture look like? and (3) What new approaches would we need to implement it? The answers are clear.

1. Business needs and technology have evolved dramatically since the first warehouse architecture. Speed of response, agility in the face of change, and a significantly wider information scope for all aspects of the business demand a new, extensive level of information and process integration beyond any previously attempted. We need a new data warehouse architecture as well as a new enterprise IT architecture of which data warehousing is one key part.

2. Business integrated insight (BI^2) is a proposed new architecture that addresses these needs while taking into account current trends in technology. It is an architecture with three layers—information, process, and people. Contrary to the traditional data warehouse approach, all information is placed in a single layer—the business information resource—to emphasize the comprehensive integration of information needed and the aim to eliminate duplication of data.

3. An initial step toward implementing this architecture is to describe and model a new topography of data based on broad types and uses of information. A data space mapped along three axes is proposed and a new modeling concept, the information nugget, introduced. The architecture also requires dynamic, in-flight modeling particularly of soft information to handle the expanded data scope.

Although seemingly of enormous breadth and impact, the BI^2 architecture builds directly on current knowledge and technology. Prior work to diligently model and implement a true enterprise data warehouse will contribute greatly to this important next step beyond BI to meet future enterprise needs for complete business insight. ■

References

Devlin, B. [1997]. *Data Warehouse: From Architecture to Implementation*, Addison-Wesley.

——— [2009]. "Business Integrated Insight (BI^2): Reinventing enterprise information management," white paper, September. http://www.9sight.com/bi2_white_paper.pdf

———, and P. T. Murphy [1988]. "An architecture for a business and information system," *IBM Systems Journal*, Vol. 27, No. 1, p. 60.

Eckerson, Wayne W., and Richard P. Sherman [2008]. *Strategies for Managing Spreadmarts: Migrating to a Managed BI Environment*, TDWI Best Practices Report, Q1. http://tdwi.org/research/2008/04/strategies-for-managing-spreadmarts-migrating-to-a-managed-bi-environment.aspx

Power, D. J. [2007]. "A Brief History of Decision Support Systems," v 4.0, March 10. http://dssresources.com/history/dsshistory.html