VOLUME 23 NUMBER 1

Business Intelligence Journal

THE LEADING PUBLICATION FOR DATA MANAGEMENT AND ANALYTICS

Thirty Years of Data Warehousing

Dr. Barry Devlin

Successful Analytics Leaders

Hugh J. Watson

BI Expert's Perspective: Have We Finally Gotten Self-Service Right?

Wayne Eckerson, Troy Hiltbrand, Mark Jackson, and Coy Yonce

tdwi Transforming Data With Intelligence™

VOLUME 23 • NUMBER 1

Business Intelligence Journal

THE LEADING PUBLICATION FOR DATA MANAGEMENT AND ANALYTICS

Thirty Years of Data Warehousing

By Dr. Barry Devlin

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.
barry@9sight.com

ABSTRACT

Since its first formal, public description in 1988, the data warehouse architecture has successfully provided the foundation for decision-making support across enterprises in every industry. With two main interpretations, the architecture has remained stable since the 1990s and has only recently been challenged by the data lake concept. This article traces the early history and drivers of the data warehouse before pivoting to discuss the data lake and its implications for the original architectural approach. This leads to a proposal: a production analytics platform that positions the data lake and warehouse, showing how to begin to dismantle the old operational/informational divide. This platform can extend the value of the data warehouse architecture for at least another decade.

In 1988, a cell phone weighed approximately two pounds, cost nearly $4,000, and offered 30 minutes of talk time—and no other function—having taken some 10 hours to charge. The Motorola DynaTAC 8000X was broadly known as "the brick" and was wildly popular eye-candy among the jet set of the time.

The same year, your office desk might have been adorned with an Apple Macintosh II or an IBM PS/2 with 512K of memory, a 20MB hard drive—if you were lucky—and a choice of a monochrome or color monitor with an eye-catching resolution of 640x480 pixels.

Of more interest to BI professionals would have been the Teradata DBC/1012, a massively parallel processing database system with a maximum of 1,000 processors and five terabytes of disk storage at the very top of the range. Few customers reached

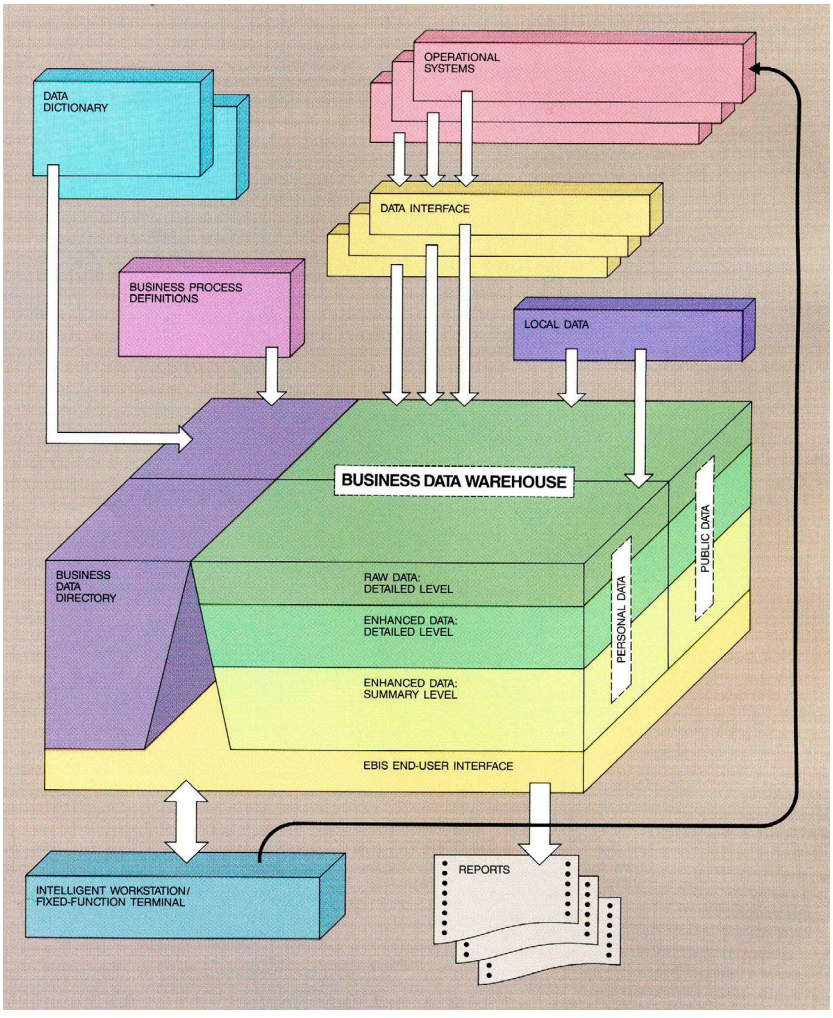

Figure 1: Overview of the EMEA Business Information System (Devlin and Murphy, 1988).

those dizzying numbers, though; the costs were eye-watering. Although Teradata is now almost synonymous with data warehousing, its marketing material spoke only of a "Data Base Computer System" because the phrase *data warehouse* was just about to be unleashed.

ENTER THE DATA WAREHOUSE

Thirty years ago in February 1988, the *IBM Systems Journal* published the first description of a data warehouse architecture, written by myself and Paul Murphy. Entitled "An Architecture for a Business and* Information System," the article summarized architectural work carried out at IBM leading to "the EMEA [Europe, Middle East, and Africa] Business Information System (EBIS) architecture as the strategic direction for informational systems [that proposed] an integrated warehouse of company data based

firmly in the relational database environment (Devlin and Murphy, 1988)."

The key architectural figure of the article, reproduced in Figure 1, would be instantly recognizable by any data warehouse practitioner today. Front and center is the business data warehouse, a single logical repository containing public and personal data at raw, detailed, and summary levels from operational and local (personal) systems. This data is described in a business data directory sourced from a data dictionary and business process definitions.

Data is made available to business users via workstations and reports. Key components of the architecture, including the update strategy and user access, are described in some detail later in this article. The structure of the data in the business data warehouse is also illustrated as a conceptual set of tables—such as customers, employees, products, orders, and so on—matching the user's perception of the subset of business information to which he or she needs access.

Embedded in this architecture, but seldom discussed, is the postulate that operational and informational systems are separated for business and technical reasons. I will return to this assumption later in this article when I discuss the future of the data warehouse.

IBM rolled the concepts of this architecture into the IBM Information Warehouse Framework in 1991, focused on a proprietary approach and trademarking the term *information warehouse*, IBM missed the opportunity to define and monetize the market for decision-making support systems based on relational databases. In retrospect, that trademark is unfortunate.

Information is surely more representative of what a warehouse should deliver, but we are left with *data warehouse* for general use and it was this term that Bill Inmon popularized in the early 1990s (Inmon, 1992).

WHAT IS A DATA WAREHOUSE?

It was Inmon who introduced the oft-quoted definition of a data warehouse: a subject-oriented, nonvolatile, integrated, time-variant collection of data in support of management decisions. The four data characteristics of this simple and memorable definition are implicit in the EBIS architecture and are emergent in Inmon's 1992 book. As the basis for much thinking about what a data warehouse looks like, these terms bear closer consideration.

- **Subject-oriented:** On the one hand, this corresponds to the idea introduced in the EBIS architecture and echoed by Inmon that the data should be represented in terms and structures, such as customer, product, order, transaction, etc., that are familiar to business people. On the other hand, a more formal interpretation aligns the term and structure to the enterprise data model described by John Zachman (Sowa and Zachman, 1992). These two views map to the perceived primary purpose of the data warehouse: the subject-oriented view is framed to directly support decision makers while an enterprise data model focuses on integrating data from diverse sources.
- **Integrated:** This characteristic springs from the understanding that data extracted from diverse operational sources may be inconsistent (and occasionally incoherent) for reasons of meaning (for example, different

definitions of profit) or timing (time zones or other reasons). Integration means reconciling these differences in various ways to deliver a "single version of the truth" (SVOT) that can be used across the enterprise. SVOT has long been recognized as an ideal that is unachievable in practice. However, integration of key data to common standards remains an important goal for data warehousing that has been underemphasized in the data lake approach.

- **Nonvolatile:** In simple terms, this reflects a long-standing business need to be able to recreate a business situation as of a particular date and time in the past, either for reporting or as a basis for what-if simulations. Therefore, unlike many operational systems and modern external data sources, a data warehouse must maintain an ongoing and stable record of both current and historical data states. Ideally, data is never deleted.
- **Time-variant:** All data records in the warehouse are timestamped. This is a consequence of nonvolatility. However, the exact nature of the timestamp has long been debated. The simplest approach is to record the time of loading into the warehouse. In general, this is insufficient for most business purposes. Therefore, bitemporal and, more recently tri-temporal, schemata—where each record carries multiple timestamps—have been implemented or promoted in the past decade to provide more and better ways of analyzing and using data over time (Johnston, 2014).

As experienced data warehouse practitioners are aware, these characteristics are neither complete nor fully congruent. At a high level they provide welcome guidance for design. Nonetheless, every data warehouse implementation ends up balancing them against one another and trading them off against business needs and the limitations of chosen or required technology.

One example of such a trade-off involves simplifying integration by focusing on a subject-oriented data warehouse for a single department, perhaps better called a data mart. Another example is the dimensional data warehouse, discussed in the following section, where trading business demands for early delivery are accommodated by redefining the concept of subject orientation.

This situation continues to this day. Despite claims to the contrary, data lakes do not eliminate the need for these compromises and in some cases, promote practices that undermine the four characteristics and introduce new challenges, as discussed in the section "Diving into the Data Lake."

COMPETING DATA WAREHOUSE STRUCTURES

What is missing from the EBIS architecture—and in Inmon's early book as well—is the hub-and-spoke structure of a centralized enterprise data warehouse (EDW) that provides the reconciliation point for data from diverse sources feeding multiple departmental data marts.

This structure, often referred to as the *Inmon data warehouse*, arose first from technological necessity. General-purpose relational databases in the 1990s weren't powerful enough to handle multiple concurrent user queries with varying data needs against a single, enterprise-level, subject-oriented database. One solution—and the solution that stuck most closely to the intent and principles of the EBIS architecture—was to

split the data warehouse into two (or sometimes more) layers. Data was integrated and reconciled in the EDW and then distributed to business users in more query-friendly, departmentally focused data marts, as shown in Figure 2.

This approach comes with two major challenges. First, its layering implies that at least some part of the EDW—often found to be quite a large part—must be built before any data marts can be delivered. Business needs come later, conceptually at least, and careful project management is a prerequisite to balancing business demand with the challenging reality of data diversity. Second, data must be moved sequentially from layer to layer, so each additional layer delays the arrival of data to where it's needed. With timeliness of decisions increasingly important, such delays are unwelcome.

Ralph Kimball took a different approach to solving these build and runtime delays. He adopted a different data model and database structure that was optimized for the most common type of analysis: slice-and-dice and drill-down. This approach is the dimensional or star-schema data warehouse (Kimball, 1996). Kimball starts from the immediate analysis needs of departmental business processes to create a performant database consisting only of relevant facts and dimensions. Departmental-level star schemas are subsequently related via conformed dimensions.

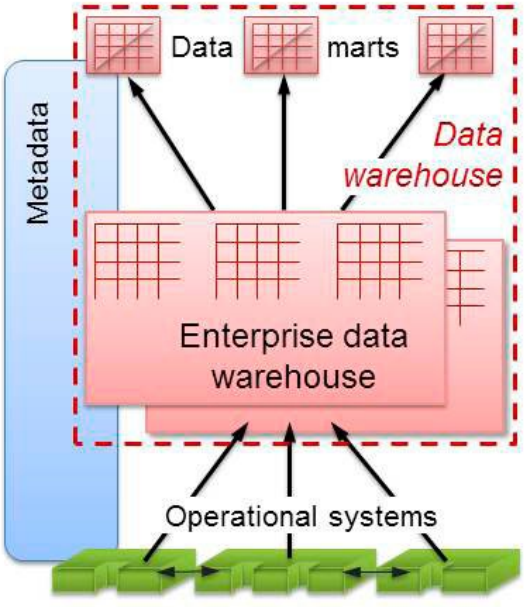

Figure 2: The layered data warehouse architecture (Devlin, 1997).

By the turn of the century, the debate between these two approaches had allegedly turned into a war (Breslin, 2004). Each approach has its strengths and weaknesses. In some cases, a hybrid approach can be taken, where the data marts are dimensional and fed from a reconciliation layer in the EDW.

A more recent development, the *data vault*, offers a balanced hybrid of the layered and star schema forms above (Linstedt and Olschimke, 2015). Consisting of a data model, methodology, and systems architecture, it provides a design basis for data warehouses that emphasizes the core data quality, consistency, and agility.

DIVING INTO THE DATA LAKE

The data warehouse is an architecture of its era. When it was designed and until the early 2000s, its main source of data by far was operational systems that managed the business processes of the enterprise. Such *process-mediated data* was (and continues to be) defined, structured, and managed within the enterprise (Devlin, 2013). As a result, it is generally well-governed and limited in scope and size. The data warehouse architecture is optimized for data with these characteristics.

However, the Internet changed the playing field, possibly forever. By the early 2000s, new types of data were blossoming in ever-increasing volumes on the Internet and at its interface to the enterprise. Businesses saw opportunities bloom and threats multiply. Collecting and using this data became an obsession, harvesting from clickstreams, social media, and—more recently—the Internet of Things (IoT). Relational databases could not handle data at such size or speed. Upfront modeling had to be replaced by schema-on-read. The data warehouse was obsolete. Enter the data lake.

In a 2010 blog, James Dixon, then CTO of Pentaho, declared, "If you think of a data mart as a store of bottled water—cleansed and packaged and structured for easy consumption—the data lake is a large body of water in a more natural state. The contents of the data lake stream in from a source to fill the lake, and various users of the lake can come to examine, dive in, or take samples (Dixon, 2010)."

Since then, data lakes have garnered widespread mindshare. Analysts, consultants, and vendors alike promote the concept. Surveys reveal that enterprises in every industry are implementing them, often declaring them to be a replacement for their existing data warehouses.

DEFINING THE DATA LAKE

Given the watery metaphor, it may be unsurprising that the definition of a data lake has remained fluid in its eight year life. Gartner's definition is a case in point: "A data lake is a collection of storage instances of various data assets additional to the originating data sources. These assets are stored in a near-exact, or even exact, copy of the source format. The purpose of a data lake is to present an unrefined view of data to only the most highly skilled analysts, to help them explore their data refinement and analysis techniques independent of any of the system-of-record compromises that may exist in a traditional analytic data store (such as a data mart or data warehouse)." (Gartner, 2015, emphasis in original)

This definition of a data lake—and many similar ones—offers little of substance on which to base a solid reference architecture describing

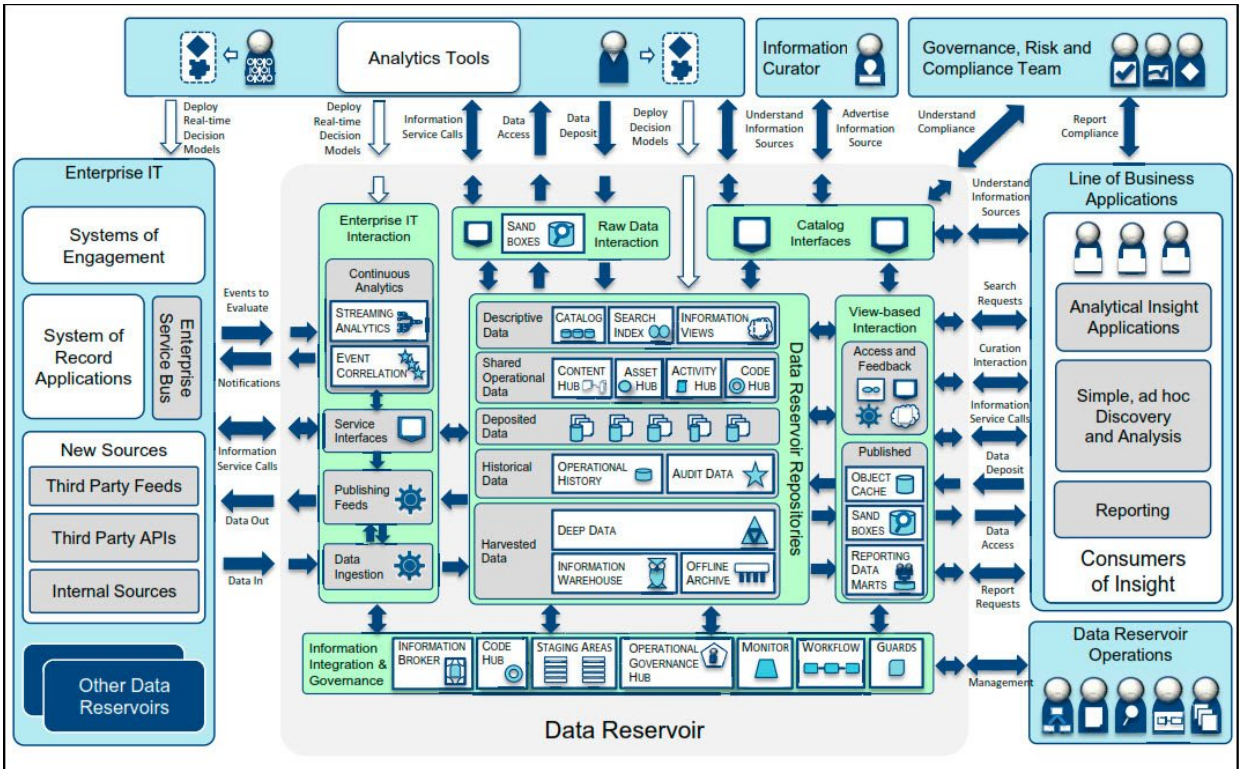

Figure 3: Summary of the components of a data reservoir (Chessell et al, 2015).

mandatory functions, components, interactions, and so on. Architectures thus range from the all-inclusive to the poetic.

At the comprehensive end of the spectrum, IBM defines an architecture for a *data reservoir*—a less popular name for a data lake that suggests more engineering—consisting of six major subsystems: data reservoir repositories, enterprise IT interaction, information integration and governance, raw data interaction, catalog interfaces, and view-based interaction (Chessell, et al, 2015). More than 30 components are documented within these subsystems, as shown in Figure 3. The result is a system of such broad scope that it even includes IBM's information warehouse.

In a less daunting and more illustrative take on the data lake, Bill Inmon offers, "The data lake needs to be divided into several sections, called data ponds. There is the raw data pond, analog data pond, application data pond, textual data pond, and archival data pond… [all of which] require conditioning in order to make the data accessible and useful (Inmon, 2016)."

DEFINING THE SHORELINE OF A DATA LAKE

The origins of the data lake concept can be traced back to the data flowing from Internet-related sources into enterprises—a stream that grew rapidly into a torrent—with the advent of the Web. From the earliest days, it was evident that a place was needed to store this raw data and analyze it at detail and summary levels

in support of business needs. It was also clear that the characteristics of such data—usually described via the 3 Vs (volume, velocity, and variety)—made it incompatible with existing data architectures and the common storage and analytics technologies of the time.

Open source technology, such as Hadoop and associated systems, emerging in the mid-2000s displayed several characteristics that made it an ideal candidate to meet these storage and analytics needs. Horizontal scaling on commodity hardware offered voluminous storage and scalable processing at low cost. Because the systems were largely file-based, rather than databases, the data could be stored and processed in its raw form without the need for upfront modeling and design work.

The programmatic and procedural approach to data processing (as opposed to the declarative approach at the heart of relational database technology and long favored by data management professionals) was attractive to the early adopters of the technology, because of their strong engineering backgrounds.

Taken together, these requirements and technology characteristics clearly indicated the need for a new system in the IT environment. In 2010, that system came to be included within the concept of James Dixon's data lake. However, as defined, the data lake was not limited to this new Internet-related "big data." Rather it was applied to all data of interest to analytic users, including data traditionally processed and made available through data marts and, by implication, the data warehouse.

At first sight, this may appear reasonable. The analytics needed for Internet-related data is similar to that required for traditional, internally sourced data. Many applications demand that both types of data be linked together for analysis. However, in my opinion, this focus on user needs unfortunately misses key differences between these data sources.

In my opinion, the data lake should have been reserved exclusively for Internet-related data.

Internally sourced, process-mediated data is central to business operations and is therefore relatively well governed, modeled, and intricately interlinked. The data warehouse/marts system through which it was made available to business people already existed and was optimized for such data characteristics.

In contrast, Internet-related data is messier, poorly described, often ancillary to core business processes, and comes from diverse and unrelated sources. Furthermore, although many proponents want to store it indefinitely—just in case it might be of use some time—it is generally most useful only in the short term.

In architectural terms, these differences in usage and characteristics strongly suggest that these two classes of data be stored and processed separately according to their differing needs before being made available, either separately or jointly, to business people for analysis and decision making. The system for internally sourced data has existed for 30 years; the data warehouse and data marts have proven successful simply through their longevity.

In my opinion, the data lake should have been reserved exclusively for Internet-related data. For clarity, I will use the term *data lough* (the Irish for lake and pronounced, as in Scottish, *loch*) to refer to a data lake used exclusively for Internet-related data in the remainder of this article.

The key drivers of such a data lough are:

- To provide cost-effective storage for raw Internet-sourced data in large volumes and at speed
- To enable high-speed processing and ad hoc analysis of such data
- To support appropriate management and governance of this data commensurate with its value
- To offer a facility for refining, modeling, and summarizing this data and the ability to link it with process-mediated data in the operational and data warehouse environments

Although the prime purpose is to support Internet-related data, it is also recognized that these attributes may benefit traditional data and systems. For example, offloading seldom used, cold data from a data warehouse to a data lake can provide improved return on investment for both systems. Similarly, running data preparation tasks in the data lake may be beneficial.

INTO THE DATA WAREHOUSE'S FUTURE

The death of the data warehouse has been proclaimed many times and with increasing frequency in this decade as the data lake has become increasingly popular. To paraphrase Mark Twain's alleged riposte, "Reports of its death have been greatly exaggerated."

Although the story of the data warehouse over the past 30 years has been marked by failures as well as successes, its current standing is a testament to the strength of the original architectural thinking and the support and extension of the architecture by many practitioners over the decades.

Although the story of the data warehouse over the past 30 years has been marked by failures as well as successes, its current standing is a testament to the strength of the original architectural thinking.

When dealing with traditional process-mediated data, the data warehouse architecture as updated in 1997 continues to satisfy the majority of business intelligence needs (Devlin, 1997). However, it is equally clear that the data warehouse cannot address all the informational needs of the modern digital business, particularly as they pertain to Internet-related data. A comprehensive extension of the original data warehouse is provided in *Business unIntelligence* (Devlin, 2013). Although the term data lake didn't appear there, Internet-related data was dealt with extensively under the topics of machine-generated data and human-sourced information. The resulting architecture was detailed at conceptual (IDEAL) and logical (REAL) levels, and also described in a previous *Business Intelligence Journal* article (Devlin, 2015).

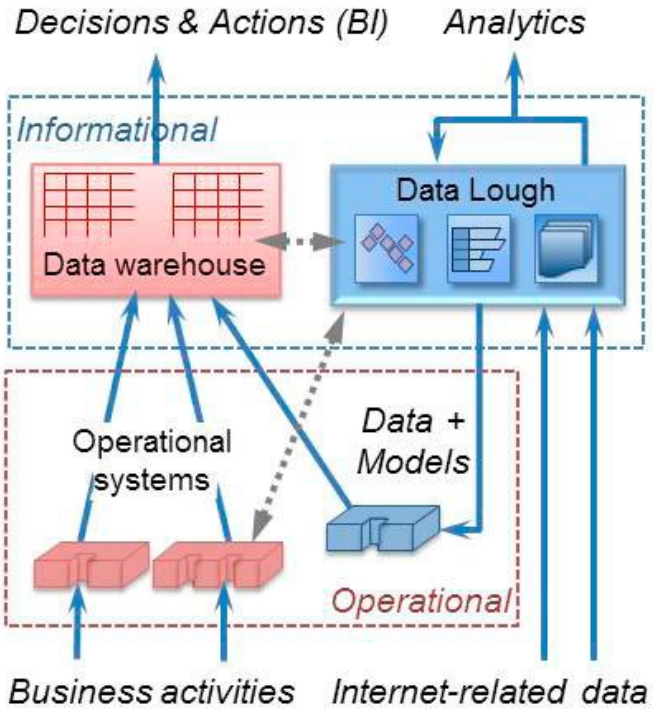

Figure 4: Positioning the data warehouse and data lough today.

The IDEAL and REAL architecture offers a longer-term vision of information preparation and use in the enterprise and in the world at large. In the shorter term, a simple picture that positions the data warehouse and the data lough can clarify much of the current confusion in the industry about both concepts, as shown in Figures 4 and 5.

As previously observed, today's IT landscape is first characterized by a division between operational ("run the business") and informational ("manage the business") concerns. The traditional data warehouse architecture, shown in simplified form on the left of Figure 4, strongly supports this long-standing view, with a separate layer of operational systems feeding process-mediated data into the warehouse. The decisions and actions output from the data warehouse are traditionally labeled BI.

On the right, the data lough receives raw Internet-related data—both machine-generated and human-sourced—as the basis for analytics. Although commonly perceived as an informational environment, deeper examination shows that a significant backflow of data and models into the operational environment is required

to deliver real-time operational analytics. In addition, as depicted by the gray, dashed arrows, substantial bidirectional data sharing is required between the data lough and the operational systems and between the data lough and the warehouse to facilitate analytics and BI.

These data flows and interactions between the data lough and the operational world, in addition to long-standing timeliness challenges encountered in operational BI, suggest that the original separation of operational and informational activities should be reconsidered. I've previously raised the possibility of reuniting the data warehouse with operational systems (Devlin, 2013). Indeed, the direction can be seen in practice in SAP HANA, for example. The growth of operational analytics driven from the data lough further extends the need to explore if and how such reunification could be achieved.

Figure 5 offers a new vision for the short-to-medium-term future that positions the data lake/lough and begins to reunite operational and informational processing, taking advantage of technology advances in both relational and nonrelational tooling.

On the left of the diagram, the data warehouse and some to-be-determined portion of the operational systems have been integrated into

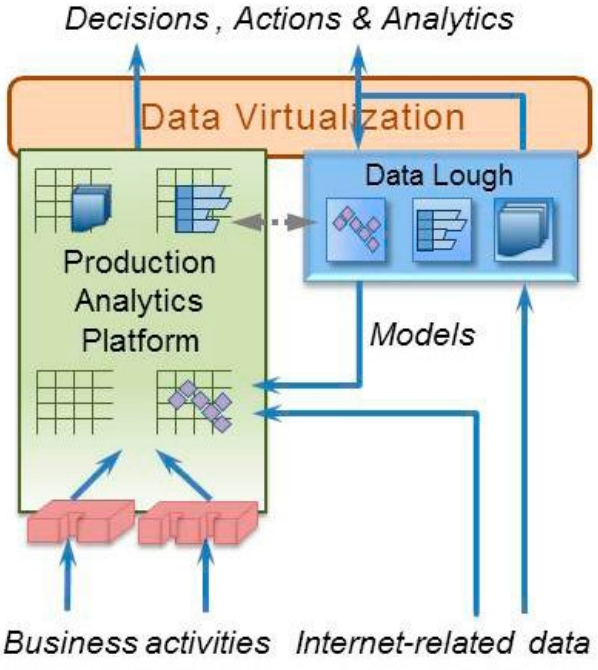

Figure 5: The production analytics platform.

an extended relational environment called the production analytics platform. This integration is facilitated by advances in relational database technology such as solid-state disks, in-memory databases, and multicore parallel processing. In addition, relational databases increasingly support storage and processing of nonrelational data formats—for example, JSON, Graph, and so on. These extensions allow selected analytics tasks to be migrated from the data lough, particularly those that demand reliability, availability, maintainability, and performance levels compatible with production use for daily operational decisions, as well as those that require access to traditional process-mediated data.

There is minimal change to the data lough itself. However, the feed of Internet-related data is split between that used in analysis and model generation (which continues to go to the data lough) and the real-time data used when models are put into production.

Where the data required for a particular analysis or decision is spread across the two environments, data virtualization allows joining of the required data at runtime.

Figure 5 does not claim to be a completely defined architectural vision for the evolution of the data warehouse. Instead, it is offered as a starting point for discussion about how concepts such as data lakes and combined operational-informational systems can extend the traditional data warehouse architecture.

FINAL THOUGHTS

At age 30, the data warehouse architecture has displayed remarkable longevity. Although the original architecture provided a strong foundation, much credit also goes to those who promoted it, built upon it, extended it, restructured it, and more. I mentioned some of the better known here: Inmon and Kimball, of course. Others were omitted for lack of space: Hans Peter Luhn's original definition of business intelligence in 1958 and Howard Dresner's reintroduction in the 1990s; Claudia Imhoff's corporate information factory, also in the 1990s; and others to whom I apologize for their omission. Vendors such as Teradata and IBM contributed powerful technologies, and, of course, there are the thousands of architects in enterprises, consulting firms, and service providers who labored at the coal face of implementation—who corrected and added to the architecture in the process.

The data warehouse architecture lives on, extended with the concepts of analytics and data lakes. Artificial intelligence and the Internet of Things will drive further growth. The "Business unIntelligence" architecture shows the shape of the longer-term evolution, providing a template for all types of information in every possible business usage, as enterprises pursue extensive augmentation and automation of decision making. The data warehouse, as the repository of integrated, core business information, will continue to beat at the heart of current and future digital transformation. ●

REFERENCES

Breslin, Mary [2004]. "Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models," *Business Intelligence Journal*, Winter, pp. 6–20.

Chessell, Mandy; Nigel L. Jones; Jay Limburn; David Radley; and Kevin Shank [2015]. *Designing and Operating a Data Reservoir*, IBM Redbooks. http://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg248274.html

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

Devlin, Barry A. [2013]. *Business unIntelligence*, Technics Publications. https://technicspub.com/business-unintelligence/

Devlin, Barry A. [2015]. "From Layers to Pillars—A Logical Architecture for BI and Beyond," *Business Intelligence Journal*, Vol. 20, No. 2, pp. 14–22.

Devlin, Barry A., and Paul T. Murphy [1988]. "An Architecture for a Business and Information System," *IBM Systems Journal*, Vol. 27, No. 1, pp. 60–80. http://www.9sight.com/1988/02/art-ibmsj-ebis/

Dixon, James [2010]. "Pentaho, Hadoop, and Data Lakes," blog post, Jamesdixon.wordpress.com, October. https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/

Gartner [2015]. "Data Lake," Gartner IT Glossary, accessed February 25, 2018. https://www.gartner.com/it-glossary/data-lake/

Inmon, William H. [1992]. *Building the Data Warehouse*, QED Information Sciences.

Inmon, William H. [2016]. *Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump*, Technics Publications.

Johnston, Tom [2014]. *Bitemporal Data: Theory and Practice*, Morgan Kaufmann.

Kimball, Ralph [1996]. *The Data Warehouse Toolkit*, Wiley and Sons.

Linstedt, Daniel, and Michael Olschimke [2015]. *Building a Scalable Data Warehouse with Data Vault 2.0*, Morgan Kaufman.

Sowa, John F., and John A. Zachman [1992]. "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 31, No. 3, pp. 590-616.