DW 2.0 represents a long term architectural blueprint

A tale of two architectures (1)

W. H. Inmon

"It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, …" Charles Dickens; "A Tale of Two Cities".

It was the best of times. It was the worst of times. From an age of applications and the confusion over application based information in the corporation arose the concept of a data architecture and data warehousing. Into the miasma came Bill Inmon's best selling book – "Building the data warehouse". And there was Kimball's software company – RedBrick Systems. And soon the world of data warehousing was born. It was the late 1980's and the world was about to witness the rise of analytical processing, business intelligence and a whole host of technologies never before seen that would change the world forever.

The corporate information factory data warehouse

The industry accepted definition of a data warehouse – "a subject oriented, integrated, non volatile, time variant collection of data for management's decision making" – appeared in "Building the data warehouse". Later books by Inmon soon appeared which described the architecture into which the data warehouse fit. The architecture – sometimes called the "corporate information factory" (or simply "Inmon's architecture") is seen in a simple form in Fig 1.

Single version of the truth

The nexus of the corporate information factory and its foundation – the data warehouse – is the notion of the single version of the truth. Centered in the data warehouse and described by the definition of the data warehouse is the granular, integrated historical data – the "single version of the truth" – which is the essence of the corporate information factory. With the corporate information factory, the data warehouse has a place where there is a "final word" as to what data is right and what data is wrong. At the heart of the confusion over information that preceded the data warehouse is the inability of the organization to understand what data is correct and what data is not correct. It is hard to make proper decisions on data that is unreliable. Prior to the corporate information factory, organizations had a plethora of data, but they had no idea what data was correct and what data was incorrect. With the corporate information factory, there was a definitive source of data to which the corporation could turn – the "single version of the truth". It is true that the corporate information factory solved many other problems. But the single most important aspect of the corporate information factory was that it contained the "single version of the truth."

The corporate information factory includes an architecture that centers around the data warehouse. It is in the data warehouse where the "single version of the truth" resides. Other features of the corporate information factory architecture include legacy, operational systems, ETL and data marts. ETL is the technology that reads in raw data from applications and writes out corporate data (or data that constitutes the "single version of the truth"). Data marts are those data bases created for the analytical needs for different departments and different groups of people doing analytical processing. In the corporate information factory the only source of data for the data marts is the data warehouse.

The biggest issue in creating the corporate information factory is that of the integration of application data into corporate data. Data that comes from applications must be recast into a corporate form and structure. That is how the "single version of the truth" is created. The integration of old legacy, operational unintegrated data is a complex and time consuming job. In many cases, old legacy data is undocumented. In many cases old legacy data lies in technologies that have been unsupported for years. In many cases old legacy applications must be merged where a merger of application data was never an objective of the designer of the legacy application. In many cases the very definition of data

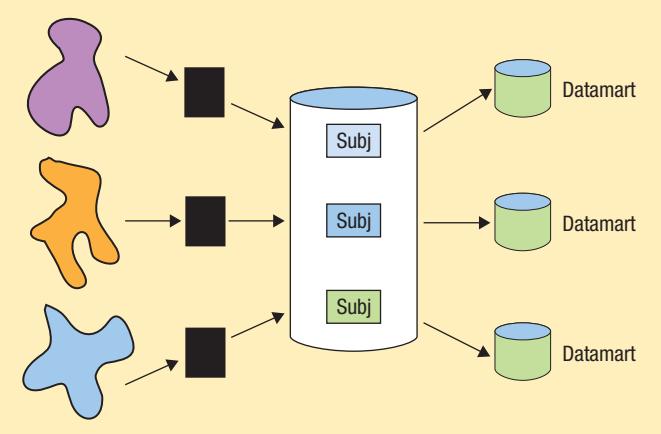

Figure 1.

sitting in an old application must be recast. All of the work required for integration is tedious and must be performed in a disciplined and in an exact manner. As such, building a data warehouse for the corporate information factory is not an easy or a fast thing to do. But the result is integrated data – a "single version of the truth" for the organization.

The focus of the corporate information factory is data across the enterprise. Data from many different places and applications – the "legacy systems environment" is all integrated and included into the data warehouse. One of the reasons why the corporate information factory is not built quickly is that data from lots of places needs to be integrated. For a small organization there may be very little integration that needs to occur. But for a large organization the process of integration can be laborious, tedious, and time consuming.

As a rule, the data in the corporate information factory data warehouse is stored in a normalized relational format. Generally speaking, the data in the corporate information factory relational data base is granular, historical and is "lightly" denormalized. Stated differently, building the corporate information factory is a long term proposition and the result is a long term infrastructure that the corporation can rely upon.

The corporate information factory (and its evolved form – DW 2.0) is the architecture that is espoused and developed by Bill Inmon and expressed as the corporate information factory in his book in 1999 and later in the book "DW 2.0 – Architecture for the next generation of data warehousing".

The dimensional model data warehouse – the Kimball approach

But there was another related architecture that arose in roughly the same time frame. That architecture is the one that can be called the "Kimball" architecture. It is the Kimball architecture that is associated with Red Brick Systems. The Kimball architecture has evolved over time, like all architectures evolve. The first stage of evolution of the Kimball architecture began with what is known as a "dimensional" (or star schema) architecture. In the context of this paper we will call the first stage of evolution of the Kimball architecture a "simple" dimensional model. Fig 2 shows the bare bone essence of a simple dimensional architecture.

Fig 2 shows a fact table surrounded by several dimensions. In general, the facts are a cluster of attributes that are physically colocated and the dimensions are the separate tables that describe the facts. The fact table and its dimensions form what is termed a "star schema". As a rule there are many facts in the fact tables and relatively few occurrences of data in the dimensions. The data that comes from applications is placed in a star schema and is used to create what is termed a "data mart".

The data that populates the simple dimensional model comes directly from applications. In fact, Kimball draws a diagram that shows how application data enters the simple dimensional model. The diagram is taken from an article published by Kimball in 2004, along with Margy Ross [1]. Fig 3 depicts the diagram showing how the simple dimensional model is used to produce multiple data marts from multiple sources.

In Fig 3 it is seen that there are legacy applications and data marts in the simple dimensional model. The many different data marts are populated directly from the many different applications. Kimball goes on to give his definition of a data warehouse. Kimball's definition relates to the first phase of the Kimball architecture – the simple dimensional model – "a data warehouse is nothing more than the union of the data marts." [3] Kimball refined the definition of the data warehouse at a later point in time, saying that the definition of a data warehouse was a "a copy of the data specifically structured for query and analysis." [2] It is easy and fast to merely copy data from one data base to the next.

Kimball's Stage 1 simple dimensional architecture was never designed for enterprise integration. The Kimball Stage 1 simple dimensional architecture was designed for immediate applications and immediate data marts, where the scope of the effort was limited. Because the scope of the Kimball Stage 1 simple dimension architecture was limited and because only the copying of data was involved, Kimball's Stage 1 simple dimensional data warehouse is fast and easy to construct.

The biggest selling point of the Kimball simple dimensional architecture is the speed with which the data marts can be constructed. Indeed, around the world, people like architectures that

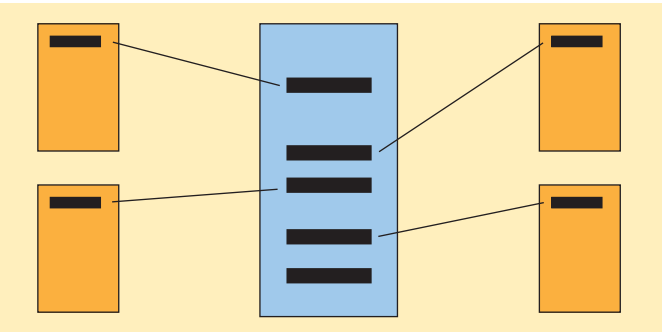

Figure 2.

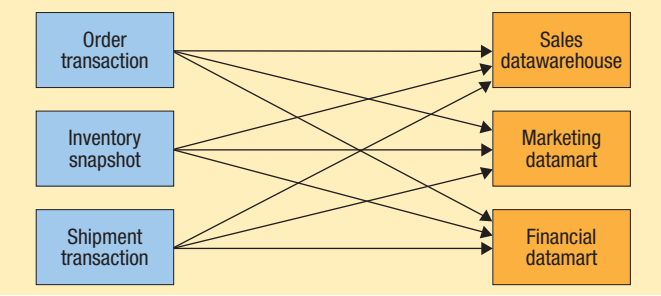

Figure 3.

are easy to construct and quick to be used. The problem with the simple dimensional architecture (and the nexus of the difference between Inmon and Kimball) is that nowhere in the Kimball Stage 1 simple dimensional architecture is there the notion of the "single version of the truth". At best, Kimball says that application data should be *copied* from the application environment. Inmon, on the other hand, suggests that a fundamental and rigorous transformation of legacy data is necessary in order to create the "single version of the truth".

When comparing the Kimball Stage 1 simple dimensional architecture versus the Inmon corporate information factory, Inmon's data warehouse requires that there be a "single version of the truth" while Kimball's data warehouse is a collection of data marts consisting of data that has been copied from applications. And therein lies the difference between the Inmon approach to data warehousing and the Kimball approach to data warehousing.

Differences between the models

The fundamental differences between the Kimball Stage 1 simple dimension architecture and the Inmon corporate information factory architecture can be summed up as:

- The corporate information factory (Inmon) addresses the need for integration of data across the organization creating what can be called the "single version of the truth." The focus of the Inmon corporate information factory is the integration of data across the corporation;
- The Kimball Stage 1 dimensional architecture is quick to build and allows reports to be built quickly but does not require a "single version of the truth" be built, only that a "copy" of data from the legacy environment be made. The focus of the Kimball Stage 1 simple dimensional model is on a few immediate applications from which data marts can be built. Since the focus in the Kimball Stage 1 dimensional architecture is on the speed with which a data mart can be produced across a few applications, there is no time to build a "single version of the truth" across the enterprise.

There is no denying that a corporate information factory requires much more time and many more resources to build than a simple dimensional architecture, primarily because the scope of the corporate information factory is enterprise wide. The Kimball style simple dimensional architecture is unquestionably faster and easier to build. But the Kimball Stage 1 simple dimensional architecture does not contain the "single version of the truth" for the enterprise.

For small organizations with a small amount of data the Kimball Stage 1 simple dimensional architecture may be perfectly adequate. But for larger organizations with larger amounts of data and a need for integration of data cross the enterprise, the Kimball Stage 1 simple dimensional architecture soon becomes problematic. When the Kimball Stage 1 simple dimensional architecture is applied to large systems, the lack of the "single version of the truth" and the lack of the ability to integrate data across the organization becomes a large issue.

The simple dimensional model in the large enterprise

Consider what happens to the simple dimensional model in the face of a lot of data – there are lots of legacy sources and lots of

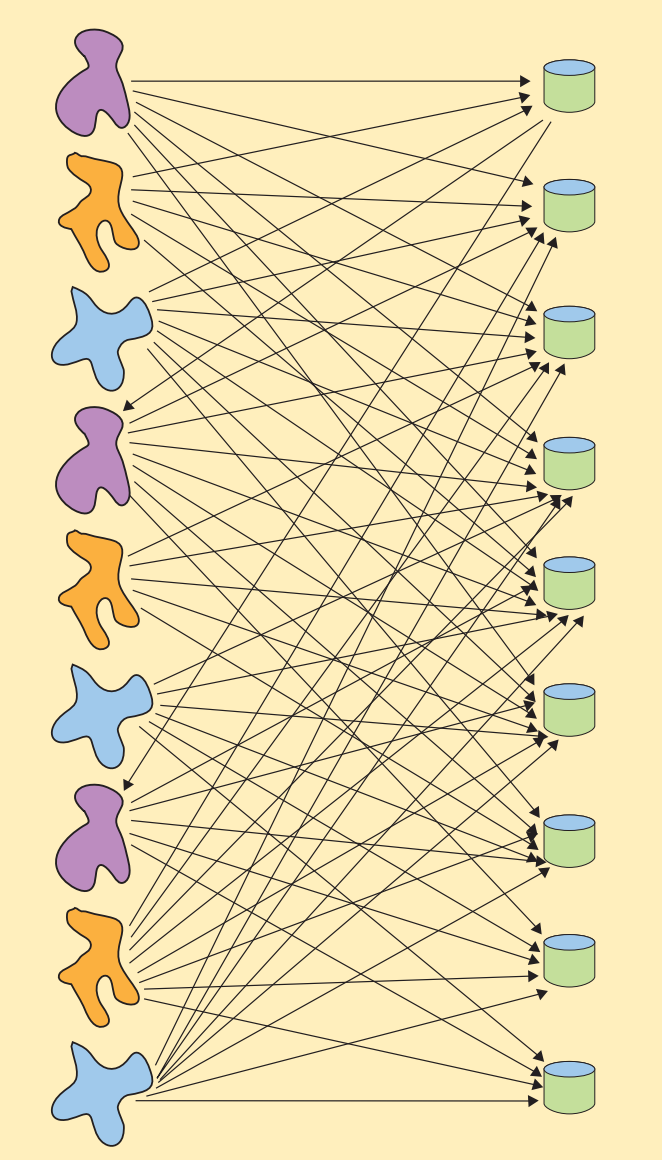

Figure 4.

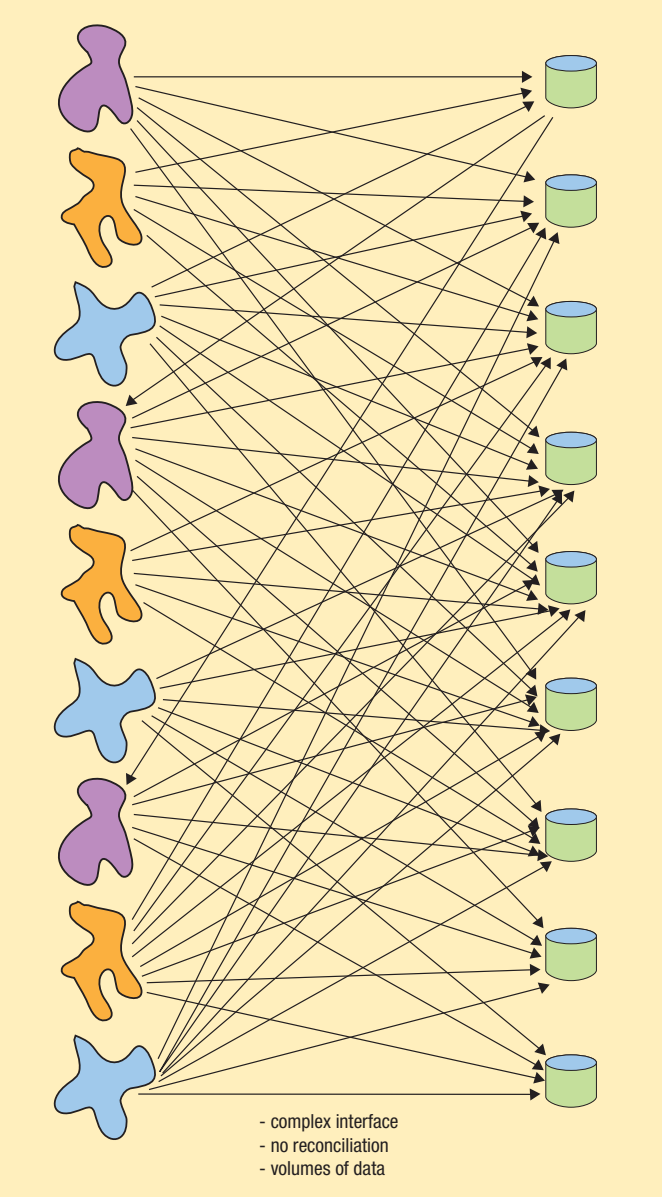

Figure 5.

data marts. The model *as illustrated by Kimball and Ross [1]* in Fig 3 merely expands. In the face of a large organization, the diagram drawn by Kimball and Ross that depicts the simple dimensional model simply grows larger. And with that expansion comes some major architectural problems. Fig 4 depicts a Kimball Stage 1 architecture for a large organization.

It is at this point that the Kimball architecture began to evolve into the next stage. Evolution occurs because of the pain of problems. And there were adequate points of pain for large organizations that tried to implement the Kimball Stage 1 simple dimensional architecture for an evolution to occur.

One of the motivations for evolution is that there are many interface programs that are needed to support a Kimball Stage 1 simple dimensional architecture in a large organization. More pain arises when it comes time to maintain those interface programs. When the Kimball Stage 1 architecture is built for a large organization, there is enormous redundancy of data, from one data mart to the next. Another motivation for evolution occurs when it is time to refresh data into the data marts. The window of opportunity for refreshment continues to shrink on a nightly basis. But perhaps the most pain with the Kimball Stage 1 simple dimensional architecture occurs because there is no corporately understood value of data, no "single version of the truth". In a large scale implementation of a Kimball Stage 1 simple dimensional architecture, when an end user wants to find a value of data, the end user literally has hundreds of places to turn to find that single value of data. In the Kimball Stage 1 simple dimensional architecture there is no one definitive place that states where a value of data is or is not. Consequently, a given value of data can reside anywhere (or nowhere) in a Kimball Stage 1 simple dimensional model. Since there is no definition of where there is a proper value of data, there can be many versions of the same value of data in a Kimball Stage 1 simple dimensional model in a large organization. Needless to say, large confusion results when large organizations turn the Kimball Stage 1 simple dimensional architecture into reality. If – as Kimball suggests (in his own words) – "a data warehouse is a union of all the data marts" – then there is a real problem with the data warehouse when it is based on the Kimball Stage 1 simple dimensional model.

Fig 5 suggests the major problems that arise with the Kimball Stage 1 data warehouse for a large organization. (Note – a small organization may not experience anywhere near the amount of grief that a large organization may experience. The size and the sophistication of the organization make a real difference in the amount of pain felt by an organization when it struggles with a Kimball Stage 1 dimensional architecture.) (*Continued in part 2.)*

Bill Inmon

William H. Inmon (binmon@inmondatasystems.com) is oprichter en CEO van Inmon Data Systems, gevestigd in Castle Rock, Colorado.

Literatuur

Inmon;
- *Building the data warehouse, John Wiley, 1991.*
- *The corporate information factory, John Wiley, 1999.*
- *Operational data store, John Wiley, 1995.*
- *Business metadata: capturing enterprise knowledge, Morgan Kaufman, 2007.*
- *Tapping into unstructured data, Pearson, 2007.*
- *DW 2.0 – Architecture for the next generation of data warehouse, Morgan Kaufman, 2007.*
- *Building the unstructured data warehouse, Technics Publications, Nov 2010.*

Kimball;
- *[2][3] Data warehouse toolkit, John Wiley, 1998.*
- *Data warehouse toolkit: complete guide to dimensional modeling, John Wiley, 2002.*
- *Data warehouse toolkit: building the web enabled data warehouse, John Wiley, 2000.*
- *[1] Differences of Opinion: Comparing the Dominant Approaches to Enterprise Data Warehousing, Intelligent Enterprise magazine, 2004.*
- *Internet – Planning MDM and EDW with Dr Kimball for 2010 – Informatica.*