Essential Steps for the Integrated EDW

A Kimball Group White Paper

By Ralph Kimball

KIMBALL GROUP
Consulting | Kimball University

Table of Contents

Executive Summary

In this white paper, we propose a specific architecture for building an integrated enterprise data warehouse (EDW). This architecture directly supports master data management efforts and provides the platform for consistent business analysis across the enterprise. We describe the scope and challenges of building an integrated enterprise data warehouse, and we provide detailed guidance for designing and administering the necessary processes that support integration. This white paper has been written in response to a lack of specific guidance in the industry as to what an integrated EDW actually is, and what necessary design elements are needed to achieve integration.

About the Author

Ralph Kimball founded the Kimball Group. Since the mid 1980s, he has been the data warehouse/business intelligence (DW/BI) industry's thought leader on the dimensional approach and trained more than 10,000 IT professionals. Prior to working at Metaphor and founding Red Brick Systems, Ralph co-invented the Star workstation at Xerox's Palo Alto Research Center (PARC). Ralph has his Ph.D. in Electrical Engineering from Stanford University.

The Kimball Group is the source for dimensional DW/BI consulting and education, consistent with our best-selling *Toolkit* book series, Design Tips, and award-winning articles. Visit www.kimballgroup.com for more information.

What Does an Integrated Enterprise Data Warehouse (EDW) Deliver?

The mission statement for the integrated EDW is to provide the platform for business analysis to be applied consistently across the enterprise. Above all, this mission statement demands *consistency* across business process subject areas and their associated databases.

Consistency requires detailed textual descriptions of entities such as customers, products, locations, and calendars to be applied uniformly across subject areas, using standardized data values. Of course, this is a fundamental tenet of master data management (MDM).

Consistency requires aggregated groupings such as types, categories, flavors, colors, and zones defined within entities to have the same interpretations across subject areas. This can be viewed as a higher level requirement on the textual descriptions described in the previous paragraph.

Consistency requires that constraints posed by BI applications which attempt to harvest the value of consistent text descriptions and groupings be applied with identical application logic across subject areas. For instance, constraining on a product category should always be driven from a field named Category found in the Product dimension.

Consistency requires that numeric facts are represented consistently across subject areas so that it makes sense to combine them in computations and compare them to each other, perhaps with ratios or differences. For instance, if Revenue is a numeric fact reported from multiple subject areas, then the definitions of each of these revenue instances must be the same.

Consistency requires that international differences in languages, location descriptions, time zones, currencies, and business rules be resolved to allow all of the above consistency requirements to be achieved!

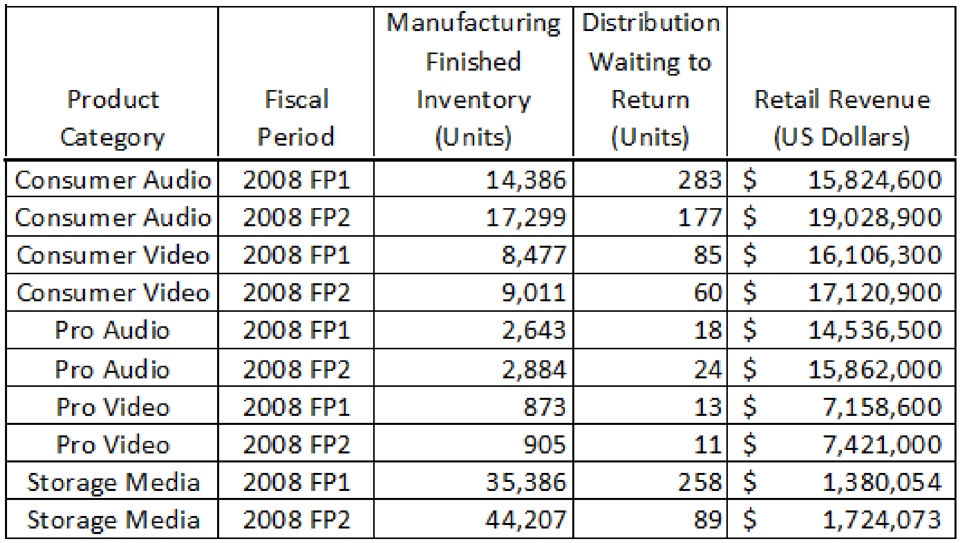
Consistency requires that auditing, compliance, authentication, and authorization functions be applied in the same way across subject areas.

Finally, consistency implies *coordination* with industry standards for data content, data exchange, and reporting, where those standards impact the enterprise. Typical standards include ACORD (insurance), MISMO (mortgages), SWIFT and NACHA (financial services), HIPAA and HL7 (health care), RosettaNet (manufacturing), and EDI (procurement).

Drilling Across is the Ultimate Litmus Test for Integration

Even an EDW that meets all of the consistency requirements described above must additionally provide a mechanism for delivering integrated reports and analyses from BI tools, attached to many database instances, possibly hosted on remote, incompatible systems. We call this *drilling across*. Drilling across is the essential act of the integrated EDW. When we drill across, we gather results from separate business process subject areas and then align or combine these results into a single analysis.

For example, suppose our integrated EDW spans manufacturing, distribution and retail sales in a business that sells audio/visual systems. We'll assume that each of these subject areas is supported by a separate transaction processing system. A properly constructed drill across report could look like Figure 1.



Product Category	Fiscal Period	Manufacturing Finished Inventory (Units)	Distribution Waiting to Return (Units)	Retail Revenue (US Dollars)
Consumer Audio	2008 FP1	14,386	283	$ 15,824,600
Consumer Audio	2008 FP2	17,299	177	$ 19,028,900
Consumer Video	2008 FP1	8,477	85	$ 16,106,300
Consumer Video	2008 FP2	9,011	60	$ 17,120,900
Pro Audio	2008 FP1	2,643	18	$ 14,536,500
Pro Audio	2008 FP2	2,884	24	$ 15,862,000
Pro Video	2008 FP1	873	13	$ 7,158,600
Pro Video	2008 FP2	905	11	$ 7,421,000
Storage Media	2008 FP1	35,386	258	$ 1,380,054
Storage Media	2008 FP2	44,207	89	$ 1,724,073

Figure 1. A Three Fact Table Drill Across Report

The first two columns are row headers from the Product and Calendar "conformed" dimensions, respectively. The remaining three fact columns each come from separate databases, namely manufacturing, distribution, and retail sales. This deceptively simple report can only be produced in a properly integrated EDW. In particular, the Product and Calendar dimensions must be available in all three separate databases, and the Category and Period attributes within those dimensions must have identical contents and interpretations. Although the metrics in the three fact columns are different, the meaning of the metrics must be consistent across product categories and times.

You must understand and appreciate the tight constraints on the integrated EDW environment demanded by the above report. If you don't, you won't understand this white paper, and you won't have the patience to study the detailed steps described below. Or, to put the design challenge in other terms, if you eventually build a successful integrated EDW, you will have visited every issue in this paper. So, with those warnings, read on!

The Organizational Challenges of Providing an Integrated EDW

The integrated EDW deliverables described above are a daunting list indeed. But for these deliverables to even be possible, the enterprise must make a profound commitment, starting from the executive suite. The separate divisions of the enterprise must have a shared vision of the value of data integration, and they must anticipate the steps of compromise and decision making that will be required. This vision can only come from the senior executives of the enterprise, who must speak very clearly on the value of data integration.

Existing master data management projects provide an enormous boost for the integrated EDW, since presumably the executive team already understands and approves the commitment to building and maintaining master data. A good MDM resource greatly simplifies, but does not eliminate, the need for the EDW team to build the structures necessary for data warehouse integration.

In many organizations, a chicken-and-egg dilemma exists, as to whether MDM is required before an integrated EDW is possible, or whether the EDW team creates the MDM resources. Often, a low profile EDW effort to build “conformed dimensions” solely for data warehouse purposes morphs into a full-fledged MDM effort that is on the critical path to supporting main line operational systems. In our classes since 1993, we have shown a backward pointing arrow leading from cleaned data warehouse data to operational systems. In the early days, we sighed wistfully and wished that the source systems cared about clean, consistent data. Now, more than fifteen years later, we seem to be getting our wish!

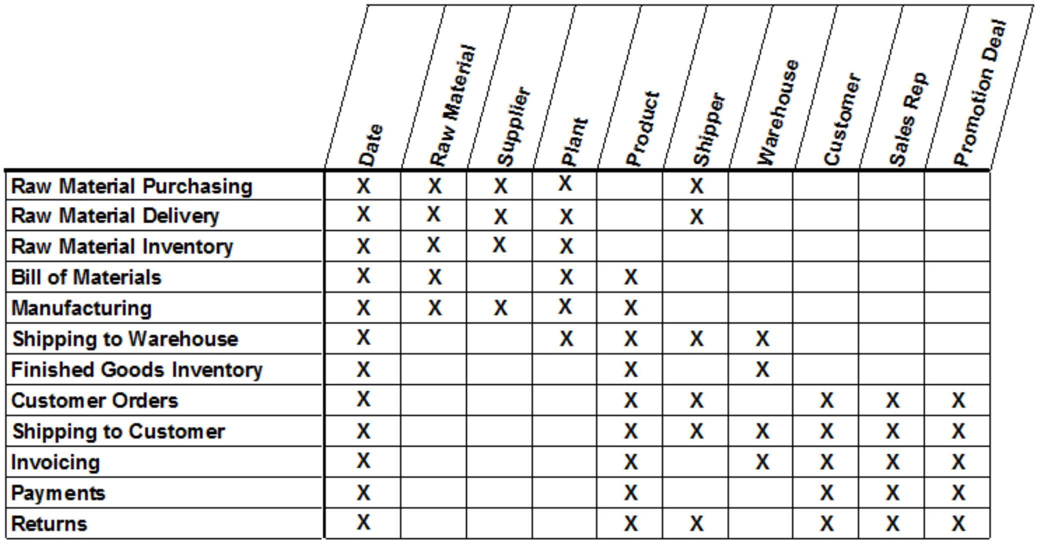
Conformed Dimensions and Facts

Since the earliest days of data warehousing, *conformed dimensions* have been used to consistently label and constrain separate data sources. We learned about conformed dimensions from A.C. Nielsen in 1983 when, at Metaphor Computer Systems, we brought Nielsen’s syndicated scanner data together with product shipments data at consumer package goods companies. The idea behind conformed dimensions is very simple: two dimensions are conformed if they contain one or more common fields, whose contents are drawn from the same domains. That results in constraints and labels having the same content and meaning when applied against separate data sources.

Conformed facts are simply numeric measures that have the same business and mathematical interpretations so that they may be compared and computed against each other consistently. Using these names, we have taught the principles of conformed dimensions and conformed facts since 1993 in our books and articles.

Using the Bus Matrix as a Way to Communicate with Executives

When you combine the list of EDW subject areas with the notion of conformed dimensions, a powerful diagram emerges, which we call the *enterprise data warehouse bus matrix*. A typical bus matrix is shown in Figure 2.



	Date	Raw Material	Supplier	Plant	Product	Shipper	Warehouse	Customer	Sales Rep	Promotion Deal
Raw Material Purchasing	X	X	X	X		X				
Raw Material Delivery	X	X	X	X		X				
Raw Material Inventory	X	X	X	X						
Bill of Materials	X	X		X	X					
Manufacturing	X	X	X	X	X					
Shipping to Warehouse	X			X	X	X	X			
Finished Goods Inventory	X				X		X			
Customer Orders	X				X	X		X	X	X
Shipping to Customer	X				X	X	X	X	X	X
Invoicing	X				X		X	X	X	X
Payments	X				X			X	X	X
Returns	X				X	X		X	X	X

Figure 2. A Bus Matrix for a Manufacturing EDW

The business process subject areas are shown along the left side of the matrix and the dimensions are shown across the top. An X marks where a subject area uses the dimension. Note that “subject area” in our vocabulary corresponds to a business process, typically revolving around a transactional data source. Thus “customer” is not a subject area.

At the beginning of an EDW implementation, this bus matrix is very useful as a guide, both to prioritize the development of separate subject areas, but also to identify the potential scope of the conformed dimensions. As we have often remarked, the columns of the bus matrix are the invitation list to the conformed dimension design meeting!

Before the conformed dimension design meeting occurs, this bus matrix should be presented to senior management, perhaps in exactly the form of Figure 2. Senior management must be able to visualize why these dimensions (master entities) attach to the various business process subject areas, and they must appreciate the organizational challenges of assembling the diverse interest groups together to agree on the conformed dimension content. If senior management is not interested in what the bus matrix implies, then to make a long story short, you have no hope of building an integrated EDW.

It is worth repeating the definition of a conformed dimension at this point to take some of the pressure off of the conforming challenge. Two instances of a dimension are conformed if they contain one or more common fields, whose contents are drawn from the same domains. This means that the individual subject area proponents do not have to give up their cherished private descriptive attributes. It merely means that a set of master, universally agreed-upon attributes must be established. These master attributes then become the contents of the conformed dimension and become the basis for drilling across.

The Kimball Group books and our article and design tip archives contain a wealth of additional material on the steps of building the bus matrix for an enterprise and establishing conformed dimensions and facts. Please see www.kimballgroup.com.

Managing the Backbone of the Integrated EDW

The backbone of the integrated EDW is the set of conformed dimensions and conformed facts. Even if the enterprise executives support the integration initiative, and the conformed dimension design meeting goes well, there is a lot to the operational management of this backbone. This management can be visualized most clearly by describing two personality archetypes: the *dimension manager* and the *fact provider*. Briefly, the dimension manager is a centralized authority who builds and distributes a conformed dimension to the rest of the enterprise, and the fact provider is the client who receives and utilizes the conformed dimension, almost always while managing one or more fact tables within a subject area.

At this point in the white paper we must make three fundamental architectural claims to prevent false arguments arising that turn into distractions:

1) The need for dimension managers and fact providers arises solely from the natural re-use of dimensions across multiple fact tables (or OLAP cubes). Once the EDW community has committed to supporting cross-subject area analysis, there is no way to avoid all the steps described in this white paper!

2) Although we describe the handoff from the dimension manager to the fact provider as if it were occurring in a distributed environment where they are remote from each other, their respective roles and responsibilities are the same whether the EDW is fully centralized on a single machine or is profoundly distributed across many diverse machines in different locations.

3) The roles of dimension manager and fact provider, although obviously couched in dimensional modeling terms, do not arise from a particular modeling persuasion. All of the steps described in this white paper would be needed in a fully normalized environment. Actually, the management of primary, durable, and natural keys described later in this white paper, are substantially more complicated in a normalized environment because of the need to propagate changing keys up and down the chain of linked normalized tables.

The next two sections describe the roles of the dimension manager and the fact provider.

The Dimension Manager

The dimension manager defines the content and structure of a conformed dimension, and delivers that conformed dimension to downstream clients known as fact providers. This role can definitely exist within an MDM framework, but the role is much more focused than just being the keeper of the single truth about an entity. The dimension manager has a list of deliverables and responsibilities, all oriented around creating and distributing physical versions of the dimension tables that represent the major entities of the enterprise. In many enterprises, key conformed dimensions include customer, product, service, location, employee, promotion, vendor, and calendar. In the following, as we describe the dimension manager's tasks, we will use customer as the example to keep the discussion from being too abstract. Here are the tasks of the customer dimension manager:

Define the content of the customer dimension. The dimension manager chairs the design meeting for the conformed customer dimension. At that meeting, all the stakeholders from the customer facing transaction systems come to agreement on a set of dimensional attributes that everyone will use when drilling across separate subject areas. Remember that these attributes are used as the basis for constraining and grouping customers. Typical conformed customer attributes include Type, Category, Location (multiple fields implementing an address), Primary Contact (name, title, address), First Contact Date, Credit Worthiness, Demographic Category, and others. Every customer of the enterprise appears in the conformed customer dimension.

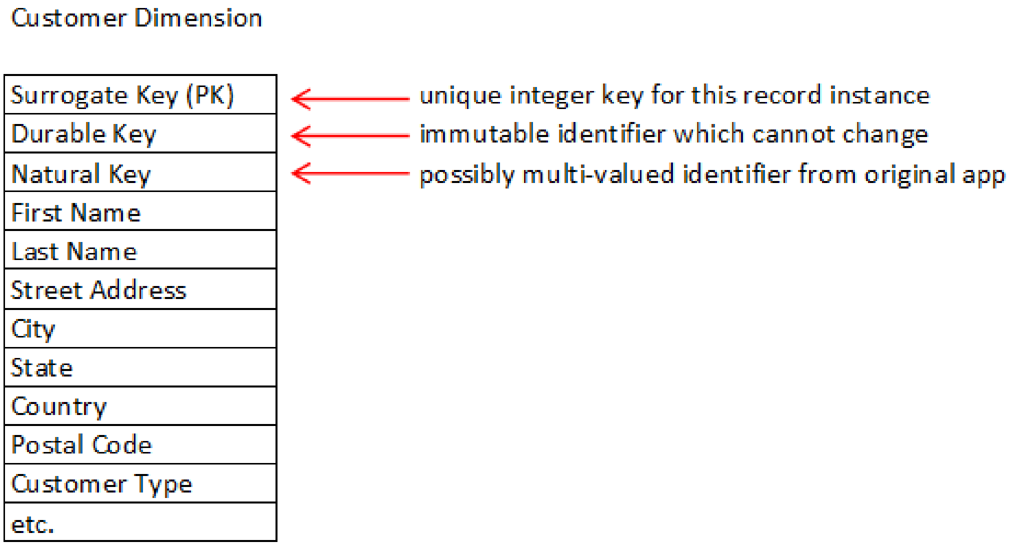
Receive notification of new customers. The dimension manager is the keeper of the master list of dimension members, in this case customers. The dimension manager must be notified whenever a new customer is registered. In a full blown MDM environment, new customers should only be registered by using an MDM-supplied process which is under the direct control of the dimension manager. In a more modest data warehouse environment without a centralized MDM facility, each remote customer facing process has the potential for registering a new customer. In these cases, the dimension manager receives notifications of new customers after the fact. Without an MDM facility, the dimension manager is forced to maintain a list of natural keys of customers from each possible source. These natural keys are the only way to reliably distinguish a new customer from an old customer.

De-duplicate customer dimension. The dimension manager must de-duplicate the master list of customers. Customer lists in the real world are nearly impossible to de-duplicate completely. Even when customers are registered through a central MDM process, it is often possible to create duplicates, either for individual customers or business entities. The de-duplication problem is much worse when no central MDM resource exists, since the separate customer facing processes are by definition not well coordinated. Even worse, these separate customer facing processes may apply different business rules and have different database structures when collecting customer identity information.

Assigns unique durable key to each customer. The dimension manager must identify and keep track of a unique durable key for each customer. Many DBAs automatically assume that this is the “natural key.” But quickly choosing the natural key may be the wrong choice. A natural key may not be durable! Using our customer example, if there is any conceivable business rule that could change the natural key over time, then it is not durable. Also, in the absence of a formal MDM process, natural keys can arise from more than one customer facing process. In this case, different customers could have natural keys of very different formats. Finally, a source system’s natural key may be a complex, multi-field data structure. For all these reasons, the dimension manager needs to step back from literal natural keys and assign a unique durable key that is completely under the control of the dimension manager. We recommend that this unique, durable key be a simple sequentially assigned integer, with no structure or semantics embedded in the key value. Note that the creation of such a unique, durable key does not preclude carrying original natural keys in the conformed dimension record, but of course this becomes complicated when there are multiple original sources registering customers, potentially with duplications.

Tracks time variance of customers with Type 1, 2, and 3 SCDs. The dimension manager must respond to changes in the conformed attributes describing a customer. Much has been written about tracking the time variance of dimension members using slowly changing dimensions (SCDs). A Type 1 change overwrites the changed attribute and therefore destroys history. A Type 2 change creates a new dimension record for that customer, properly time stamped as of the effective moment of the change. A Type 3 change creates a new field in the customer dimension that allows an “alternate reality” to be tracked. The dimension manager updates the customer dimension in response to change notifications received from various sources. See any of the Kimball Group books or our website for an extensive discussion of SCDs.

Assigns surrogate keys for the customer dimension. Type 2 is the most common and powerful of the SCD techniques since it provides precise synchronization of a customer description with that customer’s transaction history. Since Type 2 creates a new record for the same customer, the dimension manager is forced to generalize the customer dimension primary key beyond the unique, durable key. The primary key should be a simple surrogate key, sequentially assigned as needed, with no structure or semantics in the key value. This primary key is separate from the unique durable key, which simply appears in the dimension as a normal field. The unique, durable key is the glue that binds the separate SCD2 records for a single customer together. See Figure 3 showing the complete recommended set of keys for the customer dimension, including natural, durable, and surrogate keys.



Customer Dimension

Surrogate Key (PK)	unique integer key for this record instance
Durable Key	immutable identifier which cannot change
Natural Key	possibly multi-valued identifier from original app
First Name	
Last Name	
Street Address	
City	
State	
Country	
Postal Code	
Customer Type	
etc.	

Figure 3. Recommended Key Structure For a Customer Dimension

Handles late arriving dimension data. When the dimension manager receives late notification of a Type 2 change affecting a customer, special processing is needed. A new dimension record must be created, and the effective dates of the changes adjusted. The changed attribute must be propagated forward in time through existing dimension records. Please see *The Data Warehouse ETL Toolkit* book [Wiley, 2004] for a complete description of these processing steps.

Provides version numbers for the dimension. Before releasing a changed dimension to the downstream fact providers, the dimension manager must update the dimension version number if Type 1 or Type 3 changes have occurred, or if late arriving Type 2 changes have occurred. The dimension version number does not change if only contemporary Type 2 changes have been made since the previous release of the dimension. We recommend embedding the dimension version number as a field in the dimension itself, where every record in the dimension contains the same version number value. In this way, all query tools and report writers attempting to drill across separate instances of the dimension can include the version number in the SQL SELECT list, and thereby automatically avoid aligning incompatible data from different dimension versions.

Adds private attributes to dimensions. The dimension manager must incorporate private departmental attributes in the release of the dimensions to the fact providers. These are attributes that are of interest to only a part of the EDW community, perhaps a single department. Paradoxically, these attributes must be part of the master dimension release so that such departments can use the attributes for constraining and grouping when performing drill across queries. If some of the private attributes have sensitive content, then other departments must be shielded from using these attributes via the authentication and authorization functions of the EDW.

Builds shrunken dimensions as needed. The dimension manager is responsible for building various shrunken dimensions that are needed by fact tables at high levels of granularity. For example, a customer dimension might be rolled up to

Demographic Category to support a fact table that reports sales at this level. The dimension manager is responsible for creating this shrunken dimension and assigning its keys. Such a dimension cannot be created by defining a view on the lowest level customer dimension, since records in such a view would have to be drawn from the individual customer list, and these individual customers do not necessary exist over all times. Thus a shrunken dimension must be a separate, independent dimension table with its own keys.

Replicates dimensions to fact providers. The dimension manager periodically replicates the dimension and its shrunken versions to all the downstream fact providers. All the fact providers should attach the new dimensions to their fact tables at the same time, especially if the version number has changed.

Documents and communicates changes. The dimension manager maintains metadata and documentation describing all the changes made to the dimension with each release.

Coordinates with other dimension managers. Although each conformed dimension can be administered separately, it makes sense for the dimension managers to coordinate their releases to lessen the impact on the downstream fact providers.

The Fact Provider

The fact provider sits downstream from the dimension manager and responds to each release of each dimension that is attached to a fact table under the provider's control.

Avoids changes to conformed attributes. The fact provider must not alter the values of any conformed dimension attributes, or the whole logic of drilling across diverse subject areas will be corrupted.

Responds to late arriving dimension updates. When the fact provider receives late arriving updates to a dimension, the primary keys of the newly created dimension records must be inserted into all fact tables using that dimension whose time spans overlap the date of the change. If these newly created keys are not inserted into the affected fact tables, then the new dimension record will not tie to the transactional history. The new dimension key must overwrite existing dimension keys in the affected fact tables from the time of the dimension change up to the next dimension change that was already correctly administered. This process is described in more detail in *The Data Warehouse ETL Toolkit*.

Ties conformed dimension release to local dimension. The dimension manager must provide to the fact provider a mapping that ties the fact provider's local natural key to the primary surrogate key assigned by the dimension manager. In the surrogate key pipeline (see below), the fact provider replaces the local natural keys in the relevant fact tables with the conformed dimension primary surrogate keys using this mapping.

Processes dimensions through surrogate key pipeline. The fact provider converts the natural keys attached to contemporary transaction records into the correct primary surrogate keys, and loads the fact records into the final tables with these surrogate keys.

Handles late arriving facts. The surrogate key pipeline described in the previous paragraph can be implemented in two different ways. Traditionally, the fact provider maintains a current key lookup table for each dimension that ties the natural keys to the contemporary surrogate keys. This works for the most current fact table data where you can be sure that the contemporary surrogate key is the one to use. But the lookup tables cannot be used for late arriving fact data since it is possible that one or more old surrogate keys must be used. In this traditional approach, the fact provider must revert to an inefficient dimension table lookup in order to figure out which old surrogate key applies.

A more modern approach to the surrogate key pipeline implements a dynamic cache of records looked up in the dimension table rather than a separately maintained lookup table. This cache handles contemporary fact records as well as late arriving fact records with a single mechanism. See *The Data Warehouse ETL Toolkit* book for more detail.

Synchronizes dimension releases with other fact providers. It is critically important for all the fact providers to respond to dimension releases at the same time. Otherwise a client application attempting to drill across subject areas will encounter dimensions with different version numbers. See the description of using dimension version numbers in the last paragraph of this white paper.

Configuring BI Tools to Use the Integrated EDW

There is no point in going to all the trouble of setting up dimension managers, fact providers, and conformed dimensions if you aren't going to perform drill across queries. In other words, you need to sort-merge separate answer sets on the row headers defined by the values from the conformed dimension attributes. There are many ways to do this in standard BI tools, and in straight SQL.

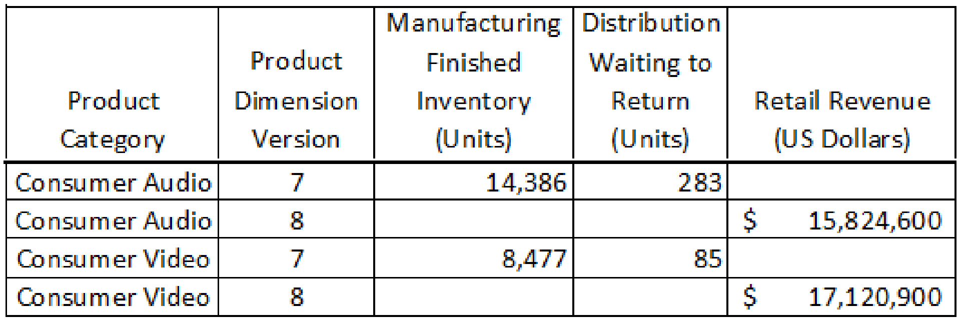
Mechanism for drill across. In SQL a drill across query bringing data from manufacturing shipments and retail sales could be implemented as follows:

```
SELECT Mfg.ProductCategory, Mfg.Year, Mfg_Amount, Sales_Amount
FROM

-- Subquery “Mfg” returns total shipments from Manufacturing
 (SELECT Category AS ProductCategory, Year, SUM(Ship_Amount) Mfg_Amount
   FROM Mfg_Shipments A
   INNER JOIN Product C ON A.Product_Key = C.Product_Key
   INNER JOIN Date D ON A.Sales_Date_Key = D.Date_Key
   GROUP BY Category, Year) Mfg

INNER JOIN


-- Subquery “Sales” returns total sales from the Sales database
 (SELECT ProdCat_Name AS ProductCategory, Year, SUM(Amount) Sales_Amount
   FROM Sales_fact F
   INNER JOIN Product C ON F.Product_Key = C.Product_Key
   INNER JOIN Date D ON F.Sales_Date_Key = D.Date_Key
   GROUP BY ProdCat_Name, Year) Sales

-- Join condition for our small result sets
  ON Mfg. ProductCategory = Sales. ProductCategory
  AND Mfg.Year = Sales.Year
```

This should perform almost as fast as doing the two individual queries against the separate fact tables because the join is on relatively small subset of data that's already in memory.

Uses dimension version numbers where sort-merge (outer join) is supported by BI tool in drill across queries. A properly instrumented BI tool that sort-merges the final separate answer sets that compose a drill across query can provide valuable protection against erroneous results that come from accessing conformed dimensions that have different version numbers. If the BI tool does include the version number in the SELECT list, and the results are sort-merged (outer joined) then the results from the fact table queries will end up on separate rows of the answer set, properly labeled by the dimension version. This isn't much consolation to the end user, but at least the problem is diagnosed in an obvious way.

In Figure 4 we show a report drilling across the same three databases as in Figure 1, but where a dimension version mismatch occurs. Perhaps the definition of certain product categories has been adjusted between product dimension version 7 and version 8. In this case, the retail sales fact table is using version 8 whereas the other two fact tables are still using version 7. By including the product dimension version attribute in the SQL SELECT list, we automatically avoid merging potentially incompatible data. Such an error would be particularly insidious because without the rows being separated, the result would look perfectly reasonable, but it could be disastrously misleading.



Product Category	Product Dimension Version	Manufacturing Finished Inventory (Units)	Distribution Waiting to Return (Units)	Retail Revenue (US Dollars)
Consumer Audio	7	14,386	283	
Consumer Audio	8			$ 15,824,600
Consumer Video	7	8,477	85	
Consumer Video	8			$ 17,120,900

Figure 4. A Drill Across Report With a Dimension Version Mismatch

Advanced Topics

In this section we describe special refinements to the challenge of EDW integration that are beyond the basic steps presented in the previous sections.

Fact provider implements local SCDs in addition to conformed SCDs. A tricky problem occurs when a locally provided dimension attribute undergoes a change at a different time than any changes downloaded from the dimension manager. This is logically equivalent to handling late arriving dimensions, but requires the fact provider to create a surrogate key for the dimension that will not be used by the dimension manager. The dimension manager may need to partition the key space to assign a band of keys to the fact provider for this purpose.

Dimension managers and fact providers resolve international representation differences. A truly international EDW presents many challenges, which are explored in significant detail in *The Data Webhouse Toolkit*, (Kimball and Merz, Wiley 2000). These challenges include:

> *Foreign alphabets and character sets*. Many of the international display and printing problems in an international EDW require being able to represent foreign characters, including not just the accented characters from western European alphabets, but Cyrillic, Arabic, Japanese, Chinese, and dozens of other less familiar writing systems. It is important to understand that this is not a font problem. This is a character set problem. A font is simply an artist's rendering of a set of characters. There are hundreds of fonts available for standard English. But standard English has a relatively small character set that is enough for anyone's use unless you are a professional typographer. This small character set is usually encoded in ASCII (American Standard Code for Information Interchange), which is an 8-bit encoding that has a maximum of 255 possible characters. Only about 100 of these 255 characters have a standard interpretation that can be invoked from a normal English keyboard, but this is usually enough for English speaking computer users. It should be clear, though, that ASCII is woefully inadequate for representing the thousands of characters needed for non English writing systems. An international body of system architects, the Unicode Consortium, has defined a standard known as Unicode for representing characters and alphabets in almost all of the world's languages and cultures. Their work can be accessed on the web at www.unicode.org. The primary use of Unicode is a 16-bit encoding that has a maximum of 65,535 possible characters. The Unicode Standard, version 5.0, which is the published version of Unicode as of the writing of this white paper, now covers the principal written languages of the Americas, Europe, the Middle East, Africa, India, Asia, and Pacifica.
>
> *Addresses and their extensions to locations and maps*. Names and addresses are the most difficult and far reaching international design problem in the international EDW. Toby Atkinson has written a remarkable book describing the intricacies of international names and addresses. In his *Merriam Webster's Guide to International Business Communications* (Merriam-Webster, 1999) he gives the following example. Suppose you have a name and address like the following:
>
> Sándor Csilla
> Nemzetközi Kiadó Kft
> Rákóczi u. 73
> 7626 PÉCS
>
> Are you prepared to store this in a database? Is this a postally valid address? Does this represent a person or a company? Male or female?

Would the recipient be insulted by anything about this? Can your system parse it to determine the precise geographic locale? What salutation would be appropriate if you were greeting this entity in a letter or on the telephone? What is going to happen to the various special characters when it is printed? Can you even enter these characters from your various keyboards? If your EDW contains information about people or businesses located in multiple countries, then you need to plan carefully for a complete system spanning data input, transaction processing, address label and mailing production, real time customer response systems, and your marketing oriented data warehouse.

Numbers. Numbers are represented differently in different cultures. The number 100.456 is slightly larger than one hundred in the United States, but slightly larger than one hundred thousand in Germany. In India, a large number may be written as 23 34 789, since they may group the digits by twos after the first group of three. In India, a lakh represents 100,000 and a crore represents 10,000,000. Other countries use periods, commas, and even apostrophes to separate the digits. An international EDW must be able to read and write numbers correctly, given an assigned cultural context.

Telephone Numbers. Telephone numbers, like postal addresses, have two basic representations. One is for domestic consumption, and one is for international use. To make matters worse, the international version is often interpreted in a different way by each international observer. A telephone number (randomly created for illustrative purposes) in South Africa for example is written as

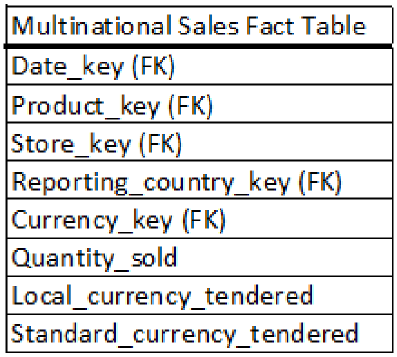
021-222-3333

but must be dialed from the United States as

011-27-21-222-3333.

The leading 011 is the way the United States dials international numbers. This will not be the same in other countries.

Currencies. Multinational businesses often book transactions, collect revenues, and pay expenses in many different currencies. A good basic design for all of these situations is shown in Figure 5.



Multinational Sales Fact Table
Date_key (FK)
Product_key (FK)
Store_key (FK)
Reporting_country_key (FK)
Currency_key (FK)
Quantity_sold
Local_currency_tendered
Standard_currency_tendered

Figure 5. A Multinational Fact Table

The primary amount of the transaction is represented in the local currency.

In some sense, this is always the "correct" value of the transaction. For easy reporting purposes, a second field included in the transaction fact record expresses the same amount in a single standard currency, such as United States dollars. The equivalency between the two amounts is a basic design decision for the fact table, and perhaps is an agreed upon daily spot rate for the conversion of the local currency into the global currency. Now all transactions in a single currency can be added up easily from the fact table by constraining the currency dimension to a single currency type. Transactions from around the world can easily be added up by summing the standard currency field. Note that currencies and countries are closely correlated but they are not the same. Countries may change the identity of their currency during periods of severe inflation.

Time of Day. The calculation of the true wall clock time in a given location around the world is surprisingly complicated. Most people think there are 24 time zones, corresponding to the 24 "possible" hours per day. But with even a little foreign travel experience, one begins to realize that this situation is much more complex. The entire country of India, for instance, sits in between these hour boundaries, since at different times of the year, it is either 5.5 or 6.5 hours ahead of Greenwich Mean Time. The rules of when various locations go on and off daylight savings time are amazingly intricate. Parts of Indiana, for example, go on daylight savings time, and other parts do not. The dates when daylight savings time goes into effect vary by location. The time difference between London, England and Sydney, Australia can vary by as much as two hours, depending on the time of year. In reality, there are more than 500 time zones in the world, and the list is constantly changing. The complexity of time zone calculations makes it clear that one cannot embed time zone assumptions in the code of applications or fixed queries. It is also pretty clear that each IT organization should not re-invent the wheel and derive all the time zone rules independently. Fortunately, the web comes to our rescue. A number of time zone conversion services, such as www.timezoneconverter.com, are available on-line that have up-to-date databases reflecting all the complexities of time zone calculations.

Calendars. Each country has a unique list of holidays. In many cases the holidays do not occur on the same day in successive years. Some holidays, such as Easter, are based on very complex rules, that involve the phases of the moon, or other events. Some religious holidays are not celebrated on the same day in various parts of the same country. Holidays are so complicated that it probably does not make sense to try to define them more than ten or twenty years into the future. Thus, much as with time zones, the technical definition of holidays in the EDW needs to be driven from a service. At the time of this writing, some of the best publicly available sources of international holiday definitions can be found on the web by searching Google for "international holiday calendar."

Reports, printing, and collating sequences. An interesting issue in multinational reporting is how to prepare a set of consistent reports for managers across such an organization in different languages. There are three basic issues that must be dealt with simultaneously: sorting (collating), grouping, and conforming.

Many language systems sort their special characters in a unique way.

Atkinson’s book discusses the specific rules for sorting in Catalan, Czech, Danish, Finnish, German, Hungarian, Norwegian, Polish, Slovenian, Spanish, Swedish, and Turkish. And these are only languages using the Roman alphabet. A report could sort the same set of customer names differently in different languages.

Great care must be taken if a set of attributes in a dimension is translated from one language to another. For instance, if the category and department names for a large number of products are translated into more than one language, then the cardinality and the detailed many-to-many and many-to-one relationships must be identical between the two languages versions of the dimension, or else the use of an attribute from the dimension as a row header (grouping criterion) will not produce the same results in the separate languages. Because the maintenance of two language versions of a large dimension table would be so subtle and difficult, we recommend against this approach.

If the same dimension table has several language versions in different countries, then it may be impossible to conform data sources across these versions, because at an SQL query level, the row headers of the separate answer sets in different languages could not be matched.

If we assume that we want a set of reports to span multiple languages, then we recommend implementing a two layer architecture. In the lower layer, we store all data and produce all reports from a single base language system. In the upper layer, the finished report is augmented with translations in auxiliary reporting columns. These auxiliary reporting columns do not affect sorting, grouping, or the ability to conform reports across data sources located in different countries. If we adopt this approach, managers from different countries should be able to sit in the same room with their own versions of the same reports, but be able to understand each other’s reports and compare them.

Dimension managers and fact providers ensure that auditing, compliance, authentication, authorization, and usage tracking functions are applied uniformly for all BI clients. This set of responsibilities is especially challenging since they are outside the scope of the steps described in this white paper. A centralized MDM resource may standardize clients’ direct access to master data, such as customer. But such direct access probably occurs over an enterprise service bus (ESB), perhaps implemented on a service oriented architecture (SOA) framework. This access directly to the MDM resource is very different than using a customer dimension in a BI report produced by the EDW. Even when modern role enabled authentication and authorization safeguards are in place when using the EDW, subtle differences in the definition of roles may give rise to inconsistency. For example, a role named “senior analyst” may have different interpretations at different entry points to the EDW. Logically, the challenge of conforming these role definitions is similar to conforming dimensional attributes, but the role definitions are stored and maintained entirely differently. In many cases, these role definitions are stored and enforced in local LDAP directory servers that intercept end users’ login requests all across the EDW landscape. And finally, the criteria for who qualifies to be a senior analyst may depend on local administration that is tied more to the human resources function than business responsibility. The best that can be said for this difficult design challenge is that personnel responsible for defining the LDAP-enabled roles should be invited to the original dimension conforming meetings so that they become aware of the scope of EDW integration.

Dimension managers and fact providers coordinate with industry standards for data content, data exchange, and reporting, such as ACORD (insurance), MISMO (mortgages), SWIFT and NACHA (financial services), HIPAA and HL7 (health care), RosettaNet (manufacturing), and EDI (procurement). The existence of industry standards is mostly good news for the EDW since each industry standard provides the definition of many conformed dimension attributes and facts. But often these standards are accompanied by legal restrictions on how the information is handled.

Conclusion

The integrated EDW promises a rational, consistent view of enterprise data. This promise has been repeated endlessly in the trade literature. But until now, there has been no specific design for actually implementing the integrated EDW. In this paper we have precisely identified the ability to drill across as the central deliverable of the integrated EDW. Then we have methodically described the required steps and responsibilities which give rise to the archetypal roles of the dimension manager and the fact provider. Although this implementation of the integrated EDW surely must seem daunting, we believe that the steps and responsibilities we have described are basic and unavoidable, no matter how your data warehouse environment is organized. Finally, this architecture represents a distillation of more than two decades' experience in building data warehouse based on conformed dimensions and facts. If you carefully consider the detailed recommendations in this paper, you should avoid re-inventing the wheel when you are building your integrated EDW.