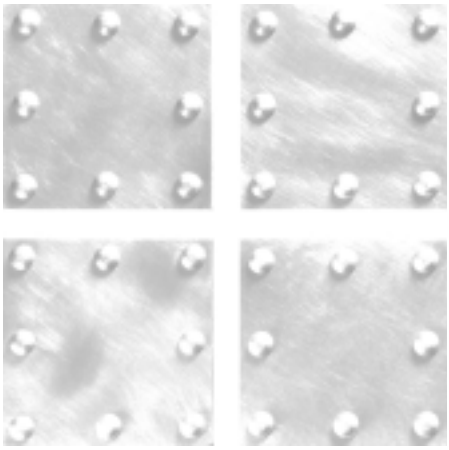

6

Metadata in Data Warehousing and Business Intelligence

In most large organizations, the deployment of IT systems over the past 20 or 30 years has taken place to service the needs of individual departments or functional groups. Typically, the systems that have been bought or built to meet these needs have been designed or evaluated in a local context, relevant to the scope laid down by the clients for the system, and highly focused on their business concerns. In many cases, this degree of focus has been critically important for ensuring that IT projects concerned with application development and deployment remain under control and that the resulting systems are well-suited to the needs of their clients.

Two significant trends have led, over recent years, to a breakdown in this insular view of data management within IT deployment activities. Increased integration between applications has gained favor in many large organizations. This integration can take the form of closer coordination of

function and data content between previously distinct business applications (e.g., production control systems and general ledgers) or of automation of the information supply chain from data capture through transaction processing to provision of management information, within a data warehouse, for example. These two forms may be referred to as *horizontal* and *vertical* integration, respectively.

In the first case, the use of highly integrated, packaged applications—ERP systems such as SAP R/3, Baan, and PeopleSoft—has focused attention on the way that data, as well as processes, can be integrated or made to conform to a common set of rules. These systems aim to offer a wide view of the business activities and combine data from diverse sources in a cohesive and reusable way. The spread of ERP systems as a practical alternative to self-built "islands of functionality" has been instrumental in forcing the IT and user community to think more carefully about the ways in which one part of their organization may interact with others.

A second factor that, although rather different in its intended purpose, has had a very similar effect, is the popularity of data warehousing. Once again, data warehouses aim to offer the user an integrated view of the organization, coupled with the ability to manipulate and analyze data from diverse sources. Clearly, to be capable of this, the data warehouse concept must be underpinned by a cohesive set of rules governing the way in which the data content behaves. This chapter explores in detail the way in which an effective understanding of metadata is a key factor in the success of data warehousing projects.

6.1 The data warehouse paradigm

The concept of data warehousing has been developed to define a model for the provision of information within typically large and complex organizations. The phrase was coined by Bill Inmon, in his seminal work on the subject [1], to describe a situation where data is brought into a central point, controlled and improved in quality, and dispersed (also in a controlled fashion) to a variety of users.

This model may not seem to differ that radically from the more traditional approach to decision support systems. For many years, separate

databases have been built downstream of major transaction processing applications, reflecting the information of those systems, but separate from them for reasons related to performance or control. To understand the real distinction between data warehousing and basic decision support, a clear understanding of the concepts is required. In fact, the phrase *data warehouse* is used to convey not just the model itself but the hardware and software technology that has made the manipulation of very large databases possible in recent years.

The basic problem that data warehousing was defined to address is that of a more sophisticated user, who may need to draw data from a variety of sources and perform complex operations on this data. These operations typically center on statistical (refining patterns from the bulk of past data) or predictive (extending these patterns to model future possibilities) analysis. In the base case, where the operational computer application had been developed separately and had its own standards for both data and metadata, such analysis has always been problematic.

Unless data can be drawn together and made to conform to a common pattern of behavior, there will be difficulties. Data must be made accessible to all users whilst preserving both the consistency (ensuring that one entity in the real world can be identified as such) and the integrity (ensuring that the business rules constraining the data are obeyed, even across multiple source systems) of the data as a whole. Without such control, anyone requiring a broader view of the corporate information base would need to build in his or her own consistency checks (effectively relying on what we might call self-built metadata).

This do-it-yourself approach will bring the benefits of enforced consistency to a broad view but will not impose any control over the consistency between views. The problem of seeing enterprise-wide data as a single cohesive picture still remains, but the possible inconsistencies are now more visible to a higher echelon of people within the organization.

Effectively, this is taking the familiar "islands of automation" problem and making the same mistake in the context of metadata. This is, of course, inherently wasteful, since the effort of imposing these rules would be duplicated across many users. It would also be error-prone and inherently incomplete, since few users, even the more sophisticated, would have sufficient knowledge or self-discipline to impose truly "neutral" business rules on the data. Nearly everyone, when faced with this

problem, falls back on their biases, built up over the course of their careers. A marketer would see a customer as someone to sell to; an accountant would see a customer as someone to bill. While not wrong or mutually incompatible, these views of the entity CUSTOMER are inherently incomplete and biased, which will inevitably lead to problems when large-scale, enterprise-wide reporting is attempted.

The analogy that Inmon chose to describe the new model for information provision was a warehouse or distribution center. In a real-world warehouse, goods are received from a supplier at a goods-in bay. At this point, they are checked to ensure that the quantity and description match the purchase order and that they have been delivered undamaged. If the quality is acceptable, they are taken into the warehouse and stored. The location and description of each batch of goods is recorded to ensure that they can be accessed and retrieved easily.

At a future point, the goods are removed from the warehouse and shipped to a customer or a smaller, satellite distribution point to meet a local need. Once again, checking and control are necessary to ensure that the correct quantity and specification of goods are dispatched. As we can see from Figure 6.1, a data warehouse environment, as envisaged by Inmon, bears a strong resemblance to this model of "real" warehousing logistics.

Data is gathered from a variety of source applications and databases (the suppliers) and taken into a staging area (the goods-in bay) where the quality is checked. Once we are assured of the data's timeliness, completeness, accuracy, and conformance with the standards laid down, they are accepted into the data warehouse proper and stored for future use. This journey from the source database into the warehouse is often described as an extract, transform, load (ETL) flow, and the availability and use of metadata are critical to the successful execution of these data flows. The data flows themselves can be triggered by a "push" from the source system (effectively, the source application gathers the data and signals to the ETL tool when it is ready) or a "pull" from the staging area environment (control of the overall transfer process is exerted from the data warehouse). As a general rule, the pull approach suits complex environments better, since it supports metadata control from a central point on the architecture. This leads to a simpler and more robust design and improved control of the transfer process. In some cases, ETL tools that

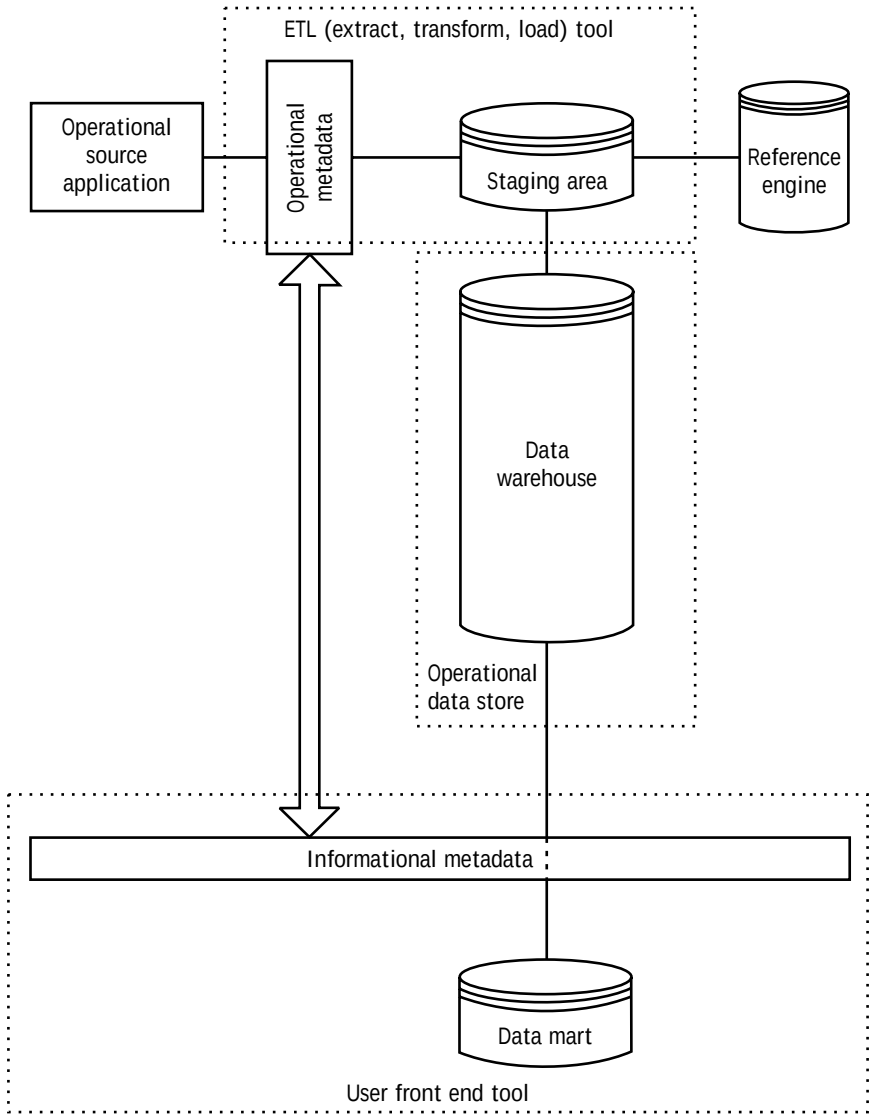

Figure 6.1 A generic data warehouse architecture.

interrogate the source system metadata directly are available. This is applicable to specific source database management system (DBMS) technologies (e.g., the ETL tool may be able to make use of Oracle system tables) and will thus not be true in the general case. It is worth noting that,

although there are not as yet any universal standards for metadata sharing, some ETL tools provide a wide variety of protocols for sharing metadata with specific source DBMS technologies.

Eventually, the data is moved out from the warehouse environment, either to a smaller, localized store (a so-called data mart, analogous to a local distribution center), or directly to the user who requires the data for analysis and reporting (the customer). Once again, if the end user is to have a clear, complete, and universally applicable view of what the data means and how they have been constructed, then metadata have a key part to play.

These two distinct types of metadata will require separate facilities to ensure their control and propagation. The first type, known as operational metadata, will be predominantly of interest to IT personnel charged with the design and control of ETL facilities. The second will require more delicate handling, as they relate to the perspective that end users will have on the information within the data warehouse. This second type is known as informational metadata.

Informational metadata also has an important part to play in the control of data marts. The key point of data warehousing is the ability to bring the whole scope of the enterprise (at least as far as information is concerned) under a common set of controls. By doing so, consistency can be achieved at both data and metadata levels.

Consistency of data, in this context, implies that the contents of the data warehouse are free from redundant data (e.g., two different records representing the same customer), code duplication (e.g., the same customer identifier being given to two different customers—very common if the data are sourced from multiple systems), and erroneous data.

Metadata consistency extends this to ensure that the data content obeys a common, consistent set of business rules. For example, it defines the meaning of the term "customer group" and ensures that it is applied consistently wherever it is used in the data warehouse. It also defines the rules for referential integrity and ensures their consistent application. If, for instance, an account must have an associated currency that is one of an approved set, then this rule is checked and enforced before the data are allowed within the warehouse environment.

The data within the data warehouse environment is quality-assured, usually within the staging area that acts as the entry point for data. The same cannot be said, however, for data marts, since the refragmentation

of data is inherent to the whole idea of marts. If the marts are designed as nonupdating environments (as is usually the case) then there should be no issue with the integrity of data breaking down, even though there are multiple copies in existence. Care must be taken, however, to ensure that the metadata are propagated into the mart environments at design time in a controlled and complete fashion.

The technique of data marting, or drawing off limited subsets from the main data warehouse and using the information they contain for focused business needs, is quite common in the context of business information provision. Data marts prove very useful where the following conditions exist:

1. The data must be segregated to improve the performance of the system from the user perspective. This is nearly always a significant issue with data warehousing. By limiting the scope of the database (and hence its size), significant advantages are achieved.

2. The data must be secured from unauthorized access by the simple expedient of maintaining a physically separate copy within a data mart and restricting the use to which data is put. This does, of course, imply that direct access to the data within the main warehouse itself will be forbidden (in the so-called mart-exclusive data warehouse architecture).

3. It is important, on occasions, to strengthen the concept of ownership within the database. Some part of the organization should feel overall responsibility for the content and structure of each mart, effectively taking ownership of it on behalf of the enterprise as a whole. The provision of a physically distinct, bounded environment can help significantly in such a case.

This reversion to the situation where databases are kept physically distinct from each other can clearly bring benefits, but there is a price to pay in the reintroduction of possible inconsistency, which can negate the whole basis for installing a data warehouse environment. By allowing physical separation of data into compartments, to be used by different groups within the business community, the ability of the organization to control data consistency from the center is diminished.

The first step that can be taken to address this problem is to ensure that the data mart environments remain read-only. By depriving users of the ability to change data, the contents of the data marts should remain as they were at the point of exit from the data warehouse. In this way, data consistency can be preserved.

The preservation of metadata consistency is more problematic. If the business rules are to be applied consistently across all data marts, then the design, construction, feeding, and use of all marts must be subject to common, central control. If the builder of such a feed decides to ignore a particular rule, the impact should be small since the rule will have been checked at the point of entry into the data warehouse. However, rules may be adapted to specific purposes. New rules may be introduced that contradict the rules within the data warehouse. Interpretations of the rules that may be misinterpreted by users (e.g., renaming customer types as customer groups, because the users like the name, even though the customer group concept has been globally defined as something different). In such cases, metadata consistency will undoubtedly suffer.

Experience shows that the data mart approach is very widespread within data warehouse installations. Many organizations recognize the particular benefits that such an approach brings, without considering the impact of uncoordinated mart development.

Marts are (quite rightly) seen as a quick and relatively inexpensive delivery mechanism for fulfilling business information needs. If these advantages are not to be outweighed by the reintroduction of inconsistent data and metadata, then steps should be taken to avoid uncontrolled deployment.

A coordinated, architected approach to data marts, bound together with a practical application of data and metadata control techniques, can yield the required benefits without endangering this consistency. The core problem that has to be faced is how to ensure this metadata consistency and what tools are available to help in this area.

6.2 Approaches to data mart deployment

The data mart has been described above as a subset of the data contained within a warehouse, extracted into a separate environment. Although it

helps us to understand the basic problem that the possible fragmentation of metadata will bring, this top-down approach is by no means the only method used to deploy data marts, or even the most common.

In many cases, the data marts are designed in the absence of a full-blown data warehouse from which to source data. Indeed, the deployment of data marts can be seen as the primary activity, an end in itself. It is easy to see how such an approach can be defended. After all, the marts are fundamentally user-focused and as such deliver the basic benefits of a data warehouse environment in manageable chunks. This splitting up of the problem also has the significant advantage of enabling each part of the development effort to be economically justified. In most commercial environments, the possibility of building a full-scale data warehouse environment is limited less by technical constraints than by the perceived cost of implementation. If this cost can be spread across a number of efforts, each with a tangible, identified, and quantified benefit, then the chances of success are significantly increased. By maintaining a tight focus on particular business problems and working toward a new architecture in evolutionary stages, the costs involved are distinctly easier to justify, than by seeking financial support for grandiose schemes.

In this context, the data warehouse becomes a clearinghouse for data on its way to the marts, effectively fulfilling little more than the purpose of a universal staging area. This in itself is by no means a fundamental flaw, provided that effective metadata control is in place at the entry and exit points to this area (operational and informational metadata control, respectively). The practical difficulties of achieving this control without a physical, persistent data warehouse to impose and preserve the integrity of the overall database should not be overlooked.

It is important to point out that there is not, at present, a wealth of tools to assist in the process of metadata control and that many of the tools that do exist are either embedded within, or dependent on, the database or ETL tools themselves. This lack of openness will cause further problems in complex, diverse environments, even where metadata control tools can be identified.

Therefore, both the top-down and the bottom-up approaches to data mart deployment have significant disadvantages. The adoption of an approach predicated on a pre-existing data warehouse would undoubtedly be more expensive to implement and more complex to design and

deploy and would raise significant financial and political difficulties along the way.

The mart-driven, bottom-up approach appears more pragmatic, but the dangers of reverting to uncontrolled, independent systems presenting isolated, mutually inconsistent versions of the "truth" should not be understated. Also, it must be remembered that the boundaries between these marts are liable to vary over time. Without a central, coordinated data warehouse, and the metadata to control its contents, it is unlikely that such flexibility will be achievable. In addition, the requirement to deploy independent data marts quickly can lead to a focus on sourcing from individual operational systems, rather than taking an enterprise-wide view. This will further diminish the value of the resulting system as a generic, universally applicable tool.

A third, hybrid approach, known as the *federated warehouse*, seeks to combine the advantages of both the bottom-up and top-down deployment techniques, while limiting the dangers. In this model, the concept of autonomous data marts, controlled by a common core of metadata, may coexist with an enterprise data warehouse (EDW) dependent on the same metadata. The possibility for building further independent marts or for extending the EDW and drawing further, dependent marts from it, can both be supported. This approach is more complex that the other two but has the virtue of incorporating the advantages of both.

However, because of its inherent complexity, it is also the most thoroughly dependent on metadata control for its successful, continued evolution. Metadata will be used to ensure that a common picture of the data is defined in all the destination data marts, whether they are dependent on or independent of the EDW.

Generally speaking, metadata fulfill three major functions at each stage of the data life cycle within a federated data warehouse environment. In the first place, they provide a common language for defining the behavior of data. The formal metadata syntax allows robust definition of the business rules and meanings associated with the data that is being transferred from the source operational systems to the EDW and eventually to dependent or independent marts. This is closer to the traditional view of metadata, describing how, for instance, sales orders and customers are related to each other, ensuring that the term customer group has a

consistent, universal definition and that the set of values to be used for currency codes are all defined.

Second, metadata control the data flows themselves. The operational metadata will contain not only definitions of the data but formal descriptions of the flows which occur, the rules governing data transformations, dependencies between transformations, and timing factors affecting the execution of the data transfers. The fact that the customer masterfile within the EDW is loaded from the sales order processing system on a daily basis and that it must follow the daily load of general ledger data and be followed in turn by the load of sales orders together with details of how any possible duplicate customers are removed from the flow in the process are all described in the metadata.

Third, metadata will communicate the nature of the data within the warehouse environment to users and IT developers with a need to understand the business basis for warehouse data. In one sense, this is merely the ability to access the definition and control data described above, but it is vital that the proliferation of metadata throughout the technical and nontechnical communities is managed in an effective and understandable fashion. Figure 6.2 expands the generic architechture (shown in Figure 6.1) to describe a federated data warehouse in operation and to indicate the points at which metadata is critical to the success of the system.

The EDW can fulfill a number of distinct roles in this type of architecture. In the first instance, it can simply be the superset of all data required by the dependent marts that are extracted from it. This provides a very safe, thorough mechanism by which the quality of data can be put under consistent control, within the marts, but evidently represents a significant level of data redundancy.

Each data item included in the EDW will, in such a case, occur also within at least one of the mart structures. While it must be remembered that a significant amount of data redundancy is to be expected in a data warehousing environment, it may represent an unnecessary overhead in this case.

In such a case, the EDW will start with a narrow scope of data, sufficient to ensure that the first mart to be implemented has the contents it requires and that these contents will obey the metadata constraints that

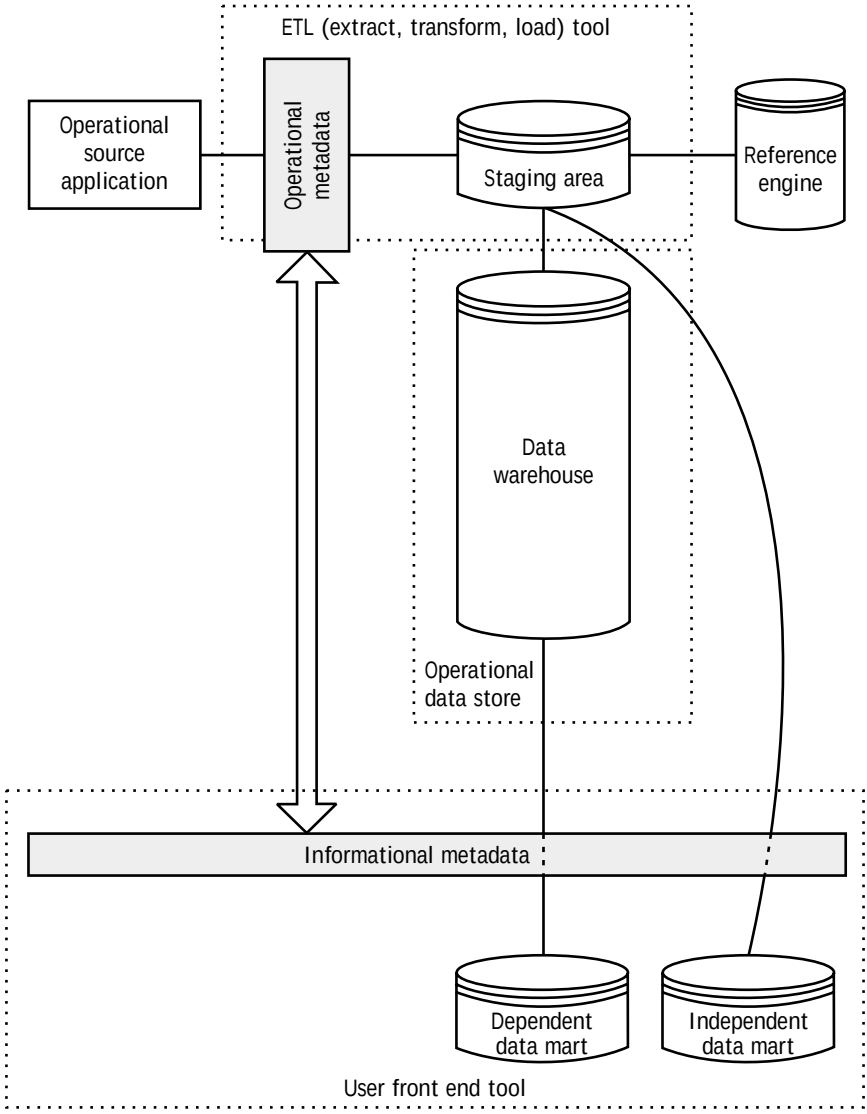

Figure 6.2 The federated data warehouse.

will be imposed by future extensions of the ETL boundary. Figure 6.3 diagrams the way in which this type of EDW is built up.

In Figure 6.3, the *x* (horizontal) axis represents the business scope of applications, or marts, within the scope of the data warehouse. The

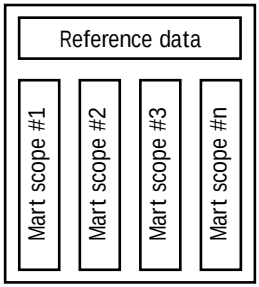

Figure 6.3 Mart data scope.

y (vertical) axis represents the amount of detail that is stored within the data warehouse environment for each part of the scope. Each vertical slice represents the data associated with a mart, and the total scope of detailed information provided by these marts evolves incrementally over time. The horizontal block, *basic reference data*, at the top of Figure 6.3 is a small portion of common data that effectively future-proofs the first mart deployments without implementing a full-scale warehouse in the first instance. As more marts come on stream, the vertical blocks within the EDW fill up and complete the true enterprise-level scope for the data content.

Alternatively, the EDW can form an enterprise-level data mart, providing a broad but shallow view of the data. It will help to understand this concept if we consider the basis on which a large proportion of data warehouse access is based.

In many cases, data warehouse applications are built to enable users to navigate their way through various levels of the database, seeing different tiers or levels of detail. A user may wish to drill up and down between a national-, regional-, or city-based view of its business, or between annual, monthly, and daily sales totals.

By defining the EDW as a "broad but shallow" mart, the implication is that it will cover a large swathe of the enterprises business but be limited to the higher levels of detail. The marts drawn from it will be restricted in the same way but will be more focused on specific functional areas or business activities. These marts contrast with the independent marts, which will allow significantly greater "drilling up and down" through the various data hierarchies. In either case, it must be emphasized that

the compatibility of these marts with each other, and with the EDW, is dependent on the quality and extent of metadata used at each stage in the data warehouse process.

All business rules implied by the EDW, or by the operational metadata applied within the staging area, will be inherited by the lower-level marts, in an ideal case. Thus, the possibility for putting a different, conflicting interpretation on the data in the marts from that implied by the operational metadata is minimized.

In summary, the operational metadata define and control the global business rules that are enforced on the data as it enters the data warehouse environment. Once within this environment, the interpretation that is put upon the data by users is governed by informational metadata, which also ensures that the behavior of data in the separate data marts is mutually consistent.

In practice, this implies that metadata must be explicitly managed at a number of levels, described as follows:

- The rules contained within the operational metadata must be controlled to ensure that they are correct, that no necessary rules are omitted, and that no rules that are included as universal are, in fact, partial.
- The rules within the informational metadata must be presented in a form that is easily understood by the end user of the data—the prime purpose of this metadata is to ensure that the user can access and manipulate the data efficiently and correctly. This will not happen unless informational metadata is robust and simply presented in business terms.
- In addition, a control mechanism should be in place to ensure that the operational and informational levels of metadata are consistent *with each other!* If this is not the case, then the metadata is acting to degrade data quality rather than improve it.

6.3 Operational metadata

Operational metadata exerts an influence at the incoming boundary to the data warehouse environment. It determines the acceptability or

otherwise of data that is sent to the warehouse and controls the process by which such transmissions are made. Significantly, it also offers the means by which systems developers, seeking to extend the data warehouse, can understand the developments that have gone before.

It is fundamental to the nature of a data warehouse environment that the scope of the system will be in a state of constant flux. Changes in business circumstances, new front-end reporting and analysis tools offering new opportunities for examination of the data, and the realization that the consistency and completeness offered by a data warehouse can significantly benefit many parts of the enterprise will all contribute to this need for constant evolution.

This being so, it is vitally important that these changes, which are so fundamental to the nature of a data warehouse can be made efficiently—using techniques that have been developed for other purposes—and safely—remaining consistent with the business rules defined for previous "slices."

Systems developers (and, to a lesser extent, users) will rely on the pool of metadata developed for previous slices during the building process. Operational metadata to support this constant extension of the data warehouse should include the following entities (at minimum):

- APPLICATION ENTITY, defining the tables that exist within the source application and that are accessed to find the data to be transferred into the warehouse;
- APPLICATION ATTRIBUTE, recording the individual fields that are used as source data from the APPLICATION ENTITY;
- TARGET (WAREHOUSE) ENTITY, providing a neutral, enterprise definition of each entity within the data warehouse environment;
- TARGET (WAREHOUSE) ATTRIBUTE, providing a neutral, field-level definition of each data item;
- TRANSFER DATAFLOW, defining the end-to-end flow relating to an independent segment of work within the data warehouse load activity;
- TRANSFER JOB STEP, a subdivision of the TRANSFER DATAFLOW, enabling the developer to understand how the

transfer will work, in what order the steps are taken, and under what circumstances they are considered successful;

- APPLICATION, defining any computer system acting as a data source;
- TRANSFORMATION, an activity, generally at the TRANSFER JOB STEP level, that changes the structure or content of some portion of the data being transferred;
- PHYSICAL DATA STORE, recording the actual source, intermediate, and destination files used within the transfer;
- RULE, (at its simplest) describing the constraints that are placed on each item of data (generally at the attribute level) as it passes into the data warehouse environment, possibly including validity checks, range checks, complex dependencies with other attributes, date and time dependencies, or other external effects;
- DEPENDENCY—specifically dependencies between TRANSFER JOB STEPs—adding further sophistication to the work flow concept, enabling the developer to define the order in which job steps take place (also records the need to finish one work step before another is started or, conversely, the possibility for performing the work in parallel, where appropriate). Clearly, a more sophisticated implementation of this feature would be possible within project management software, such as Microsoft Project™. It is also worth considering the extent to which the ETL tools will enable the recording and control of these work flow models.

These metadata elements are all basically static (in other words, they will not change unless there is some change made to the data load process itself). As such, they are not affected by the content or size of the source and target databases, nor by other external influences.

Several companies offer tools that can significantly enhance the capability to manage this operational metadata. In some cases, the tools (such as those offered by Prism, ETI, and Acta) combine the operational metadata management with sophisticated ETL capabilities, effectively subsuming the staging area within their scope.

These types of tools rely heavily on the capture and management of metadata. They are sold as efficient mechanisms for developing and controlling the ETL sequences but are in fact significant advances in the area of automated metadata management. In some cases, the basic metadata model has been extended to include dynamic features that provide information on such factors as data volumes, volatility, load run times, and flags set in the source systems in readiness for data transfer.

Whether static or dynamic, the metadata that controls the extraction, transformation, and loading of data is captured via a graphical interface, enabling the developer to see the data loads in diagrammatic form, building interdependent jobs and identifying and choosing source and target data fields from menus, etc. Without a robust, underlying metadata model, however, it would not be able to control the loads effectively, no matter how sophisticated the front end of the ETL tool may appear.

6.4 Informational metadata

By contrast, there are very few tools that are aimed at controlling and presenting the informational metadata. The information required by end users to make a balanced judgment about how to use the contents of a data warehouse to service their business needs is far more nebulous than the specification and control of dataflows.

Essentially, informational metadata is provided to ensure that each user has a consistent, well-understood picture of the contents of the data warehouse environment and that they can use this to formulate an appropriate method for meeting their business information needs. In a particular case, a business analyst may have a requirement for reporting sales volume and revenue figures over time, for a variety of customers and products.

Data warehousing software will provide analysts with the means of constructing such analyses and manipulating them in a variety of sophisticated ways. This will only be inherently useful to the analysts, however, if the assumptions that they are making about the data are correct and well-understood. The informational metadata must therefore incorporate answers to questions such as the following:

- Does the official sales volume figure net out returns of faulty goods?
- Does the revenue figure include or exclude sales taxes?

- Are the figures for a particular time period based on the calendar, or on financial closings?
- Do customer groups refer to holding companies (i.e., real owners) or to some generic similarity between customers?
- When someone in another department talks about product types, to what is he or she referring?

Clearly, even for such a relatively simple report, the potential for misunderstanding and misanalysis is great if the answers to these questions are unknown and unobtainable. It is the job of the informational metadata to control and proliferate answers to these questions, thus ensuring that the users of the data warehouse and data mart environments are working according to a consistent set of rules.

In practice, there are few tools on the market that handle the informational metadata-base in a generic manner. Most offerings in this area are deeply embedded within the tools that provide the decision support facilities to the end user.

Recently, however, a new type of metadata management tool has been introduced to the market. These tools, typified by the MetaExchange™ system from Pine Cone Systems, aim at the large-scale data warehouse market and seek to respond to a need for companies to control metadata along the entire chain of BI systems.

MetaExchange™ endeavors to integrate the metadata from a variety of systems, taking different perspectives into account. The source systems for data, the data warehouse, and the marts spawned from it will all have autonomous data models that are managed as a single library. In addition, the technical, operational, and informational metadata are managed in a cohesive fashion to ensure that the rules that are embedded within them remain compatible.

The physical sharing of metadata is achieved by means of middleware interactions with a so-called metadata hub. Standard protocols, such as CORBA and DCOM, are used to enforce object-level consistency, and emerging standards for metadata definition, including XML, are supported. The purpose of this hub is to control sharing of metadata (via these mechanisms) and business data (by supporting a common neutral business data model). The control of business data is reinforced by the

concept of identified *systems of record* for each part of the overall business database.

6.5 Decision support and OLAP

Once the data have arrived within the data warehouse environment, there is clearly a need to extract them in a form that can be easily and powerfully manipulated by end users, the consumers of information. These information consumers come in a variety of types, broadly defined as follows:

1. Basic users, requiring standardized reports from the warehouse environment on a regular basis to highlight fixed types of information.
2. Exploratory users, who need to be able to view the data at a variety of levels of detail, manipulating them to determine the patterns of behavior within the business. This user typically makes heavy use of the "drill-up" and "drill-down" facilities within an online analytical processing (OLAP) tool. The work performed is used to make or contribute to tactical level decisions within the organization.
3. Analytical users, who need to be able to perform sophisticated statistical analyses on the data, determining the correlation between different business factors and making or contributing to strategic level decisions as a result.

These different types of users form a BI continuum, effectively defining a spectrum of needs, ranging from the basic to the sophisticated. The manager of a data warehousing project has the responsibility for identifying which of these groups of data consumers exist within the organization and what tools can service their needs.

This brings us back to the inherent problem with informational metadata, which is that each BI tool will generally have an embedded mechanism for controlling the metadata that fulfills a focused purpose but does little to guarantee consistency across the enterprise. Bearing in mind the likelihood that the data consumers will be spread widely across the BI

continuum and will thus require different tools, this can present a major problem.

6.6 BI tool type 1: Multidimensional OLAP

The first category of tool that should be considered is the so-called multidimensional OLAP (MOLAP) facility. These tools utilize a multidimensional model of the data and allow navigation from one level of detail to another online, in real time. As an example, these tools would offer the ability to see sales figures broken down by sales area by year. The user would be allowed to change the level of detail dynamically, by drilling down into a given year to see monthly figures and into months to see daily figures. Similarly, the user could break down the sales areas into districts and individual customers. The point of OLAP tools is that this flexibility is provided in real time, *regardless of the amount of recalculation needed*. Bearing in mind that data warehouses typically contain vast amounts of data, the ability to be able to present summaries at different levels, while maintaining acceptable response times, is clearly a nontrivial problem.

Imagine a data consumer who would like to view a worldwide sales matrix, giving volumes by country, by product line, and by time, along three axes, or dimensions. This user has a further requirement to be able to "drill down" from a country into a regional level and from there into a city level, having the sales figures along the other two dimensions dynamically updated in real time. This user may further require the facility to drill down on more than one dimension simultaneously, jumping from a national product line view to a regional, individual product view of the same base data, once again in real time. Bearing in mind the number of records involved in making these calculations for a large company (assuming the ability to drill all the way down to individual sales order lines), there is a considerable problem in delivering the necessary speed to the end user.

For this reason, some vendors have abandoned the traditional relational database technology and have implemented OLAP tools based on a multidimensional model. The database engines for these arrays or "cubes" of data are generally proprietary, close architectures. It is not possible to

access the data using SQL, since the structures employed within the multidimensional databases are not relational. In most cases, the vendors have implemented proprietary mechanisms within the DBMS to ensure that the online response times can be minimized. As a consequence, neither the data nor the metadata (in the form of systems tables) are directly accessible via industry-standard relational access tools based around SQL. Table 6.1 shows a typical multidimensional model.

Each dimension listed in Table 6.1 represents a characteristic type by which the user may wish to analyze the data. The tiers are the levels of detail to which the user may navigate, and the measures are numeric data that may be aggregated across any or all of the dimensions.

Because of the closed nature of such tools, and because of the targeting of these tools toward particular groups of analytical users, facilities to manage metadata on an enterprise-wide basis are rarely incorporated.

6.7 BI tool type 2: Relational OLAP

A further category of decision support system (DSS) tools have evolved to service the needs of data consumers typically at the lower end of the BI continuum; these tools draw directly from a data warehouse or mart environment. These relational OLAP (ROLAP) tools have the advantage over MOLAP tools in that they do not require massive extra storage, since the data is not physically copied from the mart environment. From our perspective, this means that the metadata within the mart level can be accessed and are directly relevant to the eventual use made of it.

Table 6.1
Multidimensional Model

Dimension	Time	Product	Customer
Tier 0	All time	All products	All customers
Tier 1	Year	Product group	Customer type
Tier 2	Quarter	Product category	Customer
Tier 3	Month	Product	
Measures	Sales revenue, sales volume, number of items returned		

ROLAP tools do, however, have significant performance disadvantages over MOLAP, which is often a critical factor for large data warehouse implementations.

ROLAP tools, such as Business Objects™, have limited metadata access as part of their report design and construction facilities. Since they rely on direct access to their source relational database for their reporting abilities, they must in turn have some degree of access to the native metadata within the source DBMS itself. In the case of Business Objects, this metadata is accessed to build so-called universes, or views of the data from which formal or ad-hoc reports can be produced. The universes can in themselves provide a means for formalizing and recording metadata, at the risk of introducing yet another version of the "truth" embedded within it.

6.8 Informational metadata summary

If the relational approach is adopted, the possibility for "self-built" metadata control and proliferation should be considered, since the metadata themselves is both relatively simple and easily accessible within the data dictionary of most industrial strength relational database management systems (RDBMS). The following entities will need to be controlled and proliferated around the enterprise to ensure that users have a clear view of the informational metadata:

- Entity within application;
- Aliases—alternative names for entities and attributes—ensuring that there is common understanding of the business rules, despite the use of different jargon within the enterprise;
- System of record (the single system that represents the authoritative source for each item of data);
- Attribute within application;
- Global entity;
- Global attribute;
- Domains;

- Volumes;
- Cardinality;
- Volatility;
- Dimensions;
- Tiers;
- Measures;
- Nonmeasure attributes.

6.9 Metadata management and proliferation within a data warehouse

Most current implementations of data warehousing solutions include separate metadata repositories within the application from which the data is sourced, within the ETL tool, and embedded within the OLAP or DSS front end.

Often, certain parts of this metadata-base are shared across tools (for instance, the source data dictionary may be accessed directly by the ETL tool to define source attributes), while others are independent and isolated.

In theory, there are several approaches that can be adopted to ensure metadata consistency:

1. Adopt a universal metadata control tool that will maintain active consistency between the various repositories—unfortunately no such tools exist at the time of writing.
2. Rely on the bilateral sharing of metadata between the various components along the information supply chain—this will provide a partial solution but is likely to exclude some of the more "user-facing" elements of metadata control.
3. Build automated tools that check on the consistency of the various repositories and produce control reports for the data architect to examine and use to take action. This may well prove effective but will require companies to build a profound knowledge of

the internals of all the individual metadata repositories along the information supply chain.

4. Define a combination of manual procedures, documentation standards, and automated facilities that will enable effective consistency to be managed. For many companies, this will be the most practical approach in the medium term.

If we adopt the "hybrid" approach (4), the whole process will be driven by the documentation and interlinking of metadata entities. This can be achieved in a number of ways:

- *On paper:* The simplest to implement but the least controllable and updatable mechanism—likely to be significant drawbacks;
- *Using a work flow tool (e.g., Lotus Notes) to record the various metadata entities as separate document types and to utilize the built-in look-up features within the tool to maintain some sort of integrity:* Will also ensure that the metadata is accessible to developers and users alike;
- *Doing the same sort of thing as an intranet implementation—once again, a good method of sharing metadata—by setting up linked pages, or possibly a simple intranet-accessible database, containing metadata for the enterprise:* Open access that is simple and quick to set up and reasonably easy to use (if well-designed);
- *CASE tools (likely restricted to the "technical" end of the metadata spectrum):* Contain extensive metadata but tend to be tailored for use by systems analysts and people with a solid understanding of data modeling, rather than the general user population, which can severely restrict their use for proliferation of informational metadata;
- *Building a dedicated metadata management utility from scratch:* Probably overkill, and it requires a very peculiar combination of skill to design and build such a tool!

6.10 Conclusions

In the longer term, the management of metadata across a data warehouse environment will always present a significant problem, certainly with the existing technology. The basic issue is how to ensure that the behavior of the data—which arrives within the warehouse environment and is transformed and loaded into separate data marts, restructured by OLAP and DSS tools, and then separately interpreted by users—remains consistent.

At each stage, the possibility exists for reinterpretation of the business rules that govern this behavior, introducing inconsistencies. These inconsistencies will be carried forward into the reports, leading to errors of judgment, misreporting of statistics, and time wasted in trying to reconcile apparent conflicts in the figures.

All of the above are included in the list of problems that data warehousing has set out to solve! We therefore need to ensure that the metadata, which defines and controls the behavior of data at each stage, is under control and consistent.

Reference

[1] Inmon, W. H., *Building the Data Warehouse*, Wiley-QED [Technical Publishing Group] New York, 1993.