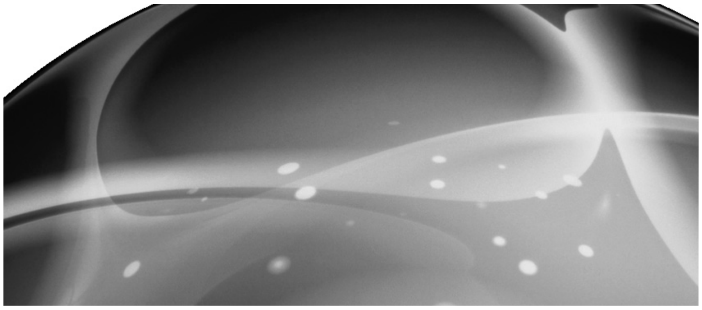

CHAPTER

6

Data Management Concerns of MDM-CDI Architecture

IN THIS CHAPTER

Data Strategy

Managing Data in the Data Hub

The preceding chapters discussed the enterprise architecture framework as the vehicle that helps resolve a multitude of complex and challenging issues facing MDM and CDI designers and implementers. As we focused on the Customer Data Integration aspects of the MDM architecture, we showed how to apply the Enterprise Architecture Framework to the service-oriented view of the CDI platform, often called a Data Hub. And we also discussed a set of services that any Data Hub platform should provide and/or support in order to deliver key data integration properties of matching and linking detail-level records—a service that enables the creation and management of a complete view of the customers, their associations and relationships.

We also started a discussion of the services required to ensure the integrity of data inside the Data Hub as well as services designed to enable synchronization and reconciliation of data changes between the Data Hub and surrounding systems, applications, and data stores.

We have now reached the point where the discussion of the Data Hub architecture cannot continue without considering issues and challenges of integrating a Data Hub platform into the overall enterprise information environment. To accomplish this integration, we need to analyze the Data Hub architecture components and services that support cross-systems and cross-domain information management requirements. These requirements include challenges of the enterprise data strategy, data governance, data quality, a broad suite of data management technologies, and the organizational roles and responsibilities that enable effective integration and interoperability between the Data Hub, its data sources, and its consumers (users and applications).

An important clarification: as we continue to discuss key issues and concerns of the CDI architecture, services, and components, we focus on the logical and conceptual architecture points of view. That means that we express functional requirements of CDI services and components in architecture terms. These component and service requirements should not be interpreted literally as the prescription for a specific technical implementation. Some of the concrete implementation approaches—design and product selection guidelines that are based on the currently available industry best practices and state of the art in the MDM and CDI product marketplace—are provided in Part IV of this book.

Data Strategy

This chapter deals primarily with issues related to data management, data delivery, and data integration between a Data Hub system, its sources, and its consumers. In order to discuss the architecture concerns of data management we need to expand the context of the enterprise architecture framework and its data management dimensions by introducing key concerns and requirements of the enterprise data strategy. While these concerns include data technology and architecture components, the key insights of the enterprise data strategy are contained in its holistic and multidimensional approach to the issues and concerns related to enterprise-class information management. Those readers already familiar with the concepts of data strategy, data governance, and data stewardship can easily skip this section and proceed directly to the section titled "Managing Data in the Data Hub."

The multifaceted nature of the enterprise data strategy includes a number of interrelated disciplines such as data governance, data quality, data modeling, data management, data delivery, data synchronization and integrity, data security and privacy, data availability, and many others. Clearly, any comprehensive discussion of the enterprise data strategy that covers these disciplines is well beyond the scope of this book. However, in order to define and explain Data Hub requirements to support enterprise-level integration with the existing and new applications and systems, at a minimum we need to introduce several key concepts behind data governance, data quality, and data stewardship. Understanding these concepts helps explain the functional requirements of those Data Hub services and components that are designed to find the "right" data in the "right" data store, to measure and improve data quality, and to enable business-rules-driven data synchronization between the Data Hub and other systems. We address the concerns of data security and privacy in Part III of this book, and additional implementation concerns in Part IV.

Data Governance

Let's consider the following working definition of data governance.

Data Governance

Data governance is a process focused on managing the quality, consistency, usability, security, and availability of information. This process is closely linked to the notions of data ownership and stewardship.

Clearly, according to this definition, data governance becomes a critical component of any Data Hub initiative. Indeed, an integrated CDI data architecture contains not only the Data Hub but also many applications and databases that more often than not were developed independently, in a typical stovepipe fashion, and the information they use is often inconsistent, incomplete, and of different quality.

Data governance strategy helps deliver appropriate data to properly authorized users when they need it. Moreover, data governance and its data quality component are responsible for creating data quality standards, data quality metrics, and data quality measurement processes that together help deliver acceptable quality data to the consumers—applications and end users.

Data quality improvement and assurance are no longer optional activities. For example, the 2002 Sarbanes-Oxley Act requires, among other things, that a business entity should be able to attest to the quality and accuracy of the data contained in their financial statements. Obviously, the classical "garbage in—garbage out" expression is still true, and no organization can report high-quality financial data if the source data used to produce the financial numbers is of poor quality. To achieve compliance and to successfully implement an enterprise data governance and data quality strategy, the strategy itself should be treated as a value-added business proposition, and sold to the organization's stakeholders to obtain a management buy-in and commitment like any other business case. The value of improved data quality is almost self-evident, and includes factors such as the enterprise's ability to make better and more accurate decisions, to gain deeper insights into the customer's behavior, and to understand the customer's propensity to buy products and services, the probability of the customer's engaging in high-risk transactions, the probability of attrition, etc. The data governance strategy is not limited to data quality and data management standards and policies. It includes critically important concerns of defining organizational structures and job roles responsible for monitoring and enforcement of compliance with these policies and standards throughout the organization.

Committing an organization to implement a robust data governance strategy requires an implementation plan that follows a well-defined and proven methodology. Although there are several effective data governance methodologies available, a detailed discussion of them is beyond the scope of this book. However, for the sake of completeness, this section reviews key steps of a generic data governance strategy program as it may apply to the CDI Data Hub:

- *Define a data governance process.* This is the key in enabling monitoring and reconciliation of data between Data Hub and its sources and consumers. The data governance process should cover not only the initial data load but also data refinement, standardization, and aggregation activities along the path of the end-to-end information flow. The data governance process includes such data management and data quality concerns as the elimination of duplicate entries and creation of linking and matching keys. We showed in Chapter 5 that these unique identifiers help aggregate or merge individual records into groups or clusters based on certain criteria, for example, a household affiliation or a business entity. As the Data Hub is integrated into the overall enterprise data management environment, the data governance process should define the mechanisms that create and maintain valid cross-reference information in the form of Record Locator metadata that enables linkages between the Data Hub and other systems. In addition, a data governance process should contain a component that supports manual corrections of false positive and negative matches as well as the exception processing of errors that cannot be handled automatically.

- *Design, select, and implement a data management and data delivery technology suite.* In the case of a CDI Data Hub both data management and data delivery technologies play a key role in enabling a fully integrated CDI solution regardless of the architecture style of the Data Hub, be it a Registry, a Reconciliation Engine, or a Transaction Hub. Later in this chapter we will use the principles and advantages of service-oriented architecture (SOA) to discuss the data management and data delivery aspects of the Data Hub architecture and the related data governance strategy.
- *Enable auditability and accountability for all data under management that is in scope for data governance strategy.* Auditability is extremely important as it not only provides verifiable records of the data access activities, but also serves as an invaluable tool to help achieve compliance with the current and emerging regulations including the Gramm-Leach-Bliley Act and its data protection clause, the Sarbanes-Oxley Act, and the Basel II Capital Accord. Auditability works hand in hand with accountability of data management and data delivery actions. Accountability requires the creation and empowerment of several data governance roles within the organization including data owners and data stewards. These roles should be created at appropriate levels of the organization and assigned to the dedicated organizational units or individuals.

To complete this discussion, let's briefly look at the concept of data stewards and their role in assessing, improving, and managing data quality.

Data Stewardship and Ownership

As the name implies, data owners are those individuals or groups within the organization that are in the position to obtain, create, and have significant control over the content (and sometimes, access to and the distribution of) the data. Data owners often belong to a business rather than a technology organization. For example, an insurance agent may be the owner of the list of contacts of his or her clients and prospects.

The concept of data stewardship is different from data ownership. Data stewards do not own the data and do not have complete control over its use. Their role is to ensure that adequate, agreed-upon quality metrics are maintained on a continuous basis. In order to be effective, data stewards should work with data architects, database administrators, ETL (Extract-Transform-Load) designers, business intelligence and reporting application architects, and business data owners to define and apply data quality metrics. These cross-functional teams are responsible for identifying deficiencies in systems, applications, data stores, and processes that create and change data and thus may introduce or create data quality problems. One consequence of having a robust data stewardship program is its ability to help the members of the IT organization to enhance appropriate architecture components to improve data quality.

Data stewards must help create and actively participate in processes that would allow the establishment of business-context-defined, measurable data quality goals. Only after an organization has defined and agreed with the data quality goals can the data stewards devise appropriate data quality improvement programs.

These data quality goals and the improvement programs should be driven primarily by business units, so it stands to reason that in order to gain full knowledge of the data quality issues, their roots, and the business impact of these issues, a data steward should be a member of a business team. Regardless of whether a data steward works for a business team or acts as a "virtual" member of the team, a data steward has to be very closely aligned with the information technology group in order to discover and mitigate the risks introduced by inadequate data quality.

Extending this logic even further, we can say that a data steward would be most effective if he or she can operate as close to the point of data acquisition as technically possible. For example, a steward for customer contact and service complaint data that is created in a company's service center may be most effective when operating inside that service center.

Finally, and in accordance with data governance principles, data stewards have to be accountable for improving the data quality of the information domain they oversee. This means not only appropriate levels of empowerment but also the organization's willingness and commitment to make the data steward's data quality responsibility his or her primary job function, so that data quality improvement is recognized as an important business function required to treat data as a valuable corporate asset.

Data Quality

Data Quality

Data quality is one of the key components of any successful data strategy and data governance initiative, and is one of the core enabling requirements for Master Data Management and Customer Data Integration.

Indeed, creating a new system of record from information of low quality is almost an impossible task. Similarly, when data quality is poor, matching and linking records for potential aggregation will most likely result in low match accuracy and produce an unacceptable number of false negative and false positive outcomes.

Valuable lessons about the importance of data quality are abundant, and data quality concerns confronted data architects, application designers, and business users even before the problem started to manifest itself in the early data integration programs such as Customer Information Files (CIF), early implementations of data warehouses (DW), Customer Relationship Management (CRM), and Business Intelligence (BI) solutions. Indeed, if you look at a data integration solution such as a data warehouse, published statistics show that as high as 75 percent of the data warehouse development effort is allocated to data preparation, validation, and extraction, transformation, and loading (ETL). Over 50 percent of these activities are spent on cleansing and standardizing the data.

Although there is a wide variety of ETL and data cleansing tools that address some of the data quality problem, data quality continues to be a complex, enterprise-class challenge. Part of the complexity that needs to be addressed is driven by the ever-increasing performance requirements. A data cleansing tool that would take more than 24 hours to cleanse a customer file is a poor choice for a real-time or a web-based customer service application. As the performance and throughput requirements continue to increase, the functional and technical capabilities of the data quality tools are sometimes struggling to keep up with the demand.

But performance is not the primary issue. A key challenge of data quality is an incomplete or unclear set of semantic definitions of what the data is supposed to represent, in what form, with what kind of timeliness requirements, etc. These definitions are ideally stored in a metadata repository. Our experience shows that even when an enterprise adapts a metadata strategy and implements a metadata repository, its content often contains incomplete or erroneous (poor quality) definitions. We'll discuss metadata issues in more details later in this chapter.

The quality of metadata may be low not because organizations or data stewards do not work hard on defining it, but primarily because there are many data quality dimensions and contexts, each of which may require a different approach to the measurement and improvement of the data quality. For example, if we want to measure and improve address information about the customers, there are numerous techniques and reference data sources that can provide an accurate view of a potentially misspelled or incomplete address. Similarly, if we need to validate a social security number or a driver license number, we can use a variety of authoritative sources of this information to validate and correct the data. The problem becomes much harder when you deal with names or similar attributes for which there is no predefined domain or a business rule. For example, "Alec" may be a valid name or a misspelled "Alex." If evaluated independently, and not in the context of, say, postal information about a name and the address, this problem often requires human intervention to resolve the uncertainty.

Finally, as the sophistication of the data quality improvement process grows, so do its cost and processing requirements. It is not unusual to hear that an organization would be reluctant to implement an expensive data quality improvement system because, according to them, "…so far the business and our customers do not complain, thus the data quality issue must not be as bad as you describe." This is not an invalid argument, although it may be somewhat shortsighted from the strategic point of view, especially since many aspects of data quality fall under government- and industry-regulated requirements.

Data Quality Tools and Technologies

There are many tools that automate portions of the tasks associated with cleansing, extracting, loading, and auditing data from existing data stores into a new target environment, be it a data warehouse or a CDI Data Hub. Most of these tools fall into several major categories:

- **Auditing tools** These tools enhance the accuracy and correctness of the data at the source. These tools generally compare the data in the source database to a set of business rules that are either explicitly defined or automatically inferred from a scan operation of the data file or a database catalog. Auditing tools can determine the cardinality of certain data attributes, value ranges of the attributes in the data set, and the missing and incomplete data values, among other things. These tools would produce various data quality reports and can use their output to automate certain data cleansing and data correction operations.
- **Data cleansing tools** These tools would employ various deterministic, probabilistic or machine learning techniques to correct the data problems discovered by the auditing tools. These tools generally compare the data in the data source to a set of business rules and domain constraints stored in the metadata repository or in an external rules repository. Traditionally, these tools were designed to access external, reference data such as a valid name and address file from an external "trusted" data provider (e.g., Acxiom or Dun & Bradstreet), or an authoritative postal information file (e.g., National Change of Address [NCOA] file), or to use a service that validates social security numbers. The data cleansing process improves the quality of the data and potentially adds new, accurate content. Therefore, this process is sometimes referred to as *data enrichment*.
- **Data parsing and standardization tools** The parsers would break a record into atomic units that can be used in subsequent steps. For example, such a tool would parse one contiguous address record into separate street, city, state, and zip code fields. Data standardization tools convert the data attributes to what is often called a *canonical* format or canonical data model—a standard format used by all components of the data acquisition process and the target Data Hub.

Canonical Data Format

Canonical data format is a format that is independent of any specific application. It provides a level of abstraction from applications' native data formats by supporting a common format that can either be used by all applications or may require transformation adapters that convert data between the canonical and native formats. Adding a new application or a new data source may only require a new adapter or modifying an old one, thus drastically reducing the impact on applications. A canonical format is often encoded in XML.

- **Data extraction, transformation, and loading (ETL) tools** are not data quality tools in the pure sense of the term. ETL tools are primarily designed to extract data from known structures of the source systems based on prepared and validated source data mapping, transforming input formats of the extracted files into a predefined target data store format (e.g., a Data Hub), and loading the transformed data into a target data environment, e.g., the Data Hub. Since ETL tools are aware of the target schema, they can prepare and load the data to preserve various integrity constraints including referential integrity and the domain integrity constraints. They can filter out records that fail a data validity check, and usually produce exception reports used by data stewards to address data quality issues discovered at the load stage. This functionality helps ensure data quality and integrity of the target data store, which is the reason we mentioned ETL tools in this section.
- **Hybrid packages** These packages may contain a complete set of ETL components enriched by a data parser and a standardization engine, the data audit components, and the data cleansing components. These extract, parse, standardize, cleans, transform, and load processes are executed by a hybrid package software in sequence and load consistently formatted and cleansed data into the Data Hub.

Managing Data in the Data Hub

Armed with the knowledge of the role of the enterprise data strategy, we can discuss CDI Data Hub concerns that have to deal with acquiring, rationalizing, cleansing, transforming, and loading data into the Data Hub as well as the concerns of delivering the right data to the right consumer at the right time. In this chapter, we also discuss interesting challenges and approaches of synchronizing data in the Data Hub with applications and systems used to source the data in the first place.

Let's start with the already familiar Data Hub conceptual architecture that we first introduced in Chapter 5. This architecture shows the Data Hub data store and supporting services in the larger context of the data management architecture (see Figure 6-1). From the data strategy point of view, this architecture depicts data sources that feed the loading process, data access and data delivery interfaces, Extract-Transform-Load service layer, the Data Hub platform, and some generic consuming applications.

However, to better position our discussion of the data-related concerns, let's transform our Data Hub conceptual architecture into a view that is specifically designed to emphasize data flows and operations related to managing data in and around the Data Hub.

Data Zone Architecture Approach

To address data management concerns of the Data Hub environment, we introduce a concept of the data zones and the supporting architectural components and services. The Data Zone architecture illustrated in Figure 6-2 employs sound architecture principles of the separation of concerns and loose coupling.

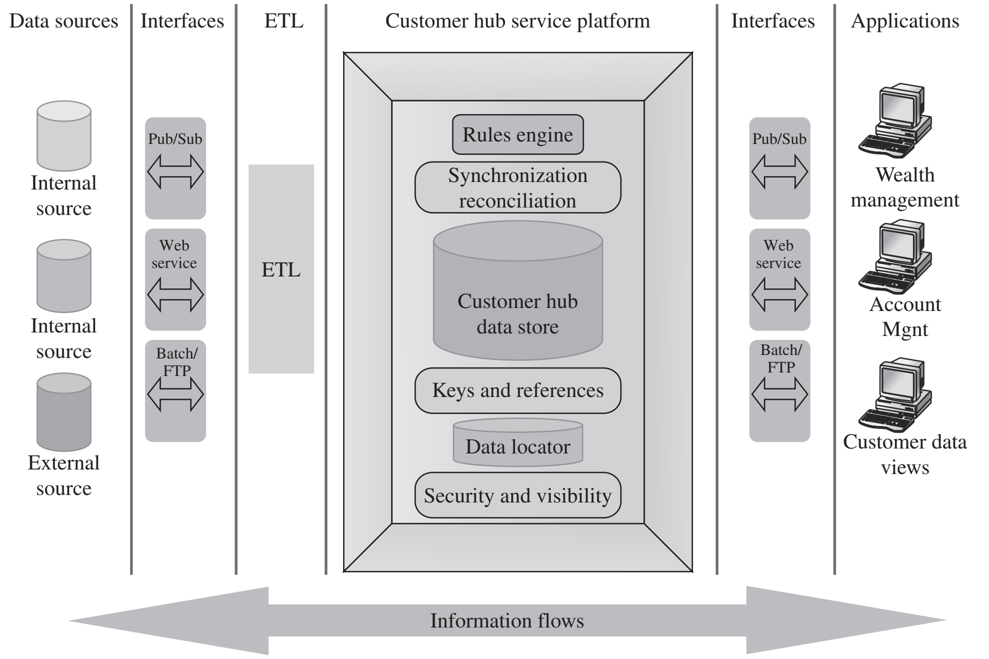

Figure 6-1 *Conceptual Data Hub components and services architecture view*

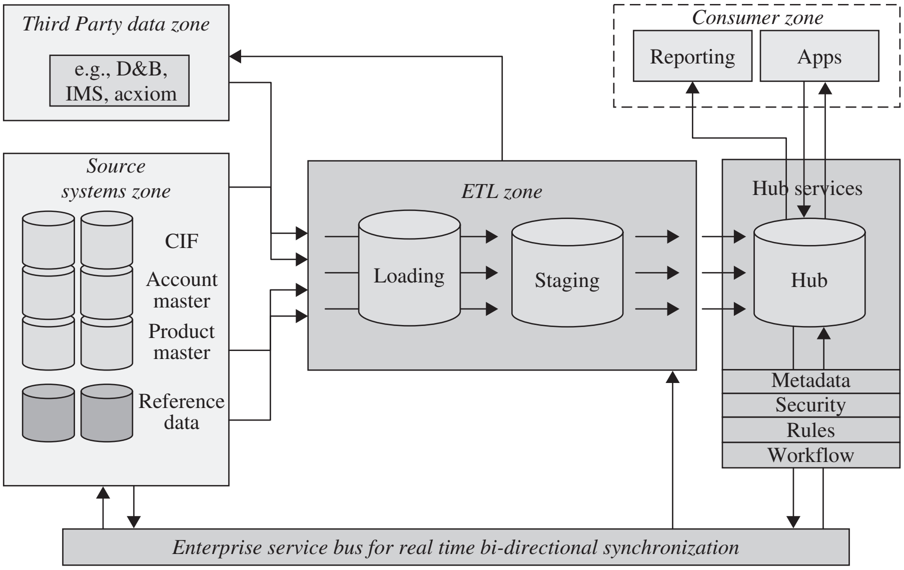

Figure 6-2 *Data Hub architecture—Data Zone view*

Separation of Concerns

In software design, the principle of *separation of concerns* is linked to specialization and cooperation: When designing a complex system, the familiar trade-off is between a few generic modules that can perform various functions versus many specialized modules designed to work together in a cooperative fashion. In complex systems, specialization of components helps address the required functionality in a focused fashion, organizing groups of concerns into separate, designated, and specifically designed components.

Turning to nature, consider the difference between simple and complex organisms. Where simple organisms contain several generic cells that perform all life-sustaining functions, a complex organism (e.g., an animal) is “built” from a number of specialized “components” such as heart, lungs, eyes, etc. Each of these components performs its functions in a cooperative fashion together with other components of the body. In other words, when the complexity is low to moderate, having a few generic components simplifies the overall design. But, as the complexity of the system grows, the specialization of components helps address the required functionality in a focused fashion, by organizing groups of concerns into separate specifically designed components.

We briefly discussed the principle of Loose Coupling in the previous chapter when we looked at service-oriented architectures.

Loose Coupling

In software design, *loose coupling* refers to the design approach that avoids rigid, tightly coupled structures where changes to one component force that change to propagate throughout the systems, and where a failure or a poor performance of one component may bring the entire system down.

When we apply these architecture principles to the data architecture view of the Data Hub, we can clearly delineate several functional domains, which we call *zones*.

The Data Zones shown in Figure 6-2 include the following:

- Source Systems zone
- Third-Party Data Provider zone
- ETL/Acquisition zone
- Hub Services zone
- Information Consumer zone
- Enterprise Service Bus zone

To make it very clear, this zone structure is a logical design construct that should be used as a guide to help solve the complexity of data management issues. The Zone Architecture approach allows architects to consider complex data management issues in the context of the overall enterprise data architecture. As a design guide, it does not mean that a Data Hub implementation has to include every zone and every component. A specific CDI implementation may include a small subset of the data zones and their respective processes and components.

Let's review the key concerns addressed by the data zones shown in Figure 6-2.

- **The Source Systems zone** is the province of existing data sources, and the concerns of managing these sources include good procedural understanding of data structures, content, timeliness, update periodicity, and such operational concerns as platform support, data availability, data access interfaces, access methods to the data sources, batch window processing requirements, etc. In addition to the source data, this zone contains enterprise reference data, such as code tables used by an organization to provide product-name-to-product-code mapping, state code tables, branch numbers, account type reference tables, etc.

This zone contains "raw material" that is loaded into the Data Hub and uses information stored in the metadata repository to determine data attributes, formats, source system names, and location pointers.

- **The Third-Party zone** deals with external data providers and their information. An organization often purchases this information to cleanse and enrich input data prior to loading it into a target environment such as a Data Hub. For example, if the Data Hub is designed to handle customer information, the quality of the customer data loaded into the Data Hub would have a profound impact on the linking and matching processes as well as on the Data Hub's ability to deliver an accurate and complete view of a customer. Errors, use of aliases, and lack of standards in customer name and address fields are most common and are the main cause of poor customer data quality. To rectify this problem an organization may decide to use a third-party data provider that specializes in maintaining an accurate customer name and address database (for example, Acxiom, D&B, etc.). The third-party data provider usually would receive a list of records from an organization, would match them against the provider's database of verified and maintained records, and would send updated records back to the organization for processing. Thus the third-party zone is concerned with the following processes:
    - Creating a file extract of customer records to be sent to the provider
    - Ensuring that customer records are protected and that only absolutely minimal necessary information is sent to the provider in order to protect confidential data
    - Receiving an updated file of cleansed records enriched with accurate and perhaps additional information
    - Making the updated file available for the ETL processing
    - Making appropriate changes to the content of the metadata repository for use by other data zones
- **The ETL/Acquisition zone** is the province of data extract, transformation, and loading (ETL) tools and corresponding processes. These tools are designed to extract data from known structures of the source systems based on prepared and validated source-to-target data mapping; transforming input formats of the extracted files into a predefined target data store format (e.g., a Data Hub); and loading the transformed data into the Data Hub using either a standard technique or a proprietary one. The transformations may be quite complex and can perform substitutions, aggregations, and logical and mathematical operations on data attribute values. ETL tools may access an internal or external metadata repository to obtain the information about the transformation rules, integrity constraints, and target Data Hub schema, and therefore can prepare and load the data while preserving various integrity constraints. Many proven, mature solutions can perform ETL operations in an extremely efficient, scalable fashion.

They can parallelize all operations to achieve very high performance and throughput on very large data sets. These solutions can be integrated with an enterprise metadata repository and a BI tool repository.

  - An effective design approach to the data acquisition/ETL zone is to use a multistage data acquisition environment. To illustrate this point, we consider a familiar analogy of using loading dock for "brick-and-mortar" warehouse facility. Figure 6-2 shows a two-stage conceptual Acquisition/ETL data zone where the first stage, called *Loading zone*, is acting as a recipient of the data extraction activities. Depending on the complexity and interdependencies involved in data cleansing, enrichment, and transformation, a Loading zone may serve as a facility where all input data streams are normalized into a common, canonical format. The third-party data provider usually receives an appropriate set of data in such a canonical format. The Loading zone is a convenient place where the initial audit of input records can take place.
  - The *Staging zone*, on the other hand, is a holding area for the already cleansed, enriched, and transformed data received from the Loading zone as well as the data processed by and received from a third-party data provider. The Staging zone data structure could be similar to that of the Data Hub. The benefits of having a Staging zone include efficiency in loading data into the Data Hub (most often using a database utility since the transformations are already completed). The Staging zone offers access to a convenient area to perform a record-level audit before completing the load operation. Finally, a Staging zone provides for an easy-to-use, efficient, and convenient Data Hub reload/recovery point that does not depend on the availability of the source systems.

- **The Hub Service data zone** deals with the data management services that create and maintain the structures and the information content inside the Data Hub. We discussed several of these services in the previous chapter. In this chapter, we discuss Data Hub services that support data synchronization and reconciliation of conflicting data changes. Some Data Hub services use a metadata repository to enforce semantic consistency of the information. Other services include linking, matching, record locator, and attribute locator services.
- **The Information Consumer zone** is concerned with data-delivery-related issues such as formats, messages, protocols, interfaces, and services that enable effective and easy-to-use access to the required information whether it resides in the Data Hub or in the surrounding systems. The Information Consumer zone is designed to provide data to support business applications including Business Intelligence applications, CRM, and functional applications such as account opening and maintenance, aggregated risk assessment, and others. The Information Consumer zone enables persistent and virtual, just-in-time

data integration technologies including Enterprise Information Integration (EII) solutions. Like other data zones, the information consumer zone takes advantage of the metadata repository to determine data definitions, data formats, and data location pointers.

- **The Enterprise Service Bus (ESB)** zone deals with technologies, protocols, message formats, interfaces, and services that support a message-based communication paradigm between all components and services of the CDI data architecture. The goal of ESB is to support the loosely coupled nature of the Data Hub service-oriented architecture (SOA) by providing a message-based integration mechanism that ensures guaranteed, once and only once, sequence-preserving, transactional message delivery.

Now that we have reviewed the content and purpose of the architecture zones, we can expand these concepts by including several high-level published services that are available to various data architecture components including Data Hub. These services include

1. Data acquisition services
2. Data normalization and enrichment services
3. Data Hub management services
4. Data synchronization and reconciliation services
5. Data location and delivery services

We discuss some of these services in Chapter 5. The following sections offer a discussion on additional data management services in the context of the Data Zone architecture.

Loading Data into the Data Hub

Data architecture concerns discussed in the beginning of this section have a profound impact on the overall Data Hub architecture and in particular, its data management and data delivery aspects. The Data Zone architecture view shown in Figure 6-2 can help define new effective design patterns, additional services and components that would support any generic data integration platform, and in particular, a Data Hub system for Customer Data Integration.

The level of abstraction of the data zone architecture is sufficiently high to be applicable equally well to all major styles of the Data Hub design including Registry style, Reconciliation Hub style, and ultimately, full Transaction Hub style. However, as we take a closer look at these design styles, we discover that the way the data is loaded and synchronized varies significantly from one style to another.

Indeed, consider the key difference between these styles—the scope of data for which the Hub is the master, specifically:

- **The Registry style of a Data Hub** represents a master of unique identifiers of customer "match groups" and all key attributes (often called identity attributes) that allow Data Hub Linking and Matching services to generate these unique persistent identifiers. The Registry-style Data Hub maintains links with data sources for the identity attributes to provide a clear synchronization path between data sources and the Data Hub. The Registry-style Data Hub allows the consuming application to either retrieve or assemble an integrated view of customers or parties at run time.
- **The Reconciliation Engine style** (sometimes also called Coexistence Hub) supports an evolutionary stage of the Data Hub that enables coexistence between the old and new masters, and by extension, provides for a federated data ownership model that helps address both inter- and intraorganizational challenges of who controls which data. The Data Hub of this style is a system of record for *some* but not all data attributes. It provides active synchronization between itself and the systems that were used to create the Hub data content or still maintain some of the Hub data attributes inside their data stores. By definition of the "master," the data attributes for which the Data Hub is the master need to be maintained, created, and changed in the Data Hub. These changes have to be propagated to the upstream and downstream systems that use these data attributes. The goal is to enable synchronization of the data content between the Data Hub and other systems on a continuous basis. The complexity of this scenario increases dramatically as some of the data attributes maintained in the Data Hub are not simply copied but rather *derived* using business-defined transformations on the attributes maintained in other systems.
- **The Transaction Hub** represents a design style where the Hub maintains *all* data attributes about the target subject area. In the case of a CDI Data Hub, the subject area is the customer (individuals or businesses). In this case, the Data Hub becomes a "master" of customer information, and as such should be the source of all changes that affect any data attribute about the customer. This design approach demands that the Data Hub is engineered as a complete transactional environment that maintains its data integrity and is the sole source of changes that it propagates to all downstream systems that use this data.

A conceptual Data Hub architecture shown in Figure 6-1 and its Data Zone viewpoint shown in Figure 6-2 should address several *common* data architecture concerns:

- **Batch and real-time input data processing** Some or all data content in the Data Hub is acquired from existing internal and external data sources. The data acquisition process affects the Source System zone, theThird-Party Data Provider zone, and the Data Acquisition/ETL zone. It uses several relevant services including data acquisition services, data normalization and enrichment services, and Data Hub management services such as Linking and Matching, Key Generation, Record Locator, and Attribute Locator services (see Chapters 4 and 5 for more details). Moreover, the data acquisition process can support two different modes—initial data load and delta processing of incremental changes. The former implies a full refresh of the Data Hub data content, and it is usually designed as a batch process. The delta processing mode may support either batch or real-time processing. In the case of batch design, the delta processing, at least for the new inserted records, can leverage the same technology components and services used for the initial data load. The technology suite that enables the initial load and batch delta processing has to support high-performance, scalable ETL functionality that architecturally "resides" in the Acquisition/ETL data zone and usually represents a part of the enterprise data strategy and architecture framework. Real-time delta processing, on the other hand, should take full advantage of service-oriented architecture including the Enterprise Service Bus zone, and in many cases is implemented as a set of transactional services that include Data Hub management services and synchronization services.
- **Data quality processes** To improve the accuracy of the matching and linking process, many Data Hub environments implement data cleaning, standardization, and enrichment preprocessing in the Third-Party Data Provider and Acquisition/ETL zones before the data is loaded into the Data Hub. These processes use data acquisition and data normalization and enrichment services, and frequently leverage external, industry-accepted reference data sources such as Dun & Bradstreet for business information, or Acxiom for personal information.

NOTE

A note about mapping Data Hub service to the data zones. Using a service-oriented architecture approach allows Data Hub designers to abstract and isolate services from the actual location of the methods and functions that execute them regardless of which architecture zone these methods reside.

Data Synchronization

As data content changes, a sophisticated and efficient synchronization activity between the "master" and the "slaves" has to take place on a periodic or an ongoing basis depending on the business requirements. Where the Data Hub is the master, the synchronization flows have to originate from the Hub toward other systems. Complexity grows if an existing application or a data store acts as a master for certain attributes that are also stored in the Data Hub. In this case, every time one of these data attributes changes in the existing system, this change has to be delivered to the Data Hub for synchronization. One good synchronization design principle is to implement one or many unidirectional synchronization flows as opposed to a more complex bidirectional synchronization. In either approach, the synchronization process may require transactional conflict-resolution mechanisms, compensating transaction design, and other synchronization and reconciliation functionality.

A variety of reasons drive the complexity of data synchronization across multiple distributed systems. In the context of a CDI Data Hub, synchronization becomes difficult to manage when the entire data environment that includes Data Hub and the legacy systems is in a peer-to-peer relationship. This is not a CDI-specific issue; however, if it exists, it may defeat the entire purpose and benefits of building a CDI platform. In this case, there is no clear master role assigned to a Data Hub or other systems for some or all data attributes, and thus changes to some "shared" data attributes may occur simultaneously but on different systems and applications. Synchronizing these changes may involve complex business-rules-driven reconciliation logic. For example, consider a typical non-key attribute such as telephone number. Let's assume that this attribute resides in the legacy Customer Information File (CIF), a customer service center (CRM) system, and also in the Data Hub, where it is used for matching and linking of records. An example of a difficult scenario would be as follows:

- A customer changes his/her phone number and makes a record of this change via an online self-service channel that updates CIF. At the same time, the customer contacts a service center and tells a customer service representative (CSR) about the change. The CSR uses the CRM application to make the change in the customer profile and contact records but mistypes the number. As the result, the CIF and the CRM systems now contain different information, and both systems are sending their changes to each other and to the Data Hub for the required record update.
- If the Data Hub received two changes simultaneously, it will have to decide which information is correct or should take precedence before the changes are applied to the Hub record.
- If the changes arrive one after another over some time interval, the Data Hub needs to decide if the first change should override the second, or vice versa. This is not a simple "first-in first-serve" system since the changes can arrive into the Data Hub after the internal CIF and CRM processing is completed, and their timing does not have to coincide with the time when the change transaction was originally applied.

- Of course, you can extend this scenario by imagining a new application that accesses the Data Hub and can make changes directly to it. Then all systems participating in this change transaction are facing the challenge of receiving two change records and deciding which one to apply if any.

This situation is not just possible but also quite probable, especially when you consider that the Data Hub has to be integrated into an existing large enterprise data and application environment. Of course, should the organization implement a comprehensive data governance strategy and agree to recognize and respect data owners and data stewards, it will be in a position to decide on a single ownership for each data attribute under management. Unfortunately, not every organization is successful in implementing these data management structures. Therefore, we should consider defining conceptual Data Hub components that can perform data synchronization and reconciliation services in accordance with a set of business rules enforced by a business rules engine (BRE).

Overview of Business Rules Engines

Let's first define a business rules engine.

Business Rules Engine

A *business rules engine (BRE)* is a software application or a system that is designed to manage and enforce business rules based on a specified stimulus, for example, an event of attribute value changes. Business rules engines are usually architected as pluggable software components that separate the business rules from the application code. This separation helps reduce the time, effort, and costs of application maintenance by allowing the business users to modify the rules as necessary without the need for application changes.

In general, a BRE may help register, classify, and manage the business rules it is designed to enforce. In addition, a BRE can provide functionality that detects inconsistencies within individual business rules (for example, a rule that violates business logic), as well as rule sets.

Rule Set

A *rule set* is a collection of rules that apply to a particular event and must be evaluated together.

In the context of the CDI Data Hub, BRE software manages the rules that define how to reconcile the conflicts of bidirectional synchronization. For example, if a date-of-birth attribute is changed in the CRM system supporting the service center and in the self-service web channel, an organization may define a business rule that requires the changes to this attribute that came from the self-service channel to take precedence over any other changes. A more complex rule may dictate to accept changes to the date of birth only if the resulting age of the customer does not exceed the value of 65. There may be another business rule that would require a management approval in the case when the age value is greater than 65. The BRE would evaluate and enforce all rules that apply to a particular event.

At a minimum, a full-function BRE will include the following components:

- **Business Rule Repository** A database that stores the business rules defined by the business users
- **Business Rule Designer/Editor** An intuitive, easy-to-use, front-end application and a user interface that allows users to define, design, document, and edit business rules
- **A Query and Reporting Component** Allows users and rules administrators to query and report existing rules
- **Rules Engine Execution Core** Actual code that enforces the rules

There are several types of business rules engines available today that differ by at least the following two dimensions: by the way they enforce the rules and by the types of rules they support. The first dimension differentiates the engines that *interpret* business rules in a way similar to a script execution, from the engines that "compile" business rules into an internal executable form to drastically increase the performance of the engine. The second dimension is driven by the types of rules—inference rules and reaction rules:

- **Inference Engines** support complex rules that require an answer to be inferred based on conditions and parameters. For example, an Inference BRE would answer a question like "Should this customer be offered an increased credit line?"
- **Reaction Rules Engines** evaluate reaction rules automatically based on the context of the event. The engine would provide an automatic reaction in the form of real-time message, directive, feedback, or alert to a designated user. For example, if the customer age in the Data Hub was changed to qualify for mandatory retirement distribution, the reaction BRE would initiate the process of the retirement plan distribution by contacting an appropriate plan administrator.

Advanced BRE solutions support both types of business rules in either translator / interpreter or compilation mode. In addition, these engines support rules conflict detection and resolution, simulation of business rules execution for "what-if" scenarios, and policy-driven access controls and rule content security. Clearly, such an advanced BRE would be useful in supporting complex data synchronization and conflict reconciliation requirements of the Data Hub. Architecturally, however, a BRE may be implemented as a component of a Data Hub or as a general business rules engine that serves multiple lines of business and many applications. The former approach leads to a specialized BRE that is fine-tuned to effectively process reconciliation rules of a given style and context of the Data Hub. The latter is a general-purpose shared facility that may support a variety of business rules and applications, an approach that may require the BRE to support more complex rules-definition language syntax and grammar, and higher scalability and interoperability with the business applications. To isolate Data Hub design decisions from the specifics of the BRE implementation, we strongly recommend that companies take full advantage of the service-oriented approach to building a Data Hub environment and to encapsulating the BRE and its rules repository as a set of well-defined services that can be consumed by the Data Hub on an as-needed basis.

Data Delivery and Metadata Concerns

The complexities and issues of populating Data Hub give rise to a different set of concerns. These concerns have to be solved in order to enable data consumers (systems, applications, and users) to find and use the right data and attest to its quality and accuracy. The Information Consumer zone addresses these concerns by providing a set of services that help find the right data, package it into the right format, and make available the required information to the authorized consumers. While many of these concerns are typical for any data management and data delivery environment, it is important to discuss these concerns in the context of the CDI Data Hub and its data location service.

As we look at the overall enterprise data landscape, we can see the majority of data values spread across the Data Hub and many heterogeneous source systems. Each of these systems may act as a master of some data attributes, and in extreme cases, it is even possible that some data attributes have many masters. Every time a consumer requests a particular data record or a specific data attribute value, this request can be fulfilled correctly only when the requesting application "knows" what system it should query to get the requested data. This knowledge of the master relationship for each data attribute, as well as the knowledge of the name and location of the appropriate masters, is the responsibility of the *Attribute Location service*. Architecturally, it is an enterprise-wide data service that hides the implementation details from the applications and other service consumers. Conceptually, this service acts as a directory for all data attributes under management, and this directory is active, that is, the directory is continuously updated as data attributes migrate or get promoted from old masters to the Data Hub—a natural evolution of a CDI environment from a Registry style to the Transaction Hub. Logically, however, this service is a subset of a larger service framework that represents an enterprise-wide *metadata repository*—a key component of any enterprise-wide data strategy and data architecture. As we mentioned in Chapter 5, the Metadata Repository role is much broader than just providing support for the Attribute Locator service, and also includes such internal Data Hub services as Record Locator and even Key Generation services.

Although a detailed discussion of metadata is beyond the scope of this book, we briefly discuss the basic premises behind metadata and the metadata repository in the section that follows. This section describes how a metadata repository helps enable just-in-time data delivery capabilities of some Data Hub implementations as well as some end-user applications such as real-time or operational Business Intelligence applications.

Metadata Basics

In simple terms, *metadata* is "data about data," and if managed properly, it is generated whenever data is created, acquired, added to, deleted from, or updated in any data store and data system in scope of the enterprise data architecture.

Metadata provides a number of very important benefits to the enterprise, including:

- **Consistency of definitions** Metadata contains information about data that helps reconcile the difference in terminology such as "clients" and "customers," "revenue" and "sales," etc.
- **Clarity of relationships** Metadata helps resolve ambiguity and inconsistencies when determining the associations between entities stored throughout data environment. For example, if a customer declares a "beneficiary" in one application, and this beneficiary is called a "participant" in another application, metadata definitions would help clarify the situation.
- **Clarity of data lineage** Metadata contains information about the origins of a particular data set and can be granular enough to define information at the attribute level; metadata may maintain allowed values for a data attribute, its proper format, location, owner, and steward. Operationally, metadata may maintain auditable information about users, applications, and processes that create, delete, or change data, the exact timestamp of the change, and the authorization that was used to perform these actions.

There are three broad categories of metadata:

- **Business metadata** includes definitions of data files and attributes in business terms. It may also contain definitions of business rules that apply to these attributes, data owners and stewards, data quality metrics, and similar information that helps business users to navigate the "information ocean." Some reporting and business intelligence tools provide and maintain an internal repository of business-level metadata definitions used by these tools.
- **Technical metadata** is the most common form of metadata. This type of metadata is created and used by the tools and applications that create, manage, and use data. For example, some best-in-class ETL tools maintain internal metadata definitions used to create ETL directives or scripts. Technical metadata is a key metadata type used to build and maintain the enterprise data environment. Technical metadata typically includes database system names, table and column names and sizes, data types and allowed values, and structural information such as primary and foreign key attributes and indices. In the case of CDI architecture, technical metadata will contain subject areas defining attribute and record location reference information.
- **Operational metadata** contains information that is available in operational systems and run-time environments. It may contain data file size, date and time of last load, updates, and backups, names of the operational procedures and scripts that have to be used to create, update, restore, or otherwise access data, etc.

All these types of metadata have to be persistent and available in order to provide necessary and timely information to manage often heterogeneous and complex data environments such as those represented by various Data Hub architectures. A metadata management facility that enables collection, storage, maintenance, and dissemination of metadata information is called a metadata repository.

Topologically, metadata repository architecture defines one of the following three styles:

- Centralized Metadata repository
- Distributed Metadata repository
- Federated or Hybrid Metadata repository

The centralized architecture is the traditional approach to building a metadata repository. It offers efficient access to information, adaptability to additional data stores, scalability to capture additional metadata, and high performance. However, like any other centralized architecture, centralized metadata repository is a single point of failure. It requires continuous synchronization with the participants of the data environment, may become a performance bottleneck, and may negatively affect quality of metadata. Indeed, the need to copy information from various applications and data stores into the central repository may compromise data quality if the proper data validation procedures are not a part of the data acquisition process.

A distributed architecture avoids the concerns and potential errors of maintaining copies of the source metadata by accessing up-to-date metadata from all systems' metadata repositories in real time. Distributed metadata repositories offer superior metadata quality since the users see the most current information about the data. However, since distributed architecture requires real-time availability of all participating systems, a single system failure may potentially bring the metadata repository down. Also, as source systems configurations change, or as new systems become available, a distributed architecture needs to adapt rapidly to the new environment, and this degree of flexibility may require a temporary shutdown of the repository.

A federated or a hybrid approach leverages the strengths and mitigates the weaknesses of both distributed and centralized architectures. Like a distributed architecture, the federated approach can support real-time access of metadata from source systems. It can also centrally and reliably maintain metadata definitions or at least references to the proper locations of the accurate definitions in order to improve performance and availability.

Regardless of the architecture style of the metadata repository, any implementation should recognize and address the challenge of semantic integration. This is a well-known problem in metadata management that manifests itself in the system's inability to integrate information properly because some data attributes may have similar definitions but have completely different meanings. The reverse is also true. A trivial example is the task of constructing an integrated view of the office staff hierarchy for a company that was formed because of a merge of two entities. If you use job titles as a normalization factor, a "Vice President" in one company may be equal to a "Partner" in another. Not having these details explained clearly in the context becomes a difficult problem to solve systematically. The degree of difficulty grows with the diversity of the context. Among the many approaches to solving this challenge is the metadata repository design that links the context to the information itself and the rules by which this context should be interpreted.

Enterprise Information Integration and Integrated Data Views

Enterprise Information Integration (EII) is a set of technologies that leverage information collected and stored in the enterprise metadata repository to deliver accurate, complete, and correct data to all authorized consumers of such information without the need to create or use persistent data storage facilities.

The fundamental premise of EII is to enable authorized users to just-in-time and transparent access to all information they are entitled to. Part III of this book discusses the concepts of the "authorized user" and "entitlements."

Conceptually, EII technologies complement other solutions found in the Information Consumer zone by defining and delivering virtualized views of integrated data that can be distributed across several data stores including a Data Hub.

EII data views are based on the data requests and metadata definitions of the data under management. These views are independent from the technologies of the physical data stores used to construct these views.

Moreover, advanced EII solutions can support information delivery across a variety of channels including the ability to render the result set on any computing platform, including various mobile devices. Looking at EII from a CDI Data Hub architecture viewpoint, and applying service-oriented architecture principles, we can categorize EII technologies as components of the Information Consumer zone. The EII components that deliver requested data views to the consumers (users or applications) should be designed, implemented, and supported in conjunctions with the data location and delivery services depicted in Figure 6-2.

Although, strictly speaking, EII is not a mandatory part of the Data Hub architecture, it is easy to see that using EII services allows a Data Hub to deliver the value of an integrated information view to the consuming applications and users more quickly, at a lesser cost, and in a more flexible and dynamic fashion.

In other words, a key part of any CDI Data Hub design is the capability of delivering data to consuming applications periodically and on demand in agreed-upon formats. But being able to deliver data from the Data Hub is not the only requirement for the Information Consumer zone. Many organizations are embarking on the evolutionary road to a Data Hub design and implementation that makes the Data Hub a source for analytical and operational data management including support for the Business Intelligence and Servicing CRM systems. This approach expands the role of the Data Hub from the data integration target to the master data source that feeds value-added business applications. This expanded role of the Data Hub and the increased information value of data managed by the Data Hub require an organizational recognition of the importance of enterprise data strategy, broad data governance, clear and actionable data quality metrics with specially appointed data stewards that represent business units, and the existence and continuous support of an enterprise metadata repository.

The technical, business, and organizational concerns of data strategy, data governance, data management and data delivery that were discussed in this and the previous chapter are some of the key factors necessary to make any CDI initiative a useful, business-value-enhancing proposition.