"Go, Set, Ready, Repeat" : Lessons Learned for Successful Data Integration

David Loshin
Knowledge Integrity, Inc.
Business Intelligence Solutions

Table of Contents

Introduction

The number of enterprise projects demanding shared data is on the rise; yet why does their success continue to be forestalled by data integration complexity? In contrast to years of conventional wisdom and technology development driving distribution of business applications across divisional workgroup computing resources, there is a resurgence in centralizing organizational data to support enterprise applications and techniques such as enterprise resource planning (ERP), data warehousing for business intelligence, customer relationship management (CRM), and customer data integration (CDI) and master data management (MDM). But while these enterprise imperatives need to integrate data from decentralized/distributed data systems, their development is often stymied, in terms of decreased efficiency, increased costs, and critically, extenuated time to value from the start by the hurdles and complexity of data consolidation.

Over time, the degeneration in establishing standards and agreeing to common business data definitions has led to repeated failed attempts to integrate data for these types of enterprise objectives. All too often, with the objective of sweeping data from all the nooks and crannies into a single warehouse, the result of structural, semantic, and functional variations in the source data sets is inconsistency, inaccuracy, and incompleteness of the integrated data sets. Errors introduced by blindly extracting, transforming, and loading data into a target model lead to questionable results, reconciliations, and the need to begin the data integration process from the start. There is a clear impact in terms of increased effort, costs, and delays in developing trustworthy data assets supporting enterprise analytical and operational business systems.

This is akin to pulling the trigger prior to finding the target and setting the aim. In this paper our goal is to learn from these lessons and propose that an alternate approach to assess the model characteristics of both the target environments as well as the existing operational data stores to organize and rationalize a standardized, common model for information sharing that will simplify the data integration process. Employing tools and techniques such as data profiling, metadata management, and data modeling, the information analysts and developers can set their aims on the target prior to pulling the trigger to change the “go, set ready” paradigm to a “ready, set, go” initiative.

Organic Information Architecture vs. Data Centralization

Two technology trends over a long time period have significantly, albeit inadvertently, influenced the current state of an organization’s information architecture. Reduced costs for hardware components in the late 1970’s and early 1980’s led to smaller “work-group” computing systems able to support the business application needs for individual divisions or groups within an organization. A consequence of investments in these lower-cost work-group computing resources was the accompanying need for work-group information technology support, ranging from programmers, database analysts, and data modelers. In turn, decentralized business systems developed at the group level were engineered to address the transactional and operational needs of the group’s business users.

At the same time, structured programming and relational database systems emerged in tandem as the development paradigm for business applications and the data systems employed by these applications. Yet as data modelers were directly supporting divisional business processing needs, their objectives in model development focused on the structures and, correspondingly the *semantics* specific to the designed application. This organic approach to developing business applications suited the immediate needs of the business.

However, decentralized development and organic evolution of business application has created a situation where consistency in information models has been abandoned in lieu of expediency in application deployment. In most organizations, the application architecture has evolved in support of acute needs for transactional or operational applications combined with increased decentralization driven by the economics of workgroup computing. Because of this, there has been a natural tendency towards “data entropy,” especially as variant data models have emerged without any benefit of enterprise oversight.

The Needs for Data Integration

Despite the perceived value in the (short term) value proposition of decentralized application and data development, the organization's need for recentralizing data has never truly diminished, as reflected in concurrently-developed technologies (such as the three-tier architecture, object sharing via CORBA, and data warehouses in particular). From an operational standpoint, there are many opportunities for exploiting economies of scale, and the desire for actionable intelligence at the enterprise level requires the collection of data from multiple sources in preparation for analysis.

Whether the intent is an ERP system, business intelligence, or complex analysis and embedded analytics, distributed data systems will not satisfy the business need of the organization as a whole. More to the point, when data is distributed across the environment, enterprise applications create the need for the accessibility and availability of data sets sourced from multiple "islands of information" combined into a centralized framework. Any initiative intended to analyze overall customer behavior, product performance, employee productivity, procurement efficiency, etc. must have a broad view of all the information that is relevant to the analysis. This means that customer data from any customer-facing application must be accumulated together in order to have a full view of customer interactions. The same is true for product data, employee data, spend data, vendor data, or any other subject of review – in no uncertain terms, there is a need for integrating data from multiple sources and making that information available for secondary purposes.

"Go, Set, Ready" and Data Challenges

Most enterprise applications are complex – their intent crosses lines of business, their value proposition is based on a community of use, and their implementations rely on cooperation of application owners, stewards, and developers from across the organization. Often, the absence of planning, organization, and clear success criteria and associated metrics complicates the matter.

Yet when implementing an enterprise project, clear steps must be taken to demonstrate that progress is being made. From the Information Technology perspective, this need for demonstrating progress is manifested in terms of concrete tasks, including presumption of need, software procurement, and product implementation, followed by an uncontrolled collection of data into the centralized environment. Without a well-defined roadmap to direct the effort, it is not always clear that this will achieve the expected goal. While a set of tasks are completed, the overall result may not meet the needs of the business.

Unfortunately, years of dependence on the organically-developed business applications has created a knowledge gap when it comes to data integration. Business application/data developers are relatively adept at building systems meeting specific vertical business needs, but are less comfortable considering the collected needs of identified (as well as *potential future*) downstream data consumers. On top of that, repurposing legacy data sets for new uses forces reinterpretation of what the data sets truly represent. Yet that is the critical point of the enterprise application: accumulating data from various sources into a target repository that will be used to satisfy a variety of end-user expectations.

Meeting the objective of creating this target repository through ungoverned data set extraction followed by arbitrary transformations is tantamount to putting the horse before the cart. And once the data issues emerge and lead to continual reconciliations, flawed decision-making, and general mistrust, the data integration cycle starts again. No wonder that this "Go, Set, Ready" approach accounts for the estimates that 70-80%, or even as much as 90% of the effort involved in data mining, data analysis, and data warehousing involves data cleaning, preparation, and integration!

Technology acquisition and tool implementation are the easy parts; getting the data to meet the needs of the consumers of the centralized data is much more challenging. It requires recognizing the existence of common data issues occurring as a result of distributed development and executing a plan to finesse those issues to maintain data consistency. Some examples of these common data issues include:

1. Structural and Semantic inconsistency: Differences in data types, data element lengths, value formats, structures, and semantics presumed by downstream data consumers that lead to diverse conclusions even when drawn from similar analyses;
2. Inconsistent validations: Data validation rules are inconsistently applied at various points in the business processes, with a variety of impacts downstream;
3. Replicated functionality: Repeatedly applying the same (or similar) data parsing, standardization, and/or cleansing and identity resolution applications to the same data items multiple times increases costs but does not ensure consistency;
4. Data entropy: Continuously making copies of the same data leads to more data silos where the quality of the data continues to degrade, especially when levels of service for consistency and synchronization are not met or are not even defined.

These issues manifest themselves when there is no awareness regarding data set existence, usage, meanings, and ultimately, where the semantic differences come from. In most organizations, where there is a lack of knowledge about common data concepts and how these data concepts are employed across multiple business processes, the data integration process requires numerous iterations to get to a point where the data can be trusted. Any time one downstream user interprets the data in a different way than another downstream user, there is bound to be confusion, followed by an acute need for reconciliation, in which a team of analysts must trace back the lineage associated with the resulting report values back through the information production flow. This is a tedious and challenging effort that stalls productivity until the source of the semantic variance can be identified, resolved, at which point the entire process of analysis has to be restarted.

In turn, the data management practitioners must reevaluate the tools and methods used for data integration, without addressing the root cause: the absence of a rationalized view of those data concepts that are ubiquitous across the organization. Increased demands for data sharing and repurposing means that most enterprises today will require an enumeration of common data concepts and knowledge of how these data concepts are employed across multiple business processes in order to rationalize semantics as a prelude to consistency.

“Ready, Set, Go”

Using the “Go, Set, Ready, Repeat” approach, the teams extract the data, manipulate it for some specific purpose, load it into a data warehouse or other analytical environment, perform the analysis or generate the reports, see the inconsistencies, perform the reconciliations, determine that the data was not extracted or transformed or loaded correctly, and then it is back to the drawing board. If this process leads to confusion, inconsistency, and scrap and rework, then the alternative is to best determine what the downstream needs for data (as well as its consistency and quality) are, and then proactively develop the data integration processes to anticipate those needs.

In other words, understanding organizational data requirements, especially in relation to commonly used data concepts, properly prepares the organization for efficient and effective data integration. This requires that key stakeholders in the organization bite the bullet by rationalizing the enterprise understanding of the different ways that common data concepts are defined, used, and in some circumstances, differentiated. Once there is clarity on what data concepts are represented in different applications along with their underlying structures and semantics, the data management practitioners can work with the data consumers to make informed decisions regarding those core data concepts that are relevant to both the transactional/operational applications and the necessary reporting and analytics performed downstream. In turn this allows those practitioners to determine the best approach for creating a standardized model that can accommodate both a shared representation and a shared understanding.

Examples

These (and similar) issues emerge as the byproduct of the absence of historical standards for, and lack of oversight for the ways that different stakeholders model their core data concepts. Each of the following real-world examples demonstrates how model and data rule inconsistencies can hinder data integration.

Structural Inconsistency

Structural inconsistencies appear at the data element level occur when the same conceptual data element is represented using variant data element lengths and data types, such as a conceptual data element for a postal code implemented in a variety of ways to accommodate different structural variations:

- A version for a 5-digit US ZIP codes that uses a length 5 character string;
- A version to store a US ZIP+4, allocating a length 9 character string;
- Another version for a US ZIP+4, instead allocating a length 10 character string that holds a hyphen to separate the first 5 digits from the last 4 digits;
- A data element that allows US, Canadian, and UK postal codes, using a length-6 character string; and
- A version that uses a length 10 character string with no format constraints.

Inconsistent Validation

There is a risk of applying inconsistent validations occurs when the definitions for two versions of what should be the same data element contain conflicting information. As an example, look at the difference between these two definitions for "wage amount," adapted from data element layouts provided via conflicting US government documents:

1. Length 11 alphanumeric value that contains the information as provided from the *Quarterly Wage* record submitted to the Employee Directory.
2. Length 11 numeric value containing the amount of a person's wages during a Reporting Quarter. The last two positions are implied to be to the right of the decimal point.

While both definitions describe a length-11 field, the difference lies not only in the data types but also in the precision. In the first definition there is no indication that there is any implicit decimal place, while the second explicitly states that there is. In essence, there is a precision difference of two orders of magnitude, yet one rule validating that a proposed value fits the defined range would have inconsistent results depending on which definition is used.

Replicated Functionality

Many organizations rely on location as critical input to business processes. In the property insurance industry, location is used for numerous business processes such as marketing and sales, assessing risk, underwriting, and catastrophe management. It is not uncommon that locations are represented in different ways in different applications. Frequently these organizations are using a variety of tools and procedures for address cleaning and geocoding, and this is an example of replicated functionality, where different software tools supported by different teams, using different rules, all intended to achieve the same goal, is not only inefficient, it also creates new inconsistencies in the enhanced data sets and increased costs due to replicated work.

Semantic Inconsistency

As each business system has essentially grown in a vacuum, it is likely that there are multiple business definitions for business concepts like "customer," "product," etc. or attributes like "postal code," "wage amount," etc. across the

application landscape. Even worse, sometimes is the situation in which everyone believes that there is agreement to a common definition, yet no clear definition has ever been documented!

In the absence of common agreement on the meanings of common data concepts and their associated attributes, it is not surprising to find inconsistencies in consolidated data sets. Whether there are precision issues (such as an undocumented assumption of decimal placement for numeric currency attributes), data domain issues (such as reliance on different reference data sets for common concepts such as "Country code" or "State"), or definition issues that map to slightly different conceptual data sets, forcing the data consumer to reinterpret the meaning of an attribute will often lead to trouble.

Data Entropy

There are two aspects to the tendency for data to go from a high state of organization to a greater degree of disorganization. The first reflects the rigor applied in assuring that data values observe the data element specifications, and how that degrades in relation to changes in the real world that are not mirrored in the underlying model. An example is attribute overloading, in which one data element is used for multiple purposes, such as storing email addresses in the FAX number field, or inserting asterisks as indicators at the end of a customer's name.

The second is a byproduct of the increasing data needs over time that had not been identified when the system was originally designed. An example is a roadside assistance club membership database that did not accommodate a *birth date*, which is an issue years later when that roadside assistance club is integrated into an automobile insurance company, in which driver age contributes to underwriting and risk determination.

Tools and Techniques for Information Rationalization

In each of these examples, the variance in the models underlying the data to be centralized becomes the roadblock to data integration success. Forearmed with the knowledge of the common data integration issues, the informed data integration analysts can reduce the risks of replicated functionality and rework by assessing the existing variances and creating standards by developing a rationalized set of "canonical data models" intended to accommodate data instances representing the same concepts in a way that establishes both structural and semantic consistency during the data integration process.

The result of the upfront effort invested in organizing the rationalized canonical models is greater efficiency and predictability in engineering the data integration procedures. The design of any canonical model requires that these tasks be performed:

1. **Analyze** – Analyze the data to understand the "ground truth" and collect data element types, lengths, structures, and definitions for both the target and the candidate sources;
2. **Synthesize** – Harmonize the data element definitions, consider the similarities and differences, and then synthesize a shared view of common data element concepts;
3. **Map** – Ensure that there is a way to map source data elements into the common view as well as map the common data elements to the target uses;
4. **Publish** – Provide a means for capturing and then publishing the shared metadata; and
5. **Model** – Employ the common data elements to develop models that everyone can use while standardizing communications and data sharing, so that those models can be kept up to date.

To facilitate the development of the common canonical models, these tasks are supported by an array of data management technologies, particularly data profiling, metadata management, and data modeling tools. Each of these technologies provides a continuous contribution to aspects of the analysis of existing enterprise data assets prior to data integration to prepare for, and bypass potential consolidation pitfalls. This is not to imply that these techniques must be

applied in a specific sequence; rather, their value is magnified when the process employs all three in an integrated, iterative approach.

Data Profiling

Data profiling tools provide a set of algorithms for statistical analysis and evaluation of the data values within a data set, as well as exploring relationships that exist between data elements within the same table or across data elements in different data sets. A data profiler will scan the values in a column and provide a frequency distribution of that column's values as well as that column's value patterns. The tool will also capture the maximum length of the values, and makes inferences so as to suggest potential data types. Some of the more sophisticated data profiling tools are able to expose embedded value dependencies across columns within a table, while inter-table analysis explores overlapping values sets to identify foreign key relationships between entities.

Cross-table analysis is also employed in identifying mappings between value domains that refer to the same conceptual domains. As an example, the techniques can help determine that *male* and *female* are respectively represented using 'M' and 'F' in one table, and using '0' and '1' in another table. The analysis maps corresponding value sets and helps in harmonizing variations in data types used for the same data element concepts instantiated across different business applications. Cross-table analysis will provide details about relationship cardinality across data entities, which can be used in scrutinizing model consistency.

Metadata Repository

A metadata repository provides a central storage point for common business term definitions, data element definitions, data element data types, table layouts, reference data domains, as well as a directory of applications that use the assorted data element artifacts. While many groups may maintain independent data dictionaries that provide some of the services a metadata repository provides, the value of a metadata repository grows when it is used as more than a glorified data dictionary. Collecting data element metadata and organizing the conceptual data element definitions allows the analyst to review whether the definitions are identical, whether they are compatible, and determine when the definitions are dissimilar, indicating a semantic inconsistency.

After harmonizing data element definitions, the data analysts can consider the physical representations (as manifested by identified data types and lengths) to synthesize a common data element representation. Linking the existing data element instances from across a number of business applications will inform the source-to-target mapping process, which transforms source data elements into the common data element representation in a way that retains semantic consistency. The metadata repository then acts as the shared forum in which the definitions, data types, and corresponding transformations are published.

Data Modeling

The nexus of the information rationalization is the data model, and this may be the most important aspect of the process. The data modeling process is both the main framework for documenting the shared vision, as well as the means for its communication. The data model can be used to help reach agreement on shared representations through a number of avenues, including providing a means for capturing an inventory of the current state, allowing analysts to reverse engineer the existing representations, employing comparative features to discern differences across what are presumed to be common models, and a variety of other techniques.

Once there is agreement on shared representations for frequently-used data elements, the data modelers can examine the similarities and differences between the entity layouts and corresponding relational structure. A logical model for each relevant data subject area can be developed using data modeling tools that incorporates the common data elements, and these logical models also drive the definition of the core business definitions for shared data as well as their associated business rules.

This model becomes the core of the data integration processes for a number of reasons. First, it simplifies the transformations necessary to bring data into the downstream business processes (such as a data warehouse or an ERP system). Second, because the initial analysis will have identified those source data sets whose structures and semantics are aligned with the target business applications, the developers can be proactive in both selecting data sources for consolidation and developing data integration procedures that preserve the structural and semantic consistency across the information flow.

A good data modeling environment allows the modelers to synchronize with the metadata management tools, provides details about the data element concepts and how they are incorporated within logical and physical models, and may even provide traceability mapping data elements to the business application in which they are referenced.

Benefits of Model-Driven Data Integration

Taking a proactive, mode-driven approach yields a number of benefits beyond the simplification of the data integration process:

- Knowledge Discovery – Instituting metadata analysis exposes corporate lore embedded within legacy data sets over years of distributed application development.
- Rationalization – the process of organizing and rationalizing data element definitions, structures, and semantics not only aligns meaning across an enterprise architecture, it engages the subject matter experts to explore the reasons for variation and consider resolutions.
- Knowledge Sharing – Metadata repositories and the associated practices for metadata management make discovered knowledge available to the entire organization.
- Improved Data Quality – Reducing the potential for introducing data flaws as a byproduct of data extraction and transformation errors means that the resulting consolidated data set will be more trustworthy for downstream application purposes.
- Improved Efficiency and Decreased Development/Maintenance Costs – By improving the standardization of structure and semantics for data subject to reuse, one can reduce the complexity of implementing enterprise applications.

Establishing a robust set of good data management techniques relies on the right mixture of people, process, and technology. Data analysts and developers can achieve these benefits through governed procedures for metadata analysis and synthesis, data modeling, and data integration using data profiling, metadata management, and modeling tools and technologies. By applying the best practices associated with data model development, one can eliminate the repeated misfires of the “Go-Set-Ready-Repeat” paradigm, enable the right level of preparedness, and be ready to achieve data integration success the first time around.