Data Migration – Understanding the Complexities

February 19, 2015

Data Migration – Understanding the Complexities

Data Migration

Data migration is the process of preparing and moving data from a source to a target data store. The data store can be a database, file, COTS product, spreadsheet, etc. Data migration is comprised of six steps: data mapping, data collection, data profiling and analysis, data assessment, data cleansing, and loading the target data store. The five steps involved in preparing the data for migration are cyclical (mapping, collection, profiling and analysis, assessment, cleansing).

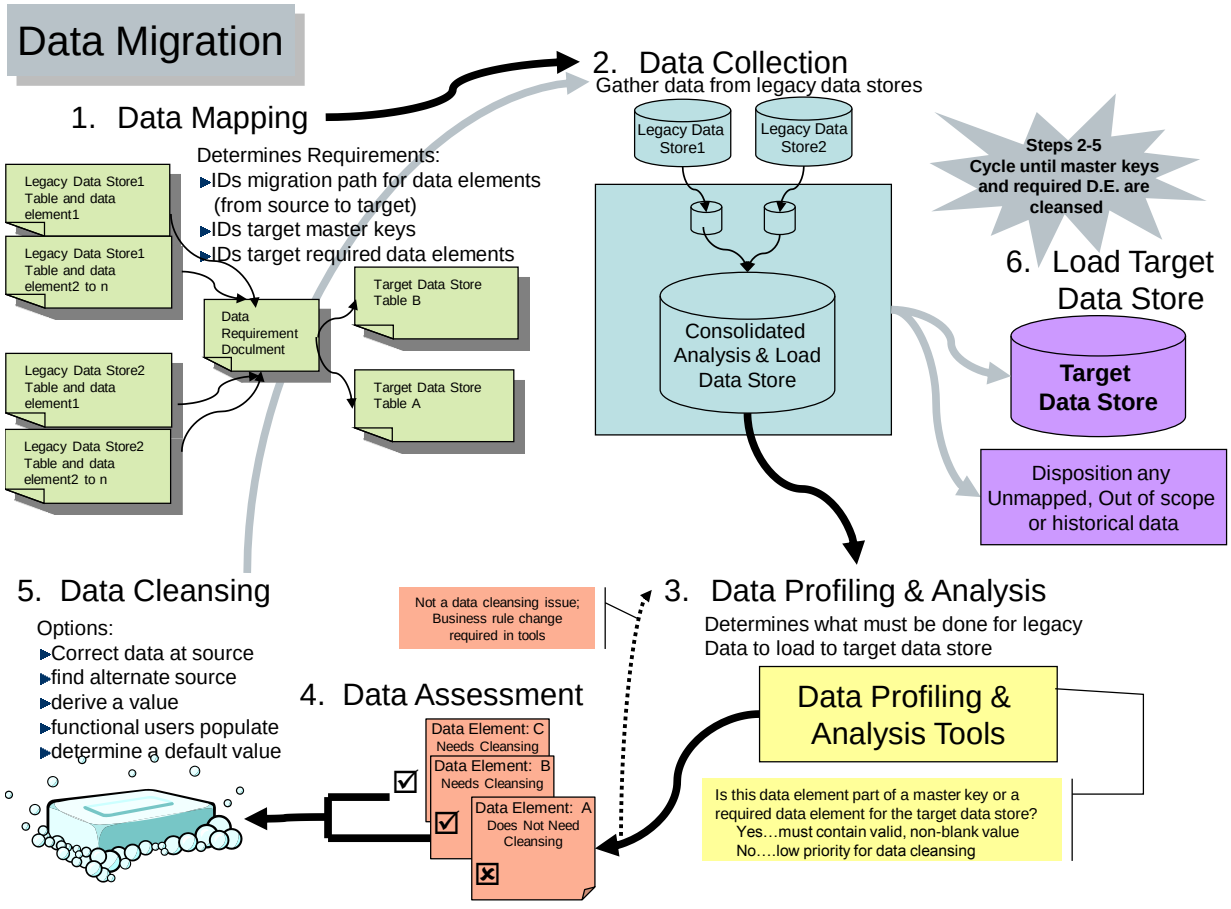

Figure 1- Data Migration Process

Step 1. Data Mapping

Data mapping is the processing of identifying to which target data element a source data element will be moved. This can be simple and straight forward if the two data sources have similar data standards and constructs. But often this is very complex, requiring a concerted effort by individuals on both the source and target data store sides of the effort to collaborate and make the determination. The discussion is not just based on name and definition, but on context (how the data element is used in the source environment and how that same functionality is performed in the target environment).

To illustrate a few of the potential complexities, the example below is for the custom-built legacy MyCompany (MyC) database which is being migrated to the Office for Everyone (O4E) COTS product. The first table to be reviewed for migration is the legacy Employee_Info_Table. Due to the structure of the new database, the data in this single MyC table will be migrated to multiple tables in the O4E database.

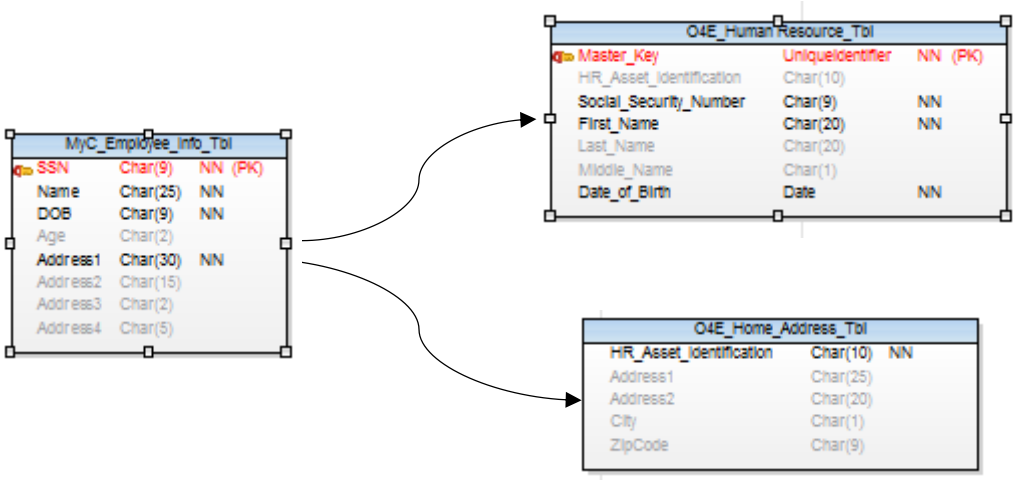

Figure 2 - Data Mapping Example

Identify the Migration Path (Disposition the Data)

To complete the data mapping, each data element from the source data store must be dispositioned. Dispositioning data elements means determining if a data element 1) will be migrated, 2) to which table and data element name it will be migrated, and 3) how it will be migrated. This can appear straight forward and sometimes it will be, but we must look closer at the description and the context (how the data element is used), to determine the accurate disposition and ensure meaningful data in the source data store will continue to be meaningful in the target data store. Completing the full disposition of a data element not only accounts for which target will receive the data, it also sets the business rules for movement of the data. The table below illustrates how employee data elements may be migrated from the MyC database to the O4E database. The example is simplistic and uses easily understandable information that for the most part can be taken at face value.

Source Table	Source Data Element	Target Table	Target Data Element	Disposition / Data Migration Business Rule
N/A (data element does not exist in the MyC DB)	N/A	Human_Resource_Tbl	HR_Asset_ Identification (master key)	Generate this identification number during initial database load
Employee_Info_Tbl SSN is stored in an 11 characters format: xxx-xx-xxxx; SSN is the master key	SSN	Human_Resource_Tbl SSN is stored as 9 characters, numerals only; SSN is not a key data element	Social_ Security_ Number	Migrate by removing dashes between groups of numbers
Employee_Info_Tbl Name is stored in 25 characters, first name space last name. Middle name is not stored	Name	Human_Resource_Tbl Name is stored in three distinct data elements	First_Name Last_Name Middle_ Name	Migrate by Parsing Name 1. put all characters before the space into First Name 2. put all characters following the space into Last Name 3. leave Middle_Name blank
Employee_Info_Tbl DOB is stored in a text field as ddMMMyyyy	DOB	Human_Resource_Tbl Date_of_Birth is stored as a date field in yyyymmdd format	Date_of_ Birth	Migrate by converting text formatted date into date field format
Employee_Info_Tbl Age is calculated in a batch job that runs on the first day of the month	Age	Human_Resource_Tbl Age is not stored. It will be derived	N/A	Do not migrate. This data element will be derived by subtracting DOB from today's date at the time of query
Employee_Info_Tbl Address1 contains the house number and street name	Address1	Home_Address_tbl Address1 contains the house number and street name	Address1	Migrate by copying
Employee_Info_Tbl Does not exist in the legacy database	N/A	Home_Address_tbl Address2 contains Apartment or Suite number	Address2	Leave Address2 blank
Employee_Info_Tbl Address2 contains the city name	Address2	Home_Address_tbl City contains the city name	City	Migrate by copying
Employee_Info_Tbl Address 3 contains the state name	Address3	Home_Address_tbl State is not stored, it is looked up in a USPS table based on Zip_Code	N/A	Do not migrate; this data element will be looked up using a USPS table at time of query
Employee_Info_Tbl Address4 contains the five position zip code	Address4	Home_Address_tbl Zip_Code contains the 9 position zip code, when available	Zip_Code	Migrate by copying

Figure 3 - Data Mapping and Disposition Example

Identify Translation of Encoded Data

Another complicating factor is encoded data. Codes and values used in a source data store may not easily translate into the target data store. An example of this might be storing the language a person speaks. Below is an abbreviated example of a potential problem in which the legacy data sources contains codes that do not exist in the target data store. So, the migration of data, in the case of codes "AD" and "AE" will lose the granularity of dialect that is available in the source data store.

Legacy Language Code	Legacy Language Value	Target Language Code	Target Language Value
AA	AFRIKAANS	AFR	Afrikaans
AB	ALBANIAN	ALB	Albanian
AC	AMHARIC	AMH	Amharic
AD	ARABIC-MODERN STANDARD	ARA	ARABIC
AE	ARABIC-EGYPTIAN	ARA	ARABIC

Figure 4 - Code and Value Mismatch Example

Identify Master Keys

During data mapping it is important to identify the target master keys and any other key data elements within the target data store. These data elements will require special handling to ensure the correct source data element or elements are mapped to the target keys. Step 4, Data Assessment provides more information on handling key data elements.

Identify Required Data Elements

Required data elements are not keys, but must contain a value according to the rules established for the target data store (not null). The source data store was built with data rules and constraints and the target data store will have data rules and constraints as well. When the rules and constraints between the two are not the same, a methodology must be used to resolve conflicts. An example is when a source data element contains a valid blank and the target data element requires a value. Step 4, Data Assessment provides more information on handling required data elements.

All data elements from the source data store must have a documented disposition, even if the disposition is that the data will not be migrated. Documenting the results of all data dispositions will allow easy answers, when the inevitable questions are asked, "has all of the data is been migrated", or, "what happened to data element x"?

Step 2. Data Collection

Data Collection provides specified source data on a periodic basis to the target development team to use in preparation for the initial database load (IDBL) of the target data store. Data is provided periodically, so the target development team can develop and test data load scripts. The data will also be used to accomplish data profiling with the goal of improving the quality of the source data and developing business rules for the IDBL to accommodate special situations when data cannot be easily migrated

directly from the source data store. Depending on the number of source data stores and the operational requirements for those data stores, data collection may involve extensive coordination.

Step 3. Data Profiling and Analysis

Data profiling determines the suitability of individual data elements in the source data store to be loaded in the target data store. Data profiling is the next step in the cyclical process of preparing data to be migrated from source data stores to the target data store. Data profiling and analysis normally utilizes automated tools to identify source data elements that *appear* to be invalid for loading in the target data store. Part of the process is done by comparing the content of a source data element against the valid values in the source data store's data dictionary. The tools to execute these processes can be referred to as "filters".

During the initial stages of data profiling and analysis be prepared to hear that the source data store has "bad data". It indeed may contain invalid data, but not all source data that appears invalid when measured against the target data store rules and constraints really is invalid.

Step 4. Data Assessment

Data assessment involves reviewing the results of the profiling step to determine if data elements which *appear* to be invalid, actually are invalid (and require data cleansing) or if adjustments need to be made to the profiling filters and the data migration business rules. Below is a table showing potential causes for the appearance of invalid data and suggested action.

Cause for Appearance of Invalid Source Data	Suggested Action
The value in the source data element is valid, but is not loaded in the source data store's data dictionary causing a false error	Add the value to the source data dictionary and rerun the profile report
The source data element contains a valid blank/null value AND the data element IS NOT mapped to a target data store key or required data element	Adjust the profiling filters to show that blank is a valid value for the data element and rerun the profiling report. Any blanks in the source data element will be loaded to the target data store as blanks at IDBL.
The source data element contains invalid data and IS NOT a key or required data element	Determine the best approach to identify a value for the data element: • Find an alternate source that contains accurate data • Use other data elements to derive a value • Research and manually enter the correct data • Determine a default value • Blank-fill the data element If possible, update the source data store using the identified method*

Cause for Appearance of Invalid Source Data	Suggested Action
The source data element contains a valid blank/null value or contains invalid data AND the data element IS mapped to a target data store key or required data element	Key and required data elements must contain a value for the target system to function. Determine the best approach to identify a value for the data element: • Find an alternate source that contains accurate data • Use other data elements to derive a value • Research and manually enter the correct data • Determine a default value • Blank values are not an option If possible, update the source data store using the identified method*

Figure 5 - Cause and Action for Invalid Data

**Ideally the actions prescribed will be performed on the source data store to ensure the best traceability of data from source to target. This is not always practical or desired since it may cause operational problems for legacy applications, cause confusion for current customers, funds may not be available for the changes, or the source data store may not be within the target project's sphere of influence. In this case, the actions prescribed will need to be built into business rules for the IDBL and the data mapping documents updated to reflect these business rules.*

Step 5. Data Cleansing

Data cleansing involves correcting values in the source data store which have invalid entries. If possible, without causing operational problems or customer confusion, data elements which are mapped to target key and required data elements can be updated according to identified business rules. Usually, key and required data elements are the priority for cleansing, since these data elements must be "correct" for a record to be established in the target data store. As noted above, if the data changes cannot be made in the source data store, they must be built into the data migration business rules and be applied during IDBL.

Steps 1-5 form a continuous cycle. The steps are repeated until all source data has been dispositioned and all data identified to be migrated to the target data store has been successfully loaded and verified during multiple test load runs.

Step 6. Loading the Target Data Store

The culmination of the work done in the previous steps is the IDBL of the target data store during the cut-over period when the source data store is retired and the target data store becomes operational in production.

What About the Left Overs?

When source data is dispositioned, there may be some data elements and records that are not migrated to the target data store. The records may be historical and not included in the new data store. The data element may be derived in the new data store, so it is not migrated. The data may be out-of-scope for the purpose of the new data store. The data element may be obsolete and the purpose it served is no longer relevant. Each data element or group of records that is not migrated should be reviewed to determine if there is a need to maintain the data or set of records. Does law require the historical records be maintained for a specified set of time? Does another customer need data that is not being migrated? Once the non-migrating data has been analyzed, decisions must be made: 1) whether to maintain the data, 2) how/where to maintain it and 3) for how long.