Demand-Based Data Migration to Enterprise Systems: The Case of United Nations Economic Commission for Africa

Mesfin Tolossa
HiLCoE, Software Engineering Programme, Ethiopia
United Nations Economic Commission for Africa (UNECA), Addis Ababa, Ethiopia
mesfintol@gmail.com

Berhanu Borena
HiLCoE, Ethiopia
berhanuborena@gmail.com

Abstract

Data migration is usually part of a larger project deliverable, and typically the majority of business attention is focused on the package selection and configuration rather than on ensuring that the data that populates the new system is fit for the purpose. Nevertheless, there is a gap in the literature regarding data migration under the influence of the specific circumstances and the majority of publications address data migration from a perspective of a service provider that builds a onetime solution for a data owner. When the data owner re-engineers a data intensive product, a data migration solution is needed. However, the situation is different than that of classical data migration because the owner has in-house knowledge of the source, is itself building the target and must provide replicable deployment to all data. Thus, the owners, such as, in this research case, the United Nations Economic Commission for Africa (UNECA), are left to tailor data migration to a specific framework without any guidance.

The purpose with a data migration portal is to gather all data migration to one common area, Umoja, without a third-party migration tool and minimize the technical complexity associated with data migration projects. We have developed two modules for the portal. The first module concerns migration from an Excel document to Umoja. The second module handles upload of records directly from Integrated Management Information System (IMIS) to the new Enterprise Resource Planning (ERP) system called Umoja. The portal has functionality like data mapping, validation and loading data to the target. A data migration portal which can visually monitor, filter, transform and import various types of data would facilitate the migration process.

Keywords: Integrated Management Information System; Umoja; ERP; Data Migration; Schema Translation; Data Migration Strategy; Data Migration Framework

1. Introduction

The purpose of the Integrated Management Information System (IMIS), which was approved by the General Assembly of the United Nations on 21 December 1988, is to develop an integrated system for the processing and reporting of administrative actions at all major duty stations. The main areas covered by the system are personnel management, post management, accounting, procurement, travel, and payroll and budget execution. IMIS is installed at the eight major duty stations, namely, New York, Addis Ababa, Bangkok, Beirut, Geneva, Nairobi, Santiago and Vienna. It is also being used by the United Nations Development Programme (UNDP), the United Nations Population Fund (UNFPA), and the United Nations Office for Project Services (UNOPS).

Currently, IMIS remains the only fully integrated system available to the United Nations system of organizations, meeting their specific requirements in most of their administrative needs in personnel management, budget implementation, treasury, accounting, payroll, procurement and travel until the upcoming new system, the Umoja solution, with its core implementation of an Enterprise Resource Planning (ERP) solution, as well as a thorough

streamlining of UN administrative processes, which is proceeding on schedule.

Umoja, a Swahili word which means “unity”, will provide a harmonized, simplified, and real-time approach to the Organization’s management of finances, resources and assets. In particular, it becomes a decision support system, providing dashboards for all sorts of performance indicators allowing management quick synopsis of any given situation, and allowing quicker decision making [1]. During Umjoa implementation, data migration is one technique which involves planning, scoping the project, extracting data from the source application, cleansing to repair corrupt data or invalid records, removing duplicates, transforming the source data to conform to data requirements, translating source values to new data based on translation tables, loading the data, and verifying the data for accuracy.

This paper analyzes the data migration processes while Umoja is implemented at UNECA to ensure the ongoing viability, robustness of the system and it continues to meet the requirements of the Organization. The paper also aims to develop a framework that UNECA can use in order to manage data migration for Umoja.

2. Related Work

Drumm *et al.* [2] offer a first definition for data migration. Data migration is defined as the process of transforming and integrating data from one or more source legacy data stores to a new target data store. During the data migration process, the data needs to be extracted from the source, and transformed and loaded into the target.

Haller [3] also provides a definition for data migration. The main reason of the author for defining data migration is in order to separate it from the concept of database migration which encompasses migrating database specific objects such as triggers or procedures together with the data. Thus the author states that data migration refers only to migrating data out of one schema to a new schema. The new schema can be structured completely differently from the existing one.

Matthes and Schulz [4] compile a definition of data migration based on the state of the art in academic and practitioner’s literature and their own research. The definition they offer states that data migration is a “tool supported one-time process which aims at migrating formatted data from a source structure to a target data structure whereas both structures differ on a conceptual and/or physical level”. This definition is preferred by the authors because it touches on all the relevant concepts of data migration. Because data migration is a process, it encompasses a control and data flow. The aim of the process is to migrate formatted data from a source data structure to a target structure.

Hainaut *et al.* [5] present several migration strategies by combining techniques in the database layer with techniques in the program layer. Two possible strategies are identified on the database layer. The first strategy, Physical schema conversion, implies only that the schema of the legacy system is simulated, or mirrored, in the target database system. Thus, no changes are made on the concept level. The second strategy is concerned with re-conceptualizing the legacy data model. The conceptual schema conversion employs schema refinement and data structure conceptualization in order to define additional structures and constraints within the model. Mapping is used in both strategies in order to define the transformations that are required to migrate the data. Two types of mapping can be defined: structural mapping, which is concerned on schema modifications, and instance mapping, which describes how to instantiate a source object in a target data model.

The authors present an ETL approach to the data conversion dimension. This approach consists of extracting the data from the source. Employing a converter that uses the mappings for transforming legacy data to the target structure and finally, loading the data to the target system. The required steps for data conversion are, in order, implementing the target

schema, defining the mappings between source and target and implementing the mappings in a converter in order to translate the legacy data to the target structure.

Thalheim and Wang [6] define three possible implementation strategies for data migration as described below.

One of the earliest migration approaches is the Chicken Little Methodology. It is a gateway-based eleven-step approach which allows both the legacy and target system operate in parallel during migration. The approach is also incremental as the target system though small at the onset, continues to grow as the migration progresses until it replaces the legacy system [7]. Two major problems of this approach which had provided a research challenge and which we intend to solve are: 1) the complexity of the gateway modules and 2) the lack of prioritization of the data to be migrated.

The Butterfly Methodology challenged the iterative and approach of the Chicken Little. Also questioned was the parallel run of both legacy and target system in the previous approach. This necessitated the definition of a five-phase approach which involves the following steps:

Step 1: Determination of the semantics of the candidate legacy system and development of the target schema;

Step 2: Construction of a sample data store in the target system based upon target sample data;

Step 3: Migration of all the information system components while leaving out the data;

Step 4: Migration of the legacy data to the target system and the training of users' and

Step 5: Decommissioning of the legacy system and switchover to the new system.

Each of these steps is further broken down into sub-steps and specific activities [7].

Another early model is the "Big-Bang" or "Cold Turkey" approach [4]. This is basically a Forward and Reverse Migration process where the legacy system must be shut down for a considerable time to facilitate data migration before the target system is made available. One problem associated with this approach is that the proposed framework is presented at too high a theoretical level to be useful in practice and no consideration is given practically to the actual migration of the data.

The primary concern of a research was the minimization of the number of adapters to be used during the integration of multiple legacy applications with several new target applications. The research introduced an intermediate data format in place of software programs that provide different translational layers between data formats.

The technical details of the migration strategy of image data from a series of legacy applications were not made available. This is primarily due to the fact that the migration process was proprietary and the target systems were commercial applications. However, the documentations available emphasized the importance of the planning process in order to minimize the financial impact of a large data migration. In contrast, the documentation of the technical process involved in our migration strategy will be made available in the public library in order to provide guidance for enterprises who may wish to migrate their legacy data to new infrastructures. The authors in addition, provided a sketch of the overall project plan, including interim solutions that were needed to address various stages during their migration but from a technical perspective this paper offers little assistance, especially for IT personnel considering a large data migration dealing with medical image files or with large numbers of image files.

Thalheim and Wang [6] applied a data re-engineering strategy to the Chicken Little approach to create an incremental model for migrating legacy information systems. The data migration segment involves a comprehension step where the data to be migrated are first understood in order to separate relevant data from the redundant ones; and their structures re-engineered and migrated to the target platform.

All the approaches above are limited to simply migrating legacy data to target systems. Most involve bringing business operations to a halt while the migration is ongoing and very little consideration is given to prioritization of critical sensitive data in the migration process.

3. The Proposed Framework

We re-defined the ETL approach in [5] using Content Analysis and adopted it in this paper. The methodology is a stepwise adaptation of the Structured Systems Analysis methods hybrid with Object Oriented Design for data migration tool.

We have defined a new framework in this work. Using content analysis, this framework was adopted to develop a strategy for migrating legacy data to enterprise databases. The structural view of the framework is depicted in Figure 1. The structure shows the interaction and interdependence among the various stages in the migration process allowing the separation of relevant and redundant data from the source legacy candidate, establish critical nature of data to be ported and develop priorities based on the critical sensitivities, build knowledge schemas from user experiences and construct the condenser.

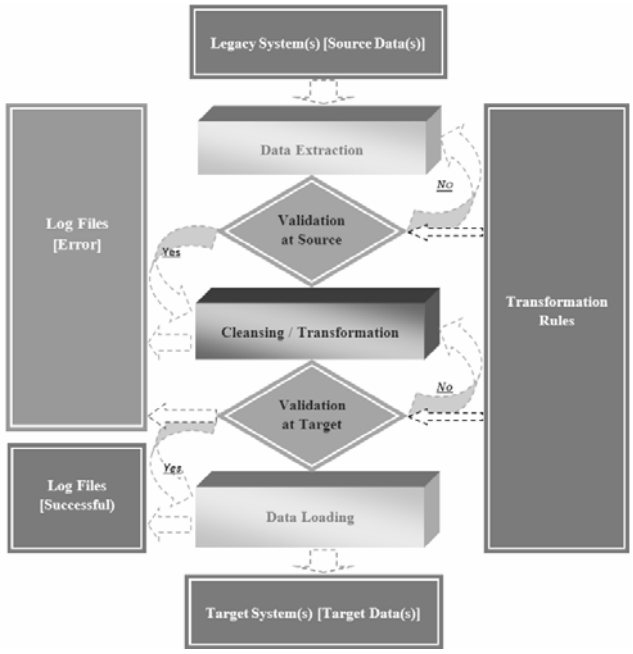

Figure1: The Proposed Data Migration Framework

The framework consists of the following steps:

i. *Data Extraction*: read and gather data from source data store(s) into another storage, and if required, convert to the data format of the Target System.

ii. *Validation and Cleansing*: to confirm content and structure of extracted data in light of business rules and fulfills integration rules based on the referential rules of Target System. Data Cleansing is performed at this time based on requirements identified during the Analysis phase.

iii. *Transformation*: convert the extracted data from its previous form into the target form. Transformation occurs by using Transformation Rules defined in Data Map (Transformation specification) and lookup tables.

iv. *Validation-Target System*: confirm content and structure of transformed data is valid for target.

v. *Data Loading*: Write the data into the target database, either through script or copying data using system utilities.

4. Implementation

The migration software developed in this research contains modules to extract data from a source system, correct errors, reformat, restructure and then load the data into a replacement target database. The system consists of a legacy archive extraction phase and a target database injection subsystem, implemented as a condenser. Priorities are assigned using the critical sensitivities of each data thereby enabling the movement of high-critical data and ensuring business continuity during the migration process. In addition to the transfer and reformatting of data, clean-up, redirection and allocation, as well as methods to carry out batch export of documents and associated indexes are also embedded into the software system.

We also developed a part of the data migration portal as a proof of concept. The portal could be

applied while migrating data from IMIS to Umoja (Figure 2). The idea of the portal is to eliminate a lot of the technical complexity associated with data migration.

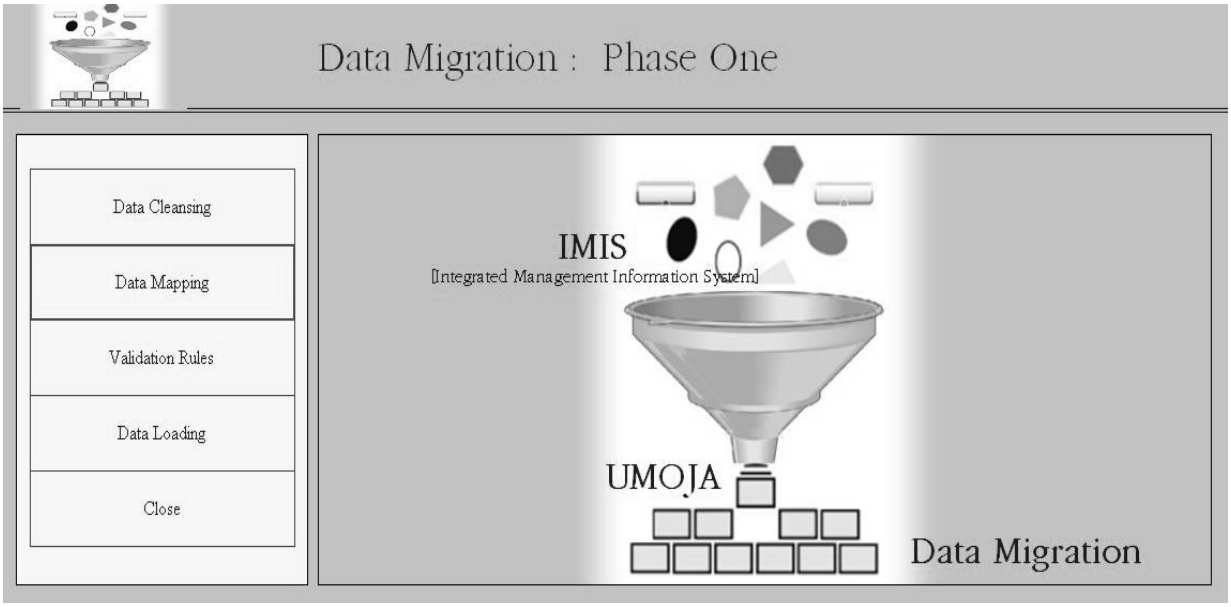

Figure 2: Data Migration Portal

The question is that the portal can be made both general and still has all the required functionality that is specific for each migration type. We think that at some point one will have to compromise between the general solution and the solution with all the specific and desired functionality. We tried to make it flow as general as possible and it will follow the same pattern independent of which type of migration project the user chooses. The portal is dynamic in the sense that the user can have a lot of various options.

5. Results and Discussions

We have defined a new design framework, using content analysis. The framework was adopted to develop a strategy for migrating legacy data to enterprise databases.

Preliminary stages include separation of the legacy system into information system data and historical (user) data, establishment of critical data and their prioritization and provision of the justification and definition of the high level requirements for the migration. Other preliminary activities include an analysis of the source data repository in order to determine the legitimate data to be migrated. The physical data structures are also frozen for both migration source database and the target data source.

The idea is to be able to illustrate the techniques used in the execution of the methodology involved in the development of the model. Legacy data will be accessed from a legacy store of any kind of data format and carefully studied. After which, such legacy data will undergo manipulation and redirection at the generic main lines area. They also undergo some sorts of transformation in accordance to certain mapping rules. The portal, a kind of utility program, carries out the actual migration of the data to the target database, Excel or XML files, usually an enterprise one. Along the line of process, data definition criteria is also done using data dictionary. Error reports generated are kept in error log files. Finally, migrated data are loaded into the enterprise service provider's database, for the organization's use.

6. Conclusion

Data migration is a process that most companies bump in to sooner or later. Although it might be tempting to postpone the first data migration due to all the risks involved, once it is done most companies perform data migration on a regular basis.

Several factors can necessitate the need to migrate data: different system architectures and data formats, differences in bit orders, alignment and padding, etc. Whatever is the reason, it is important that business activities don't come to a halt during the process of migration and that different data formats be harmonized to fit into those of the target data.

We proposed re-engineering through simplification of the existing Legacy Data Migration framework. We have specified an iterative process and implemented a smart module, which take the transfer to the data instead of the current approaches which took data to the transfer.

References

[1] Umoja Team, "Briefing for the UN on Umoja".

[2] Drumm, C., Schmitt, M., Do, H. H., and Rahm, E. „QuickMig - Automatic Schema Matching for Data Migration Projects", ACM Press, 2007.

[3] Haller, K., "Data Migration Project Management and Standard Software -- Experiences in Avaloq Implementation Projects. Architecture", 2008, pp. 391-406.

[4] Matthes, F., and Schulz, C., "A detailed process model for large scale data migration projects". 15th International Conference on Business Information Systems BIS2012, 2012.

[5] Hainaut, J.-L., Cleve, A., Henrard, J., and Hick, J.-M., "Migration of Legacy Information Systems", In T. Mens & S. Demeyer (Eds.), Software Evolution, Springer, 2008.

[6] Thalheim, B., and Wang, Q., „Towards a theory of refinement for data migration", In Proceedings of the 30th International Conference on Conceptual modeling (ER'11). pp. 318-331. Springer-Verlag, Berlin, Heidelberg, 2011.

[7] Horstmann, J., "Migration to Open Source Databases", Diploma Thesis, Technical University of Berlin, 2005.