Volume 6, No. 11, November 2015

Journal of Global Research in Computer Science

RESEARCH PAPER

http://rroij.com/global-research-in-computer-science.php

ISSN-2229-371X

RE-ENGINEERING LEGACY DATA MIGRATION METHODOLOGIES IN CRITICAL SENSITIVE SYSTEMS

Francisca O. Oladipo[1], Jude O. Raiyetumbi[2]

[1]Computer Science Department, Nnamdi Azikiwe University, Awka Nigeriafauzialias@unikl.edu.my

of.oladipo@unizik.edu.ng

[2]Computer Science Department, Kogi State University, Ayingba Nigeria

raitumbi@yahoo.com

Abstract: This aim of this research is to solve the portability problems of legacy data repositories during migration through the implementation of a data migration strategy for legacy databases. Our research revealed that provision was not made for migrating Enterprise databases from obsolete DBMS as they tend to become overgrown, excessively large and complex leading to a mutability and an inability to conform to the technology of contemporary architectures. We have carried out a re-engineering of the traditional data migration methodologies (the chicken-little and butterfly) by building a middleware, *Olaray* through the development and incorporation of a software module (the condenser) into the design stages to achieve a 100% migration. Using Content Analysis techniques, a stepwise adaptation of the Structured Systems Analysis methods hybrid with Object Oriented Design for forward engineering was deployed to implement the migration tool. Microsoft Access 2010 provided the backend, while the ASP.Net, Visual Basic.Net and Java Applets were deployed for the frontend. The developed methodology was tested using different homogenous and heterogeneous databases from MYSQL, PregSQL and DbaseIV. A populated result for the migration exercises into remote database environment was obtained in this case which implied a 100% successful data migration to the cloud.

INTRODUCTION

Data Migration is a very important activity during the transformation of Legacy investments in software to contemporary architectures. While the process can be cumbersome and time consuming, it is believed that the benefits far outweigh the eventual costs (TIMS, 2015).

Present day computing dictated the need for organizations to focus more on business processes and less on storage infrastructures. This gave rise to techniques such as information technology services outsourcing and virtual infrastructures provision and in addition would require that a common format be established for all entities involved in the outsourcing process.

Outsourcing data storage via storage-as-a-service is one major infrastructure-as-a-service outsourcing facility provided by third parties to an organization. One major concern of providers of this service is the problem of data portability which resulted from several factors chief among which are:

i. different system architectures for the source data
ii. the fact that data formats are not generally portable
iii. differences in bit orders, padding, alignment, etc.

In the same regard, believed that legacy systems cannot simply be replaced, but needed to be an integral part of the migration process as they represent substantial investments which cannot simply be disposed of as well as they are often the only place where certain business logic is documented [1]. One major consideration by the same researcher is the condition that business activities must continue during the migration process. Substantial downtime is often not an option since the business may be dependent on the legacy system.

The original process of importing legacy data to a target system generally involves manually entering the data into the system, transfer of disks and folders from one computer to another or introducing query statements for database insertion. In addition, because of the differences in the nature and state of the data being migrated as well as the type of applications that run the front-ends, there exist different migration strategies operating at different abstraction levels. There is therefore a need for a generic data migration system -a middleware that supports all data formats across different data sources.

This research examined the state-of-the-art in Legacy data Migration and created a new strategy for porting data from legacy DBMSs through a re-factory of the default migration strategies -Chicken-little and butterfly. In order to optimize the migration process and facilitate the priority migration of critical data, a condenser which is a software module is developed and incorporated into the migration system. In addition, a data access locator was designed and implemented to enable tabular item position-memory mapping. The final system incorporated prioritization into the creation of data extractions, cleaning, importation, and verification activities to ensure the continuity of business processes while transformation is ongoing.

The rest of this paper is organized as follows: After the provision of a brief background to data migration and its justifications in this section, a review of previous research and the state-of-the-art in Legacy Data Migration is presented. This is followed by a discussion of the adopted methodology and research materials and finally, a discussion and evaluation of the obtained results as well as the conclusions drawn were presented.

THE STATE OF THE ART IN LEGACY DATA MIGRATION

Legacy data generally results from using legacy information systems. A discussion on legacy data migration will therefore be incomplete without a detailed one on legacy system migration.

Legacy information system migration is a major research work but there are only a limited number of general migration methods available [2]. In this context, Tilley (1995) presented five perspectives to legacy system re-engineering [3]. These perspectives are: engineering, system, software, managerial, evolutionary and maintenance perspectives and serve as a framework for placing reengineering in the context of evolutionary systems.

In technology, data migration is the process of making a copy of data and moving it from one device or system to another, after which processing uses the new device or system. The key, or challenge, is to do this without disrupting or disabling active business processing and, most importantly, maintain the integrity of the data [4]. Migrations can be on a small or large scale that is planned in terms of months and even years. Regardless of the scope, steps must be taken to ensure data integrity, compatibility, low downtime and a secured migration process. This requires clear strategy, requirements determinations, and well laid-out plans for design and implementation.

Levine (2009) identified some business drivers that may necessitate data migration to include changes to the structure of databases, mergers and acquisitions which may give rise to new users or businesses, replacement, upgrade or consolidation of hardware and software infrastructures such as servers, data storage, applications and software, etc. [5]. Other factors include relocation of data centres or general performance-related tuning. This was further supported by Nowak et al., who in addition opined that structural differences in the source and target systems or inconsistencies across multiple data sources were also factors that necessitated the need for establishing a compatibility format for migration [6].

A set of "how to" was presented by Powell (2011) to provide a step-by-step guide to data migration to modern architectures [7]. The author in particular suggested discarding certain features of the legacy data such as annotations, workflows or images that cannot be converted. This research is set to address this concern by providing alternative representation of such features.

A3-part Legacy data migration strategy presented by Standen (2009) identified understanding and mapping as the key considerations in Legacy Data Migration [8]. The model involves determining data quality as the first step in its data migration, mapping the legacy system to the target data system and building a data dictionary as reference. While this model considered other factors such as location, dependencies and legal considerations, it fails to specifically address the concerns of critical sensitivity of the data.

Matthes et al. (2011) delineated a practice-based risk model, for migrating data [9]. The research was aimed at mitigating risks in data migration projects and it is essentially concerned with a data migration process-model that emerged and was continuously refined in the course of several industry data migration projects. The authors believed that relational databases are still relevant in business and end-user contexts in spite of the presence of other techniques for visualising persistent data. Hence the model migrates to relational data structures.

RELATED RESEARCH

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 the migration operations. 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 [10]. 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 datastore 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

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 [2].

Another early model is the "Big-Bang" or "Cold Turkey" approach by Bateman and Murphy [11] 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 while one problem associated with this approach is that, the proposed framework is presented at too high a theoretical level to be useful in practice, no consideration is given practically to the actual migration of the data.

The XTIVIA model was developed in response to the request by Sun Country airline to migrate and integrate their legacy data. The data flow algorithm was defined using the community edition of talend open studio and a sub-system was built to migrate and synchronise the data. The approach is essentially a three-step process with sub-steps in-between (Figure 1). A preliminary analysis is first conducted during which the client's requirements are established. This step is followed by the assessment phase where the approach, timeframe, and estimated schedule and costs are defined. Finally, migration, validation and deployment are carried out in the third and last phase.

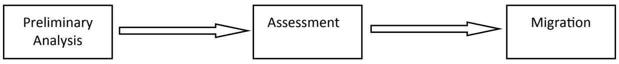


Figure 1. The XTIVIA Legacy Data Migration Methodology.

Another vendor-specific data migration strategy was developed by (Rosum, 2006) at The Data Warehousing Institute (TDWI) [12]. The strategy believed that data migration should hardly be a one way process where data is transformed from one point to another, but rather is essentially an iterative and cyclical process comprising of 7 steps where some of the steps may be repeated (Figure 2). The author likened data migration to the deportation of a nation of people and destroying their original country to prevent their ever going back -this must be done with utmost sensitivity. Thus the migration must successfully pass through each phase of the methodology.

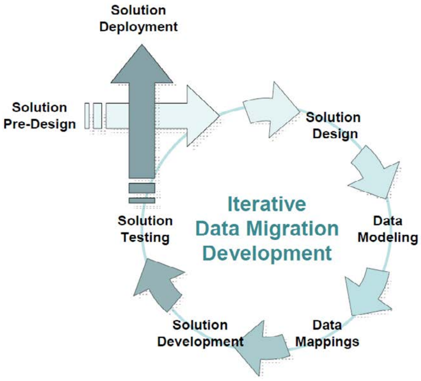


Figure 2. Cyclical Legacy Data Migration [12].

The primary concern of a research by (Emmrich et al. 2001) was the minimization of the number of adapters to be used during the integration of multiple legacy applications with several new target applications [13]. The research introduced an intermediate data format in place of software programs that provide different translational layers between data formats.

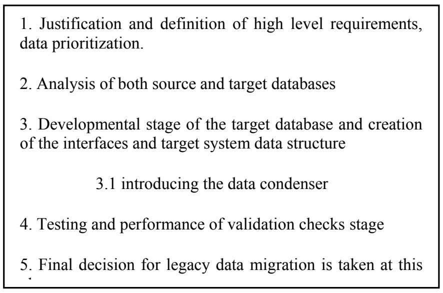
The technical details of the migration strategy of image data from a series of legacy applications by Ratib et al. were not made available [14]. This is primarily due to the fact that the migration process was proprietary and the target systems were commercial applications. However, the documentations 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 article offers little assistance, especially for IT personnel considering a large data migration dealing with medical image files or with large numbers of image files. These IT personnel may also find this discussion useful for the early planning stages.

Bianchi et al. applied a data re-engineering strategy to the Chicken Little approach to create an incremental model for migrating legacy information system [15]. 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 involves 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.

MATERIALS AND METHODS

We re-defined the Olaray methodology using Content Analysis and adopted it in this paper [16]. The methodology is a stepwise adaptation of the Structured Systems Analysis methods hybrid with Object Oriented Design for forward engineering of the migration tool (Figure 3). Object oriented design specification were done using UML class, activity and use case diagrams, and the resulting migration implementation provides support for point and click mapping.



1. Justification and definition of high level requirements, data prioritization.

2. Analysis of both source and target databases

3. Developmental stage of the target database and creation of the interfaces and target system data structure

3.1 introducing the data condenser

4. Testing and performance of validation checks stage

5. Final decision for legacy data migration is taken at this

Figure 3. The itemized Olaray migration method [16].

Other materials deployed in this research are:

i. One homogenous and heterogeneous databases each from MYSQL, PregSQL and DbaseIV as the source DBMS to test the migration model

ii.ASP.Net, Visual Basic. Net and java Applets were deployed for implementing the front-end of the software module (the condenser)

iii. Microsoft Access 2010 deployed as the Target DBMS backend

RESULTS AND DISCUSSIONS

We have defined a new design methodology, the Olaray in this work. Using content analysis, this methodology was adopted to develop a strategy for migrating legacy data to enterprise databases. The structural view of the Olaray methodology is depicted below (Figure 4). 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.

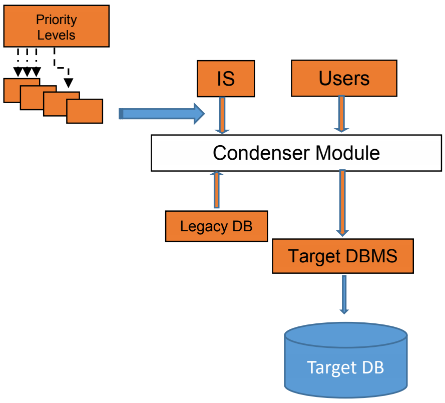


Figure 4. Structural view of the Olaray Methodology.

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 legacy data is passed through the condenser module and using the data access allocator (DAA) in the pre-known target DBMS, the legacy data is migrated into a source system.

A logical view of the above methodology is presented in Figure 5 below. The big 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 DBMS 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 condenser, a kind of utility program, carries out the actual migration of the data to the target database, usually an enterprise one. Along the line of process, data definition criteria, is also done using the data dictionary. Error reports generated are kept in error log files. Finally, migrated data are loaded in the enterprise service provider's database, for the organization use.

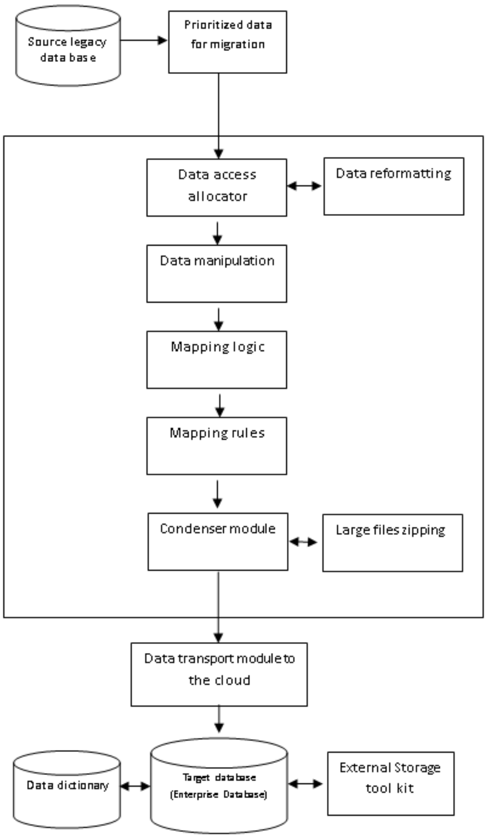


Figure 5. Logical View of the Olaray methodology.

Olaray Design and Implementation Details

The migration software developed in this research contains modules to extracts data from a source system, correct errors, reformat, restructure and then load the data into a replacement target database. As seen in the High Level Model (Figure 6), 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.

Supporting Figure 6 is the top-level UML activity diagram for the legacy data migration system (Figure 7). Data from a legacy store of any kind of DBMS format are loaded into the system and priorities are set for each data set.

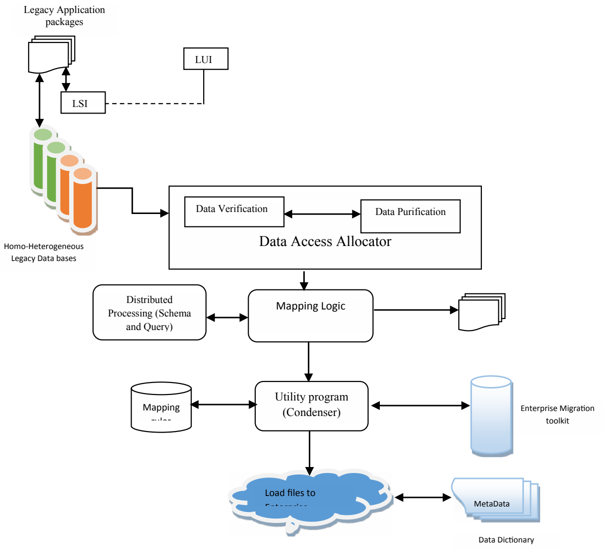

Legacy Application packages
LUI
LSI
Homo-Heterogeneous Legacy Data bases
Data Verification
Data Purification
Data Access Allocator
Distributed Processing (Schema and Query)
Mapping Logic
Mapping rules
Utility program (Condenser)
Enterprise Migration toolkit
Load files to Enterprises
MetaData
Data Dictionary

Figure 6. High Level Model of the Migration Software.

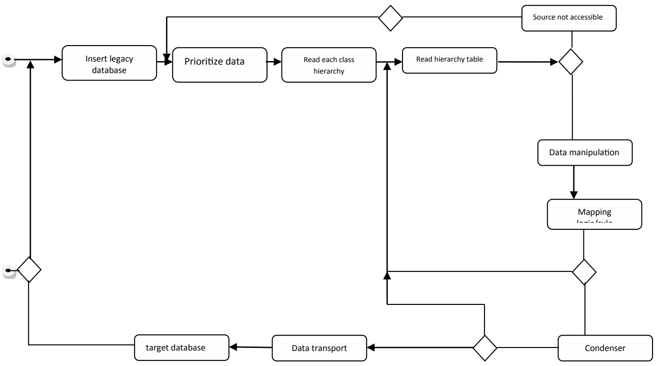


Figure 7. Activity diagram for the Legacy Migration System.

The developed migration software was tested using two legacy candidates: an inventory control databases in DBIV and the open source relational database in MySQL. The DBIV application was developed in 1998 by an undergraduate of Computer Science student to manage inventory for a multi-level marketing chain. Due to the age of the application, and the lack of technological support for storage system in use at that time, only partial data was available. The MySQL DBMS candidate data on the other hand was developed in 2004 using the Version 5.1 of 2008 release which had some similar features with modern DBMSs. The overall menu system for the migration tool is presented in Figure 8.

CONCLUSION AND RECOMMENDATION

There will always be old applications, with legacy data. These applications represent substantial corporate investments and discarding them when new technology surfaces imply losing the corporate investments and knowledge embedded therein.

Several factors can necessitate the need to migrate data: different system architectures and data formats, differences in bit orders, alignment and padding, etc. Whatever, 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 DBMS.

We recommend the re-engineering through simplification of the existing Legacy Data Migration methodologies as done in this work (Figure 9). As we have specified an iterative process and implemented a smart module, 'the condenser' which take the computation (transfer) to the data instead of the current approaches which took data to the transfer.

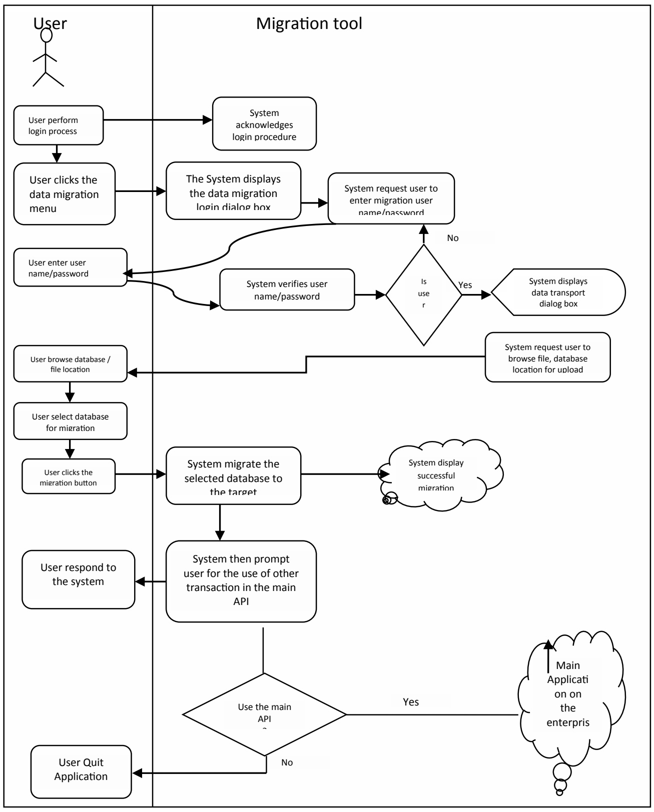

User
Migration tool
User perform login process
System acknowledges login procedure
User clicks the data migration menu
The System displays the data migration login dialog box
System request user to enter migration user name/password
No
User enter user name/password
System verifies user name/password
Is use r
Yes
System displays data transport dialog box
System request user to browse file, database location for upload
User browse database / file location
User select database for migration
User clicks the migration button
System migrate the selected database to the target
System display successful migration
User respond to the system
System then prompt user for the use of other transaction in the main API
Use the main API
Yes
Main Applicati on on the enterpris
No
User Quit Application

Figure 8. The overall menu system for the migration Tool.

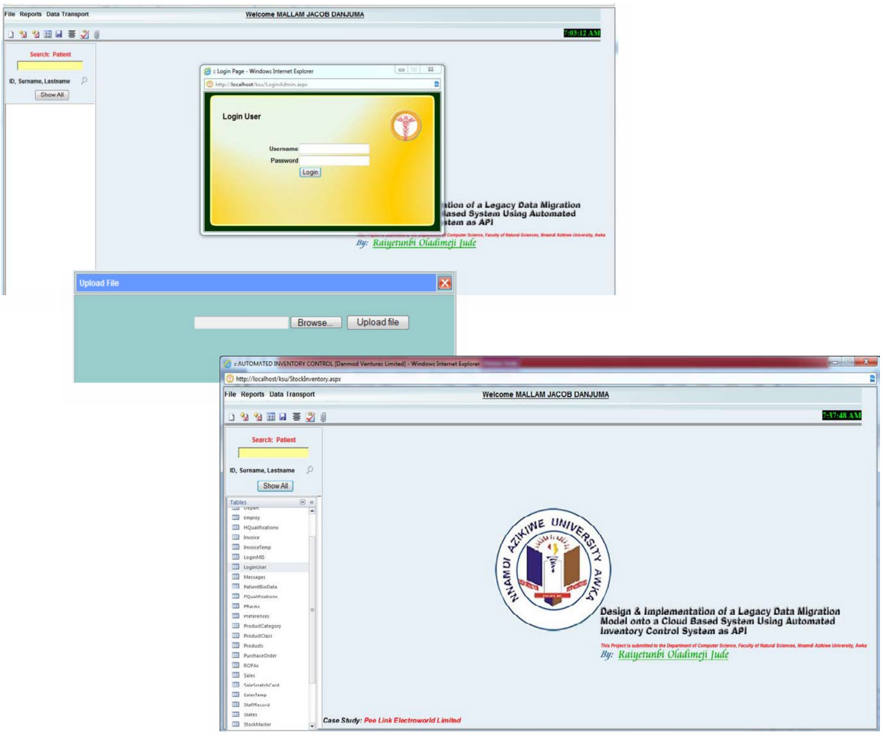

File Reports Data Transport
Welcome MALLAM JACOB DANJUMA
Search: Patient
ID, Surname, Lastname
Show All
Login Page - Windows Internet Explorer
Login User
Username
Password
Login
By: Raiyetunbi Oladimeji Jude
Upload File
Browse...
Upload file
AUTOMATED INVENTORY CONTROL [Danmod Ventures Limited] - Windows Internet Explorer
http://localhost/ksu/StockInventory.aspx
Tables
NNAMDI AZIKIWE UNIVERSITY AWKA
Design & Implementation of a Legacy Data Migration Model onto a Cloud Based System Using Automated Inventory Control System as API
By: Raiyetunbi Oladimeji Jude
Case Study: Pee Link Electroworld Limited

Figure 9. It shows some sample application's interface. The upper box showed an empty target database, the browse and upload box is shown next and finally a populated database is shown.

REFERENCES

[1] Hasselbring W, Reussner R, Schlegelmilch J, Teschke T. and Krieghoff S, "The Dublo Architecture Pattern for Smooth Migration of Business Information Systems: An Experience Report." Proceedings of the 26th International Conference on Software Engineering, pp.117-126, 2004.

[2] Wu B, Lawless D, Bisbal J, Richardson R, Grimson J, Wade V, O'Sullivan D, "The Butterfly Methodology: A Gateway-free Approach for Migrating Legacy Information Systems." In Proceedings of the 3rd IEEE Conference on Engineering of Complex Computer Systems (ICECCS97), Villa Olmo, Como, Italy, 1997.

[3] Tilley SR, "Perspectives on Legacy System Reengineering." Software Engineering Institute Carnegie Mellon University, Pittsburgh, PA, 1995.

[4] Yourself LM, Batrico D, and Da Silva, "Towards a unified ontology of cloud computing." Grid Computing Environments Workshop, pp. 1-10, 2013.

[5] Levine R, "Data migration strategies. The Wikibon Project." 2009.

[6] Nowak A, Leymann F, and Mietzner R, "Towards Green Business Process Reengineering," in Proceedings of the Workshop on Services, Energy, & Ecosystem, pp. 44-52, 2010.

[7] Powell JE, "Q&A: How to Successfully Migrate Legacy Data." Enterprise Systems Journal, 2011.

[8] J. Standen, "Data migration Part 3- Mapping the legacy systems," Datamartist, 2009.

[9] Matthes F and Schulz C, "Testing & Quality Assurance in Data Migration Projects Software Engineering for Business Information Systems." In Proceedings of 27th IEEE International Conference on Software Maintenance (ICSM'11), Williamsburg, VA, 2011.

[10] Brodie M and Stonebraker M, "Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach." San Francisco: Morgan Kaufmann Publishers Inc, 1995.

[11] Bateman A and Murphy J, "Migration of Legacy Systems." School of Computer Applications. Dublin: Dublin City University: Dublin, 1994.

[12] Rossum P, "Best Practices in Data Migration." Informatica, 2006.

[13] Emmrich W, Ellmer E and Fieglein H, "An Architectural Style for Enterprise Application Integration." Proceedings of the 23rd International Conference on software Engineering (ICSE-01), Gaithersburg, Maryland, pp. 567-576, 2001.

[14] Ratib O, Liu B, Kho HT, Wenchao T, Wang C and McCoy J, "Multigeneration Data Migration from Legacy systems." Proceedings of the SPIE – The International Society for Optical Engineering, Integrated Medical Information Systems Design and Evaluation. Medical Imaging, San Diego, pp. 285-288, 2003.

[15] Bianchi AD, Caivano V, Marengo and G Vissagio, "Iterative reengineering of Legacy Systems." IEEE Transactions on Software Engineering, Vol. 29, no. 3, pp. 225-241, 2003.

[16] Raiyetunbi OJ, "Design and Implementation of a model for legacy data migration onto a cloud based system." MSc Thesis, Department of Computer Science, Nnamdi Azikiwe University, Awka Nigeria, 2014.