A Seamless Network Database Migration Tool for Insititutions in Zambia

Mutale Kasonde
Department of Electrical and Electronic Engineering
University of Zambia
Lusaka, Zambia

Simon Tembo
Department of Electrical and Electronic Engineering
University of Zambia
Lusaka, Zambia

***Abstract*—The objective of the research was to efficiently manage migration process between different Database Management Systems (DBMS) by automating the database migration process. The automation of the database migration process involved database cloning between different platforms, exchange of data between data center and different clients running non-identical DBMS and backing up the database in flexible format, such as eXtensible Markup Language (XML). This approach involved development of a "Database Migration Tool". The tool was developed on a windows platform using Java Eclipse™ with four non-identical dummy Relational Databases (Microsoft Access, MySQL, SQL Server and Oracle). The tool was run in a controlled environment over the network and databases were successfully migrated from source to targeted destination option specified. The developed tool is more efficient, timely, as well as highly cost effective.**

Keywords*—Database management system; database migration; database structure; database migration toolkits and database cloning

I. INTRODUCTION

Advancements in technology usually result in database migration, and an example would be for the National Pension Scheme Authority (NAPSA). A database (DB) is a persistent, logically coherent collection of inherently meaningful data, relevant to some aspects of the real world [1]. A collection of these databases is what forms the Database Management Systems (DBMS).

Database Management Systems perform a wide variety of roles such as allowing concurrency, controlling security, maintaining data integrity, providing for backup and recovery, controlling redundancy, allowing data independence, providing a non-procedural query language as well as performing automatic query optimization.

Migrating a database involves migrating the tables and records from one database management system to another. The Transvive white paper, 2014 defines the term "Migration" as the movement of technology from older, or proprietary systems to newer, more versatile, feature-rich and cost-effective applications, and operating systems [2]. Data migration is usually undertaken for the purpose of replacing, upgrading server or storage equipment for a website consolidation, so as to conduct server maintenance or to relocate a data center.

However, because different database management systems have different formats for storing the database, the exchange of database tables and records between different database systems usually results in compromising the quality, or authenticity of the data in the transformation process. According to an Oracle White paper, 2011, up to 75% of new systems fail to meet expectations, often because of flaws in the database migration process, which in turn result in data that is not adequately validated for the intended task [3].

Some of the challenges associated with database migration processes include data loss particularly in a case of Poor Legacy Data Quality, having Wrong Data Migration Tools, inadequate knowledge in using the precise Data Migration Tools, failure to Test and validate Data Migration Process, and Absence of Data Governance Policies [4].

Currently, there is no comprehensive system to comportment the data migration process. The existing procedure for migration is semi manual, and involves fragmentary procedures, which entails using different tools in order to achieve a comprehensive process. As such, maintaining the structure of a database when migrating it from one database management system to another is quite a challenging task for most Organizations. Often times, organizations have to design a new database for the different database management system it wants to adopt. This current system of database migration is not only costly, but it's also ineffective, and may sometimes result in the loss of essential data in the migration process. This is because this system involves hiring a database designer every time the Organization has to switch to a different database management system. Therefore, by developing an automated database migration process, the challenges being experienced with the current migration process required to be addressed and eliminated.

This study intended to explore ways in which data migration process could be improved through the development of a new Seamless Database Migrator. This was expected to help overcome challenges associated with the network database, and deliver the data with such accuracy. Specifically, the new database migrator was expected to.

1) Eliminate the need for script writing when transferring tables in the database with the records.

2) Prompt the user to select the destination and source Database Management System, and specify what to migrate i.e. either the entire database with structure and data or just structure or data, thereby allowing different database management systems to exchange database tables and records in them, without any loss of data details in the transformation process.

The paper is organized as follows: Section 2 deals with literature review, which covers existing tools as well as the theoretical literature bordering on policy issues regarding database migration. The methodology is presented in Section 3. Section 4 brings out the results and the discussion of the baseline study conducted to identify challenges in the database migration process. System testing is presented in Section 5, and the last Section 6 contains the conclusion.

II. Literature Review

In order to cope with a fast changing business environment, it is necessary to update the technological infrastructure constantly, and database migration is a routine part of this technology. Barron. C et al. suggested that the core reason for the need of database migration is mainly to upgrade the existing system into a developed system that conforms to the Industry requirements [5].

The tasks of a migration workflow are diverse and complicated, executing all these processes manually requires plenty of time, as well as a highly experienced migration team in both the source, as well as the target system. In a paper, reviewing database migration strategies, tools and techniques, Elamparithi, M and Anuratha, V singled out relational database migration (RDM) as an example. The authors stated that relational database migration was always a complex, time consuming, and magnified process due to heterogeneous structures and several data types of relational database [6]. This gives rise to certain risks and challenges in the data migration process.

The Arbutus Software Whitepaper summarized the risks of database migration as follows: Unrealistic estimates of data quality, inaccurate, missing or out of date source system documentation, as well as the inability to reconcile the target systems data to the source system [7].

To counter the above challenges of database migration, businesses have seen the need to develop effective methodologies of migrating databases. Several migration tools and strategies have since been developed in the software industry [8]. However, finding the effective methodologies for database migration still remains a challenge, and many current approaches to data migration suffer from a consistently low success rate. Arbutus Software White Paper approximates that between 70 and 90% of data migration projects either fail outrightly, or run over budget, with an average cost overrun of 10 times the original estimate [9]. This is mainly due to the unplanned issues that often occur at the later stages of a project.

Several researches have since been undertaken to address some of the problems associated with database migration, although no absolute solution has come forth. For example, Joseph R. Hudicka, provided a complete solution of data

migration methodology, which deals with row counts, column counts and related statistics to the source databases [10]. However, the problem with this methodology is that it does not migrate null and numeric values and error occurs for key constrain keys.

TABLE I. DEVELOPED DATA MIGRATION TOOL [15]

S. No	Name	Company	Source	From	To	Operating System
1	OSDM Toolkit	Apptility	Open	Oracle, SyBase, Informix, DB2, MS Access, MS SQL	PostgreSQL & MYSQL	Windows, Linux, Unix & Mac OS
2	DB Migration	Akcess	Closed	Oracle & MS SQL	PostgreSQL & MYSQL	Windows
3	Mssql2 Pgsql	OS Project	Open	MS SQL	PostgreSQL	Windows
4	MySQL Migration Toolkit	MySql AB	Open	MS Access & Oracle	MySQL	Windows
5	MySQL Migration Toolkit	Intelligent Convertors	Closed	MS Access, MS SQL, Dbase & Oracle	MySQL	Windows
6	Open DBcopy	Puzzle ITC	Open	Any RDB*	Any RDB*	OS Independent
7	Progression DB	Versora	Open	MS SQL	PostgreSQL, MySQL & Ingres	Linux & Windows
8	Shift2Ingres	OS Project	Open	Oracle & DB2	Ingres	OS Independent
9	SQLPorter	Real Soft Studio	Closed	Oracle, MS SQL, DB2 & Sybase	MySQL	Linux, Mac OS & Windows
10	SQLWays	Ispirer	Closed	All Relational Databases	PostgreSQL & MySQL	Windows
11	SwisSQL Data Migra-tion Tool	AdventNet	Closed	Oracle, DB2, MS SQL, Sybase & MaxDB	MySQL	Windows
12	SwisSQL SQLOne Console	AdventNet	Closed	Oracle, MSSQL, DB2, Informix & Sybase	PostgreSQL & MySQL	Windows
13	MapForce	Altova	Closed	SQL Server, DB2, MS Access, MySQL & PostgreSQL	SQL Server, DB2, MS Access & Oracle	Windows, Linux & Mac OS
14	Centerprise Data Integrator	Astera	Closed	SQL Server, DB2, MS Access, MySQL & PostgreSQL	SQL Server, DB2, MS Access, MySQL & PostgreSQL	Windows
15	DBConvert	DB Convert	Closed	Oracle, DB2, SQLite, MySQL, PostgreSQL, MS Access & Foxpro	Oracle, DB2, SQLite, MySQL, PostgreSQL, MS Access & Foxpro	Windows

Ramaswamy, V.K. argued that effective and efficient migration of data is one of the cornerstones for the success of the process [11]. He further emphasized the fact that significant planning needed to be done before the actual process of data migration commences. He outlined a strategy for data migration in which he listed down the type of data to be migrated, timing of the data load, templates and tools for use in the migration process.

Sait S.A. et al. on behalf of Amazon Web Services (AWS) provided comprehensive strategies for migrating Oracle databases in which they stated that there is no absolute formula for migrating databases but that there are certain factors one needed to put into consideration before undertaking database migration [12]. These factors include the size of the database, the network connectivity between the source server and the target service, the version and edition of the oracle database software, the database options, tools and utilities that are available, as well as the time available for the migration process. Based on the above factors, the authors divided the migration process into two methods namely the One Step migration, which is ideal for small databases, and the two-step migration, which can be used for any size of the database.

Currently, a number of prototypes and tools have been developed to facilitate the migration of relational databases (RDBs) into target databases. Senior researchers Bin Wei, and Tennyson X. Chen, developed Data Migration Tool (DMT) for US National Oceanic and Atmospheric Administration (NOAA), outlining the criteria that need to be considered when evaluating a DMT [13]. However, while the criteria outlined by these authors may be adequate for the complex project of developing DMT, the complexity of a general extract, transform, and load (ETL) system may go beyond what these criteria can evaluate. Still the investigations are needed on dealing with complex files [14].

Jutta Hortsmann, J. suggested some examples of database migration tools as shown in Table I.

The data migration process under goes through several stages, which include planning, designing, cleansing, loading as well as verifying of the data. Fig. 1 shows the Data migration process.

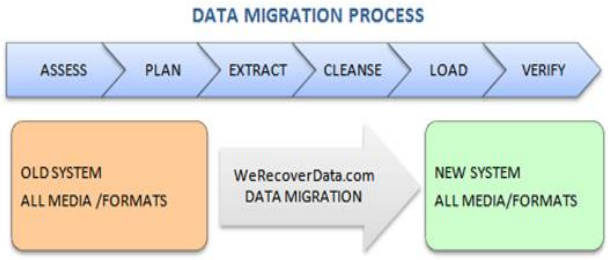

Fig. 1. Data migration process [16].

III. Methodology

The method of analysis for this project was divided into two categories, guided mainly by the objectives of the study.

The first method of analysis involved getting expert opinion from IT personnel that were purposefully selected from the Applications and Database Administration sections of three institutions namely NAPSA, ZANACO and ZAMTEL. These IT experts were meant to capture as much qualitative data as possible regarding the existing database migration procedures, as well as identifying the challenges associated with it.

The second method of analysis involved designing the modules that made up the system. The overall purpose of this system design was to provide efficiency and effectiveness in data migration process. To achieve this process, the Java Programming Language and Database Management Systems technologies were used.

This approach involved developing the tool "Database Migrator Tool". The tool was developed on a windows platform using Java Eclipse with four non-identical dummy databases (Microsoft Access, MySQL, SQL Server and Oracle). The automation of the database migration process involved database cloning between different platforms, exchange of data between data center and different clients, running non-identical DBMS and backing up the database in flexible format such as eXtensible Markup Language (XML). The tool was run in a controlled environment over the network. The following java support tools were used to support the technologies used in the system:

1) MySQL connector (mysqlconnector.jar); which was used to connect MySQL database management system from java.

2) SQL server (sqljdbc.jar); which was used to connect SQL server database management system from java, and

3) Jackcess (jackcess.jar); which was used to connect SQL server database management system from java.

4) Oracle connector (OJDB5.jar); this java library was used to connect Oracle database Management system from Java.

The Database Management Systems used included:

1) Microsoft SQL Server is a database management system whose primary function in this case was to store the database in SQL server format (.mdf) and retrieve data as requested by other software applications.

2) MySQL database management system was used to store the database in MySQL format (.frm). This database management system stored data in separate tables whose structures were organized into physical files.

3) Microsoft Access Database Management System combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It is a member of the Microsoft Office suite of applications, included in the Professional and higher editions. Microsoft Access was used to store database data in its own format (.accdb) based on the Access Jet Database Engine.

4) Oracle database Management system was used to store and retrieve related information. Oracle database management system was used to store database data in its Oracle database format (.dbf)

IV. Results and Implementation of New System

This segment presents the results obtained from the baseline study as well as development and testing of system prototype. In order to confidently and significantly address the challenges associated with the old system, baseline study was conducted and proposed prototype application was developed.

The data collected from the baseline study was analyzed using descriptive statistics and the results were presented in form of charts. From the responses obtained from the respondents, 58% admitted that the tools that were currently being used for database migration were wrong tools. 42% said the tools were not wrong per ser. However, 71% of the responses indicated that the old system had no provision to test and validate the data migration process. Additionally, 87% had experienced data loss in the process of migrating the database when using the old system. 93% of the respondents admitted that because of the technical challenges associated with the old database migration process, such as constant data loss, failure to test and validate the migration process, engaging a consultancy every time the need arises, etc., the old system was too expensive to manage. 88% further stated that there was inadequate knowledge among the users on the precise data migration tools to use. 55% of the respondents also stated that there were no existing data governance policies, a situation that made it difficult to manage this system. Fig. 2 shows the summary of the results from the baseline study.

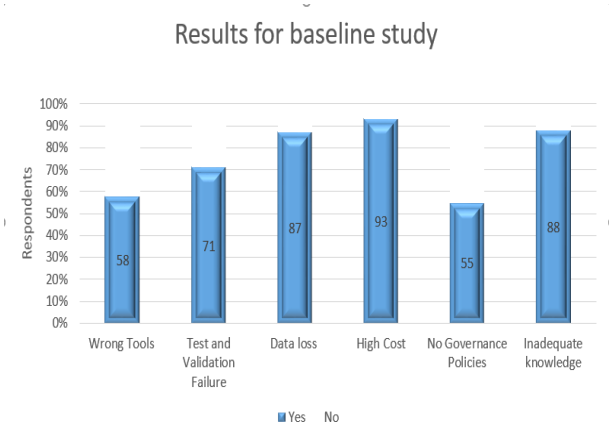

Fig. 2. Baseline study results.

A. The Architectural Design

This Database migrator was developed using a Top-Down approach. This approach involved decomposing the system into individual smaller modules, aimed at achieving the required detail. Fig. 3 shows the Internal Logic Design.

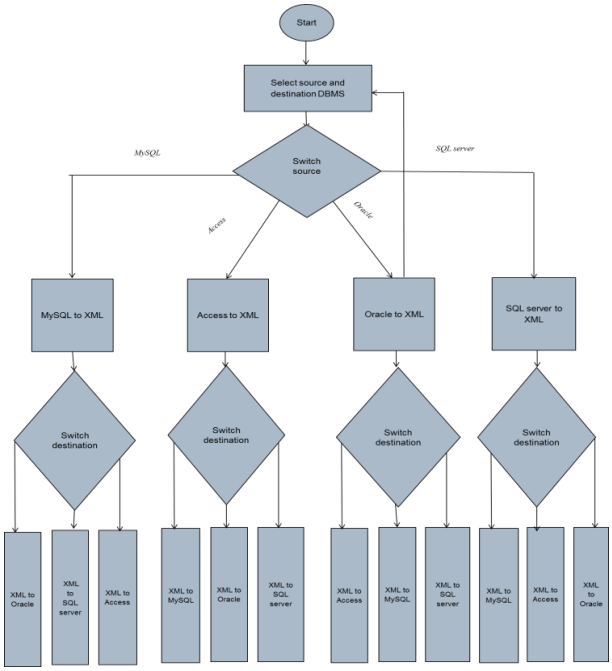

Fig. 3. Internal logic design.

B. Conversion Modules

A prototype was developed and a database migrator was effectively implemented using the technologies elaborated in the project methodology. Fig. 4 below shows the conversion module from XML to Oracle.

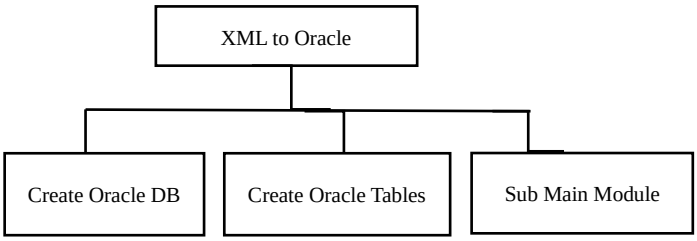

Fig. 4. XML to Oracle.

C. The Interface Design

Database migrator is a desktop application and the interface enables the user to use the system without any difficulties; Java Graphical User Interface (GUI) was used to design and develop the interface. The interface design comprises of the main interface, sub interface and the graphical interface.

D. The Main Interface

The main interface is the home interface for the system and appears when the system runs. It consists of buttons used for running the migration process where the user selects the source and destination Database management system. Fig. 5 show the Main Interface

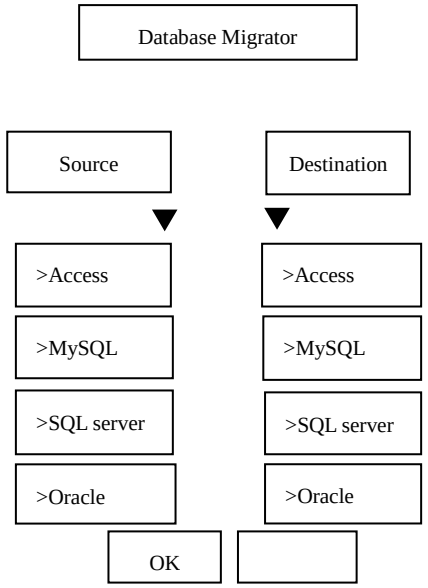

Fig. 5. The Main Interface.

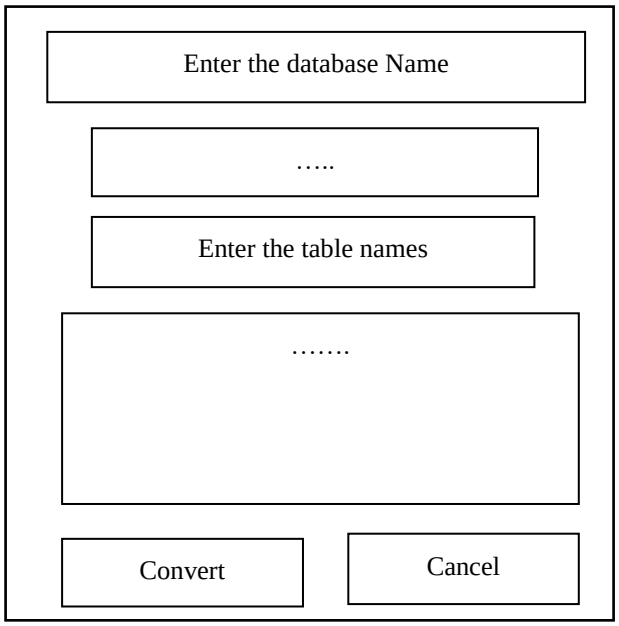

Fig. 6. Sub module.

E. Sub Interface

The sub interface is used for the actual conversion from one database system to the other. Upon specifying the details of the source database, the sub interface converts the source database system into the destination database system. In case of an error, the sub interface has a provision for cancelling the process.

All the four sub modules were converted to XML file, which in turn acts as a common ground for converting the database from one system to another. Once the design specification and project design was approved, system coding commenced. Fig. 6 shows the sub module.

F. Graphical User Interface

The graphical user interface allows the user to interact with the system; it was developed using Eclipse java (jdk 1.6.0). The user interface provided the user with a point and click interface which reduced user's errors because the user was not prompted to enter any information.

G. The Migration Process

The process involves selection of the source and destination Database Management System using the main interface by the system users. The sub interface then converts from one system to the other. Fig. 7 shows the source and destination interface.

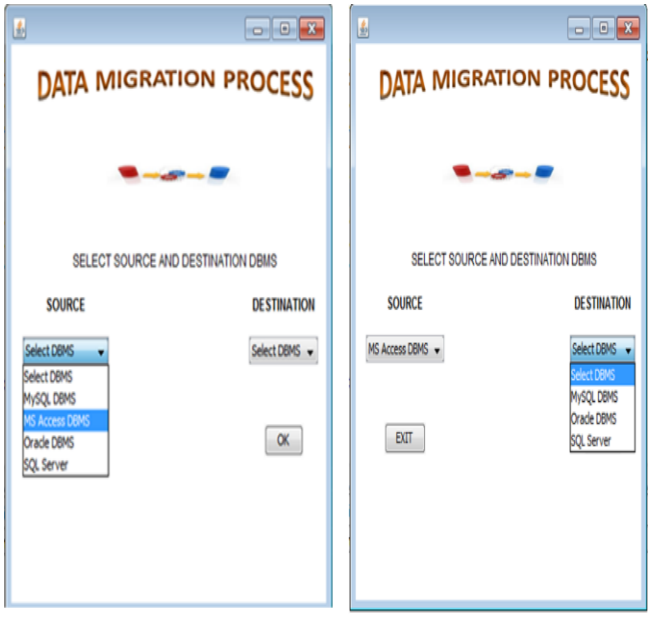

Fig. 7. Source and destination of DM.

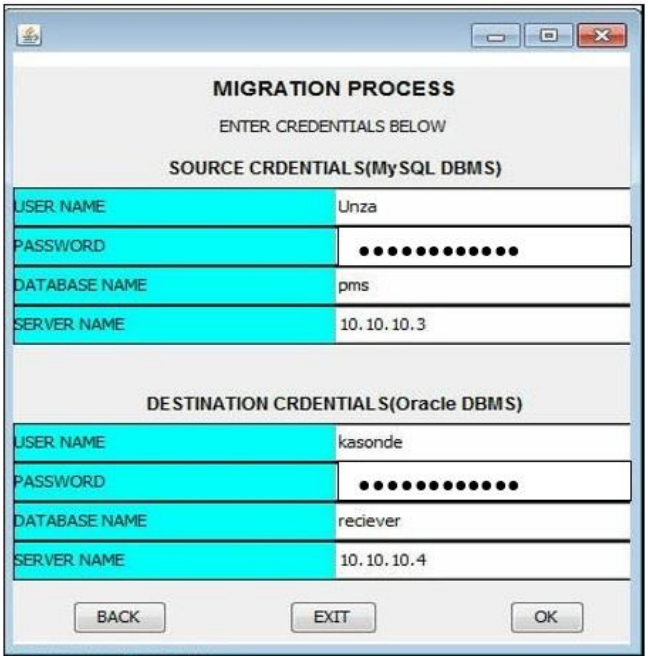

Fig. 8. Remote database credentials.

Once the user selects the source and destination databases, the tool prompts the user to specify credentials for the source and destination databases as shown in Fig. 8.

Upon authenticating the source database credentials, the tool prompts the users with options to specify what needs to be migrated, i.e. data, table or entire database. Fig. 9 shows the tables from source database.

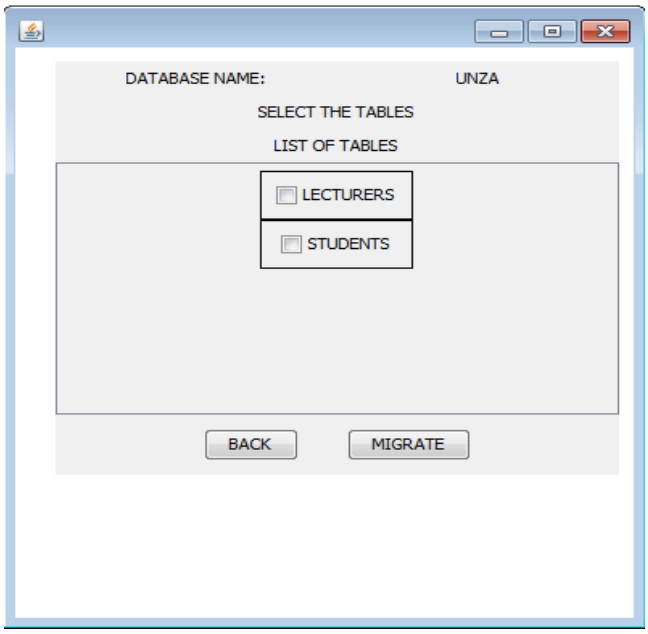

Fig. 9. Tables selected for migration process.

TABLE II. TESTING OF CLASSES FROM XML FILE TO DATABASE

	Testing Scenarios	**Expected Results**	**Actual Results**
1	XML file to Microsoft Access	Access DB	XML file was converted to Access DB
2	XML file to MySQL	MySQL DB	XML file was converted to MySQL DB
3	XML file to SQL Sever	SQL Server DB	XML file was converted to SQL Server DB
4	XML file to Oracle	Oracle DB	XML file was converted to Oracle DB

V. SOFTWARE TESTING

To ensure that the new system was working according to the intended objective, system testing was done both for the individual units as well as the integrated system. Unit testing was carried out by testing all the individual modules separately with connections to the database to see if module

performed all the expected functions. System testing was also carried out to determine how well the various components that comprise the system would interact in order to achieve the total system functionality. The main objective of testing was to detect and eliminate errors, as well as to check on whether it meets the requirements set out in the requirement specification document.

A. Integration Testing

After testing individual modules, integration testing was carried out and this involved putting the modules together and testing them to ensure that the object classes still work as expected even after being combined with other classes.

B. Testing Output

The classes for converting from database to XML file were tested separately to ensure that the selected source database management system was converted to XML. Table II shows testing of classes from Database to XML file whereas Table III shows testing of classes from XML file to Database.

TABLE III. TESTING OF CLASSES FROM DATABASE TO XML FILE

	Testing Scenarios	**Expected Results**	**Actual Results**
1	Microsoft Access to XML file	XML file	MS Database was converted to XML file
2	MySQL to XML file	XML file	MySQL Database was converted to XML file
3	SQL to XML file	XML file	SQL Database was converted to XML file
4	Oracle to XML file	XML file	Oracle Database was converted to XML file

The classes for converting from an XML file to a database were also tested separately to ensure that the xml files are converted to specified database management system. Table IV shows the system module testing.

TABLE IV. SYSTEM TESTING OF MODULES

	Testing Scenarios	**Expected Results**	**Actual Results**
1	Migrating from MS Access to MySQL.	MySQL DB	MS Access DB was successfully converted to MySQL DB
2	Migrating from MS Access to SQL Server.	SQL Server DB	MS Access DB was successfully converted to SQL Server DB
3	Migrating from MS Access to Oracle	Oracle DB	MS Access DB was successfully converted to Oracle DB
4	Migrating from MySQL to MS Access	SQL Server DB	MySQL DB was successfully converted to MS Access DB
5	Migrating from MySQL Server to SQL Server.	Access DB	MySQL Server DB was successfully converted to SQL Server DB
6	Migrating from MySQL Server to Oracle	Oracle DB	MySQL Server DB was successfully converted to Oracle DB
7	Migrating from SQL Server to MS Access.	Access DB	SQL Server DB was successfully converted to MS Access DB
8	Migrating from SQL Server to MySQL.	MySQL DB	SQL Server DB was successfully converted to MySQL DB
9	Migrating from SQL Server to Oracle.	Oracle DB	SQL Server DB was successfully converted to Oracle DB
10	Migrating from Oracle to MS Access	Access DB	Oracle DB was successfully converted to MS Access DB
11	Migrating from Oracle to MySQL Server	MySQL DB	Oracle DB was successfully converted to MySQL DB
12	Migrating from Oracle to SQL Server	SQL Server DB	Oracle DB was successfully converted to SQL Server DB

The system did the conversion successfully and the expected results were obtained. According to the minimum requirement of the system, the system performance was found to be fair. However, one important lesson learnt from this testing process was that modules which performed well when tested independently, do not always perform well when integrated with other modules. See Fig. 3, internal design.

In order to build a thorough contextual understanding of the issues at play, this project undertook an assessment of the old database migrator and challenges associated with it. Consequently, the project developed a new system that addresses the problems being experienced by the current database migration systems. The process was accomplished through planning and designing of a new system, cleansing, loading as well as verifying the new database migrator.

The new system does not support databases in Real Application Cluster (RAC) but only works for a single Database Node.

VI. Conclusion and Recommendation

The project objectives were achieved through the development of an automated database migrator. The results indicated that the new system was operating efficiently and effectively. It was therefore recommended that the new tool be adopted. The Seamless database migrator addresses the numerous challenges that are associated with the Database migration process. The implementation of the Seamless Database Migrator ensured that there was efficiency and effectiveness in the application and use of the database management system. With the use of a new DM tool, the challenges that were being experienced with the current migration process were addressed and eliminated. There was no data loss; no cost of hiring database designer to remodel the new Database (DB), time taken to migrate was reduced tremendously.

References

[1] Robbins, J.R, (1995, May), "Database Fundamentals" [Online]. pp-2. Available http://www.esp.org/db-fund.pdf [May. 16, 2017].

[2] Transvive white paper (2014, Apr)"Migration Strategies & Methodologies" pp 4. Available https://www.platformmodernization.org/transvive/Lists/ResearchPapers/Attachments/1/Transvive-MainframeMigrationStrategy-WP.pdf [May. 16, 2017].

[3] An Oracle White Paper. (2011,Oct.) "Successful Data Migration "[Online]. Pp.2-3. Available http://www.ijssst.info/info/IEEE-Citation-StyleGuide.pdf [May. 17, 2017].

[4] An Oracle White Paper. (2011,Oct.) "Successful Data Migration "[Online]. Pp.4-11. Available http://www.ijssst.info/info/IEEE-Citation-StyleGuide.pdf [May. 17, 2017].

[5] Barron C. Housel, Vincent Y. Lum, Nan Shu (1974),'Architecture to an Interactive Migration System' in SIGFIDET 1974: Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, Access and Control, New York, USA, pp. 157-169.

[6] Elamarithi M, and Anuratha. V. "A Review on Database Migration Strategies, Techniques and Tools." World Journal of Computer Application and Technology 3 (3): 41/48, 2015.

[7] Arbutus white paper. (2016,Oct.) "Top Three Data Migration risks "[Online]. Pp.2. Available https://cdn2.hubspot.net/hubfs/3336879/ArbutusSoftware-June2017/Pdfs/arbutus-wp-data-migration.pdf?t=1503679160981 [May. 16, 2017].

[8] Paul Dorsey, Joseph R. Hudicka) "Oracle Data Migration Handbook." McGraw-Hill Osborne Media, pp 23-27, 2000.

[9] Arbutus white paper. (2016,Oct.) "Top Three Data Migration risks "[Online]. Pp.3-5. Available https://cdn2.hubspot.net/hubfs/3336879/ArbutusSoftware-June2017/Pdfs/arbutus-wp-data-migration.pdf?t=1503679160981 [May. 16, 2017].

[10] Joseph R. Hudicka , 'Oracle8 Database Design Using Uml Object Modeling' McGraw-Hill Professional , ISBN-13:9780078824746, pp 216-224, 1998.

[11] Ramaswamy, V.K, Database Migration Strategy in ERP, London: United Kingdom, 2016

[12] Sait A, S, Strategies for migrating Oracle Databases to AWS, Amazon Web Services, Inc: Amazon2014.

[13] Bin Wei and Tennyson X. Chen (2012), 'Criterial for Evaluating General Database Migration Tools', [online] Research Report, RTI Press Publication No. OP-0009-1210. Research Triangle Park, NC:RTI Press. http://www.rti.org/pubs/op-0009-1210-chen.pdf (Accessed 17 July 2017

[14] S. M. Abdelsalam Amaraga Maatuk, Migrating Relational Databases into object-based and XML databases., Published PhD thesis, Nortambria University, Newcastle, United Kingdom, 2009

[15] Horstmann. J, Migration to Open Source Databases, Technical University, Berlin:Germany, 2005

[16] Data recovery labs, "Data Migration Process" Internet: www.werecoverdata.com [Mar. 10 2017]