Vol. 5, No. 3 March 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

A Novel Algorithm for Integrating E-Health Data in Distributed Low Bandwidth Environment

[1] Vitalis Ndume, [2] Yaw Nkansah-Gyekye, [3] Jesuk Ko
[1, 2] NM-AIST, Arusha, Tanzania
[3] Department of Healthcare Management, Gwangju University, Korea

[1] ndumev@nm-aist.ac.tz

ABSTRACT

Electronic Health Records System is transforming primary health care and improves quality of health facility management. It enhances immediate monitoring and evaluation of health facilities. However, integrating e-Health data in a network with low bandwidth imposes more challenges. This paper describes a technique to integrate e-Health records which is heterogeneous using Web Service Model in a low bandwidth environment. The focus is on the problem of medium–scale data integration where data source schemas are not known at the design time. The validation and test cases of the software were done using heterogeneous data set from various longitudinal studies including demographical surveillance data sets and baseline census data for sample vital registration with verbal autopsy. The results indicate that MySQL engine is faster than others. The proposed architecture and the algorithm applied for mapping allow heterogeneous data set to be integrated in a low bandwidth environment. The algorithm cuts off the transfer and execution time to (2On).It is concluded that Web service has more advantages in service oriented application to enhance data integration. Despite practical challenges in data integration, the process of integrating big data sets in a low bandwidth environment requires mapping rules that deploy mechanism of mapping from heterogeneous to homogenous and localizing the execution operation.

Keywords: *Integration, E-Health, Healthcare, Heterogeneous, Web Service.*

1. INTRODUCTION

Electronic Health Records System is transforming primary health care and improves quality of health facility management [1]. Furthermore, it enhances immediate monitoring and evaluation of health facilities. Progress in health research is needed to enable the introduction of dramatic changes in the organization of health and social care. In fact e-Health solution could assist the move towards more sustainable care system and effectively further transfers care provision to the community, by supporting home care and reducing unnecessary hospitalizations [2]. However, in developing countries no much effort on building e-Health has been done. The problem may be associated with network infrastructure with low bandwidth. In many developing countries demographic Surveillance System (DSS) has been used for determining the cause of death for those countries with no vital registration system. Moreover, epidemiologists use such data set for disease modeling and decision makers use it for planning. Proper management of DSS data set therefore, increases the safety, quality and efficiency of routine practice in health facilities and healthcare as well as in medical care. However, data coherent requires consistent data structure and vocabularies and services that enable re-use of the data for multiple purposes. This implies harmonized data capture and processing that involve multiple locations, multiple data as well as multiple technology contexts [2]. Furthermore, integrating DSS data sets or longitudinal studies with those from cross section studies which are heterogeneous in nature imposes more challenges to system programmers working in low bandwidth environment.

It is argued in [3, 4] that an e-Health is growing up, changing how healthcare is managed in the world. A major concern imposed by e-Health systems is the integration and interoperability of the different medical application and services regarding design goal, developer's philosophy, ownership or specific medical domain [5]. The problem of heterogeneous in cross-system integration is due to lack of standards in e-Health platform [6]. New generation of Internet based health information system intends to meet the challenge of new economy concerning requirement for higher efficiency, efficacy and higher quality of care and welfare [4]. However, when composing web service health data integration, it is useful to analyze and compute overall operational properties of the organization in need [7-9]. This allows organization to translate the vision into other business process more efficiently. It is observed in [9] that web service constitutes an emerging technology for which potential applications are unlimited. Since web services can be designed according to operational matrixes [10] its implementation requires understanding of the infrastructure in place. Even though web service provides a good infrastructure layer, but integration demands a higher level broker's architecture layer [7]. It is depicted in [11] that many existing efforts on e-Health records mainly focus on hospitals or medical institutions within the organization and few directly oriented to patients. The strong demand in healthcare planning and current increase in health research that builds independent data repositories call for the creation of integration system architecture that creates heterogeneous data set from operational databases to the centralized data repository in order to provide service to individual needs.

It is claimed in [9] that while in many other applications domain the web service is deployed to integrated computer applications by different enterprises, its application in healthcare is rare and currently limited. It is emphasized in [12] that when dealing with data integration in mobile computing the issue such as low bandwidth, higher latency wireless network, loss of network connection and network failure need to be dealt with. Data integration framework proposed in many of medical care or health field researches, concentrate on naming, semantic, ontology, heterogeneity or specific domain of decease. Despite research that has been done in the field no prime solution has been achieved [6]. The web service framework discussed in [9] focused on prenatal, obstetrical and neonatal clinical decision support. The Web service architecture proposed in [13] pays attention on cross- system personalization and ontology based user model for personalization of information. The service oriented data architecture conversed in [7] addresses the issue of large scale data integration on demand, where the concept of software as a service and binding mechanism is the primary focus. Integration based on ontology and semantic research of medical data has been in study for long. Both general architecture proposed in [14] and in [6] for integrating heterogeneous databases is more concerned with semantic mediation approach based on ontology. Therefore, the Ontology Web Language (OWL) was the main concerned and the scenario is run using pathology of cancer database. To this end, it is noted that data integration architecture in healthcare integration addressing the issue of bandwidth using web service architecture, as well as heterogeneity of data set platforms is rarely anticipated in building software application.

Despite a higher increase in health research in Tanzania [15], it is noted in the Millennium Development Goals report [16] that data on many of health indicators are difficult to compile for measuring of MDGs; and in specific data from cross sectional studies. The reason behind is that data are collected but not harmonized for integration for ease of access. Web service and other technologies can greatly help in fulfilling this gape. The current District Health Information System (DHIS) implemented in Tanzania is based on the Open source MySQL platform. The existing National HMIS was built on PostGreSQL, Drug Management System was built on MS Access while some data set exist in on SQL Server, Fox Pro and CS Pro as well as excel at facilities level.

This paper describes an algorithm to integrate e-Health records which is heterogeneous using Web Service Model in a low bandwidth environment. The focus is on the problem of medium-scale data integration architecture where data source schema is uncertain at design time and may evolve data type mapping between the sources. Database relation algebra has been used with T-SQL to illustrate the processing techniques. The programming was done using Visual studio 2010 where programming language was Visual Basic .NET. Divide and conquer algorithm was applied to modularize the application programming interface. Separation of functions enhances the programming techniques. The design technique and the algorithm applied for mapping allow heterogeneous data set to be integrated in a low bandwidth environment. The Data Synch- Application Programming Interface (API) was developed to integrate health heterogeneous data repository using Web Service Model. The API is a higher level tool to integrate longitudinal data set of various platforms from uncertain health research data sets. In this regard, the paper presents a) architectural services for integrating heterogeneous data in a low bandwidth environment; b) describes mapping techniques that reduce memory storage for heterogeneous data exchange, and c) presents a novel software application for efficient integration and management of health heterogeneous records in an environment with higher network disconnection. The validation and test cases of the software architecture were done using Demographical Surveillance data sets (DSS) as well as baseline census data for sample vital registration with verbal autopsy. Different heterogeneous database platforms were configured and tested. Performance of data exchange from one data set to another of different database platform was conducted. Similarly, performance analysis of database engine was carried out.

Despite the existence of many middleware architecture, and health standards in health data integration [18], Web service is regarded as one of the most significant studies in heterogeneous data integration [4]. Web service is not necessary to be Internet based but also Intranet based or even standalone PC configured with web server. The web service provides information system setting and rendered restricted and secure access to the service of data required from heterogeneous sources. Nevertheless integrating health data repository still carries an element of risk and often requires extensive planning. But still technology offers the possibility to mediate among the different actors in order to build system functional services that meet specific needs of individuals and organizations as whole. How these systems developed and more particularly, how they are integrated with existing health information system is still an open issue [11, 19, 20] and standards are essential [21] to ensure interoperability among health systems. In addition, it is virtually certain that if such repository are to support e-Health records successfully across various health sectors, they must be able to integrate health data distributed across heterogeneous data sets. It is argued in this paper that lack of health data integration is a key impediments to a proper health data planning, healthcare management as well as timely evaluation and monitoring of health systems. In fact e-Health solutions could assist the move towards more sustainable care system and effective further transfer care provision to the community, by reducing unnecessary hospitalizations [2].

It is pinpointed in [22] that integrating health data provides many opportunities to health planners and epidemiologists for deceases modeling. In addition, it fastens the delivery, efficiency and effectiveness of the health care such as preventing health care, chronically diseases treatment and prevention, association analysis and population tracking. Furthermore, e-Health System is transforming primary healthcare and improves quality of health facility management [23]. It is emphasized in [24] that e-Health records enhance immediate monitoring and evaluation of health facilities providing better health care and reduce cost in data analysis. It is imperative that e-Health records in developed countries should be fully adopted by the year 2014 [25]. As cited in [5] according to "i2010" EU Policy framework, building information society requires extensive use of IT and computer technology in healthcare system in order to improve the quality of medical services, increase the system responsiveness, and reduce cost. In general, the integration of multiple information system aims at combining selected systems so that they form a unified view and give user the illusion of interacting with one single information system [26].

1.1 General Architecture of Web Service Technology

Web service is a GUI less web application. The term web service describes a standardized way of integrating web based application using the Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web service description Language (WSDL) and Universal Description, Discovery and Integration UDDI open standards over the internet protocol backbone [27, 28]. The XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the service available and UDDI is used for listening what services are available [29]. The general architecture of web service is described in Figure 1 below.

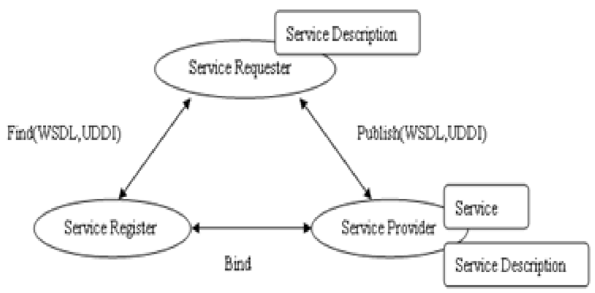

Fig 1: Web Service Architecture adopted from (Liu & Yuefan, 2009)

There are basically three major roles with the web service architecture. These are service provider, service requester, and service register. Even though there are more advantages of using web service, there are some challenges. These include issues such as web service uses plain text protocol that uses a fairly verbose method to identify data. The Hypertext Transfer Protocol (HTTP) and HTTPs which are used by web service were not meant for long term sessions like those in CORBA [17] where connection stays longer. On the other hand, HTTP and HTTPS are stateless, i.e., the interaction between the server and client is typically brief and when there is no data being exchanged the server and client have no knowledge of each other [30]. Our architecture is therefore built with understanding of those challenges.

1.2 The Concept of Heterogonous Distributed Environment

The term heterogeneity is a concept that relates to the non uniformity in data sources. Data experience different naming, relation structure, values, semantics, data model, syntax transaction and security [7]. These results are due to autonomists of data source, i.e., data source exhibits design and implementation autonomy, communication autonomy or execution autonomy. The distributed nature is due to the fact that individual data source are distributed across different organizations or sites rather than situated on the same host. The distributed system introduces new problems due to physical separation and heterogeneity of their component as well as multiple control administration. Analysis of such data sources required integrated unified view of the data. With the development of .NET Framework, the web service has become dominant in windows platform in data integration [31].The most important aspect of .NET framework is acceptance of open industrial standards that is the XML [10]. Even though it sounds simple, data integration has countless challenges in the technical implementation of integrating data from disparate sources. The main challenges lie on the entire data mapping. This problem is solved by developing an API that defines mapping libraries across multiple data sets.

2. METHODOLOGY

The Use Case Model was applied to identify the entities and class diagram. Post collected data were used to test data transfer between one network domain and another. Various database platforms were used to prototype the system. Testing was done using the same computer specification (processor speed, RAM) while varying the speed/bandwidth that is, use of physical cable or wireless modem was also used for testing. The performance of the algorithm for data integration and transfer was carried out between one platform and another and the time taken to transfer data across the internet (wireless) and across varying database engines was observed. The amount of data transferred in a time frame was also noted. The API was developed using Microsoft Visual Studio 2010 and the Visual Basic.NET was the main programming language.

The VB.Net runs on .NET framework which actually is based on ActiveX control. The ActiveX is one part of Microsoft object oriented programming interaction technology that is built on the Components Object Model [17; 18]. The ActiveX control uses the COM interface and runs in an ActiveX environment called containers on the same computer or in a distributed network consisting of windows a distributed network consisting of windows [17, 18].

A class module was developed so that to enable linked server on fly. The module is used to map any data that uses Microsoft Data Access library such as ADO, OLEDB, RDO, etc. The use of store procedures was avoided because the data sources were uncertain during the design and if exist they have different design philosophy. The techniques deployed were the use of web service with SOAP protocol at the data transfer level and use of Linked server at the execution level. The linked server was used to map the data type and allows insert and update operation of T-SQL. The techniques was to read data directly to the data table and change them to bytes then insert the byte into an xml file then compress and transfer them over SOAP protocol to the Web Services. This technique allows the service provider of the web service to transfer data while the execution and processing is done without knowledge of the provider.

In order to measure performance of each database platform four database engines were set up as an OPD and one as SDB. A set of rigorous performance evaluation using MySQL, Fox Pro, MS Access, SQL Server Compact edition (SS Ce) and SQL Server 2008 R2 were conducted. The source database originated from demographic surveillance in which five relations was chosen: i) individual relation had 133,004 tuples with 13 attributes and size of 26.04 MB, ii) location relation had 32,073 tuples with 17 attributes and size of 16.25 MB, iii) observation relation had 173,466 tuples with 10 attributes and size of 13 MB, iv) outmigration had 26,219 tuples with 23 attributes, and 12 MB, v) social group had 32,151 tuples with 8 attributes and size 4.50 MB. The choice of the relation was based on the existence of variety of data types and size. Data type date was considered most followed by long text characters. The source data were obtained from Fox Pro and transferred to the SQL Server in the staging database. From the staging database data were transferred to back to the client application. The importing of data from staging database to the local machine was not the interest of the simulation and therefore it was not recorded. Each engine was provided by the same size of megabytes of data. Data was therefore executed to the staging databases one table at a time and start and end time of transferring and execution for each a table was written in log file.

3. FINDING AND EXPERIMENTAL RESULT

A total of five heterogeneous databases engine applications were configured; i.e., MYSQL, Fox Pro MS Access, SS Ce and SQL Server. The goal of this architecture is to exchange health information between operational databases (OPD) and the staging database (SDB). The OPD are heterogeneous in nature because they contain data from different projects developed from different database platforms with different philosophy. The requirement of the OPD to SDB is to create intermediary data format that is acceptable and understood to all health research scientists.

The architecture comprised of three layers: i) The Source layer, ii) Mapping layer, and iii) Data view layer.

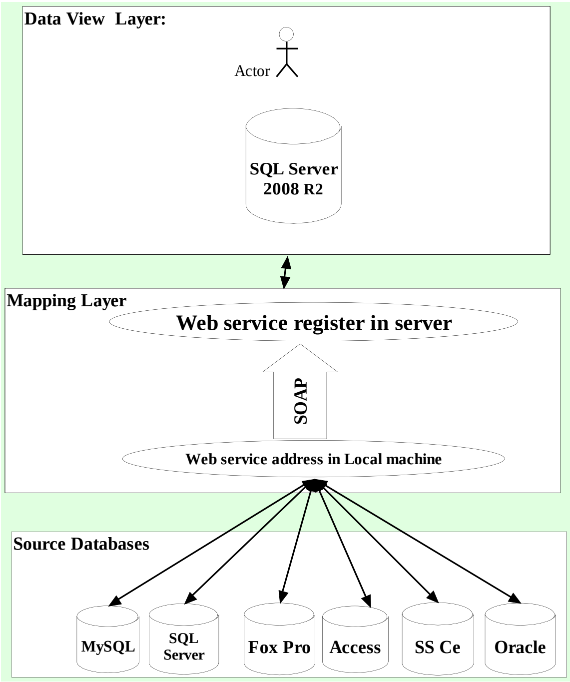

Fig 2: The Proposed Architecture for Heterogeneous Data Integration

The functionality of each layer is described as follows:

The Source layer:

This layer contains a set of heterogeneous data set built on different database platform. These data sets are commonly known as operational databases. They have different data model, and different schemas of data independent of the staging database. Because they are designed by independent database developers to meet different requirements they experience schematic heterogeneity and autonomist. For example, DSS data are stored in Fox Pro or CS Pro, SAVVY data are stored in SS Ce or SQL Server, INNES data set and IPTI data set are stored in Access and the current DHIS data set are store in MySQL. The OLEDB driver was used to act as a wrapper to connect the data source layer to the mapping layer. The wrapper provides the means to connect to and query data source and then loads onto the mapping layer.

Mapping layer:

The mapping layer is responsible for translation of the local schemas of the OPD to global schema of the SDB. The layer contains mapping libraries that translate the local schema data types and data size to the global schema. The mapping layer also ensures that the indexes of the OPD and its constraint are translated and enforced to the schema of the SDB. This is allowing the API to enforce integrity rules and constrained during schema creation. As discussed in [32] among the significant challenges is that the global schema in this case SDB is a representation of the domain of interest of data system. Integrity constraint is expressed on such schema to enhance its expressiveness thus improving its capability of representing the real world.

Vol. 5, No. 3 March 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

Data view layer:

This is an implementation layer where data is accessed for analysis. The layer provides user with a single view of the local operational databases being merged in the global schema of the staging database and into a single platform. If the mapping layer provided a correct schema, data types, data size and constraints, no loss of information is experienced in the staging database. The data view layer provides common schema in the single platform, in this case SQL Server 2008 R2. This will allow research scientists and epidemiologists to direct queries to the single schema. The integration using the global view at the SDB has the advantage of returning the widest set of relevant information. It also fastens the compatibility of data set so long as the questionnaires used to collect data have common structure and understandability. Transaction between the local sources to the staging database and vise visa is user events based. This approach ensures that data going to the staging database are cleaned and relevant for analysis. More cleaning may be required to the staging database before they are archived to the higher centralized database.

3.1 The Novel Algorithm to Transfer Data in Low Bandwidth Area

The general algorithm of integrating database is presented in Figure 3 below. The assumption made in the algorithm is that there is no such database in the staging database and therefore, modules which check if database exists runs first. If it detects that there is a database then it loops through database catalog and checks for table. Once all checks are done then the insertion and update operation is performed.

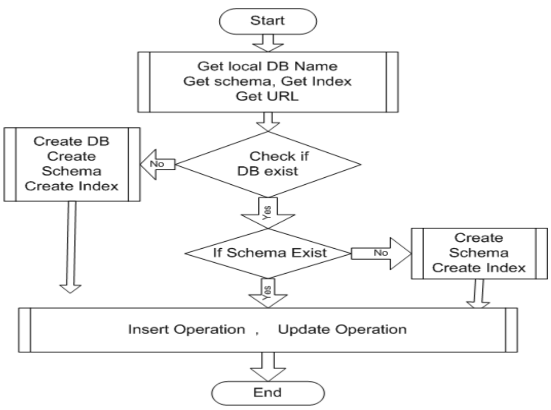

Fig 3: An Algorithm for Transferring New Operation Database to a Staging Database

The details of each part of the algorithm above is explained separately. The algorithm that performs Insertion and Update operation is presented as follows:

Let **L** have database table $\mathbf{T_L}$ at the operational database and **S** has a database table $\mathbf{T_S}$ at the staging database. Assume that the schema of $\mathbf{T_S}$ was created using the schema of $\mathbf{T_L}$ so that $\mathbf{T_L} \models \mathbf{T_S}$ at a certain time t. If both tables have the specific attribute A in their schema and constraint **C** then the attribute **A** takes its value from a given set **V**. Let $\mathbf{V_L}$ be the set of values that occur in $\mathbf{T_L}$ * A and let $\mathbf{V_S}$ be the set of values occurring in $\mathbf{T_S}$ * A. Therefore, for each inner join transaction of ($T_L \bowtie T_S$) there are some values **V** such that $\exists v_L : v \notin V_S$. These values are the ones that need the **INSERT** Operation; the rest require the **UPDATE** operation. The algorithms that filter these values from both tables run as follows:

a. Create Linked Server (SQL Connection String, DB Location):This method creates a linked server of the local database with the staging database so that a T-SQL can run on both tables.
b. For each table selected in (T_L)
c. Get Tempdata (T_i): The query runs to produce temporary table:$\mathbf{T_1 \longleftarrow \pi\{T_L \ (A_1,..,A_n)(}\sigma_c\mathbf{(\ T_L)\}}$. This gets selected column of the schema $\mathbf{T_L}$ in this case $c = \Phi$ because the aim is to select all data from source to destination
d. GetTemp2data(): The next query runs projection over equijoin to produce: $\mathbf{T_2 \longleftarrow \pi \ \{T_2 \ (A_1,\ldots,A_n)(T_L \bowtie T_S)\}}$This join operation will result on values V such that $\forall v : v \in V_S$. These are the values that need update on $\mathbf{T_S}$ and therefore allow us to perform the update operation.
e. If there are some values Von T_2 such that $v_2 \neq \Phi$ then
f. UPDATE <Table> SET <UPDATECULUMNS> FROM <Table> INNER JOIN #TEMP2 ON <INNER2JOINCOLUMNS> End if
g. Get Different Data():The next query Q runs to generate the difference of $\mathbf{T_1}$and$\mathbf{T_2}$, using the equijoin expression that is,$\mathbf{T_1 \longleftarrow \pi \ \{T_1(A_1,\ldots A_n)\} - \pi \ \{T_2(A_1,\ldots A_n)(T_1 \bowtie T_2)\} \equiv T_1 \leftarrow T_1 - (T_L \bowtie T_S \ )}$. These are the values that need to be inserted into Table T_S and therefore we perform T-SQL for Insertion
h. If there are some values **V** of $\mathbf{T_1}$ such that $v_1 \neq \Phi$ then
i. (INSERT INTO <Table> SELECT * FROM #TEMP1) // do insert operation End if
j. Loop for next Table, i.e., ($\mathbf{T_{i+1}}$)
k. Getclean Temp(): DELETE FROM #TEMP1 WHERE <WHEREIN>// This method clean all instances of the Temp file location

Analysis:

At line 3 a table schema $\mathbf{T_1}$ and its data is created and appended into a temporary file as an xml file. This is done by projection over selection. A duplicate of the Table $\mathbf{T_1}$ is created at line 4 to produce $\mathbf{T_2}$ by creating a join between source table $\mathbf{T_L}$ and destination table $\mathbf{T_S}$. This will result into tuples of $\mathbf{T_S}$ that matches tuples of $\mathbf{T_L}$. If $\mathbf{T_2}$ has records then we are assure that there are data that need update. At Line 7 we overwrite $\mathbf{T_1}$ by getting the difference of $\mathbf{T_1}$- $\mathbf{T_2}$; this result in tuples that need the Insert operation. The Insert operation is done by T-SQL at line 9. The loop continues to get the next T_{i+1} in the list. Finally, the

Vol. 5, No. 3 March 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

algorithm clears all temporary files at line 11 and terminates. The operation starting from line 4 is done on the Server side freeing the Client machine to perform other service and hence reduces the chance of Internet disconnection. It is apparent from this analysis that the primary key must be enforced on each table T_i because of the equijoin attributes.

The above algorithm executes fairly in a low bandwidth environment. In a higher bandwidth area it would be enough to open two connections at once and one can loop over the record to update and insert the data at the central server using procedure or non-query command**.** However, that approach requires reliable bandwidth and it is not cost effective in terms of SQL execution. Looping over the records could lead to having $O(n^2)$ looping instead it is reduced to (2On) by means of natural join operation during execution. The performance analysis of each engine in executing the above algorithm is explained in the next section and Query template that performs the overall operation is presented in Figure 4.

```
Private SynchQuery As String
  synchQuery = "SELECT * INTO #TEMP1_
      & " FROM <LinkedServer>...<Table> " & vbNewLine _
      & " SELECT <Columns> INTO #TEMP2 _
      & FROM #TEMP1 INNER JOIN <Table> " & vbNewLine _
      & " ON <INNERJOINCOLUMNS> " & vbNewLine & _
      " DELETE FROM #TEMP1 WHERE <WHEREIN> " & vbNewLine & _
      " INSERT INTO <Table> SELECT * FROM #TEMP1 " & vbNewLine & _
      " UPDATE <Table> SET <UPDATECULUMNS> _
      & "FROM <Table> INNER JOIN #TEMP2" & vbNewLine & _
      "ON <INNER2JOINCOLUMNS>" & vbNewLine
```

Fig 4: Query Template

3.2 Performance Analysis of Each Engine in Data Integration

Although the Black Box test was not carried out within the health sector due to lack of VPN network but a white Box test was carried out with test case of DSS data set. A test case normally consists of unique identifiers, requirements reference from a design specification, precondition, and series of steps to follow. The input and output expected are noted down. There are numerous types of system test that can be done as part of system testing and delivery process. Likewise, the testing has purposes therefore testing engineers are more relevant for Black box testing, functional test and comparison test. Other tests such as alpha test, branching/logical testing may be performed by the developer. Even though performance analysis test may be carried by developer but its result may be biased.

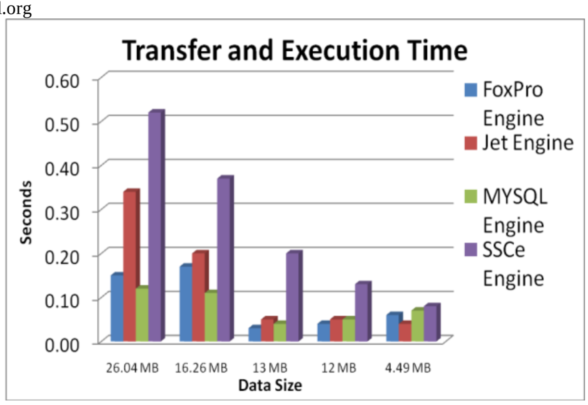

Fig 5: Transfer and execution of Various Database Engine

From Figure 5 the result indicates that MySQL engine was faster than others followed by Fox pro engineer then Jet engine. SS Ce engine was found to take long time. The long time taken by SS Ce is associated with the reasons that there is no direct communication between SS Ce and SQL Server and therefore transferring data between the two requires that data be transformed to another platform say Access before being transferred and executed to the server. From this result we argued that it is much better to collect data with MySQL open source database.

3.3 Mapping Technique to Reduce Memory Storage of Heterogeneous File

Using the idea presented in [33]a homogenous database presents 1:1 correspondence between an implementation level tables, tuples and data records while a heterogeneous database does not. Given the collection of relations$\{\mathbf{T_1}, \mathbf{T_2}, \ldots, \mathbf{T_n}\}$of an Relational Database Management implementation–level, the most natural mapping to data files is the one-to-one mapping to the collection$\{\mathbf{F_1}, \mathbf{F_2}, \ldots, \mathbf{F_n}\}$ of homogeneous data files. Each $\mathbf{F_i}$ is homogenous in the sense that all the records in $\mathbf{F_i}$ are of a single type, matching the schema of table $\mathbf{T_i}$. In certain contexts, however, we consider mapping two or more tables to a single data file F. In this case, F is said to be a heterogeneous file since it contains records of more than one type.

While heterogeneous data set do incur some overhead, in certain situations their benefits are still great enough to justify their use. Suppose for example, that the normalization process has let us to decompose interface–level table **T** into implementation–level tables

$\{\mathbf{T_1}, \mathbf{T_2}, \ldots, \mathbf{T_k}\}$. It is likely that the evaluation of many user queries will require the evaluation of the JOIN $\{\mathbf{T_1} \bowtie \mathbf{T_2} \bowtie \ldots \bowtie \mathbf{T_k}\}$. One way to support this join is to map the tables $\{\mathbf{T_1}, \mathbf{T2}, \ldots, \mathbf{T_k}\}$ to a single heterogeneous data file. The strategy is particularly appropriate if the following condition is satisfied:-
$\{\mathbf{k > 2}\}$

For some numbering of the tables in the decomposition, T_i is in 1: n relationship with $\mathbf{T_{i+1}}$, for $\{\mathbf{1 \le i \le k-1}\}$. That is, the operand tables can be numbered so that the set of attributes common to T_i and $\mathbf{T_{i+1}}$ is a super key of T_i for $\{1 \leq i \leq k-1\}$.

With heterogeneous files, however, a record of a particular type can be inserted only into a slot that is of sufficient size. Consequently, if the mix of records type presented in a file should change over time, there is the potential that the file will be left with many empty slots that cannot accommodate new records because the new records are too large for these slots. In this case, errors may occur during insertion operation otherwise if enforced by API then some information is lost. On the other hand, if we insert a record into a slot that is smaller than the existing slot, we may be left with small pieces of the old slot that is unusable in the future. To address such problem, it is necessary to employ memory management technique that reconfigures the records within their block from time to time. However, because our interest is to fit in the record; it is of great consideration that a mapping library be defined on the API to carry both record type and record size and then map the source and destination data types. This problem is experienced when down casting the application. The mapping algorithm was therefore developed to handle down casting from SQL Server engine to Fox Pro engine.

3.4 Functionality of the Data synch for Transferring Data in a Low Bandwidth

For deployment, a Service Oriented Architecture (SOA) was considered. The loose coupling principle which would potentially improve the reusability, adaptability and extendibility of the software based on the changing business was also considered. The service interface selection option is programmed with node to leaf node selector allowing a drop down of database options. The data interface interaction system has three basic functional services as shown in Figure 7 below:-

a. Entity Selector: This is drop down list of items in a source database allowing an actor to choose elements to be acted upon.
b. Export: This service allows an actor to push data items selected from a local machine to the staging database
c. Import: This service allows an actor to pull data items selected from the staging database to the local machine
d. Create schema: This service allows an actor to create schema of the selected data base elements to the local machine or to the staging database in the server.

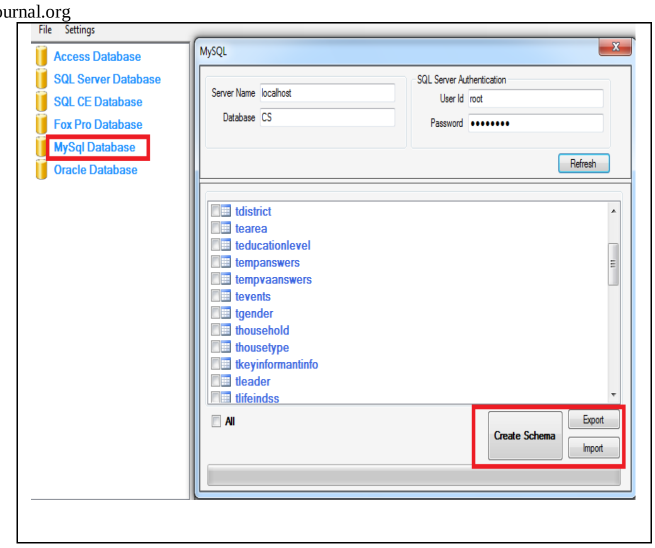

Fig 6: Integration for Database that require Server address specification

4. GENERAL DISCUSSION

4.1 Mapping

The main challenge in data integration is the mapping of the data type and size. In order to resolve the problem of translation of data type the Linked server over other applications was adopted. The attempt to use XML as a translation tool has some drawbacks that one needs to define a library that maps data type from one database to another. The looping through each node of data item in the XML schema is also a drawback. Unfortunately, there are some data types that don't find corresponding type on some database. For example "Enum or Set" in MySQL does not find correspondent data in access or Fox Pro. Data defined in SQL Server as text (maximum=400) will create loss of information when down casting to Ms Access or FoxPro with maximum of 256 sizes. Oracle and MySQL requires specific ODBC drivers and therefore must be downloaded and installed separately. Since data sources are in general autonomous, the data provided by the sources are likely not to satisfy the constraints on the global schema integration schema. Therefore, integration constraint has to be taken into account during the schema creation otherwise the system may return incorrect answers to the user in query processing that needs join operation. Another significant concern is that the source may not provide exactly the data that satisfy the corresponding portion of the global schema [32]. In particular they may provide either a subset or superset of the global schema and the mapping is to be considered sound or complete and not exact. Furthermore, new sources of data are gradually emerging in the health and social welfare sector. Therefore, any Information System developed should be capable of combining these new sets with principle data set available but not yet fully exploited. More research may be required on how to provide a solution that yield a coherent, shared information base for each individual needs.

4.2 Security concern in the Data Synchronization

Health data are very sensitive and therefore implicit or explicit details can strongly influence interpretation and action [2]. Data capture with an appropriate level of explicit details will be suitable for transformation and re-use. It follows that security becomes an important aspect in data integration. There are several levels of security and health information system implementers would be eager to enforce security at all levels. If the access control mechanism is too restricted then it could become a burden for a user who needs to access data. Likewise, if the access control mechanism is too loose then protection of information and resource cannot be guaranteed. Therefore, a precise access control mechanism is required that is able to deal with the level of unpredictability of typical healthcare application [34]. However, doing so would make the system unreliable and of less efficiency. In this case security service is provided at the data view layer requiring security aware application. Through that way, secure messages (secure objects) are exchanged providing end-to-end security also in an insecure environment. In the context of Internet based architecture as well as uncertain and insecurity in the current standard PC world, this security approach should be preferred because it ensures accountability and reliability. However, at the data view level security architecture’s access control combines discretionary access control (DAC), role based access control (RBAC), and policy based access control (PBAC) is mostly preferred. The detailed descriptions of these cases are beyond this paper.

4.3 Configuration Challenges

Configuration challenges include the accessibility to the local directories (LD) in which Temp directory of IIS can only be accessed by IIS user. If one uses windows authentication (connection string) of the database then there is no problem with access to the local directory or temp directory. If one uses SQL authentication then a problem arises that SQL server user could not access the temp directory. The solution is to define temp directory in such that SQL user and IIS user can access it and therefore instead of using common temp directory we create and place a folder in local drive where both can be accessed by IIS user and SQL authentication.

4.4 Software Testing

Among the important aspects of software development is software testing. Testing job seems to have unique characteristics of generating more work on doing work. For example, if a burg is identified and then fixed it should be re-tested to see if it does not repeat. Furthermore, it is claimed in [35] that software testing has paid little attention in offshore production. Many studies on software testing conclude that non-technical factors, such as social, technical and organization factors have a significant influence on the way software testing is performed in an organization and hence contributes to software failure or success. Software testing challenges should be seen as social-challenges [36]. Without doubt, design of software architecture is influenced by dependencies between individual requirement specifications [37].

4.5 Down casting of the Data Types

In object oriented programming down casting or type refinement is the cast of casting a reference of a base class to another of its derived class. Down casting is useful when the type of the value referenced by the parent variable is known and often is used when passing a value as parameter. Mapping of data type from higher powerful database engine such as SQL Server R2 to lower engine such as Access Jet engine or Fox Pro engine requires down casting. The text string maximum in SQL server is 400 in size while the text string maximum in jet engine is 256 while those in FoxPro has characters with maximum of 254 sizes. It follows that down casting of 400 to 256 size imposes some problem.

5. CONCLUSION AND FUTURE WORKS

It is concluded that despite practical challenges in data integration, integrating big data sets in a low bandwidth environment requires mapping rules that deploy mechanism of mapping from heterogeneous to homogenous and vice versa. The proposed architecture can be useful to provide scientists with timely data for modeling. The proposed algorithm can be used to unify and provides a broader population of data set as well as widening a set of health indicators to epidemiologists than it used to be in a traditional approach of requesting and integrating pieces of data using delimited techniques. Moreover, the harmonization of schema for heterogeneous data access requires a very skillful design that allows loose couple of health information system integration. Furthermore, 3NF decomposition is important in the process of database design. It is anticipated that the design architecture provide a novel web service infrastructure implementation of distributed data, and evaluating the web service paradigm in a low bandwidth environment. The algorithm applied can cut off the transfer and execution time to (2On) and hence it is cost effective in terms of time. It is therefore, capable of transferring data in a network with higher disconnection.

The result of evaluation performance demonstrates that the algorithm used in the architecture has huge potential execution time and cost saving under low bandwidth environment. Furthermore, the proposed architecture is expected to improve the quality of e-Health records in planning for healthcare management, reduce error in determining the burden of disease and guaranteeing the safety of records.

The architecture and the embedded algorithm may be useful for integrating e-Health records from different research group so that research scientists can access them at a single point. Even though down casting of data types is experienced in the design but the web service developed provide a possibility for individual to extract data from staging database back to the operational database with minimum effort.

The following are recommended for further works:-

a. There is a need to design a driver for SS Ce which up to the time of writing has not been attempted by Microsoft developers. This limits the linked server operations.
b. Due to federated nature of the e-Health records, security, privacy and patient's informed consent are key objective. Transferring data with SOAP protocol still give open challenges on security. Additionally, the complexity of state-of the-art security architecture can make integration of healthcare providers be expensive. Therefore, more research is required to provide secure and cost effective solution for healthcare integration using protocol that provides long time connection.
c. Due to the fact that new data sources are gradually emerging in health sectors especially on genetic data for personalized medicine more research may be required on how to provide solutions that yield a coherent, shared information base for each individual needs.
d. Currently and specifically in Tanzania, health care data are very fragmented in different hospitals or in health research institutions and not readily available for public use. Therefore, there is a higher need of research to interrelate social and health care effectively. So that all interconnected information resources should in principle be coherent across each area of responsibility.

REFERENCES

[1] M. Klompas, et al., "Integrating clinical practice and public health surveillance using electronic medical record systems," American journal of preventive medicine, vol. 42, pp. S154-S162, 2012.

[2] M. Angelo, Marta Mazzeo, Gregori Mercurio, Rita Verbicaro, "Holistic Health Predicting Our Data Future (from Inter-operability Among System to -Coperability Among People)," International Jounal of Medical Informatics vol. 82, pp. 14-28, 2013.

[3] M. A. Curran and K. E. Curran, "The E-Health Revolution: Competitive Options for Nurse Practitioners as Local Providers," Journal of the American Academy of Nurse Practitioners, vol. 17, pp. 495-498, 2005.

[4] B. Blobel, Analysis, Design, and Implementation of Secure and Interoperable Distributed Health Information Systems vol. 89: IOS Press, 2002.

[5] M. Rusu, et al., "Distributed e-health system with smart self-care units," in Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on, 2009, pp. 307-314.

[6] N. Arch-int and S. Arch-int, "Semantic information integration for electronic patient records using ontology and web services model," in Information Science and Applications (ICISA), 2011 International Conference on, 2011, pp. 1-7.

[7] F. Zhu, et al., "Dynamic data integration using web services," in Web Services, 2004. Proceedings. IEEE International Conference on, 2004, pp. 262-269.

[8] A. Tolk and S. Y. Diallo, "Model-based data engineering for web services," Internet Computing, IEEE, vol. 9, pp. 65-70, 2005.

[9] M. Frize, et al., "Towards a Web Services Infrastructure for Perinatal, Obstetrical, and Neonatal Clinical Decision Support," in Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE, 2004, pp. 3334-3337.

[10] S. Chandrasekaran, et al., "XML-based modeling and simulation: web service technologies and their synergy with simulation," in Proceedings of the 34th conference on Winter simulation: exploring new frontiers, 2002, pp. 606-615.

[11] P. Wang, et al., "Design and Implementation of a Web-Service-Based Public-Oriented Personalized Health Care Platform," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 43, NO. 4, JULY 2013, vol. 43, pp. 941-955, 2013.

[12] A. Meiappane, et al., "Latency of Web Service in Health Care System Using GSM Networks," in Machine Learning and Computing (ICMLC), 2010 Second International Conference on, 2010, pp. 22-26.

[13] F. Zhang, et al., "Web service based architecture and ontology based user model for cross-system personalization," in Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on, 2006, pp. 849-852.

[14] W. Attiaoui, et al., "Wrapping and Integrating USB Smart Card to Web Based Heterogeneous Medical Databases: Application for the Pathology of Cancer," in Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on, 2006, pp. 1403-1408.

[15] MoH, "Strengthen HMIS in Tanzania - Operational plan," Ministry of Health and Welfaire2010.

[16] URT, "MILLENIUM DEVELOPMENT GOALS REPORT: MID-WAY EVALUATION: 2000-2008: ," Dar Es Salaam2008.

[17] P. De Meo, et al., "Integration of the HL7 Standard in a Multiagent System to Support Personalized Access

Vol. 5, No. 3 March 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

http://www.cisjournal.org

to e-Health Services," Knowledge and Data Engineering, IEEE Transactions on, vol. 23, pp. 1244-1260, 2011.

[18] J. Grimson, et al., "A CORBA-based integration of distributed electronic healthcare records using the synapses approach," Information Technology in Biomedicine, IEEE Transactions on, vol. 2, pp. 124-138, 1998.

[19] A. Don de Savigny, T. (2009) System Thinking for Health system strengthening, Alliance for Health policy and system Reseach,.

[20] H. Mwanyika, Lubinski,D.,Anderson,R. Chester,K.,Makeme, M., Steel.M., and de Savigny.D., "Rational system Design for Health information system in Low-Income countries. An enterprise Architecture Aproach," Journal of Enterprise Architecture, vol. 7, 2011.

[21] S. H. El-Sappagh and S. El-Masri, "A Distributed Clinical Decision Support System Architecture," Journal of King Saud University-Computer and Information Sciences, 2013.

[22] D. C. Kaelber, et al., "A research agenda for personal health records (PHRs)," Journal of the American Medical Informatics Association, vol. 15, pp. 729-736, 2008.

[23] A. E. Sobel. (2012) The Move Toward Electronic Health Records. Computer. 22-23.

[24] D. Fridsma, "Electronic Health Records: The HHS Perspective," Computer, vol. 45, pp. 24-26, 2012.

[25] E. W. Ford, et al., "Predicting the adoption of electronic health records by physicians: when will health care be paperless?," Journal of the American Medical Informatics Association, vol. 13, pp. 106-112, 2006.

[26] P. Ziegler and K. R. Dittrich, "Data Integration—Problems, Approaches, and Perspectives," Conceptual Modelling in Information Systems Engineering, pp. 39-58, 2007.

[27] Y. Liu and L. Yuefan, "Research on Data Integration of Bioinformatics Database Based on Web Services," in The 1st International Conference on Networked Digital Technologies (NDT2009), 2009.

[28] Webopedia. (2013, Web service.http://www.webopedia.com/TERM/W/Web_Services.html. Available: http://www.webopedia.com/TERM/W/Web_Services.html

[29] C. Walker and D. Walker, "Integration and Data Sharing between WS-Based Workflows," in Web Services, 2008. ICWS'08. IEEE International Conference on, 2008, pp. 667-674.

[30] MSDN. (2009, 27 Oct.2013). Microsoft Developer Network. Available: http://social.msdn.microsoft.com/Forums/en-US/435f43a9-ee17-4700-8c9d-d9c3ba57b5ef/advantages-disadvantages-of-webservices?forum=asmxandxml

[31] A. Wigley, et al., Microsoft®. NET Compact Framework (Core Reference): O'Reilly Media, Inc., 2009.

[32] A. Calı, et al., "Query rewriting and answering under constraints in data integration systems," in Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003), 2003, pp. 16-21.

[33] P. Helman, The science of database management: Richard D. Irwin, Inc., 1994.

[34] G. Russello, et al., "A workflow-based access control framework for e-health applications," in Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on, 2008, pp. 111-120.

[35] H. Shah, et al., "Global software testing under deadline pressure: Vendor-side experiences," Information and Software Technology, vol. 56, pp. 6-19, 2014.

[36] J. Rooksby, et al., "Testing in the wild: The social and organisational dimensions of real world practice," Computer Supported Cooperative Work (CSCW), vol. 18, pp. 559-580, 2009.

[37] H. Zhang, et al., "Investigating dependencies in software requirements for change propagation analysis," Information and Software Technology, vol. 56, pp. 40-53, 2014.