CAMDIT: A Toolkit for Integrating Heterogeneous Data for enhanced Service Provisioning-ACCEPTED

CONFERENCE PAPER · NOVEMBER 2014

DOI: 10.13140/2.1.4058.1441

4 AUTHORS, INCLUDING:

Abayomi Ipadeola

University of Zululand

5 PUBLICATIONS **2** CITATIONS

CAMDIT: A Toolkit for Integrating Heterogeneous Data for enhanced Service Provisioning

[1] Ipadeola Abayomi, [2] Ipadeola Oladipupo, [3] Ahmed Ameen, [4]Sadiku Joseph

Department of Computer Science, University of Ilorin, Kwara State Nigeria.

[1]profyommexy@yahoo.ca, [2]ladiipadeola@yahoo.com, [3]aminamed@unilorin.edu.ng, [4]jssadiku@unilorin.edu.ng

ABSTRACT

Data Integration is classified as an Open and Lingering (OL) problem which must be sufficiently addressed due to its myriad benefits and significances, especially in the health sector where collaborative medicine is vital. Data integration problem evolved from the disparity in semantics and syntactic representations of medical data. It is a challenge, which must be solved to realize effective collaboration in the health sector and paramount for efficient health care service provisioning. In recent times, different approaches and software artifacts such as services, components and tools have been proposed for resolving data integration problem. However, existing approaches are faced with data inaccuracy, data unreliability, increased query response time, network bottleneck and poor system performance. The focus of this paper is to present our technique and the CAMDIT toolkit which efficiently achieves data integration, data accuracy, reliability and reduced query-response time and impacts of network bottleneck on systems performances.

I. INTRODUCTION

Data Integration refers to the problem of combining data residing at different sources, and providing the user with a unified view of these data [1].

Data Integration dates back to 1960 when a need arose to achieve data morphing from disparate sources otherwise called unified central data view. In essence, the rapid adoption of databases after the 1960s naturally led to the need to share or merge existing repositories. This merging can take place at several levels in the database architecture [2].

Data morphing or integration is a rising pursuit amongst researchers and industrialist as the need to share data explodes by the day. Wiki-pedia mentioned that the ambition of integrating disparate data sources is an essential focus of extensive theoretical work, and numerous open problems still appearing unsolved [3].

More recently, data integration is adopted in systems, applications, process developments and implementation environments for seamless service accessibility to customers in network security, application optimization, mobility services, high performance networks, IP telephony and open access networks for improved service delivery [4].

II. RELATED WORK

Three (3) prevailing approaches are adopted for integrating heterogeneous databases, and they include the Data-warehouse approach, Middleware approach and the Federated approach [5]. All of these approaches have sets of unique advantages and disadvantages in data integration and morphing.

The first is the data warehouse approach, which involves the design of centralized local repository for storing data from different data sources. Data warehouse method is based on data-translation, such that, data from different sources are modified to conform to the schema of the centralized repository prior integration [6]. Queries are executed on the warehouse rather than distributed sources of data. This approach eliminates network bottlenecks, slow-response time and the challenge posed by the unavailability of primary sources. Data warehousing promotes improved query optimization and allows for data validation, correction filtering and annotation by end-users [7].

However, the data warehousing approach suffers from data unreliability, since the centralized repository called warehouse is disconnected from its original primary data sources. In addition, data warehouse is faced with incessant schema modifications, which has adverse effect on warehouse availability [8].

The second approach is considered the most common data integration approach. It is referred to as the middleware approach and has a mediator positioned between a client (query interface) and disparate data sources [9].

Whenever a query is provided by a client, the mediator determines the concerned sources to be contacted for response to the query and decomposes the query into each required data source. Software entities called wrappers are used as query and result translators [8].

They translate both the sub queries into source-specific query language and query-output back into the common query language. The mediator combines results from the wrappers for display to client. The mediation approach suffers from network bottlenecks while incessant query translation leads to performance quandaries [10].

The third approach is referred to as the federated database approach and it differs from other two approaches due to its ability to realize cross-reference linking between disparate data sources during data integration [7]. The approach is based on a decentralized architecture, whereby one-to-one connection is implemented between all pairs of database. In the federated approach method, wrappers and mediators are utilized in the translation of disparate schemas prior integration.

Similar to the middleware approach, federated database approach also requires query translation, which is resource consuming.

III. PROPOSED TECHNIQUE: Context Aware Data Integration Technique

We propose the adoption of software agents, context-aware middleware and semantic metadata representations for addressing the data integration problem "Fig1". The technique is termed Context Aware Model (CAM) for Data Integration.

In this technique, we associated dedicated software agent with each disparate data source. Such front agents were provided with intelligent mechanisms for managing data sources and communicating with the middleware prior, during and after data integration.

Software agents performed roles such as data source schema definition, query translation, data source ontology management, which considerably improved systems performance. In order to reduce the query processing time, an ontology-based query translation mechanism was used by the software agents for translating queries for each data source. Such translation mechanisms were supported with trained data source ontology and schema information using neural network for deducing a data-source specific query. The translation mechanism is a case base reasoning method based on prior query translation output for any given translation. This approach considerably reduced the query processing time. Our output informed that this translation approach is much more efficient than the conventional approach, where the mediator performs query translation for all available data sources. Incessant schema modification of the middleware that causes performance quandaries is abated as division of labor mechanism is encouraged by allowing dedicated software agents to handle schema modification.

Additionally, all user queries were executed against a data source ontology generated by a software agent (Resource Discovery Framework (RDF) repository) as against conventional approach of direct querying of data sources. Querying a data source ontology representation is considerably more efficient than direct data source querying.

This work in addition to an RDF repository, created an object relational database metadata for every data source, such that every object or data in a known data source is associated with metadata information and stored in an object relational database. In this work, we adopted conventional medical dictionaries such as Unified Medical Language System and SNOMED for generating meta-data information for each data in the data source. This gives an alternative and achieves improved level of accuracy during integration. Meta-data repository constitutes a richer resource to answering user queries. This technique proposes a meta-data representation of user-queries prior execution.

In this work, we utilized semantic data representation during integration to achieve data accuracy. Software agents intelligently update all data source ontology representations and object relational data bases storing object meta-data during an integration exercise. Data source ontology were generated and updated based on set of trained data using neural networks.

Software Agents were notified by a trigger function whenever an update is made to underlying data sources. Such update are classified accordingly and inserted into appropriate section of the concerned data source ontology representation.

The utilization of context-aware middleware was highly important for managing context of software agents and their associated data sources and consequently in decision making for any integration task. In addition, software agents and the middleware provided information concerning data sources that must be warehoused for improved query performance. Although the warehouse approach is known to face data unreliability challenge, a warehouse is jointly managed by software agents that are associated with its original data sources.

Further to improving the level of data accuracy, we defined mechanisms for the management of real-time data. In this regard, all real-time data updates to data

sources were classified and allocated meta-data representation for updating both the ontology and object relational data base.

1. The Data Source Specification Phase
2. The Agents and Context Management Phase
3. The Query Management Phase and Integration output.

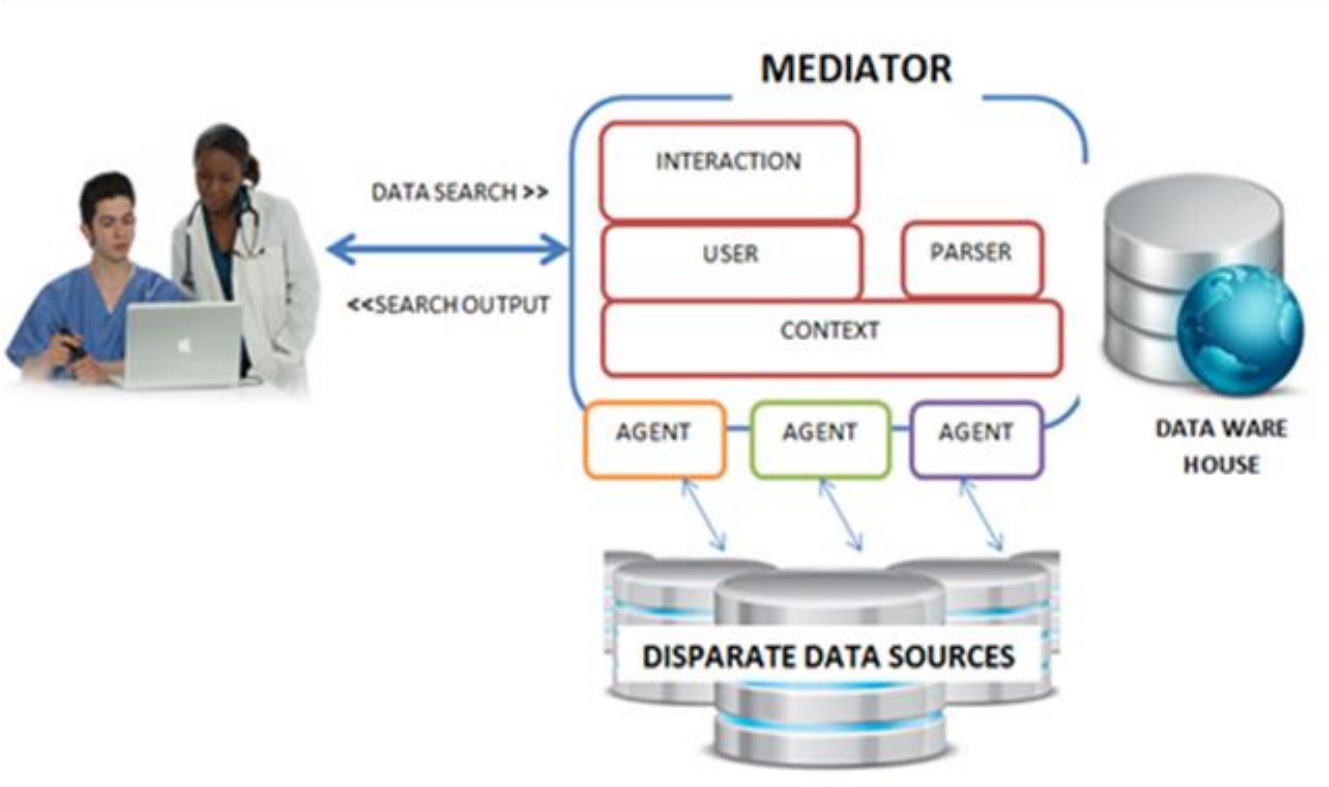

FIG1 Toolkit: Proposed Context Aware Data Integration Technique

IV. CAMDIT IMPLEMENTATION

The integration interface called toolkit was implemented for the integration of databases based on proposed Context Aware Data Integration Model. CAMDIT was implemented in Java and C# and it adopts the look and feel of most Integrated Development Environment (IDE). Query Interface and Data-source selection sections are integrated with CAMDIT for design-time specifications of data-sources. CAMDIT supports multiple and disparate data source types, through the configuration File, these datasources include, ORACLE, MySQL, PostGRes, Db2, SQLSERVER, Flat File, Microsoft Access, etc. The CAMDIT toolkit dynamically supports new data-source types by simply an addition of Key (Name of Data Source Type) and Value (Linkage Information) Parameter on the CAMDIT configuration. The CAMDIT Environment has three important phases, namely,

The various phases in the proposed CAM are represented differently on CAMDIT and discussed as follows:

- The Data source Specification Phase

In this section, the connection parameters of disparate databases or data stores are specified on CAMDIT. A user enters the name of the data source, provides the connection information such as Internet Protocol (IP) Address and the Login Credentials, which includes the username and password and clicks ‘ADD Data Source’. More data sources are specified by users and persisted on CAMDIT “Fig2”.

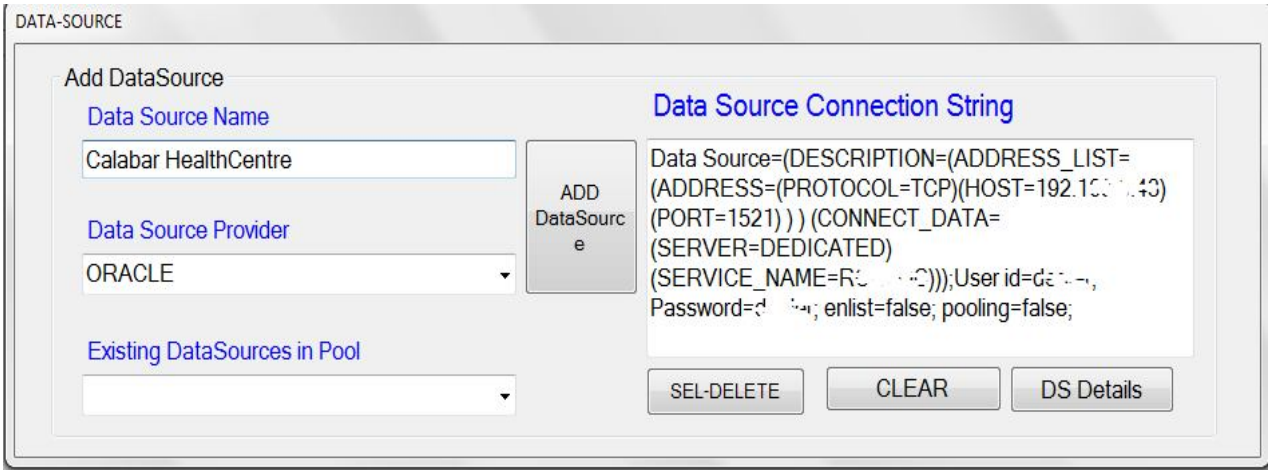

FIG2 Toolkit Data Source Specification showing Flexible Support for New Data sources

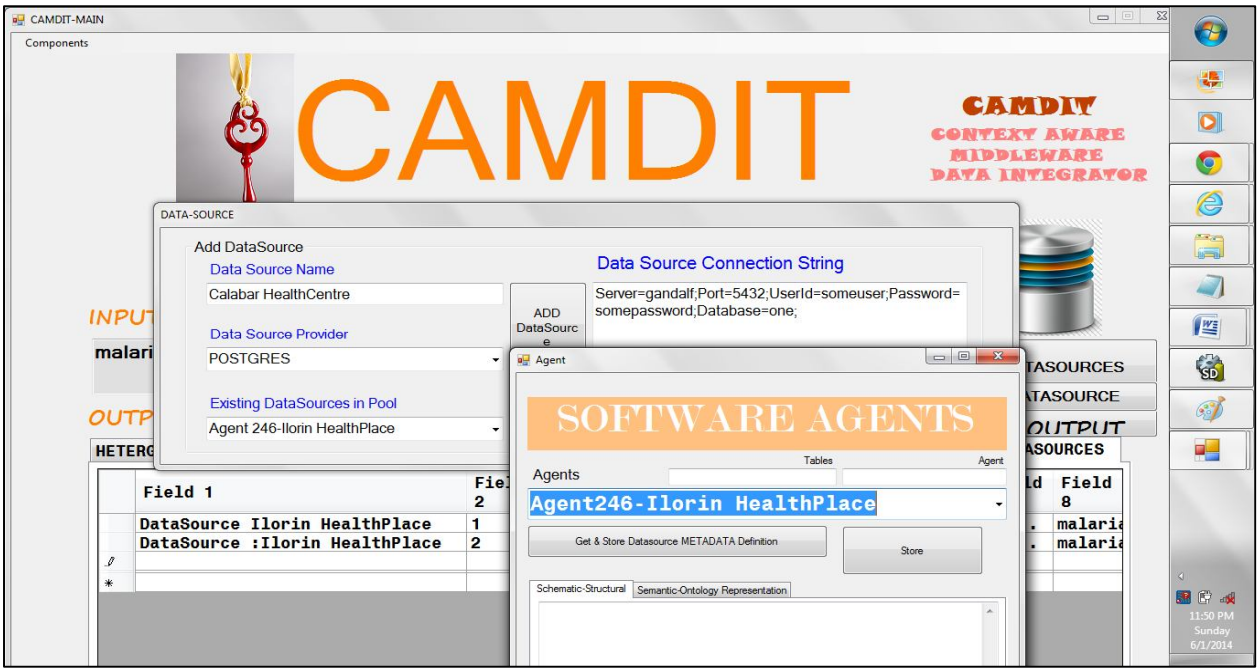

FIG3 Toolkit: Data-Source Configurations, Query Management and Context of Data sources by Agents

- Agent and Context Management Phase

This phase is significant to the realization of data accuracy in data integration. An agent is automatically created and profiled for every new data source specified in the previous section. Agents are enabled with intelligent mechanisms and tasked with the purpose of providing schematic, semantic and real time information from associated data sources.

The context information of every data source, which includes data source availability, schema definitions and metadata are stored in this phase.

- Query Management and Data Integration Output Result-set

User querying and string-searching are carried out in this section of the CAMDIT toolkit. In this phase, users specify search strings/text (Query Parameter) and the search is executed over context aware model of meta-data and ontological representations of data sources.

V. TOOLKIT EVALUATION RESULTS

The evaluation was carried out in a computer laboratory with 14 inches screen sized computer systems. Three categories of testers were considered for the evaluation,

namely the expert (E), the intermediate (I) and Novice (N) user interface designers. We considered sixty (90) testers in all, with thirty testers in each group, although Five (5) were proposed as adequate for usability [11].

The criteria for evaluating the CAMDIT Toolkit include:

- Support for Integrating Multiple and Disparate Data sources

"Fig 4" shows that 90%, 85%, 100% of expert, intermediate and novice Strongly Agreed that CAMDIT supports multiple and disparate data sources. Similarly, 8%, 10%, 0% of expert, intermediate and Novice Agreed that CAMDIT supports for integrating multiple and disparate data sources. 2%, 5%, 0% of expert, intermediate and novice could not decide, but provided additional comment information that they were software trainees and needed more training sessions to understudy the toolkit.

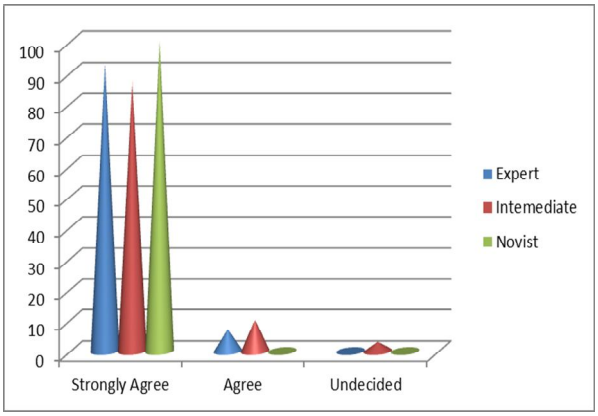

FIG4. Support for Integrating Multiple and Disparate Data Sources

- Generation of schematic and semantic representations of data Sources

"Fig 5" shows that 100%, 96%, 97% of expert, intermediate and novice Strongly Agreed that CAMDIT supports generation of schematic and semantic representations of data Sources.

Similarly, 0%, 3%, 3% of expert, intermediate and Novice agreed that CAMDIT generates the schematic and semantic representations of data Sources and the remaining, 0%, 1%, 0% of expert, intermediate and novice could not decide but provided additional comment information that they were software trainees and needed more training sessions to understudy the toolkit.

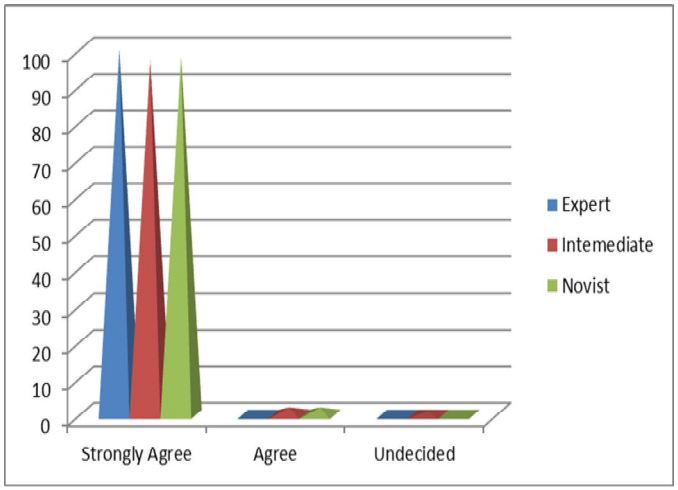

FIG5. Generation of Data source Schematic and Semantics

VI. SUMMARY AND CONCLUSION

This paper described a Context-Aware Data Integration Technique for integrating heterogeneous data sources. Our approach presented in this paper, proposes the use of software agents, context-aware middleware and semantic metadata representations for addressing the challenges posed by round-trips of query sessions, data source heterogeneities and data inaccuracy. Our technique is enabled with a prototype toolkit named CAMDIT. The evaluation results of CAMDIT Toolkit proved and informed of its support for multiple and disparate data sources and ability to integrate schematically and semantically different and heterogeneous data sources and providing users with unified data view.

REFERENCES

1. Jimison L., Adrien D., Patrick R., Stéphane S., Antoine G., Henning M: Design of a decentralized reusable research database architecture to support data acquisition in large research projects. In the Proceedings of Student Health Technology Information. (2007)

2. Feng B, Manfu M, Gou H, "An Architecture to integrate distributed information integration, ACM, Eighth International Conference Grid and Cooperative Computing, 45 – 49, (2009).

3. Wiki-pedia Contributor (June10, 2014), Data Integration [Online] Available Http://en.wikipedia.org/wiki/Data_integration.

4. Informatica Contributor (July 10, 2014) "Transforming Data Chaos into. Breakthrough Results. Expanding the Scope of Data Integration to Meet Emerging IT and Business Demands". Http://www.informatica.com/

5. Ibrahim A., Naomi. S, Database Integration: Importance and Approaches. Journal of Theoretical and Applied Information Technology Vol. 54 No.1 (2013).

6. Banek M., Tjoa A. M., Stolba N: Integrating Different Grain Levels in a Medical Data Warehouse Federation. In Proceedings of Data Warehousing and Knowledge Discovery, A. Min Tjoa, Juan Trujillo (Eds.), Krakow, Poland, LNCS, 4081, 185-94. (2006)

7. Pawel Jurczyk, Li Xiong: Towards privacy-preserving integration of distributed heterogeneous data. PIKM 65-72 (2008).

8. Stolba, N., Schanner A: eHealth Integrator - Clinical Data Integration in Lower Austria. 3rd International Conference on Computational Intelligence in Medicine and Healthcare, Plymouth. In third International Conference on Computational Intelligence in Medicine and Healthcare. (2007).

9. Critchlow, M., Ganesh, R., Terence C., Madhavan, G., Ron M: Automatic Generation of Warehouse Mediators Using an Ontology Engine, In Proceedings of the 5th International Workshop on Knowledge Representation Meets Databases. (1998).

10. Ran Z; Hai J; Qin Z, Yingshu L; Pan, C: Heterogeneous Medical Data Share and Integration on Grid: BioMedical Engineering and Informatics (BMEI). Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics – Vol 01. Issue 27-30. pp.905-909. (2008).

11. Nielsen, J. (2005). "Quantitative Studies: How Many Users to Test?" Jakob Nielsen's Alertbox June 26, 2006. Retrieved July 3, 2006 from useit.com: http://www.useit.com/alertbox/quantitative_testing.html