Quality Aware Service Oriented Ontology Based Data Integration

HEMA M S[1], CHANDRAMATHI S[2]
[1]Department of Computer Science and Engineering, [2]Dean, Electrical Sciences
[1]Kumaraguru College of Technology, [2]Hindustan College of Engineering and Technology
Coimbatore, Tamil Nadu
INDIA.
[1]ghema_shri@yahoo.co.in, [2]chandrasrajan@gmail.com

Abstract

Integration of multiple, distributed, and heterogeneous sources are essential for scientific and commercial domains. Ensuring data quality in data integration is an issue or challenge because of their varying quality levels. The existing data integration methodologies do not assure quality of data as it is difficult to assess. The Meta data of data sources do not provide quality details and it is difficult to choose best query plan. It is also difficult to predict the resultant data quality before integration. To mitigate above issues, this paper proposes Service Oriented Data Integration with Quality of Service (SODI-QoS) architecture. The SODI-QoS architecture has wrapper mediator layer, which consists of semantic conflict resolution layer and Quality of Service (QoS) layer. Semantic conflict resolution layer uses ontology to create local and global schema to resolve semantic conflicts. The QoS layer detects and resolves the incompleteness and inaccuracy of resultant data of data sources. The proposed architecture provides high-quality data results to the end-user and notification about the incompleteness and inaccuracy of the source is communicated to respective data sources. The E-shopping application has been proposed to analyze the performance of the SODI-QoS architecture. Experimental results illustrates that the accuracy and precision of SODI-QoS architecture has been improved by 12% 14% respectively than the ontology based data integration.

Keywords: Data integration, Quality of service, Semantic conflict, data completeness, source completeness, accuracy

1. Introduction

The need for accessing multiple, heterogeneous and distributed data sources are increasing for decision making applications that require comprehensive analysis and exploration of data. The data integration is solution for the above requirement. The data integration combines data residing in multiple heterogeneous data sources [22]. The three types of data integration methods are 1. Data consolidation 2. Data propagation and 3. Data federation [20]. The data federation provides a single virtual view for two or more data sources. The business applications issue a query against this virtual view to extract results. The virtual view creation among different data sources is still a challenging task due to their heterogeneities. Various types of heterogeneities are syntactical heterogeneity, data model heterogeneity and logical heterogeneity. The logical heterogeneities are further classified into semantic heterogeneity, schematic heterogeneity and structural heterogeneity [11][33]. Among these heterogeneities, the semantic heterogeneities are not resolved efficiently. The semantic heterogeneity is caused by different meaning or interpretation of data [13]. The structural semantic heterogeneities are naming conflict, identifier conflict, generalization conflict and aggregation conflict and data level semantic heterogeneities are unit conflict, representation conflict, value conflict and precision conflict [12]. The semantic heterogeneities are resolved by using ontology. Ontology is a formal explicit specification of conceptualization. Formal specification denotes machine readability with computational semantics. Explicit represents unambiguous terminological definition. Conceptualization indicate conceptual model of a domain [10][13].The uses of ontology in engineering is sharing of data or information and reusability of domain knowledge. The ontology is used to represent domain knowledge to resolve semantic heterogeneities in data integration. The ontology in data integration comprises of two components. They are names for important concepts in a domain and background knowledge or constraints on the domain such as attributes, classification and constraints.

The three ontology architectures for data integration are single ontology approach, multiple ontology approach and hybrid ontology approach [11]. The ontology is created by using Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), DARPA Markup Language (DAML) + Ontology Interchange Layer (OIL) and Ontology Web language (OWL). Among these, the OWL is more powerful than others. The OWL has well defined semantics and highly optimized implementation system.

The data quality is often defined as “fitness for use”. Data is fit for use whenever a user, (1) is able to get information, (2) is able to understand it, (3) finds it applicable to a specific domain and purpose of interest and (4) believes it to be credible.

The Key measures of data quality are data completeness, data consistency and data accuracy. Completeness is defined as the extent to which data are of sufficient granularity for the task at hand. Data consistency expresses the degree to which a set of data satisfies a set of integrity constraints. Data accuracy is defined as the closeness between the given value and the correct representation of the same in real life phenomenon.

The contribution of this paper is to implement service oriented data integration with quality of service and to illustrate the components and steps for building SODI-QoS architecture that assures the quality aspect such as semantic conflict resolution, completeness and accuracy of the result. The SODI-QoS architecture provides high-quality data results to the queries posted by the end-user to the integration system and also communicate incompleteness and inaccuracy of data to the respective data sources.

Besides the introduction section, there are five sections in this article, which are organized as follows. Section 2 describes related work in both data integration and ontology based data integration with QoS. SODI-QoS architecture is described in section 3, which includes ontology construction for data integration, query processing and quality improvement of the retrieved results. Section 4 describes the results and discussion. The conclusion is presented in section 5.

2. Related Work

Detailed surveys on ontology based data integration are found in [2][29][33] [34]. A comprehensive semantic search model is proposed in[23]. This synergizes the benefits of both keyword and semantic based search. Mediator wrapper architecture was implemented for ontology based data integration that abstracts the semantic complexity in mediator layer [3][10][35].

Methodologies to create global ontology via Local As View [LAV] or Global As View [GAV] were proposed and implemented in[11][12][16][39]. The shared vocabulary from local ontologies is created for generating global ontology is provided in [17]. An architecture called RCM is designed and implemented for mapping between local source and global ontology [37]. A method is proposed to compute similarities among various ontology specifications for ensuring reusability and accuracy [26][19]. An algorithm for ontology classification is implemented to classify the ontologies based on their domain in [4][7]. Efforts are also made to store ontology and database as separate entities [25]. An automated method for data migration from data intensive application to semantic web is developed ensuring interoperability between heterogeneous data sources [6][27]. An automated mapping between relational database and OWL ontology is implemented using mapping rule engine [28] [38]. Later ontologies are stored in relational databases for swift query processing [14] [18] [32]. The Object Relational Databases are employed to realize real time entities and mapping of ontology to ORDB is implemented in [5].An authoring tool is used to combine the intelligent techniques of assisting domain experts in constructing ontologies [31]. An algorithm is proposed to convert SPARQL query to SQL query [15]. It bridges the semantic gap between the expressive power of SPARQL and SQL. A conceptual model comprising data, service and process is used for defining mapping between different applications [30].High quality data sources are selected for data integration and prunes low quality data sources before integration. This approach creates query plans by exploring the Query Correspondence Assertions (QCAs), i.e., the cost to be paid for the query. A set of metadata features for source is defined. Source quality features include time stamp, availability and accuracy [9]. Based on minimum time stamp, availability and accuracy value in the metadata the result is processed [1]. A framework for dealing data quality in cooperative information system is implemented. This approach is to make cooperating organizations to export not only data that they intend to exchange with other organizations, but also metadata, which characterizes their quality level. Based on the quality characteristics of the data, user queries are processed [24].

A mediator system is proposed for source selection and query planning process. It ensures the completeness of the data [8]. An approach was proposed for automatic correction or editing of missing data and mutually contradictory data in very large databases [36]. An approach was proposed for data quality in data warehouse [21]

Existing methodologies improve quality of individual data sources and selects best quality data sources at the time of integration, improves quality based on choosing best query plan and metadata of data sources to provide quality integrated results. However, the problem of data quality is complex in data integration environment and data quality of each data source is not rich since they are autonomous and have a varying data quality. The data source does not provide metadata with quality to the integration system for making decision during data conflict. The extension of existing data model is costlier and not scalable. To ensure data quality of data sources, benchmark data set is required. The bench mark data is not available for all domains. Hence additional approaches are needed to ensure the quality of the data provided to the users.

To mitigate the above said problems, SODI-QoS architecture is proposed and implemented. SODI-QoS identifies the schema conflict at schema level, incompleteness and inaccuracy of the data returned by the data sources. Then, the schema conflict resolution, completeness and accuracy is implemented for the query and uses notification service to notify the data sources in case of quality lapse and provide quality results to the user.

3. Proposed System

The objectives of the proposed system are to provide quality results to end users and to improve the quality of data sources of data integration system by using SODI-QoS architecture. The SODI-QoS architecture is shown in Figure 1.

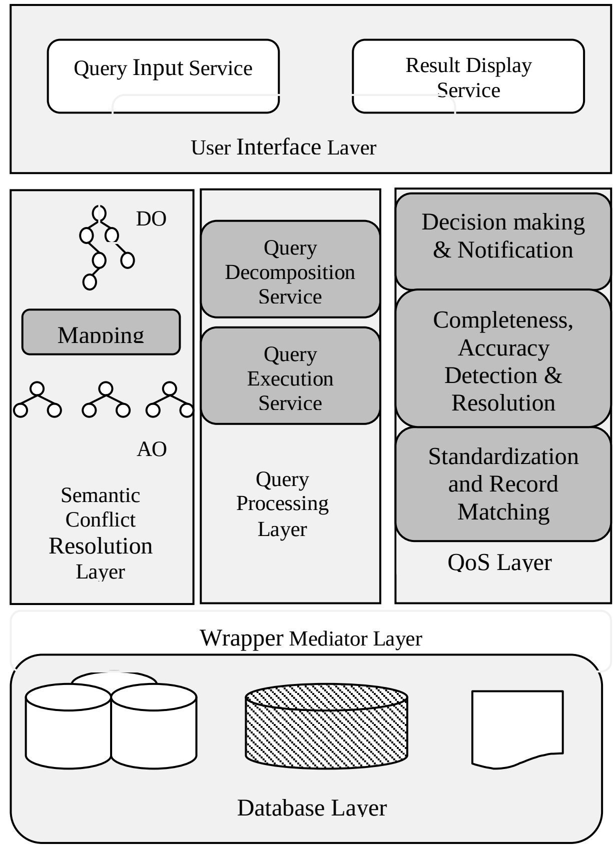

Fig 1: SODI-QoS Architecture

This architecture consists of three layers with mutually exclusive tasks. Data Source layer that is the bottom most layer contains heterogeneous data sources. Wrapper mediator layer is the middle layer that creates local and global ontology and maps them using ontology in the semantic conflict resolution (SCR) sub layer. This layer aids to retrieve the quality results from the heterogeneous data sources. This layer further is divided into local ontology service, which has been created from the local schema that is in the respective data sources. The autonomous development of data sources uses their own local ontology to represent their concept, attribute and the relation. Mapping rules are used to resolve the semantic conflicts among local ontologies for ontology based data integration. The main task of data integration is to provide a common view for the users to access data, regardless its actual organization and location. This is done by creating global ontology. The global ontology has been created by using Hybrid ontology approach [4]. The query processing (QP) layer executes the user query, which is discussed in section 3.1.

The Quality of Service (QoS) layer detects and resolves the incompleteness and inaccuracy of the retrieved results. Additionally, it notifies about incompleteness and inaccuracy of the data to the respective data sources. The components of the QoS layers are standardization and record matching service that is described in section 3.2.1. The incompleteness detection and resolution and inaccuracy detection and resolution are performed simultaneously that are described in section 3.2.1, 3.2.2 respectively. The decision making and notification service that are discussed in section 3.2.3 and 3.2.4 respectively. The top layer accepts user request for result extraction in query input service. The extracted results with QoS from QoS layer are used for decision support and analysis through result display service.

3.1 Query processing

The query processing is performed in query processing layer. In this layer, query is received from the user interface layer. The received query is posted against the global ontology.

Algorithm for query processing.

Input: Query (Q) in the form of SPARQL

Output: The results (r1, r2, r3,….., rn) from the respective local data sources.

Algorithm

Step1: The query (Q) is posted against the global ontology.

Step2: The query (Q) is decomposed into in sub queries (q1,q2,q3, …..,qn) based on the mapping rules. (Q= q1,q2,….qn)

Step3: The sub queries (q1,q2 , …..qn) are passed to respective local ontologies.

Step4: The query is converted into native database query by using wrapper program and sends to respective data sources for result extraction.

Step4: The results (r1, r2, r3,…,rn) are extracted from the respective data sources and passed to QoS layer for quality improvement.

3.2 Quality driven result integration

It is used to improve the quality for retrieved result. The data is retrieved from heterogeneous data sources. This may be poor in quality. The steps involved to improve the quality of the result are standardization and record linkage method, incompleteness detection and resolution, inaccuracy detection and resolution and decision making and notification.

3.2.1 Standardization and Record linkage method

The standardization process is essential for integrating heterogeneous databases to improve quality of the data. For instance consider a database with attribute name represented as first_name, middle_name and last_name and another database name attribute is represented as first_name and last_name. First_name, Last_name in name attribute in source database1 should be merged as first_name in the target database. After standardization, the record linkage method is used to find the similar records from various heterogeneous databases. The probabilistic record linkage method is used to find similar records from different data sources. This approach takes into account a wider range of potential identifiers, computing weights for each identifier based on its estimated ability to correctly identify a match or a non-match, and using these weights to calculate the probability that two given records refer to the same entity. The Jaro-Winkler distance [41] is a measure of similarity between two strings.

The Jaro distance d_j of two given strings S_1 and S_2 is calculated by using equation 1.

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \quad (1)$$

Where m is the number of matching characters and t is half the number of transpositions.

Two characters from S_1 and S_2 respectively, are considered matching only if they are the same and not farther than is calculated by using equation 2.

$$\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} \right\rfloor - 1 \quad (2)$$

Jaro–Winkler distance uses a prefix scale P which gives more favorable ratings to strings that match from the beginning for a set prefix length ℓ. Given two strings S_2 and S_1, their Jaro–Winkle distance d_w is calculated by using equation 3.

$$dw = dj + (\ell\ P\ (1 - dj)) \qquad (3)$$

Where: d_j is the Jaro distance for strings S_1 and S_2; ℓ is the length of common prefix at the start of the string up to a maximum of 4 characters P is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. P should not exceed 0.25, otherwise the distance becomes larger than 1.

Table 1: Merged and standardized results

Rid	Model Name	Processor	RAM	Hard Disk	OS	Brand	Price
1	Pavilion G6-2313 AX	CPU Quad Core A10	6GB	null	null	Hewlet Packard	35200
2	Pavilion G6-2005 AX	CPU Quad Core A8	4GB	500GB	Win7	HP	31990
3	Pavilion G5-2300BT	CPU Quad Core A10	6GB	1TB	Linux	HP	32000
4	Pavilion G6-2301 AX	CPU Quad Core A8	4GB	400GB	null	HP	33000
5	Pavilion G6-2103TU	CPU Quad core A10	6GB	1TB	Linux	HP	38990
6	Pavilion G6-2005 AX	CPU Quad Core A8	4GB	500GB	Win7	hp	31990
7	Pavilion G6-2313AX	CPU Quad Core A10	6GB	1TB	Linux	HP	38200
8	Pavilion G6-2301 AX	CPU Quad Core A8	4GB	500GB	Win8	hp	33000
9	Pavilion G6-2103 TU	i5	4GB	500GB	Win7	Hp	38990

The standard value for this constant in Winkler's work is P=0.1. The matching records are grouped and compared to identify incompleteness.

For example, query is to select model_name, product_description, brand, price from the products table. The query is processed based on the query processing algorithm described in section 3.1 and the result is obtained from the E-shopping data sources for laptop store. The three different set of records are retrieved from three different E-shopping data sources. The result is retrieved as nine records that are retrieved from three different data sources and each record is given a record identifier to identify it uniquely. The product description attribute is divided into processor, RAM capacity, hard disk capacity and operating system for standardization. The merged and standardized result set is shown in table 1.

The records representing the same laptop are clustered and are shown in table 2. For example record 1 and 7 are same but retrieved from different data sources.

Table 2: Record Clustering

Name of the cluster	C1	C2	C3	C4	C5
Record Grouping	1,7	2,6	3	4,8	5,9

3.2.2 Incompleteness detection and resolution

Completeness concerns the degree to which all data relevant to an application domain has been recorded in the data source. The different types of completeness measures are source completeness, tuple completeness and attribute completeness. The source completeness is measured by using the equation 4.

$$\text{Source Completeness} = NRRS/TNRR \qquad (4)$$

Where NRRS is Number of Records Retrieved from a Source and TNRR is Total Number of Records Retrieved. Tuple completeness is measured by using the equation 5.

$$\text{Tuple Completeness (TC)} = NAAT/TNAR \qquad (5)$$

Where NAAT is Number of Attributes available in Tuple and TNAR is Total Number of Attributes Required.

Attribute completeness is measured by using the equation 6.

Attribute Completeness (AC) =NNNVA/TNVA (6)

Where NNNVA is Number of Non-Null Values in Attribute and TNVA is Total Number of Values in the Attribute.The matched records are analyzed and completeness measures are obtained. The following resolutions have been made to achieve completeness.

Resolution 1: If the values of the attribute within the cluster are match exactly that are copied to resultant set without any modification.

Resolution 2: If only one or few values among the compared records have same attribute values within the cluster then the record with highest tuple completeness value is chosen and copied to resultant set.

Resolution 3: If two attribute have contradicting values and same tuple completeness within the cluster then the record with highest source completeness values is chosen and copied to resultant set.

The resolutions are passed to the decision making and notification service.

3.2.3 Inaccuracy detection and resolution

The accuracy is defined as the proximity of a value *v* to a value *v'* considered to be correct. Syntactic accuracy is the closeness of a value v to the elements of the corresponding definition domain D. In syntactic accuracy the value *v* is not compared to value *v'*, rather it is checked that whether *v* is anyone of the values in domain D, whatever it is so. Records are classified as accurate, weak inaccuracy and strong inaccuracy based on the rules shown in table 3 and the accuracy prediction for the clusters in table 2 is shown in table 4.For example according to table 3 rules, table 1 the tuple with rid 5 and 9, the processor is CPU Quad Core and i5 respectively. It is syntactically correct but tuple mismatches and hence a weak inaccuracy is identified.

The following resolutions are taken by using rules shown in table 3

Resolution 1: if the set of records in the cluster satisfies the rule 1 then it is accurate. The records are copied to resultant set.

Resolution 2: if the set of records in the cluster satisfies the rule2, rule3, rule 4 then it is not accurate. The resolutions are passed to decision making notification service.

Table 3: Accuracy Prediction Rules

Rule No	**Parameters**	**prediction**
1	Tuple matches ^ syntactically correct	Accurate
2	Tuple matches ^ syntactically incorrect	Weak inaccuracy
3	Tuple mismatches ^ syntactically correct	Weak inaccuracy
4	Tuple mismatches ^ syntactically incorrect	Strong inaccuracy

Table 4: Accuracy Prediction example

S.no	**Name of the cluster**	**Accuracy level**
1	Cluster1	Accurate
2	Cluster 2	Accurate
3	Cluster 3	Accurate
4	Cluster 4	Weak accuracy
5	Cluster 5	Weak accuracy

3.2.4 Decision Making

The record values are filled in the resultant set based resolutions from sections 3.2.2 and 3.2.3.

The following decisions have been taken

Decision 1: If all the attributes are complete and accurate then the resultant set is passed to result display service for end user.

Decision 2: If any incompleteness, inaccuracy in the resultant set then the resultant set is passes to display service and also notification service.

3.2.5 Notification Service

The notification service periodically notifies the incompleteness and inaccuracy of the respective data sources through messages in order to improve the data quality in the data sources. For example, In Table 1 the first record with rid 1 from data source 1 is incomplete because its hard disk and operating systems are NULL. Then the corresponding data source is notified to complete the record values to improve the data quality.

The updated results are forwarded to the users as shown in table 5, which is derived from table1 after applying QoS.

4. Results and discussion

For experimentation, E-shopping of a few enterprises is selected. These enterprises sell electronic gadgets like computer, laptop and television etc that are heterogeneous and autonomously developed. A unified view is created to resolve the semantic conflict among different heterogeneous databases by using ontology. This view is used by the user for shopping and business analysts for decision support.
To implement the prototype of the ontology based data integration, the following tables has been autonomously created in different enterprises.
Category (cate_id, cate_name, cate_description)
Customer(cust_id, Cust_name, Cust_address, custr_phone_no, Cut_email_id)
Products(prod_id, cate_id,model_name, product_desc, brand, price)
Order(order_id, prod_id, cust_id, no_of_products)

Table 5: Results with QoS

Rid	**Model name**	**Processor**	**RAM**	**Hard disk**	**OS**	**Brand**	**Price**
1	Pavilion G6-2313 AX	CPU Quad Core A10	6GB	1TB	DOS	HP	38200
2	Pavilion G6-2005 AX	CPU Quad Core A8	4GB	500GB	Win7	HP	31990
3	Pavilion G5-2300 BT	CPU Quad Core A10	6GB	1TB	DOS	HP	32000
4	Pavilion G6-2301 AX	CPU Quad Core A8	4GB	500GB	Win8	hp	33000
5	Pavilion G6-2103 TU	CPU Quad Core A10	6GB	1TB	DOS	HP	38990

Here three databases using MYSQL, ORACLE, SQL server are considered. In all these databases the table and attributes are using different name and are schematically heterogeneous. In these databases, for experimentation 4000 records of each data source is taken. Local and Global ontology have been constructed by using protégé 4.2 tool [40].
The local ontology and data source mapping has been implemented by using Protégé ontop plug in. The accuracy, completeness checking, resolution module, decision and notification service module has been implemented by using java. The querying and retrieved results are shown in fig 4. The query is to retrieve the products details in the product table where the product price is greater than 68000.
The proposed experimental setup involves comparison of two different systems: The classical data integration and proposed quality aware service oriented data integration. The traditional data integration is local and global schema that is constructed by using Local As View (LAV) [4] approach. The local and global ontology creation snapshots are shown in fig.2 and fig 3 respectively

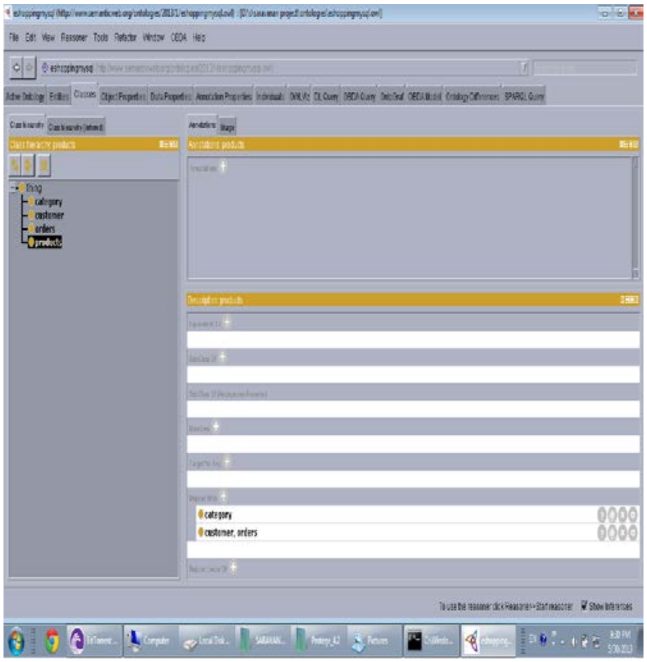

Fig 2. Local ontology creation

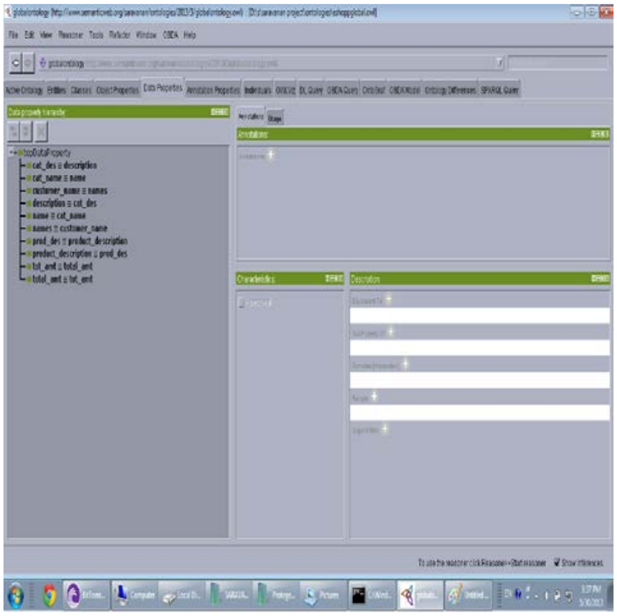

Fig.3. Global ontology creation

Fig 4: querying and result retrieval

4.1 Accuracy

The accuracy has been calculated by using formula shown in equation 7.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \quad (7)$$

tp – True Positive = Accurate records .
fp – False Positive = fn – False Negative= weak accurate records

Here tn zero as data sources does not provide any true negative values. The ontology based data integration and quality aware service oriented data integration accuracy has been calculated for combination of simple: 20 query sets, aggregated function: 20 query sets and sub query: 20 query sets that are shown in fig.5, fig 6, fig 7 respectively.

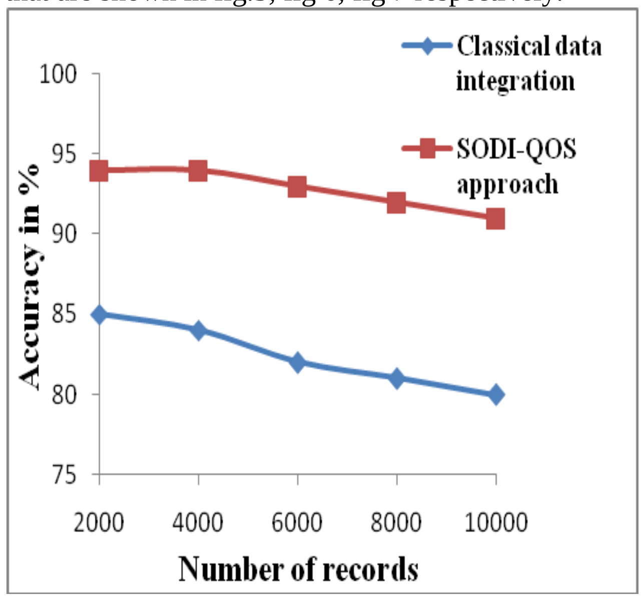

Fig 5: Accuracy comparison for simple queries

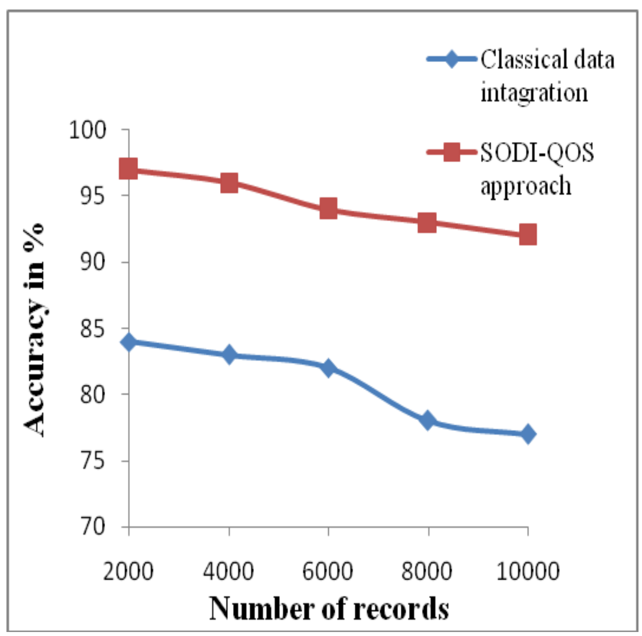

Fig : 6 Accuracy comparison for aggregate function

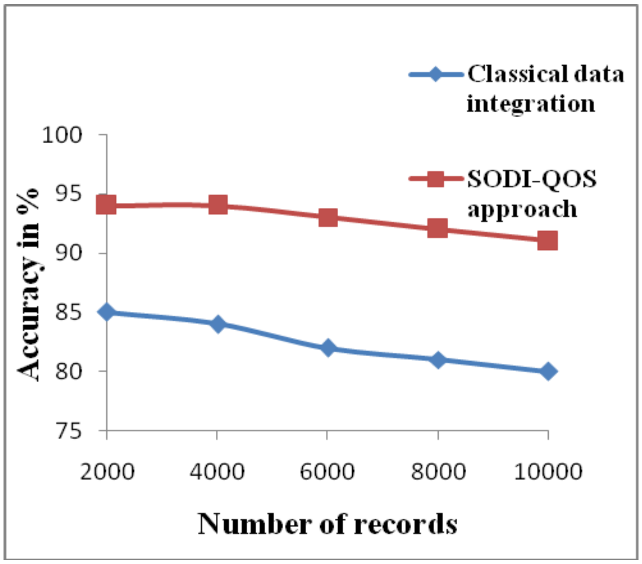

Fig 7: Accuarcy comparison for subqueries

The accuracy of classical data integration system and SODI-QoS integration system is compared. The accuracy of SODI-QoS integration system is higher due to the QoS improvement measures. Experimental results are shown in fig 5-7 concludes that SODI-QoS system provides 12% improved quality results compared to the classical data integration system

4.2 Precision

The precision is calculated by using equation 8 for the proposed SODI-QoS approach and classical data integration approach. The combination of 20 simple query sets, aggregated function query: 20 query sets and sub query: 20 query sets are taken for calculation and it is shown in fig. 8.

$$Precision = \frac{((No.of\ relevant\ records) \cap (No.\ of\ records\ retrieved))}{(No.of\ records\ retrieved)} \quad (8)$$

The precision of classical data integration system and SODI-QoS integration system is compared. The precision of SODI-QoS integration system is higher due to the QoS improvement measures. Experimental results are shown in fig 8 concludes that SODI-QoS system provides 14% improved quality results compared to the classical data integration system

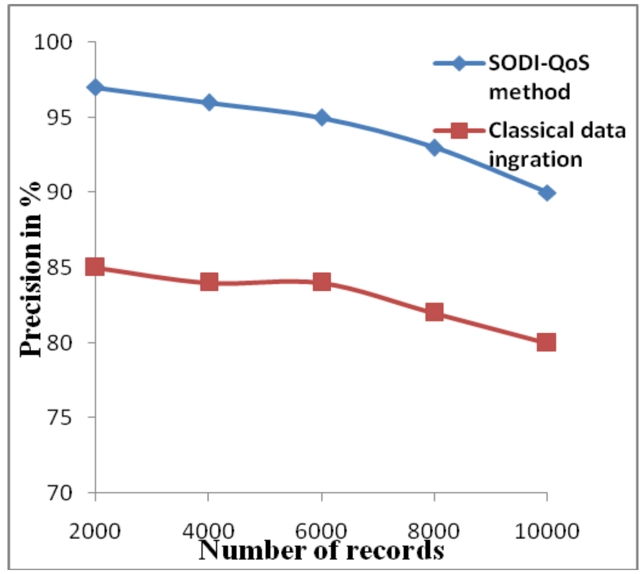

Fig 8 : Precision rate comparison

5. Conclusion

The SODI-QoS architecture has been successfully implemented and demonstrated for ontology based data integration with quality of service. The following conclusions have been achieved by using SODI-QoS architecture.

1. The local ontology has been created from the respective local schemas of the data sources.
2. Global ontology has been created by using hybrid ontology approach.
3. It addresses well known and important, yet frequently ignored problem of considering data quality such as completeness, accuracy in data integration.
4. The results offer a solution to the problem by ensuring the quality of the results before providing it to the user of the integration system. Further this method notifies the data source owners about inadequate quality of the data in case of poor data quality, which serve to enhance the quality of the data source.
5. The experimental results conclude that the proposed system has improved the accuracy, precision by 12%, 14% respectively.

In the future extension the data mining techniques shall be used for clustering the records to identify the similar records retrieved from heterogeneous data sources.

References

1. Amihai Motro, Philipp Anokhin, Fusionplex: Data quality in the integration of heterogeneous information sources, George Mason University,2003

2. Amit P. Sheth , James A. Larson, Federated database systems for managing distributed, heterogeneous, and autonomous databases, ACM Computing Surveys, vol- 22, 1990,pp.183—236

3. Andreas Langegger, Virtual Data Integration on the Web-Novel Methods for Accessing Heterogeneous and Distributed Data with Rich Semantics, International Conference on Information Integration and Web based Integration System, iiWAS2008, Linz, Austria, ACM 2008, pp 559-562.

4. Birte Glimm, Ian Horrocks, Boris Motik, Rob Shearer, Giorgos Stoilos, A novel approach to ontology classification , Journal of Web Semantics: Science, Services and Agents on the World Wide Web , 2012, pp 84-101.

5. Chen Jia, Wu Yue, Rules-based object-relational databases ontology construction, Journal of Systems Engineering and Electronics, Vol. 20, No. 1, 2009, Page(s) 211–215.

6. Dejing Dou, Paea LePendu, Shiwoong Kim, Peishen Qi (2006), Integrating Databases into the Semantic Web through and Ontology-based Framework, International Conference on Data Engineering Workshops, ICDEW'06, ACM,pp54.

7. E.Vysniauskas, L. Nemuraite, B.Paradauskas, Hybrid Method for Storing and querying Ontologies in Databases, Journal of Electronics and Electrical Engineering, vol.9, 2011,pp.67-72

8. Felix Naumann, Johann-Christoph Freytag, Ulf Leser, Completeness of integrated information sources , Journal of Information Systems , 2004,pp. 583–615

9. Felix Naumann, Ulf Leser, Johann Christoph Freytag, Quality-Driven Integration of Heterogeneous Information Systems, German Research Society GRK-316, 2002.

10. Fuqi Song , Gregory Zacharewicz, David Chen, An ontology-driven framework towards building enterprise semantic information layer, Journal of Advanced Engineering Informatics , Elseiver, 2013, pp.38-50

11. Gagnon M, Ontology-Based Integration of Data Sources, 10th International Conference on Information Fusion, IEEE, 2007, pp.1-8.

12. Gongzhu Hu, Global Schema as an inversed view of Local Schemas for Integration, International Conference on Software Engineering Research, SERA'06.

13. Huiyong Xiao, Query Processing For Heterogeneous Data Integration Using Ontologies , Ph.D thesis, University of Illinois, Chicago, 2006

14. Irina Astrova, Nahum Korda and Ahto Kalja, Automatic Transformation of OWL ontologies to SQL Relational Databases, International journal of Electrical, computer and system engineering , Vol 1, 2007,

15. Jinpeng Wang, Yafie Zhang, Zhuang Miao, Jianjiang Lu, Query Transformation in Ontology-based relational Data Integration, Asia-Pacific Conference on Wearable Computing Systems, APWCS,IEEE, 2010, pp 303-306.

16. Joao C Pinheiro, Vania M. P. Vidal, Jose A. F. Macedo, Eveline R. Sacramento, Marco A. Casanova, Fabio A. M. Porto , Query Processing in a Three-Level Ontology Based Data Integration System, International Conference of Information Integration and Web based Application Service, iiWAS2010, Paris,France, ACM,pp 283-290.

17. Laomo Zhang, Ying Ma, Guodong Wang , An Extended Hybrid Ontology Approach to Data Integration, International Conference on Biomedical Engineering and Informatics, BMEI '09, pp 1-4.

18. Lina AI-Jadir, Christine Parent, Stefano Spaccapietra, Reasoning with large ontologies stored in relational database, Journal of data and knowledge engineering, Elsevier, 2010 pp.1158-1180

19. M.A. Rodriguez, M.J. Egenhofer, Determining semantic similarity among entity classes from different ontologies, IEEE Transactions on Knowledge and DataEngineering 15 (2003) 442–456.

20. M.S.Hema, S.Chandramathi , Federated Query Processing Service in Service Oriented Business Intelligence, Communications in Computer and Information Science, Springer, 2011,pp.337-340

21. Matthias jarke, manfred a. Jeusfeld,christoph quix, panos vassiliadis, Architecture and quality in data warehouses: An extended repository approach, 1999.

22. Maurizio Lenzerini, Data Integration a Theoretical Perspective, Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Newyork, USA,pp. 233-246, 2002.

23. Miriam Fernandez , Ivan Cantador, Vanesa Lopez, Semantically enhanced Information Retrieval:

An ontology-based approach , Journal of Web Semantics: Science, Services and Agents on the World Wide Web , 2011 pp. 434-452

24. Monica Scannapieco, Antonino Virgillito, Carlo Marchetti, Massimo Mecella, and Roberto Baldoni, The DaQuinCIS Architecture: a Platform for Exchanging and Improving Data Quality in Cooperative Information Systems. Information Systems, University of Degli Studi Di Roma, 2004, 29(7):551–582.

25. Nikolaos Konstantinou, Dimitrios-Emmanuel Spanos, Michael Chalas, An approach to an intermediate layer between ontologies and relational database contents, Journal of Mobile Computing , IEEE transactions, 2009, pp.528-543

26. R.Harrison and C. Chan, Distributed Ontology Management System, Proc. 18th Annual Canadian Conference on Electrical and Computer Engineering . Aasktoon, Canada, 2005, pp. 661-664

27. R.Volz,L.Stojanovic, N.stojanovic, Migrating data intensive Web sites into the Semantic Web, ACM symposium on Applied Computing , Madrid , Spin 2002.

28. Raji, Ghawi, L.Nadine Cullot, Database-to-Ontology Mapping Generation for Semantic Interoperability , ACM VLDB 2007

29. Richard Hull , Roger King, Semantic database modeling: Survey, applications, and research issues, ACM Computing Surveys, Vol-19,1987, pp.202-260

30. Rodrigo Fernandes Calhau, Ricardo de Almedia Falbo , An Ontology-based Approach for Semantic Integration , IEEE International Enterprise Distributed Object Computing Conference, EDOC 2010,pp 111-120.

31. Ronald Denaux, Catherine Dolbear, Glen Hart, Vania Dimitrova, Anthony G. Cohn, Supporting domain experts to construct conceptual ontologies: A holistic approach, Web Semantics: Science, Services and Agents on the World Wide Web, 2011, pp 113-127.

32. Suihua Wang, Xiaodan Zhang, A high Efficiency Ontology Storage and Query Based on Relational Database, International conference on Electrical and Control Engineering, 2011, pp.4253-4256

33. Susanne Busse , Ralf-Detlef Kutsche , Ulf Leser , Herbert Weber, Federated Information Systems: Concepts, Terminology and Architectures , 1999

34. Vipul Y Kashyap, Amit P. Sheth, Semantic and schematic similarities between: a context-based approach, The VLDB Journal - The International Journal on Very Large Data Bases, Vol 5 Issue 4, 1996,pp. 276 – 304

35. Wiederhold, Mediators in the architecture of future information systems, IEEE computer, vol -25, 1992,pp.38-49

36. William E. Winkler, Quality of Very Large Databases **,** Ph.D Thesis, U.S. Bureau of the Census, Washington D.C, 2001

37. Yumeng Zhan, Shidong Zhang, Zhongmin Yan , Ontology – based Model for Resolving the Data-level and Semantic-level Conflict, *Internationa* Conference on Information and Automation IEEE,2009

38. Z. Xu, S. Zhang, Y. Dong, Mapping between relational database schema and OWL ontology for deep annotation, in: Web Intelligence, 2006, WI 2006, IEEE/WIC/ACM International Conference on, 2006, pp. 548–552

39. Ze Hua, JianMin Ban, Ontology-based Integration and Interoperation of XML Data, *Sixth* International Conference on Semantics, Knowledge and Grids, Beijing, China, IEEE,2010 pp422-423

40.The Protégé Ontology Editor and Knowledge Base Framework, http://protege.stanford.edu/.

41. WilliamW. Cohen, Pradeep Ravikuma, Stephen E. Fienberg, “A Comparison of String Distance Metrics for Name-Matching Tasks” , American Association for Artificial Intelligence, vol. 64, 2003, pp. 1183-1210