Semantic Conflicts Reconciliation (SCR): A Framework for Detecting and Reconciling Data-Level Semantic Conflicts.

Torky I. Sultan, Mona M. Nasr, Ayman E. Khedr, Walaa S. Ismail
Faculty of Computers and Information, Information Systems Department, Helwan University

Abstract

Integration of heterogeneous data sources is not an easy task that involves reconciliation of various levels of conflicts. Before we can integrate the heterogeneous data, we need to resolve these heterogeneity conflicts. Semantic conflict, if undetected, can lead to disastrous results in even the simplest information system. In this paper, we recommend system architecture to solve the semantic data level conflicts that related to different representation or interpretation of data contexts among different sources and receivers. In the proposed ontology-based approach, all data semantics explicitly described in the knowledge representation phase and automatically taken into account by Interpretation Mediation Services phase so conflicts can automatically detect and resolved at the query runtime. Data sources still independent from the integration process that is mean we can retrieve up to date data and smoothly update the data in each data source without affecting the framework.

Keywords: Data Integration, Heterogeneous Databases, Interoperability, Ontology, Semantic Heterogeneity.

I. Introduction

Information integration and retrieval assumed to be very different today, with the growth of the Internet and web-based resources, the presence of structured, semi-structured, and unstructured data - all of have added new dimensions to the problem of information retrieval and integration as known earlier. Information integration and the related semantic interoperability are becoming an even greater concern. Integration of information considers from the most important factors able to build a large-scale business applications such as enterprise-wide decision support systems. The main goal of information integration is to combine selected systems to a unified new whole and give users the ability to interact with one another through this unified view. It requires a framework for storing metadata, and tools that make it easy to manipulate the semantic heterogeneity between sources and receivers.

There are different views about classification of semantic conflicts, as in [1]

We can classify semantic conflicts to data-level conflicts and schema-level conflicts.

- *Data –level conflicts:* conflicts that arise at the data level (instance level) that related to different representation or interpretation of data values among different sources.
- *Schema –level Conflicts:* conflicts arise at the *Schema* level when schemas use different *alternatives* or definitions to describe the same information.

As we expected no single organization has complete information, intelligence information usually gathered from different organizations in different countries, so integration is necessary to perform various intelligence analyses and for decision-making. Different challenges appear when different agencies organize and report information using different interpretations. In order to illustrate the challenges of integrating information from different information sources, let us consider a simple integration scenario that involves many of the following data elements [2].Suppose we have 150 agencies and each one may use different contexts. The varieties of these contexts are summarized in Table (1) . Now we have 1,536 (i.e., 4*3*2*4*4*4) combinations from these varieties. We use the term contexts to refer to these different ways of representing and interpreting data. We can consider that each of the 150 data sources uses a different context as its data representation convention.

Table 1: Semantic Differences in Data Sources [2].

Data Types	Semantic varieties
Height	4 different units of measure: ft, in, cm, m
Weight	3 different units of measure: lbs, kg, stone
Airport	2 different coding standards: IATA, ICAO
Country	4 different coding standards: FIPS, ISO 2-Alpha, ISO 3-Alpha, ISO 3-digit
Geo-coordinate	4 different reference systems and datum parameters: MGRS_WGS84, BNG_OGB7, Geodetic_WGS84, UTM_WGS84
Date	4 different formats: mm/dd/yyyy, dd/mm/yyyy, dd.mm.yyyy, dd-mm-yyyy.

An analyst from any of the 150 agencies may need information from all the other agencies for intelligence analysis purposes . As shown in Table (1), when information from other agencies is not converted into the analyst's context, it will be difficult to identify important patterns. Therefore, a total of 22,350 (i.e., 150*149) conversion programs would be required to convert data from any source's context to any other source's context, and vice versa. Implementing tens of thousands of data conversions is not an easy task; but maintaining them to cope with changes in data sources and receiver requirements over time is even more challenging [2].

II. Related Work

Achieving semantic interoperability among heterogeneous and disparate information sources has been a critical issue within the database community for the past two decades [1]. Technologies already exist to overcome heterogeneity in hardware, software, and syntax used in different systems (e.g., the ODBC standard, XML based standards, web services and SOA-Service Oriented Architectures) While these capabilities are essential to information integration, they do not address the issue of heterogeneous data semantics that exist both within and across enterprises [2]. We can manage such heterogeneities by adopting certain standards that would eliminate semantic heterogeneity in the sources all together or by developing and maintaining all necessary conversions for reconciling semantic differences.
We can achieve semantic interoperability in a number of ways. We discuss the traditional approaches to Semantic Interoperability and the most important ontology-based systems for data integration.

III. Traditional Approaches to Semantic Interoperability Brute-force Data Conversions (*BF*)

In the Brute-force Data Conversions (BF) approach all necessary conversions implemented with hand-coded programs. for example, if we have N data sources and receivers, N (N-1) such conversions need to be implemented to convert the sources context to the receiver context. These conversions become costly to implement and very difficult to maintain When N is large. This is a labor-intensive process; nearly 70% of integration costs come from the implementation of these data conversion programs. A possible variation of the (BF) approach is to group sources that share the same set of semantic assumptions into one context. The approach allows multiple sources in the same context to share the same conversion programs, so the numbers of conversion programs will be reduced. We refer to the original approach and this variation as BFS and BFC, respectively [2]. These approaches are illustrated schematically in Fig 1.

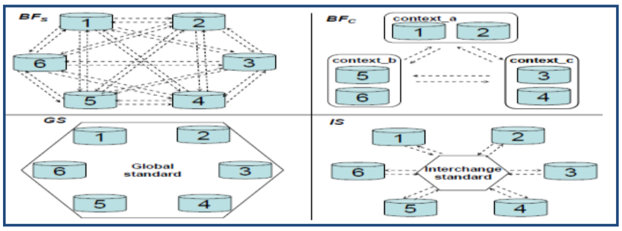


Fig. 1: Traditional approaches to Semantic Interoperability [4].

Global Data Standardization (GS)

If we could develop and maintain a single data standard that defines a set of concepts and specifies the corresponding representation, all semantic differences would disappear and there would be no need for data conversion. Unfortunately, such standardization is usually infeasible in practice for several reasons. There are legitimate needs for having different definitions for concepts, storing and reporting data in different formats. Most integration and information exchange efforts involve many existing systems, agreeing to a standard often means someone has to change his/her current implementation, which creates obstacles and makes the standard development and enforcement extremely difficult [3].

Interchange Data Standardization (IS)

The data exchange systems sometimes can agree on the data to be exchanged, i.e., standardizing a set of concepts as well as their interchange formats. The underlying systems do not need to store the data according to the standard; it suffices as long as each data sender generates the data according to the standard. That is, this approach requires that each system have conversions between its local data and an interchange standard used for exchanging data with other systems. Thus, each system still maintains its own autonomy. This is different from the global data standardization, where all systems must store data according to a global standard. With N systems exchanging information, the Interchange Standardization approach requires 2N conversions. The IS approach is a significant improvement over the brute-force approach that might need to implement conversions between every pair of systems [4]. Although this approach has certain advantages, it also has several serious limitations [2]. From which, all parties should reach an agreement on the data definition and data format. Reaching such an agreement can be a costly and time-consuming process besides; any change to the interchange standard affects all systems and the existing conversion programs. Lastly, the approach can involve many unnecessary data conversions

IV. Ontology-Based Data Integration Approaches

Most of the shortcomings of the previous traditional approaches can be overcome by declaratively describing data semantics explicitly and separating knowledge representation from conversion programs (implementation). We will explain the most important ontology-based systems for data integration, which are SIMS, OBSERVER, KRAFT, SCROL and COIN with respect to the role and use of ontologies.

SIMS is based on a wrapper/mediator, that is, each information source is accessed through a wrapper [5]. The SIMS mediator component is used to unify the various available information sources and to provide the terminology for accessing them. The core part of the mediator is the ability to intelligently retrieve and process data [6].Each information source is incorporated into SIMS by describing the data provided by that source in terms of the domain model. This model is contained in the mediator. SIMS uses a global domain model that also can be called a global ontology. The work presented in [6], classifies SIMS as a single ontology approach. An independent model of each information source must be described for this system, along with a domain model that must be defined to describe objects and actions. Further, the authors address the scalability and maintenance problems when a new information source is added or the domain knowledge changes [7].There is no concrete methodology for building ontologies in The SIMS [6].

OBSERVER uses the concept of data repository, which might be seen as a set of entity types and attributes. Each repository has a specific data organization and may or may not have a data manager [8]. The different data sources of a repository might be distributed. The architecture is based on wrappers, ontology servers and Interontology Relationships Manager (IRM) [9]. Here, a wrapper is a module, which understands a specific data organization and knows how to retrieve data from the underlying repository hiding this specific data organization. IRM is Synonym relationships relating the terms in various ontologies are represented in a declarative manner in an independent repository. This enables a solution to the vocabulary problem. OBSERVER is classified as a multiple ontology approach. OBSERVER defines a model for dealing with multiple ontologies avoiding problems about integrating global ontologies [6]. The different ontologies (user ontologies) can be described using different vocabularies depending on the user's needs.

KRAFT was created assuming dynamic information sources [5]. The KRAFT system is the hybrid ontology approach [6]. In order to overcome the problems of semantics heterogeneity, KRAFT defines two kinds of ontologies: a local ontology and a shared ontology. For each knowledge source there is a local ontology. The shared ontology formally defines the terminology of the domain problem in order to avoid the ontology mismatches that might occur between a local ontology and the shared ontology [10]. any modification or addition in any source, require changes in the local ontology, which represents this source, and the mappings between the local and the shared ontology have to be performed [8].

SCROL is a global schema approach that uses an ontology to explicitly categorize and represent predetermined types of semantic heterogeneity [1]. It is based on the use of a common ontology, which specifies a vocabulary to describe and interpret shared information among its users. It is similar to the federated schema approach. However, an ontology-based domain model captures much richer semantics and covers a much broader range of knowledge within a target domain . SCROL assumes that the underlying information sources are structured data that may reside in the structurally organized text files or database systems. However, the unprecedented growth of Internet technologies has made vast amounts of resources instantly accessible to various users via the World Wide Web (WWW)[1].

COIN Project was initiated in 1991 with the goal of achieving semantics interoperability among heterogeneous information sources. The main elements of this architecture are wrappers, context axioms, elevation axioms, a domain model, context mediators, an optimizer and a executioner. A domain model in COIN is a collection of primitive types and semantic types (similar to type in the object-oriented paradigm), which defines the application domain corresponding to the data sources that are to be integrated COIN introduces a new definition for describing things in the world. It states that the truth of a statement can only be understood with reference to a given context. The context information can be obtained by examining the data environment of each data source [11].

We discussed different approached for semantic information integration, the Traditional and Ontology-Based Approaches. The problem of semantic interoperability is not new, and people have tried to achieve semantic interoperability in the past using various approaches. Traditional approaches have sometimes been reasonably successful in limited applications, but have proven either very costly to use, hard to scale to larger applications, or both. Traditional approaches have certain drawbacks

that make them inappropriate for integrating information from a large number of data sources.
Existing ontology-based approaches for semantic interoperability also have not been sufficiently effective because there is no systematic methodology to follow, no concert methodology for building ontologies and all existing ontology-based approaches use a static data model in reconciling the semantic conflicts.

V. System Architecture

The Semantic Conflicts Reconciliation (SCR) framework is considered as an ontology based system aims to solve semantic data level conflicts among different sources and receivers in a systematic methodology. SCR based on domain specific ontology to create user queries. The user can browse the merged ontology and selects specific terms and conditions to create global query. The user query terms are mapped to the corresponding terms in each data source to decompose the global query to set of sub naïve queries. The decomposed sub-queries converted to well-formed sub-queries before sending it to the suitable database.Finally the SCR combine and resend the well-formed query results to the users that matches their contexts.

SCR consists of two phases, the knowledge representation phase and the interpretation mediation service phase.

5.1 Knowledge Representation

The knowledge representation phase consists of the following components:

- **Ontology Extraction:** Extract local ontology from each database.
- **Global/Merged Ontology:** Merge all local ontologies to construct a global one that contain all major concepts and the relationships between them.
- **Contexts:** Explicitly describe the sources and receivers assumptions about data.
- **Mapping:** Linking between the constructed merged ontology and the corresponding terms in the data sources to produce the semantic catalog**.**

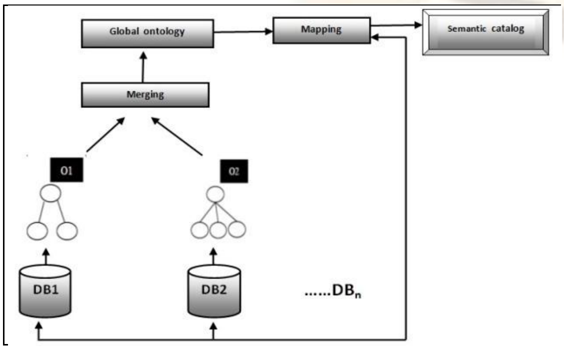


Fig. 2: Knowledge Representation phase

In the knowledge representation phase we have multiple database sources as input, and produce global ontology and semantic catalog as output. This process is done once when the integration process is started. Any changes in schemas can be easily added to the global ontology and the semantic catalog.
Fig. (2) describes the Knowledge representation phase. Assumes that, there are two or more heterogeneous sources in the same domain that are semantically related. It takes the databases as input and extracts local ontology for each source. Then, using a reasoning system to determine the corresponding terms based on suitable matching algorithm. The corresponding terms in the local ontologies are merged into new one global ontology. Now we have a constructed merged ontology for all sources that we can add annotations to explicitly describe semantics of each data source. Finally, the mapping system maps the merged ontology with the corresponding data sources to facilitate retrieving up to date data from the integrated databases.

5.1.1 Database to Ontology Extraction:

In the ontology extraction step, we have multiple databases to extracts a local ontology from each one. A local ontology contains all database information like tables, columns, relations, constraints. Moreover, it contains an intentional definition to represent a high-level abstraction than relational schema.

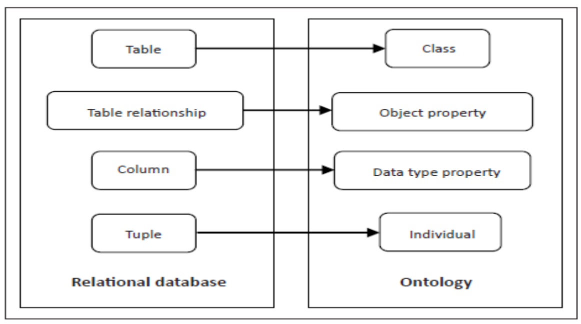


Fig. 3: Database to local ontology extraction

The local ontology represents a relational database tables as concept and columns as slots of the concept. The local ontologies are represented in a formal standard language called OWL (Ontology Web Language). OWL is the most popular ontology representation language [12]. The OWL Web Ontology Language designed to be used by applications that need to process the contents and meaning of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary for describing properties and classes along with a formal semantics. Creating local ontology for each database saves them independent. Any changes in the schema or relations can be added easily to its local ontology. The local

ontology includes only the metadata and additional semantics; however, the database instances or members still in the data source separated from its ontology.
We can use DataMaster to import schema from a relational database into ontologies; it is protégé plug-in [13]. DataMaster uses IDBC/ODBC to connect to database (the user chooses one of them). It allows variety of translating database schema into ontology depending on the user's application requirements [14].

5.1.2 Global Ontology Construction:

Merging process aims to create one global (merged) ontology contains multiple local ontologies contents. The ontology merging process is the creation of the new ontology from two or more ontologies. It contains all the knowledge of the initial ontologies. To create a merged ontology, the corresponding objects will be matched from two or more local ontologies. Subsequently, suitable matching algorithm should choose. Matching is the core of merging process to make one vantage point of view from multiple ontologies, where some concepts and slots will be represented as a new concept and new slots, or some slots may be merged and follow another concept. We can say that, there is a new structure will be created in the merged ontology. This structure does not affect the information sources, because each local ontology is independent. Creating a standard formal model (merged ontology) makes query multiple databases satisfy the user requirements at the semantic level.
The proposed framework uses a string-matching algorithm. The string similarity matching calculates the string distance to determine the matching of entity names. Before comparing strings, some linguistic technologies must be performed. These linguistic technologies transform each term in a standard form to be easily recognized.
Based on the result of matching, the system presents some suggestion to the expert about merging some concepts. The expert may take concern to some or all these suggestions. Then, some concepts and their slots may be merged or copied as it is, or some overlapping slots may be merged under a new concept. Hence, a new structure will be created from multiple databases based on semantics of terms and relations.
The SCR framework uses PROMPT tool to matching and merging local ontologies. PROMPT is a semi-automatic tool. It is protégé plug in. It guides the expert by providing suggestions. PROMPT provides suggestions about merging and copying classes. Fig. (4) explains the PROMPT algorithm [15]. PROMPT takes two ontologies as input and guide the user to create a merged ontology as output. PROMPT generates a list of suggestions based on the choose matching algorithm. Framework uses PROMPT lexical matching algorithm.

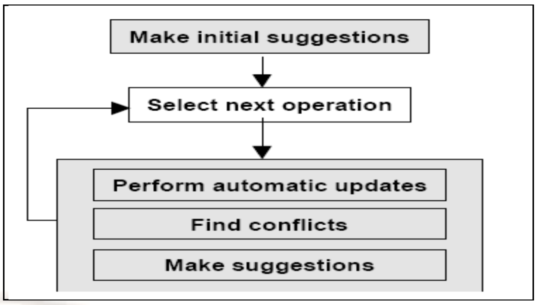


Fig. 4: The flow of PROMPT algorithm.

The PROMPT suggestions and conflicts are shown in a new interface, where the expert can modify the suggestions. It uses a semi automatic tool to add the expert's perception to the merging process. As a reason for, the matching techniques are considered as helper, but the human view makes the matching deep and increase conflicts detection. Moreover, the SCR aims to produce one unified view from multiple ontologies to be agreed among different systems. Hence, it can query the merged ontology and satisfy the user requirements. Matching and merging classes and slots make query processing faster, because we query one term instead of query many.

5.1.3 Explicitly Define Contexts

Contexts explicitly describe assumptions about data sources and receivers in a formal way. We consider contexts as a collection of value assignments or value specifications for terms in the global ontology that we need to describe their semantics.
Once we described the contexts for each source, the SCR framework can be automatically detecting and reconciling the data-level semantic conflicts among these different data sources and receivers using the interpretation mediation service that can handle these conflicts based on analyzing and comparing the differences in contexts among sources and receivers.
SCR assigns contexts descriptions about sources using the following two steps.

Annotation: Adding annotation properties to the global ontology slots to denote their contexts. We consider annotation properties as special properties that affect the interpretation of data values.
Value assignment: Assign value to each annotation property created in the previous step. We can associate more than one annotation property to the same slot (data type property) in the merged ontology. We can easily adding, removing and changing the assigned values in the ontology whenever the context changed in the sources over time.

5.1.4 Mapping Global Ontology to Multiple Databases:

The main purpose of the proposed mapping tool is to find and match between semantically similar terms in the global query with the corresponding terms in the data sources of integrated system. The output of the mapping process is the semantic catalog.
We created the merged ontology, which is a standard model, represents different database systems. However, there is not any link between the merged ontology and databases. The merged ontology does not contain databases instances as well. Thus, we need to link between the merged ontology and the integrated databases in order to retrieve up to date data from multiple sources. Each term in the merged ontology must be linked to the corresponding terms in the integrated databases. The output from the mapping process is the Semantic Catalog of the integrated databases. The Semantic Catalog contains a mapping data and a metadata, which has been collected automatically during the mapping process. We developed a semi automatic mapping tool used to map a merged ontology to multiple databases Fig. (5).

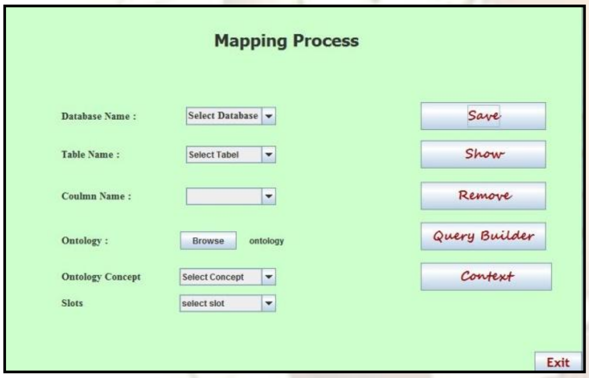
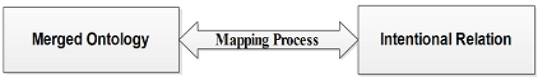


Fig 5: The proposed mapping tool

The mapping process in our mapping tool follows the following steps:

- The mapping process started by creating a database with two tables, to save the mapping data in the first table, and saving the metadata of the database system in the second table. This process is done once when the mapping process is started.
- After creating the database, the expert selects the first database to link its schema (intentional relation) with the terms in the global ontology.



- When the user select database from a list of all databases existed then all tables in the selected database will be listed. Then, press to select columns, all columns in the selected table will be listed and saved in the table created in the first step along with the correspondence terms in the global ontology. All the primary keys, foreign keys, and referenced tables for each table in the selected database automatically retrieved and saved in the second created table as metadata, to use it in query processing.
- At the end of mapping process, a Semantic Catalog has created. The Semantic Catalog contains a semantic mapping data among multiple databases and their Metadata. Hence, multiple database systems can be queried through a merged ontology using the Semantic Catalog, and retrieves data up to date. Any changes in the integrated sources can easily reflected in the semantic catalog.

5.2 Interpretation Mediation Service

Knowledge representation phase is not enough to solve semantic conflicts among data sources and receivers. The second phase in our system is the interpretation mediation service in which the user interacts with the system through graphical user interface (GUI).The GUI displays the global ontology terms to the user to facilitate finding the global query terms easily and quickly .User browse the global ontology to selects specific terms for his query. User submit query without take into account that both sources and receivers may have different interpretation of data contexts. In order to retrieve correct results from different integrated data sources queries should be rewritten into mediated queries in which all semantic conflicts between sources and receivers automatically detected and solved but it is a big challenge for users. We cannot suppose that users have intimate knowledge about data sources being queried especially when the number of these sources are big. User should remain isolated from semantic conflicts problems and no need to learn one of the ontology query languages .The **SCR** framework helps user to query integrated sources with minimal efforts using Query-by-Example language (QBE) as a result he needs only to know little information about the global ontology terms.
Interpretation Mediation Service as in Fig. (5) consists of the following main three components:

- Query preprocessor
- SQL generator
- Converter (query engine)

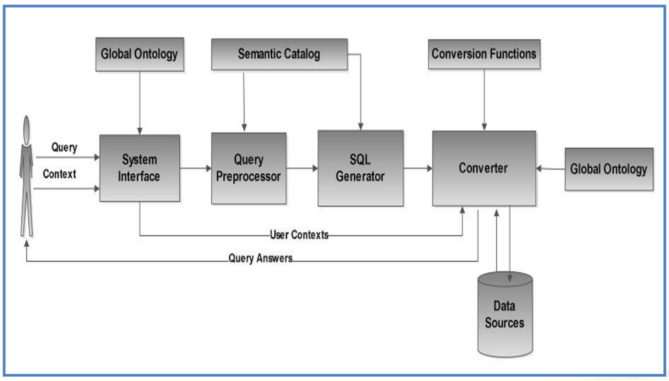


Fig. 6: Architecture of the SCR system.

5.2.1 Query Preprocessor

Query preprocessor accepts a naive user query and semantic catalog as input and produce blocks of user's data based on the selected items from the user global query. Each block represents a query but without any language format. Once the user selects terms and conditions from the system the query preprocessor does the following actions.

- Query preprocessor access the semantic catalog (mapping file) to retrieve the database name, table name and columns names that mapped to the selected terms and conditions in the user query.
- The query preprocessor reorganizes the retrieved data from the previous step into blocks according to the database name.

5.2.2 SQL Generator

SQL generator turns the query blocks received from the query preprocessor into SQL queries and directs them to the converter. It uses the semantic catalog (metadata) to translate the previous blocks into SQL correct syntax. To transform the blocks to correct syntax the generator add select, from and where clauses. In addition, if the query needs to retrieve instances from more than one table the primary keys, foreign keys and referenced tables from the integrated databases may be added from the semantic catalog metadata file as well.

5.2.3 Converter

We consider converter as the query engine that takes SQL query from the SQL generator and the user context as input. Converter transforms the user naïve query (that ignores differences in assumptions between sources) into well-formed query that respect differences among sources and receivers contexts.

Converter reconciles semantic conflicts at query time before sending the query to the suitable database, and then resends the correct result to the user that matches his contexts. Fig. (7) describes the general form of the conversion functions.

A function converts (Cs, Ct, Vs) this function returns Vt

Where Cs: Context of the source.

Ct: Context of the target.

Vs: Value of the source.

Vt: Equivalent value for Vs in user context.

Fig. 7: General form of the conversion functions.

Conversion functions represent the conversions among all annotation property values or contexts described in the merged ontology. In the SCR there is no relation between the number of sources or receivers and the number of conversion functions, it depends on the contexts or the annotation property values for each source whether it match the user query context or not .

```
Get global query from user based on global ontology
    Decompose query based on semantic catalog
    Convert global query Based on database structure
    Get Annotation for each attribute based on global ontology
    Compare attribute value and annotation value
If  match then send the query to suitable database
     and retrieve combine result
        (if (Iscontext) then convert result and display it
                  Else display default context)
Else  convert it into database context and send the query to suitable database
     and retrieve combine result
        (if (Iscontext) then convert result and display it
                  Else display default context)
```

Fig. 8: Pseudo code describes the basic specifications of the Interpretation Mediation Service algorithm.

Fig. (8) describes the basic specifications of the Interpretation Mediation Service algorithm in which converter receives well-defined quires from SQL generator and before sending it to the suitable database; it connects to the merged ontology and compare each attribute value context in the submitted query with the correspondence annotation property value in the ontology. If contexts mismatch with each other converter has to connect with the external conversion functions to reconcile it before sending the query to the suitable database. Converter receives and recombines the sub queries results from the integrated database and finally resends the query results to the user that matches his contexts.

VI. Conclusion

In this paper, we recommended system architecture for detecting and reconciling data-level semantic conflicts. The Ontology-based system provides separation between mediation services and the knowledge representation, which support sharing and reusing of semantic encoding and allowing independence of the contexts in ontology. As the result, we can easily modify or adding new contexts in ontology if users change their assumption over time without effecting the mediator. With the SCR framework user in any context can ask queries over any number of data sources in other contexts as if they were in the same context without burden them with the semantic conflicts in data sources. Ontologies are represented in a formal standard language called OWL, which makes them possible to be exchanged and processed by other applications easily. The SCR preserve local autonomy for each data source to change and maintain independently.

References

[1] Ram ,S., Park, J., Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflicts, *IEEE Transactions on Knowledge and Data Engineering, v.16 n.2* , p.189-202, 2004.

[2] Madnick S., Gannon T., Zhu, H., Siegel M., Moulton A.,Sabbouh M., Framework for the Analysis of the Adaptability, Extensibility, and Scalability of Semantic Information Integration and the Context Mediation Approach, MIT,2009.

[3] Rosenthal, A., Seligman, L. and Renner, S. From Semantic Integration to Semantics Management: Case Studies and a Way Forward, *ACM SIGMOD* Record, 33(4), 44-50, 2004.

[4] Zhu, H., *Effective Information Integration and Reutilization: Solutions to Deficiency and Legal Uncertainty*, PhD Thesis, *Massachusetts Institute of Technology Cambridge, MA, USA, 2005*.

[5] Hajmoosaei, A. and Abdul Kareem, S., An ontology-based approach for resolving semantic schema conflicts in the extraction and integration of query-based information from heterogeneous web data sources. In Proc. *Third Australasian Ontology Workshop (AOW 2007)*, Gold Coast, Australia. CRPIT, 85. Meyer, T. and Nayak, A. C., Eds. ACS. 35-43.

[6] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G.Schuster, H. Neumann and S. Hubner, Ontology-Based Integration of Information - A Survey of Existing Approaches , *In Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, pp. 108-117,2001.

[7] Arens, Y., Ciiee, Y., Knoblock. A. ,SIMS:Integrating data from multiple information sources. *Information science institute, University of Southern California*, U.S.A.,1992.

[8] Buccella,A., Cechich, A., Nieves R. Brisaboa, Ontology-Based Data Integration Methods: A Framework for Comparison. *Revista Comlombiana de Computació*n, 6(1), [doi],2008.

[9] Mena, E., Kashyap, V., Sheth, A. and Illarramendi, A.Observer , An approach for query processing in global information systems based on interoperation across pre-existing ontologies, *Distributed and Parallel Databases,Volume 8 Issue 2,* Pages 223 – 271, April 2000.

[10] Visser, P. R. S., Jones, D. M., Bench-Capon, T. J. M. and Shave, M. J. R. Assessing heterogeneity by,classifying ontology mismatches. In Proceedings of the *International Conference on Formal Ontology in Information Systems* (FOIS'98), IOS Press, 148–162, 1998.

[11] Zhu,H.,Madnick,S., Reconciliation of temporal semantic heterogeneity in evolving information systems", *ESD-WP-2009-03*, Massachusetts Institute of Technology Cambridge, MA, USA,2009.

[12] V. Devedizic ,*Model Driven Architecture and ontology development*, Springer_Verlag Berlin Hekleberg, second edition,2006.

[13] Nyulas, M. Connor, S. Tu, DataMaster_a plug in for Relational Databases into protégé, Stanford University, *10th international protégé conference*,2007.

[14] http://protogewiki.stanford.edu/wiki/DataMaster, last visit December 2011.

[15] Noy, N. F. and Musen, M. A., PROMPT: Algorithm and tool for automated ontology merging and alignment, *National Conference on Artificial Intelligence - AAAI* , pp. 450-455, 2000