ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Ontology based Approach for ETL Mapping Maintenance

M.Thenmozhi

Assistant Professor, Dept. of Computer Science and Engineering Pondicherry Engineering College, Puducherry, India

ABSTRACT: Organizations need to integrate with others to improve their performance and productivity. Data warehouse provides one of the best solutions for data integration. In data warehouse integration is achieved through ETL (extract trasform and load) which is a crucial task in data warehouse. In order to transform data from source to target format an ETL mapping need to be produced by the designer. As today, the data source is of semi-structured or unstructured format, integrating data faces several heterogeneity problems. With the popularity of semantic web, ontologies can provide a better way to integrate data between organizations. In this paper an ontology based approach has been proposed which handles the impact of source evolution over the ETL mapping. It provides an automatic way of computing the new mapping to reflect the source changes. This feature helps the ETL designer to predict the impact on transformations to be carried out for the ETL task.

KEYWORDS: ETL Mapping, ETL design, Semantic Data Warehouse, Ontology for ETL

I. Introduction

A data warehouse is a centralized repository that stores data from multiple information sources and transforms them into a common, multidimensional data model for efficient querying and analysis. A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process [1]. ETL is the process of moving and transforming data into target database. In simple terms, we take data from a source S, transform it and move it to a target T [8]. Extract is the process of reading data from a database. Transform is the process of converting the extracted data from its previous form into the form it needs to be in so that it can be placed into another database. Load is the process of writing the data into the target database. Conceptual design of ETL processes is a critical task performed at the early stages of a data warehouse project. It describes the integration of data from heterogeneous sources into the data warehouse [8].

The process of transforming the data values from one system into new values for another system is referred to as mapping. Establishing the appropriate mappings between the attributes of the data sources and the attributes of the data warehouse for specifying the required transformations are carried out in the ETL mapping.

Ontology is an explicit formal conceptualization of some domain of interest [5]. They contribute on reducing the syntax and semantic conflicts that may occur during the data integration process. Ontologies are increasingly used in various fields such as Data Integration, knowledge management, information extraction and the semantic web. The explosion of semantic databases sources (SDB) such as ontologies has become candidate for building the semantic data warehouses (SDW).

Traditional database models such as E-R and UML models have the methods for designing the multidimensional schema [4]. But these models capture only the syntactic formats present in the domain and developer should expert's in domain knowledge for design the data warehouse. Now, ontology could be used in data warehouse development since they provide the crucial context knowledge relevant to interpret semantics [6]. As ontology represents common conceptualization of a domain it can solve syntactic and semantic conflicts. Another advantage of ontology compared to UML or ER models is that it can be used for querying and reasoning [6].

In the literature ontologies have been used for data warehouse schema design and for identifying ETL operations [2][3][4][7]. As the data sources that exist for a data warehouse are autonomous in nature, they may change their source schema. Hence, the ETL mapping between the data source and the data warehouse schema becomes invalid. Thus, it is necessary to handle the data source evolution and maintain the ETL mapping. In this paper an automated way of maintaining the ETL mapping using ontology has been proposed.

DOI: 10.15680/IJIRCCE.2016. 0407054

II. RELATED WORK

In this paper [9] the proposed approach used Business Process Modelling Language for modelling conceptual ETL workflows and mapping them to real execution primitives through the use of a domain-specific language. This allow for the generation of specific instances that can be executed in an ETL commercial tool. In [10] the authors proposed a method to predict the most likely exceptions that happen during ETL process and they tried to resolve it. They have provided a set of best practices and methodologies modeled as knowledge to the benefit of the ETL worker. They have also instantiated a prototype as an initial validation of their approach. In [11] a new ETL approach called TEL (Transform-Extract-Load). Based on data virtualization has been proposed. The TEL has the capacity of directly migrating data from multisource heterogeneous databases, without the traditional data staging area. The TEL approach reduces the workload of data migration that may never be used. In [12] the authors proposed a method that contains Two-ETL phases. One treats the pre-treatment phase and another deals with the actual ETL. This method consists on determining the correspondence table, modeling new operations using the Business Process Modeling Notation and implementing these operations with Talend Open Source (TOS). It also allows the design of ETL process in an earlier stage, which enormously facilitates the implementation of this process .In [13] the authors have proposed and developed a programmable framework called semantic ETL(SETL). SETL facilitates users to build a semantic Data warehouse. SETL uses ontology as an underlying schema to integrate heterogeneous data sources. Apart from traditional data format the SETL can process semantic-aware data. Hence, it stores the data as RDF triples and also allows to link the internal resources with external resources. Though the existing works focus on various issues in ETL design, they have not addressed the ETL mapping maintenance problem in case of data source evolution. This, the focus of the paper is to develop and maintain ETL mapping using ontology.

III. PROPOSED APPROACH

The objective of the proposed approach is to develop an automatic ETL mapping for the source and target ontology using wordnet based algorithm. Then identifying and specifying the ETL operations which are required for the actual transformation. Finally, maintaining ETL mapping when source evolves by analyzing the changes. Figure 1 represents the steps involved in the proposed approach.

The data source and the data warehouse schema are converted to the ontology format to represent source ontology and target ontology respectively. The conversion has been carried out to DB2OWL tool. The ontology are represented in owl format. Next, the mapping between the entities of the source and target ontology is derived using wordnet algorithm. This mapping represents the ETL mapping for the data source and the data warehouse schema. After the mapping process the ETL operations required for transformations are identified. When data source schema changes the corresponding source ontology is updated. A log is maintained in this approach in order to capture any changes happening in the source ontology. In order to analyze the change, the recent changes are extracted from the log. For a particular change, affected entities between the source and target ontology is identified. If required the corresponding mappings are updated between them. Finally for the new mapping, the ETL operations are updated. Following section provide the details of each step involved in the approach.

A. *Mapping using Wordnet*

Mapping the source and target ontology using wordnet based algorithm. WordNet also relates words based on colloquial usage, pertinence to other words. Wordnet is hardly the only electronic source of word relationships. Riwordnet algorithm has been used here [14]. Riwordnet is distance based algorithm.

```
RiWordnet wordnet = new RiWordnet(this, "d:\\Wordnet3.0\\");
```

Methods used in Riwordnet:

- getAllHyponyms() – Returns an unordered String[] of hyponym-synsets(each a colon-d delimited String), or null if not found.
- getAllHypernyms() – Return an order String[] of hypernym-synsets(each a semi-colon delimited String) up to the root of Wordnet for the 1st sense of the word, null if not found.
- getAllSynsets() – Return String[] of words in each synset for all senses of word with pos or null if not found.

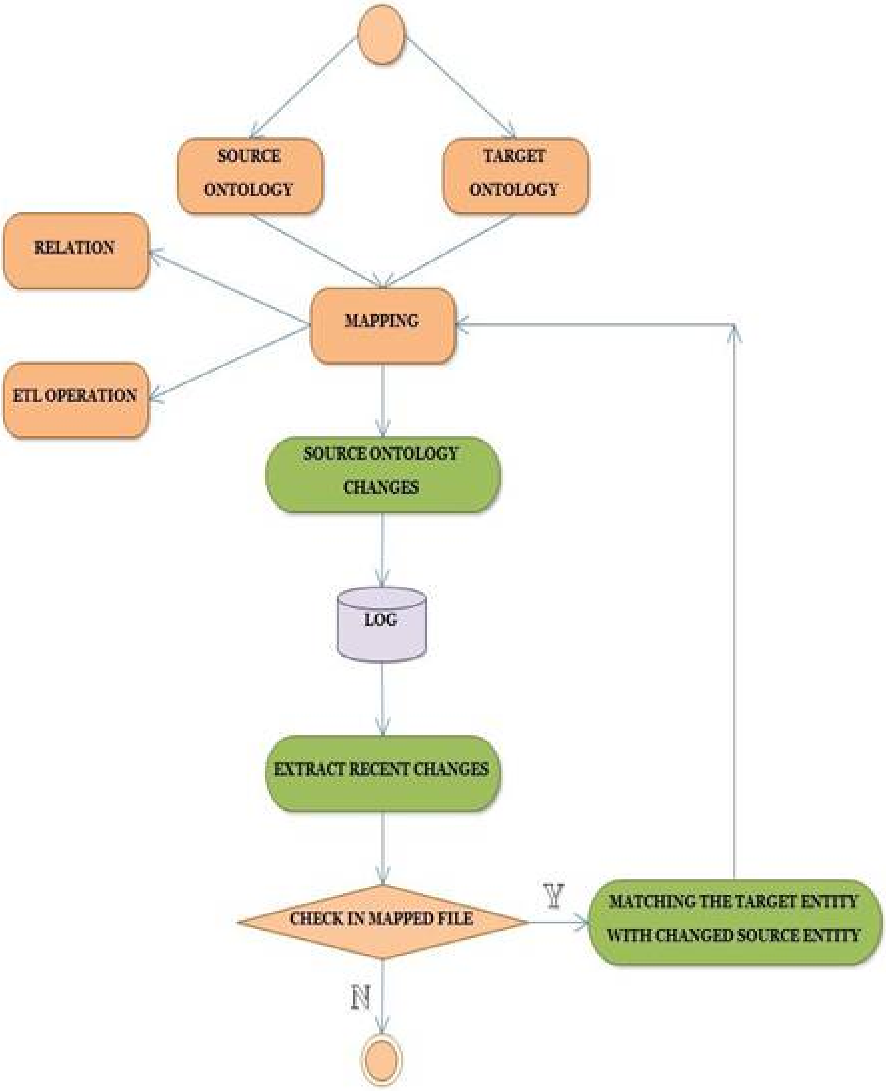


Fig 1. Steps of the Proposed Approach

A hyponym is a word that describes things more specifically. Hypernyms are words that refer to broad categories or general concepts. Synsets are interlinked by means of conceptual-semantic and lexical relations. Load the source and target ontology. Then check the corresponding concepts and properties of source ontology with target ontology using wordnet based algorithm. If any matches found then do the mapping between these two entities.

B. *Identifying ETL Operations*

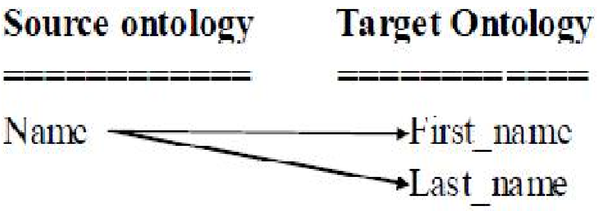
ETL operation done by mapping and relating the source and target ontology. Using the mapping value find whether the source entities is hyponyms(<=) or hypernyms(>=) or equivalent(=). Based on the relation ETL operation perform.

ETL operations:

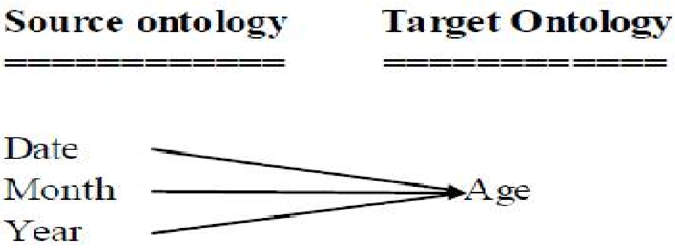
- Retrieve : Retrieves entities from a source element.
- Extract : Extract parts of entites from a source element
- Merge : Merges entities from several source elements
- Join : Joins entities from several source elements
- Filter : Selects entities based on a specified value

- Aggregate: Aggregates the value of an attribute
- Store : Stores entities to target element

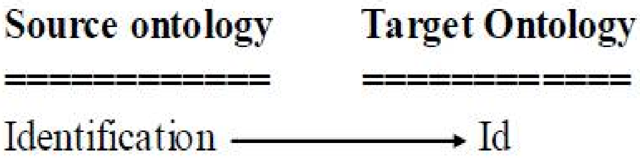
Ex:



Based on the above mapping perform the ETL split operation.



Based on the above mapping perform the ETL merge operation.



Based on the above mapping perform the ETL store operation.

C. *Capturing Source Changes using Log*

A number of changes, ranging from concepts to properties, can affect the ontology. So those changes are captured using log. The changes need to represented properly to correctly handle explicit and implicitly change requirements. To address this we use ChAO plugin.

First to configure this ChAO plugin in protégé to capture changes. It capture the changed details such as who did the change, which one is change, what action performed , which time change occurred and what is the status of the change. ChAO – Change Annotation Ontology. Using ChAO tab get the exact details about the changed entities of source ontology.

Methods used for extract details of changed entities:

- Change.getAuthor()
- Change.getAction()
- Change.getTimestamp().getDate()
- Change.getApplyTo().getComponentType()
- Change.getContext().

D. *Extraction of Changes*

Extract the recent changes from Change Annotation Ontology(ChAO). Following steps are used for the extraction process:

```
Project prj = Project.loadProjectFromFile(path, new ArrayList());
```

```
KnowledgeBase kb = prj.getKnowledgeBase();
ChangeFactory factory = new ChangeFactory(ChAOKbManager.getChAOKb(kb));
Collection<Change> changes = factory.getAllChangeObjects(true);
For(Change change:changes) { /get all changed details }
```

Select the option which kind of change want to get from source ontology such as class or attribute. Then get the specific changes such as delete, add, name_changed by using rules. We need to extract the specific details from the changed entity i.e. change.getContext() and then store it in an array. Afterwards check the source ontology changes with mapped file whether it affects the mapped file or not. If affected then it is searched with the target ontology.

E. *Mapping the Changed Entitiy*

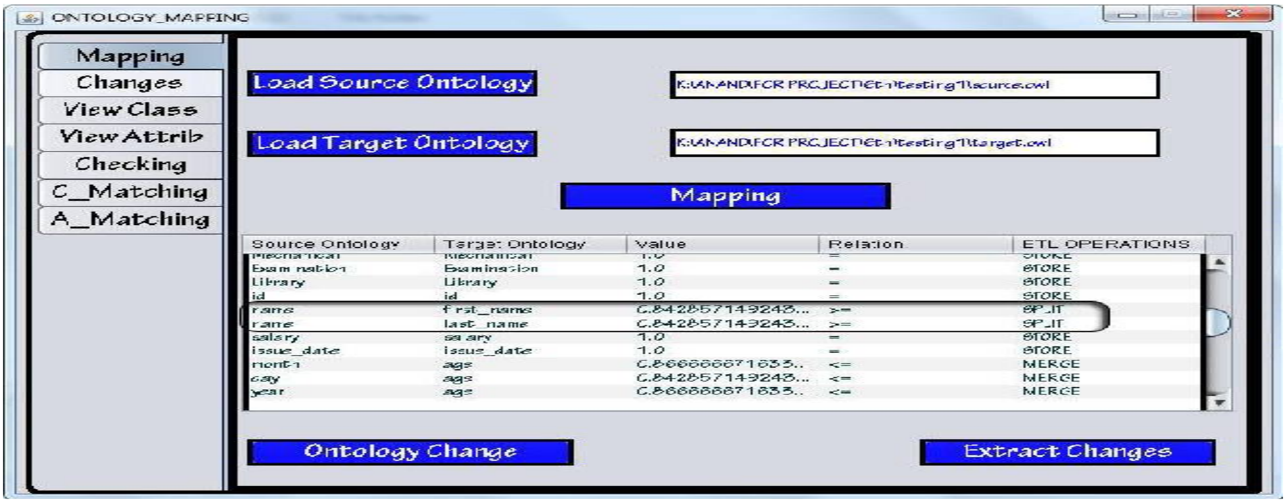
Using wordnet do the mapping between changed entity and target entity. For this to get the changed entity from source ontology using the log file. Then identify the matching between changed entity and target entity using wordnet. Wordnet use Riwordnet algorithm to match the two entity and give the value for the matched entity. For example: Create one class in source ontology. So the log file captured the changes and give the status i.e., class is created. Get the superclass of created class. Then find out whether that class already affected in mapped file or not. If affected then matched with target ontology using wordnet and get the value of the changed entity and target entity.

F. *ETL Operatinos for new Mapping*

Using the relation between changed entity and target entity identify the ETL operation for new mapping. Some of the relations are Hyponym(<=), Hypernym(>=) and Equilty(=). A hyponym is a word that describes things more specifically. Hypernyms are words that refer to broad categories or general concepts. Equilty describes the same meaning for different naming conventions. For example: Changed one of the attribute name in the source ontology. Those changes are captured in the log. Extract those changes using jena api. Then find out whether that changed name attribute already affected in mapped file or not. If affected then matched with target ontology using wordnet and get the value and relation of the changed name attribute and target attribute.

IV. IMPLEMENTATION DETAILS

The proposed approach has been developed in Net Beans environment using Java and Jena API. For editing the ontology protégé ontology editor has been used. First the source ontology and target ontology are loaded. Then the mapping between source and target ontology is computed. Using wordnet based algorithm the relation between source and target ontology are derived. Then using the relation the ETL operations are identified. When source changes after mapping, the recent changes are extreacted from the log using jena api and matching between changed entity and target entity are computed. Finally, relation and ETL operations for those changed entities are obtained. The sample screen shots of the proposed system has be given in figure 2 to figure 5.

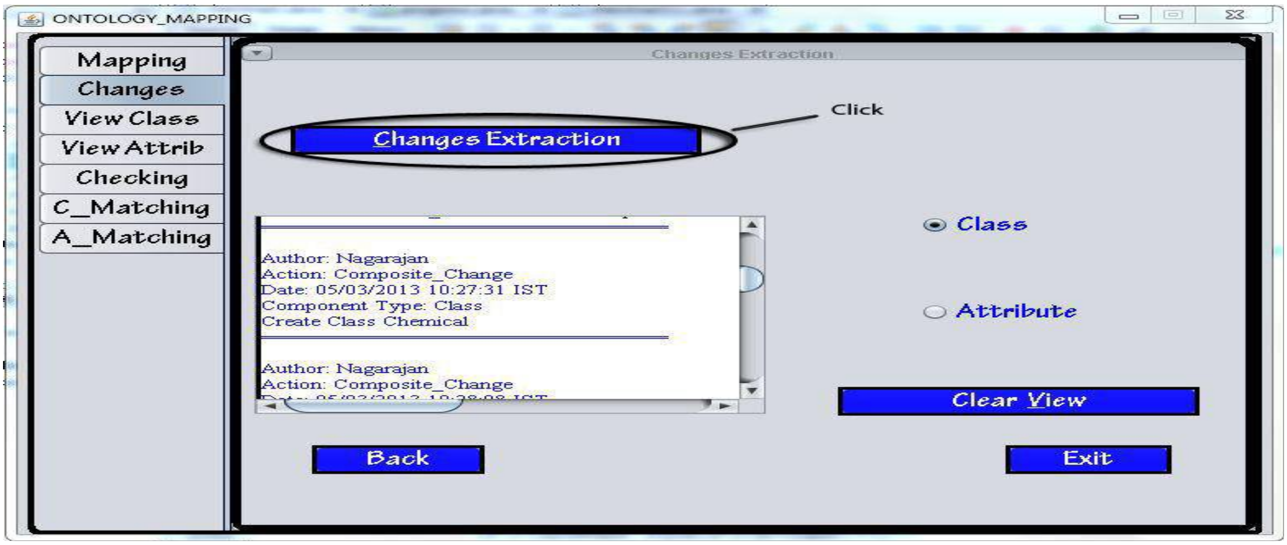


K:\ANAND\FOR PROJECT\Ext1\testing1\source.owl

K:\ANAND\FOR PROJECT\Ext1\testing1\target.owl

Source Ontology	Target Ontology	Value	Relation	ETL OPERATIONS
Mechanical	Mechanical	1.0	=	STORE
Examination	Examination	1.0	=	STORE
Library	Library	1.0	=	STORE
id	id	1.0	=	STORE
name	first_name	0.842857149243...	>=	SPLIT
name	last_name	0.842857149243...	>=	SPLIT
salary	salary	1.0	=	STORE
issue_date	issue_date	1.0	=	STORE
month	age	0.8666666671633..	<=	MERGE
day	age	0.842857149243...	<=	MERGE
year	age	0.8666666671633..	<=	MERGE

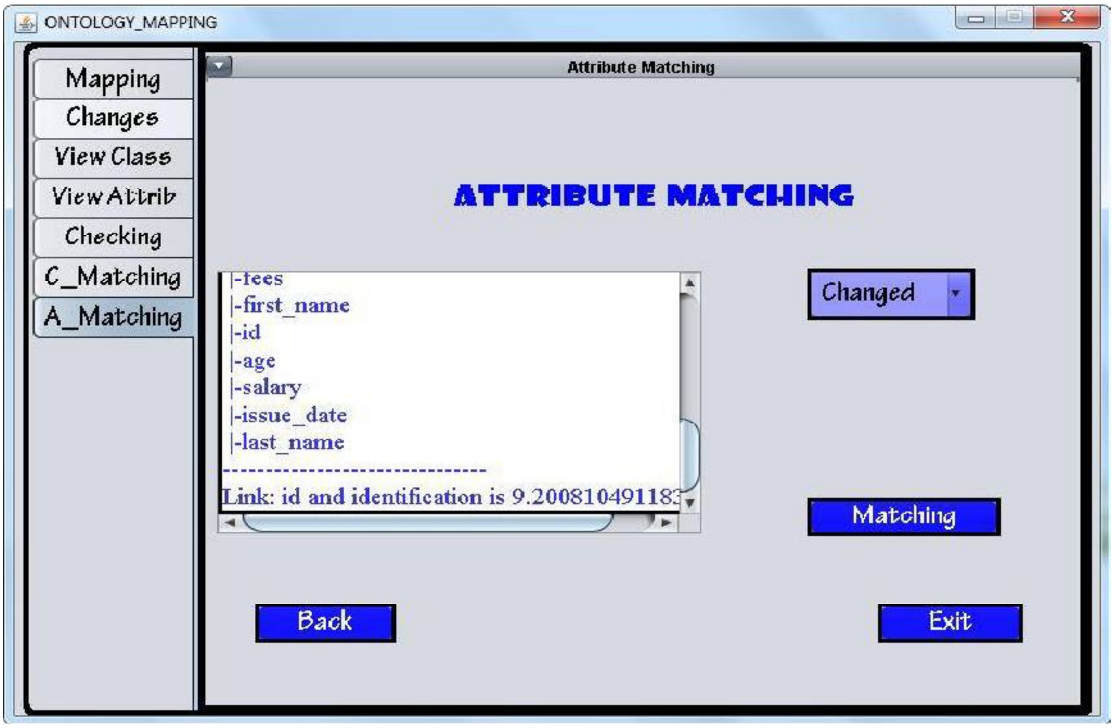
Fig. 2 Marking the ETL split operation before changing



Author: Nagarajan
Action: Composite_Change
Date: 05/03/2013 10:27:31 IST
Component Type: Class
Create Class Chemical

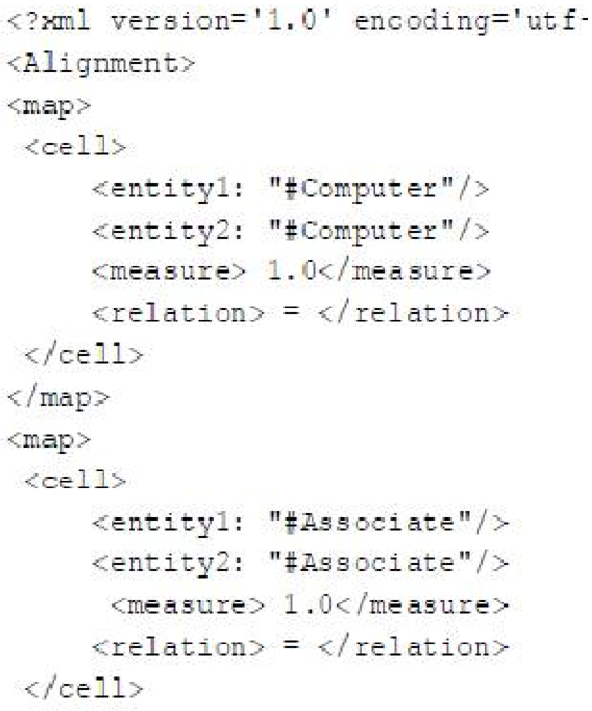
Author: Nagarajan
Action: Composite_Change

Fig. 3 Extraction of recent Change



```
|-fees
|-first_name
|-id
|-age
|-salary
|-issue_date
|-last_name
-------------------------------
Link: id and identification is 9.20081049118
```

Fig. 4 Matching the changed name attribute in source ontology with target ontology



```
<?xml version='1.0' encoding='utf-
<Alignment>
<map>
 <cell>
     <entity1: "#Computer"/>
     <entity2: "#Computer"/>
     <measure> 1.0</measure>
     <relation> = </relation>
 </cell>
</map>
<map>
 <cell>
     <entity1: "#Associate"/>
     <entity2: "#Associate"/>
      <measure> 1.0</measure>
     <relation> = </relation>
 </cell>
```

Fig. 5 Sample Mapping File in XML format

V. EXPERIMENTAL RESULTS

In order to analyse the effectiveness of the proposed approach different change set is considered. Each change set consists of elementary changes. The set of evolution operations or changes that are considered in the source ontology includes, addition of attributes and table, renaming of attributes and table, and, deletion of attributes and table. In Table 1, the results are summarized for different kinds of changes. It is observed that most of the activities are affected by attribute additions and renaming since these kinds of operations are the most common. Most important, it is observed that the proposed evolution approach can effectively adapt activities to the examined kinds of operations. A total of 126 evolution events or changes were considered. It could be observed from the Table 1 that out of 126 changes 124 mappings were automatically corrected using the proposed approach

Table 1 Results for different changes

Change Type	Total Affected	Total Corrected
Attribute Addition	45	44
Attribute Deletion	10	10
Attribute Rename	52	51
Table Addition	5	5
Table Deletion	2	2
Table Rename	12	12

Different events or changes on the ETL have a different impact on the overall effectiveness of our the proposed approach, as they vary both the number of the affected activities and the number of the adjusted activities on the mapping. The effectiveness of the proposed approach is derived using the following metrics:

$$Effectiveness = \frac{Number\ of\ Entities\ Corrected}{Number\ of\ Entities\ Affected}$$

Figure 6 shows the comparison of no. of attributes affected and that are corrected by using the proposed approach. It has been inferred that for a number of attribute changes in the source ontology, the proposed system could properly adjust the mapping. Thus the proposed system could achieve 98% effectiveness for attribute addition, 100% for attribute deletion and 98% for attribute rename. The comparison of number of tables affected and corrected by using the proposed approach is given in Figure 7. It has been observed from the figure that the system achieved 100% effectiveness for table addition, 100% for table deletion and 93% for table rename.

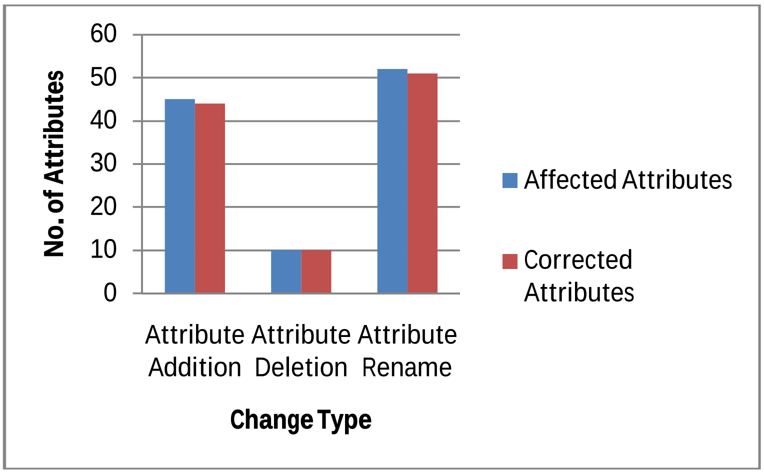


Fig. 6 No. of Attributes Affected and Corrected using proposed approach

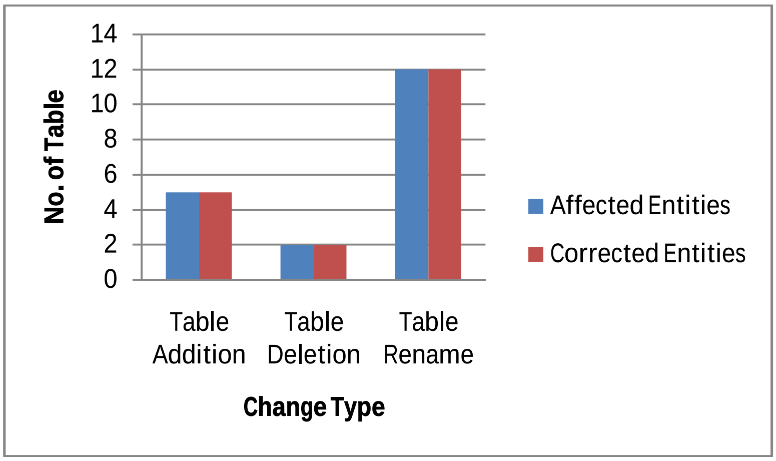


Fig. 7 No. of Tables Affected and Corrected using proposed approach

VI. Conclusion and Future Work

In this paper the problem of impact prediction of source schema changes in data warehouse environments has been addressed. The proposed work provides an efficient way to perform ETL mapping and ETL operations automatically using ontology. The proposed system could detect changes in schemata of data sources and adapt the ETL mapping accordingly. Hence, the designer need not have the full knowledge of different sources of a particular domain. If source evolves the proposed system automatically computes new mapping and also identify the ETL operation. This helps the

DOI: 10.15680/IJIRCCE.2016. 0407054

designer to modify the required transformations in the ETL work flow. In the future work, the focus will be on applying the ontology mapping maintenance approach for large ontologies.

REFERENCES

1. Inmon, William H. Building the data warehouse. John wiley & sons, 2005.
2. El-Sappagh, Shaker H. Ali, Abdeltawab M. Ahmed Hendawi, and Ali Hamed El Bastawissy. "A proposed model for data warehouse ETL processes."Journal of King Saud University-Computer and Information Sciences 23, Vol. no. 2, pp. 91-104, 2011.
3. Skoutas D, Simitsis A. Ontology-based conceptual design of ETL processes for both structured and semi-structured data. International Journal on Semantic Web and Information Systems, Vol .3(4), pp. 1-24, 2007.
4. Muñoz, Lilia, Jose-Norberto Mazón, Jesús Pardillo, and Juan Trujillo. "Modelling ETL processes of data warehouses with UML activity diagrams." In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pp. 44-53. Springer Berlin Heidelberg, 2008.
5. Pardillo, Jesús, and Jose-Norberto Mazón. "Using ontologies for the design of data warehouses." arXiv preprint arXiv:1106.0304, 2011.
6. Gruber, Thomas R. "A translation approach to portable ontology specifications." Knowledge acquisition 5, Vol no. 2, pp.199-220, 1993.
7. Romero, Oscar, Alkis Simitsis, and Alberto Abelló. "GEM: requirement-driven generation of ETL and multidimensional conceptual designs." InInternational Conference on Data Warehousing and Knowledge Discovery, pp. 80-95. Springer Berlin Heidelberg, 2011.
8. Kimball, Ralph, and Richard Merz. The data webhouse toolkit. Wiley, 2000.
9. Oliveira, Bruno, and Orlando Belo. "A Domain-Specific Language for ETL Patterns Specification in Data Warehousing Systems." In Portuguese Conference on Artificial Intelligence, pp. 597-602. Springer International Publishing, 2015.
10. Hamed, Imen, and Faiza Ghozzi. "A knowledge-based approach for quality-aware ETL process." In Information Systems and Economic Intelligence (SIIE), 2015 6th International Conference on, pp. 104-112, IEEE, 2015.
11. Guo, Shu-Sheng, Zi-Mu Yuan, Ao-Bing Sun, and Qiang Yue. "A New ETL Approach Based on Data Virtualization." Journal of Computer Science and Technology 30, Vol. no. 2, pp.311-323, 2015.
12. Nabli, Ahlem, Senda Bouaziz, Rania Yangui, and Faiez Gargouri. "Two-ETL Phases for Data Warehouse Creation: Design and Implementation." In East European Conference on Advances in Databases and Information Systems, Springer International Publishing, pp. 138-150, 2015.
13. Deb Nath, Rudra Pratap, Katja Hose, and Torben Bach Pedersen. "Towards a programmable semantic extract-transform-load framework for semantic data warehouses." In Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP, pp. 15-24, ACM, 2015.
14. Howe, D. C. Rita wordnet. Java based API to access Wordnet, 2009. [Online]. Available: http://www.rednoise.org/rita/wordnet/documentation/