ISSN 2047-3338

Ontology Evolution Methodology for Change Management

Khadim H. Asif[1,*], Syed Muhammad Ahsan[2] and M. Junaid Arshad[3]
[1,2,3]Department of Computer Science and Engineering,
University of Engineering and Technology, Lahore, Pakistan
asifkhad@yahoo.com

***Abstract*– So far various approaches and frameworks have been proposed to address the problem of ontology evolution management. They use different methods and techniques to tackle its various aspects like change management, inconsistency administration and incorporation of temporal information. Despite all the efforts made in this direction, the problem of ontology evolution management still needs to be resolved. Having identified the limitations and drawbacks associated the current existing approaches, we have proposed an approach which is based on mathematical model of ontology and makes use of an adjacency matrix to handle the changes in the concepts as well as in the relationships among the concepts of ontology.**

***Index Terms*– Ontology, Management, Approaches, Evolution and Matrix**

I. INTRODUCTION

ONTOLOGIES are used to describe and represent the knowledge of a domain. In the dynamic environment like web, domains change when changes occur in the universe of discourse, in the user's requirements, and in the view of the domain. The underlying ontologies which model these domains have to be modified in order to keep the knowledge up to date [1] [2] [3] [4], if these changes are not incorporated timely, then the reliability, accuracy and efficiency of the ontology-based applications are downgraded [5]. Hence, the management ontology evolution is an essential requirement for retrieving useful and time dependent information from ontology-based systems.

To handle the critical issue of ontology evolution management, researchers have devised various approaches and frameworks [4] [6] [7] [8] that use diverse methods and techniques. Some of them have tried to address the problem by using the works already conducted in the area of database schema evolution and schema versioning [9] [10]. Others have borrowed techniques, methods, tools and ideas from belief change [11] and revised the concepts of belief change in order to apply it to the ontology evolution management. Still others have made use of the concepts and ideas from the database management systems to create ontology-based management systems.

In spite of the plethora of approaches and frameworks devised so far, the task of management of changes in the underlying ontologies has still not been resolved efficiently as well as effectively. It our opinions, the main obstacles in the way to the solution of the problem are the inappropriate techniques and methods used to handle them. In this paper we have used a different technique to mange the changes that cause the relationships to be inconsistent in the underlying ontology. Our technique uses adjacency matrix of the weighted directed acyclic graph of the ontology that needs to be up to date.

The remainder of this paper is organized as follows. In Section 2, a survey of the ontology evolution approaches and their analysis are given. Details of the proposed methodology are given in Section 3. Section 4 narrates the discussion of the methodology. Finally, in Section 5, we give concluding remarks and future work

II. LITERATURE SURVEY

Ontology evolution has been an unavoidable requirement for an ontology-based application [12]. Researchers have proposed a wide spectrum of approaches and frameworks to deal with the critical issue of ontology evolution management. A number of researchers have tried to handle it by devising various tools.

OntoView [13] is a web-based tool that helps users to manage changes in ontologies. It has a transparent interface to various versions of ontologies. It maintains not only the transformations among different versions but also the conceptual relations among concepts in these versions. In addition, it also keeps various versions of ontologies interoperable. OntoManager [14] is ontology management tool that guides users to modify ontology when users' needs are changed.

The authors have proposed a six-phase ontology evolution approach [15], its phases are: the change capturing, the change representation, and the semantics of change, the change

Khadim H. Asif, Associate Professor, Department of Computer Science and Engineering, U.E.T., Lahore-Pakistan, (Email: asifkhad@yahoo.com).

Dr. Syed M. Ahsan, Associate Professor, Department of Computer Science and Engineering, U.E.T., Lahore-Pakistan, (Email: ahsancs@hotmail.com).

Dr. M. Junaid Arshad, Assistant Professor, Department of Computer Science and Engineering, U.E.T., Lahore-Pakistan, (Email: junaidarshad@uet.edu.pk).

implementation, the change propagation, and the change validation.

In [13] the researchers have proposed a component-based framework for handling changes in ontology. It can represent not only various representations of the changes but also integrates these representations. It is has a number of salient features such as ontology update, data transformation, reasoning, verification of data and approval of data.

Klein and Noy [3] are of the view that the information regarding the ontology changes can be represented in a number of ways. The framework proposed in [16] represents and integrates different representations of the changes.

Ontology change management approaches described in [15] and [17] help identify the semantics of change and ensure consistency in achieving the semantics of change problem. In their procedural approach [15], the researchers have presented the concept of evolution strategy that ensures the evolution process according to the users' requirements. This approach enables the users to customize the semantics of change. The declarative approach described in [17] model ontology evolution as reconfiguration-design problem solving. It reduces the problem to a graph search where the nodes act as evolving ontologies and the edges describe the changes which map the source node into the target node.

To handle the problem of ontology evolution management, the researchers have proposed an *evolution log* [14] that is based on *evolution ontology* for the KAON ontology model. The evolution ontology described in this approach deals with different kinds of changes, dependencies among changes and the decision making process.

Ontology evolution approach identified in [17] not only manages dependent but also distributed ontologies.

The problems that the users have to face handling ontologies in the distributed environments like the Web are narrated in [18]. Some researchers have made use of the research already conducted in area of database schema versioning as well as schema evolution to handle the problem of ontology evolution. In [18] the researchers have identified the problems associated with ontology versioning. In [18], the researchers have compared the evolution of ontologies with that of database schemas.

In spite of all the approaches, frameworks and tools that are devised to tackle the problem of ontology evolution, the problem still needs to be explored. We have observed that most of the approaches that deal with the change management of ontology are either complex or give birth to new problem of inconsistency. Moreover, the management of relationships among the concepts of ontology is not handled in the appropriate and effective way. We have proposed a methodology to address the problem of ontology evolution. Our methodology makes use of the graph to represent ontology. The graph is stored as an adjacency matrix. All changes to concepts as well as relationships of ontology are made through the adjacency matrix.

III. PROPOSED WORK

To overcome the drawbacks and limitations associated the current change management approaches described in Section 2 we have proposed a methodology to handle the critical issue of change management in ontology evolution. Our technique is based on mathematical model of ontology and makes use of the adjacency matrix of a weighted directed acyclic graph for the given ontology. Mathematical model of ontology is defined as:

$$O=\{C, R\} \quad \ldots\ldots\ldots (1)$$

In Equation (1), O is the ontology name, C is a set of concepts (nodes of the ontology graph) and R is a set of relationships (edges of the graph) among the concepts (nodes of the graph). These two sets, i.e., C and R, are further defined as follows:

$$C = \{c_1, c_2, c_m\} \quad \ldots\ldots\ldots\ldots (2)$$

In Equation (2), we assume that there is m number of concepts in the ontology. Now we define the set R of relationships among the concepts given in Equation (2) as follows:

$$R= \{R_I \cup R_A, \cup R_T, \cup R_S, \cup R_{AS}\} \ldots (3)$$

In Equation (3), R_I, R_A, R_T, R_S, *and* R_{AS} are sets of relationships such as *.is-a* (inheritance), *has-a* (aggregation) transitive, symmetric and association relationships among the concepts of the ontology, respectively.

These sets of relationships can be written as follows:

$$R_I = \{R_{I1}, R_{I2}, \ldots, R_{Ij}\} \ldots\ldots\ldots\ldots (4)$$

$$R_A = \{R_{A1}, R_{A2}, \ldots, R_{Ak}\} \ldots\ldots\ldots\ldots (5)$$

$$R_T = \{R_{T,1}, R_{T,2}, \ldots, R_{Tl}\} \ldots\ldots\ldots\ldots (6)$$

$$R_S = \{R_{S1}, R_{S2}, \ldots, R_{Sr}\} \quad \ldots\ldots\ldots (7)$$

$$R_{AS}= \{R_{AS1}, R_{AS2}, \ldots, R_{AS}\} \ldots\ldots\ldots\ldots (8)$$

After we have defined ontology in terms of concepts and their relationships, we create a weight directed acyclic graph of the ontology in which nodes represent the concepts of the ontology and its edges the relationships between the concepts. To store the graph, we create an adjacency matrix... the rows and column headings represent the concepts and elements stored at the intersection of row and column the relationships among the concepts.

To manage the changes in the relationships we modify the value of the element in the adjacency matrix. In order to handle changes in the concept, we alter the concepts and rearrange them.

IV. DISCUSSION OF THE METHODOLOGY

Our methodology as compared to the others described in Section 2 is simple and easy to use. Its visual support helps users analyze the relationships among concepts within ontology. Through this methodology, we are able to resolve ontology mismatching problems.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a technique to tackle the problem of change management pertaining to ontology. This methodology is based upon the adjacency matrix which is created from the weighted directed acyclic graph of the ontology that needs to be modified. The most salient feature of this methodology is that it helps users manage the relationships among the concepts of ontology in an excellent way providing the opportunity to visualize the change management process. In addition, it will prove to the most effect tool for analyzing the relationships of the ontology.

In the future, we are going to enhance the technique to incorporate temporal information to maintain the history of changes in the ontology.

REFERENCES

[1] Stojanovic, L., Maedche, A., Stojanovic, N.and R. Studer, Ontology Evolution as Reconfiguration-Design Problem Solving. In KCAP, pp.162-171. ACM, October 2003.

[2] Plessers, P. and Troyer O, Ontology Change Detection Using a Version Log, In Proceedings of the 4th International Semantic Web Conference, Proceedings. Lecture Notes in Computer Science 3729 Springer, ISBN 3-540-29754-5, Galway, Ireland, 2005.

[3] Noy, F., Klein. M., Ontology Evolution: Not the Same as Schema Evolution. Knowledge and Information Systems, 6(4):428-440, 2004.

[4] Parsia, B., Sirin, E., Kalyanpur, A. Debugging owl Ontologies, in Proceedings of the 14th International World Wide Web Conference (WWW2005), Chiba, Japan, 2005.

[5] McGuinness, D., Fikes, R., and Rice, J., An environment for merging and testing large Ontologies, In Proceedings of KR-2000, Principle of Knowledge Representation and Reasoning, Morgan-Kaufman, 2000.

[6] Flouris, D. Plexousakis, G. Antoniou. Evolving Ontology Evolution. In Proceedings of the 32nd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-06), Invited Talk, 2006.

[7] Quan T.T., Hui S.C., Fong A.C.M. and Cao H.T, (2006): Automatic Fuzzy Ontology Generation for Semantic Web, IEEE Transactions on Knowledge and Data Engineering, 18(6): 842- 856, 2006.

[8] Novacek, V., Laera, L., Handschuh, S.: Semi-automatic integration of learned ontologies into a collaborative framework. In: IWOD (2007), Data Engineering, 18(6): 842-856, 2006.

[9] Zablith, F., Sabou, M., d'Aquin, M., Motta, E.: Using background knowledge for ontology evolution, In IWOD Workshop at ISWC (2008).

[10] Sabou, M., d'Aquin, M., Motta, E., Exploring the Semantic Web as Background Knowledge for Ontology Matching, Journal on Data Semantics XI, (2008).

[11] Michel, K. Change Management for Distributed Ontologies. PhD thesis, Vrije Universities Amsterdam, 2004.

[12] Klein, M., Fensel, D., and Kiryakov, A. Ontology versioning and change detection on the web. In 13th InternationalConference on Knowledge Engineering and Knowledge Management (EKAW02), number 2473 in LNCS, page: 197, (2002).

[13] Stojanovic. L. Methods and Tools for Ontology Evolution, PhD Thesis, University of Karlsruhe, 2004.

[14] Stojanovic. N. Stojanovic, L., Evolution in the ontology-based knowledge management system, To appear in Proceedings of the Xth European Conference on Information Systems - ECIS 2002, Gdañsk, Poland, 2002.

[15] Michel Klein and Natalya F. Noy. A component-based framework for ontology evolution. In Proceedings of the Workshop on Ontologies and Distributed Systems, IJCAI '03, Acapulco, Mexico, August 9, 2003.

[16] Maedche, A., Motik, B., and Stojanovic. L. Managing multiple and distributed Ontologies in the semantic web, VLDB Journal, 12(4): 286–302, 2003.

[17] Heflin, J., and Hendler, J. Dynamic Ontologies on the web, In Proceedings of the Seventeenth National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, pages 443–449, The MIT Press, 2000.

[18] Klein, M., and Fensel, D. Ontology versioning for the Semantic Web, In Proceedings of the First International Semantic Web Working Symposium (SWWS), pages: 75–91, (2001).