A Critical Analysis of the Existing Ontology Evolution Approaches

Khadim H. Asif, Syed Muhammad Ahsan

Department of CS & Engg, U. E. T., Lahore, Pakistan
asifkhed@yahoo.com, ahsansc@hotmail.com

Abstract: Because of the rapid improvements occurring in the dynamic environment of web applications, ontologies have to be modified to reflect the changes made to the applications. Management of the changes within ontologies is one of the most crucial tasks that needs to be resolved. Various approaches and frameworks have been devised by the researchers to handle it. Despite all the efforts made in this direction, the problem still requires to be researched. To address the problem, we have critically analyzed a number of existing ontology evolution approaches against a criterion we have defined in this paper. Having identified the limitations and weaknesses along with their strengths, we have proposed some requirements that must be incorporated in the design of ontology management approaches.
[Khadim H. Asif, Syed Muhammad Ahsan. A Critical Analysis of the Existing Ontology Evolution Approaches. Journal of American Science 2011;7(7):584-588]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: Ontology, Semantic Web, Evolution, management.

1. Introduction

The current web is unfathomable sea information. Because of huge volume of its content, it is hard to relate and retrieve relevant information from various dispersed sources and to extract knowledge efficiently as well as effectively. Having explored the limitations and drawbacks of the current web, Tim Berners-Lee, the creator of World Wide Web, floated the idea of Semantic Web [30]. With the advent of vision of Semantic Web, the researchers started developing ontologies to overcome weaknesses of the current web.

Ontologies have now been an integral part of the Semantic Web. They are used to model the real world. They encode knowledge of a particular domain and provide a shard understanding of conceptualization to exchange information among different applications as well as machines [19].They are being used to cope with heterogeneous representations of web resources, providing a common understanding of a domain to be shared among human beings and software agents. Ontologies provide not only semantics but also the structure to content of the domains. They enable content-based access, interoperability among different applications and communication between various agents [8].

Because ontologies capture the domain knowledge, they have to be modified when changes occur in the real world. In order to sustain the compatibility between ontology and its corresponding domain, the management of such changes is a critical task and it is referred to as the ontology evolution.

In order to tackle the crucial task of ontology evolution, the researchers have proposed many approaches and frameworks [20] [21] [22]. Despite all the efforts made to overcome the limitations and drawbacks of the existing approaches, the problem of ontology evolution still needs to be addressed.

In this paper we have critically analyzed some of the existing ontology evolution management approaches to explore both their strengths and weaknesses so as to seek the better solutions to the problems associated with them. On the bases of the outcome of our analysis, we have proposed the requirements that a new approach must have for better results.

The remainder of this paper is organized as follows. In Section 2, a review of the existing ontology evolution approaches. In Section 3, we have critically analyzed the approaches reviewed in Section 2. Details of the proposed requirements for the design of the new approaches are described .in Section 4. Finally, in Section 5, we give concluding remarks and future work

2. Literature Survey

Various researchers groups have been working on the critical problem of ontology evolution in order to cope with the changes occurring in the ontology-based applications. Some researchers have made use of the research already conducted in the field of database schema evolution and schema versioning [23] in their approaches. Others have borrowed the concepts and principles from the field of belief change [17] in their research to handle the task. Still others, inspired from natural evolutions, have insisted on adopting biological concepts in the ontology evolution approaches [31].

Noy and Klein have identified two modes of ontology evolution: *traced* and *untraced* [25]. In traced mode; the evolution is defined in terms of series of changes made to the ontology. After every change operation, the effect of each change on the instance data as well as on the depending artifacts is observed. Finally, the resulting effect is evaluated by combining these operations. In the untraced mode, the differences between two versions of the ontology are determined in a semi-automated way.

In [26], the researchers have categorized the ontology evolution approaches into two types: *modification-centered* and *fact-centered*. In the former approaches, the fact causing the change is unimportant. The actual change that is to be performed is fed into the system. In the later one, a fact rather than the actual change is entered into the system which identifies the change to be performed.

Some approaches [27] [28] deal ontology evolution on the instance level as well as on the conceptual level. An event-based approach described in [29] supports evolution on the instance level. In [29], the authors have discussed manual as well as computer-based approaches. In the next section we have a look on the various existing ontology management approaches in order to learn about their salient features. This will also help us identify the limitations and shortcomings associated with them.

2.1. Existing Ontology Evolution Management Approaches

In this section, we narrate various existing ontology evolution management approaches proposed by different researcher groups.

2.1.1. Ontology Evolution Approach based on Documents Clustering

In [22], the authors have proposed a semi-automatic approach based upon documents clustering for the ontology evolution. It uses the model introduced by Motik et al in [23] based on the THESUS system [24]. The THESUS system organizes web documents as semantic clusters. Its clustering algorithm not only updates the domain of the real world but also utilizes a hierarchical thesaurus to map the web documents to ontology concepts. The Document Clustering approach suggests to the user four types of changes: new concept insertion, new instance insertion, concept merge and concept decline. The approach handles elementary/simple as well as composite changes to a single ontology only.

2.1.2. Ontology Evolution Approach Based on Version Log

To tackle the problem of change management in the underlying ontologies, the researcher has proposed an approach that maintains a *version log* [25] to keep all versions of each concept ever defined in the ontology from its creation until its final stage. In this approach a change request is formulated in term of simple and composite changes which is made to the ontology whenever the ontology engineer requires. Changes to ontology are defined by using a language called Change Definition Language. The approach has the following limitations:

(i) The language used to represent changes is limited in its expressiveness as well as in scope.
(ii) It does not support complex dependencies that exist among the concepts of multiple ontologies.
(iii) It lacks support for collaborative development. Only a single engineer is responsible for the management of ontology.
(iv) The tools developed using this approach do not cover all aspects of the ontology evolution.

2.1.3. Component-based Approach

This approach described in [26] comprises of a meta-ontology of change operations, complex change operations, transformations sets and the specification of relations between different ontology versions. The Approach not only identifies changes but also provides semantic specifications for the changes amongst different versions. The methods described in the framework can find changes when only two versions of the ontology are available. The other limitation is that it does not handle the problem of inconsistency after changes are made to the ontology.

2.14. Six- Phase Ontology Evolution Approach

This is the most popular approach for ontology evolution management. It is proposed and adopted by a number of researchers for KAON ontology language [14] [15] [12] [16] [7] [23]. Change capture phase, change representation phase, semantics of change phase, change propagation phase, change implementation phase and change validation phase constitute this approach. The Six-Phase Ontology Evolution Approach not only tests consistency but also handles inconsistency in the ontology. The drawback associated with this approach is that it does not provide mechanism for detecting complex changes which are defined in terms of simple changes.

A number of researchers have extended the six-phase ontology evolution in [27] [11] to enable it to support the evolution for OWL ontology language as well. The extended approach not only tests consistencies but also resolves all types of inconsistencies: structured, logical and used-defined. It also supports collaborative and usage-driven evolution of ontologies. In addition, annotation of both concepts and axioms of ontologies is also possible that helps suggest ontology changes to other users.

2.1.5. Ontology Evolution Approach based of Belief Change

The approach [17] makes use of the extensive research already conducted in the mature field of belief change. It borrows the techniques, methods, tools and ideas from belief change and applies them to the handling of ontology evolution. Belief change deals with the automatic adaptation of knowledge base to new knowledge [28] without human participation. The major drawback of using belief change is that we cannot directly apply its theories to ontology evolution because such approaches focus on classical logic.

2.1.6. Ontology Evolution through Editors

Not all existing editors are capable of handling changes to ontology. There are special types of editors dedicated for this task. In their paper [29], the authors presented a comparison of three well known ontology editors: Protégé, Onto Edit, and OilEd. They described both strengths and drawbacks of each one. They also suggested and expounded the following requirements that an ontology editor handling ontology evolution must possess; (i)Functional requirements, (ii)User's supervision requirements, (iii)Transparency requirements, (iv)Reversibility requirements, (v)Auditing requirements, (vi)Ontology refinements Requirements, and (vii)Usability requirement. It is of the view of the researchers that editors having these requirements would perform well.

2.1.7. Event-Based Approach

The event-based approach discussed in [30] handles ontology evolution on the instance level only and deals with a single ontology. It maintains two types of ontologies: evolution ontology and event ontology. The former contains a log of changes and the later a log of event types and events itself. When a change is made to instance base, the evolution ontology is updated to keep track of the changes. The approach utilizes event types to maintain consistency between an instance base and depending artifacts. It supports evolution on the instance level. It keeps track of changes and resolves inconsistencies.

2.1.8. A Process-based Approach

In [16] the authors have attempted to resolve the issue of ontology evolution by proposing a process-based approach that focuses on ontology consistency problem under complex changes during evolution... The authors have introduced the notion of ontology strategy that lets users customize the process according to their requirements. The researchers also have identified a set of design requirements [31] for ontology evolution. The approach not only analyzes the requirements but also drives an evolution process that fulfills them. Moreover, it suggests users to make additional changes in the ontology so as to make it more suitable for user's needs. It also supports semi-automatic discovery of changes to ontology

3. Critical Analysis of the Existing Ontology Evolution Approaches

Before we analyze the existing ontology evolution approaches, we need to define the criteria against which we can evaluate them. Following are the questions we have proposed for their evaluation:

Does the approach resolve the elementary changes?
Does the approach handle composite changes?
Does the approach detect inconsistency?
Does it resolve inconsistency?
Does it support at instance level?
Does it support at concept level?
Does it support single ontology evolution?
Does it support multiple ontology evolution?
Does it handle the problem manually?
Does it handle the problem semi automatically?

Having set the criteria, we now evaluate the existing ontology evolution approaches against it. To show the results of our findings, we have constructed a table. The columns headings contain various approaches to be evaluated and row headings give the criteria to be used. The value "yes" in the element indicates that the approach in the column meets the criterion in the row. The value blank indicates that the approach in the column does not meet the criterion in the indicated row. So we can visualize the strength that each approach can have in Table 1.

In the approaches described above, the authors have used different methods to capture the changes made to the underlying ontology. Some of them use version log to store changes. Others create ontology log for this purpose. These methods are not only complex but also impractical in a dynamic and de-centralized environment. Some of the approaches do not handle the problem of insistency which causes retrieval of incorrect information.

In these approaches no proper mechanism is devised to incorporate temporal information. To detect changes, some of them have to create a number of versions of the same ontology which makes the process inefficient.

Table 1: Analysis of ontology evolution approaches

Approaches / Criteria	Document Clustering	Version log	Component Based	Six - Phase	Belief Change	Using Editors	Event-based	Process based
Elementary Change Detection	yes	yes	yes	yes	yes	yes	Yes	yes
Composite Change Detection	yes	yes	yes	yes	yes	yes	Yes	
Support to detect Consistency		yes	yes	yes	yes	yes	Yes	yes
Support to resolve consistency	yes							
Support for Instance Level		yes	yes			yes	Yes	
Support for Concept Level	yes	yes	yes	yes	yes	yes	Yes	yes
Supporting Single Ontology	yes	yes	yes	yes	yes	yes	Yes	yes
Supporting Multiple Versions of Ontology			yes	yes				
Manual handling	yes	yes	yes	yes		yes	Yes	yes
Semi-Automatic	yes	yes			yes			

4. Requirements for the Future Approaches

Having learned the flaws in the existing approaches, we suggest the following requirements that would help design more appropriate mechanisms for the management of changes in the underlying ontologies:

- To revolve inconsistencies, we suggest that the visual mechanism be adopted.
- To detect and handle changes in the ontology, the mechanism must be as simple as possible.
- To keep the history of changes, temporal mechanism should be devised for ontology evolution approaches.
- In order to tackle the evolution process efficiently, the number of versions of the same ontology should be avoided.

5. Conclusions and Future Work

In this paper we have presented the review of the some of the existing ontology evolution management approaches to summarize the works done in this field so that the reader or researcher could learn the state of the art of the research to further it. We have defined criteria against which we made extensive and critical analysis of the existing approaches in order to explore new dimensions towards better solutions to the problem of ontology evolution. In the end we have suggested some requirements that each approach must have to accomplish the required results.

In the future, we are going to develop a new system of ontology evolution management that would not only address change management but also keep track of the events over time.

References

1. T. Bray, J. Paoli, and C.M. Sperberg- McQueen (eds.): Extensible Markup Language (XML)
2. J. McHugh, S. Ambitious, R. Goldman, D. Quass, and J. Widom: Lore: A Database Management System for Semi structured Data. SIGMOD Record, 26(3):54-66, September 1997
3. J. Robie, J. Lapp, and D. Schach: XML Query Language (XQL). In: Proceedings of the W3C Query Language Workshop (QL-98), Boston, December, 3-4 1998 http://www.w3.org/TandS/QL/QL98/pp/xql.html
4. J. Robbie (ed.): XQL (XML Query Language). Working draft. August 1999.
5. A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu: XML-QL: A Query Language for XML. *W3C Note*, August 19, 1998. http://www.w3.org/TR/NOTE-xml-ql/
6. D. Florescu, A. Deutsch, A. Levy, M. Fernandez, D. Suciu: A Query Language for XML: *in Proceedings of Eighth International World Wide Web Conference* (WWW'8), 1999
7. R. Goldman, J. McHugh, and J. Widom: From Semistructured Data to XML: Migrating the

Lore Data Model and Query Language. In: *Proceedings of the 2nd International Workshop on the Web and Databases (WebDB '99)*, Philadelphia, Pennsylvania, June 1999. ftp://db.stanford.edu/pub/papers/ xml.ps
8. Fensel, D. (2001). Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag. Berlin.
9. Y. Ding and S. Foo. Ontology Research and Development, Part 2 - A Review of Ontology Mapping and Evolving. *Journal of Information Science, nr 25, pp. 375-388*, 2002.
10. M. Fernandez, A. G´omez-P´erez, and N. Juristo METHONTOLOGY From Ontological Art Towards Ontological Engineering. In *Symposium on Ontological Engineering of AAAI. Stanford (California)*, 1997.
11. V. Sugumaran and V. C. Storey. Ontologies for conceptual modeling: their creation, use, and management. *Data & Knowledge Engineering*, 2002.
12. M. Uschold. Building Ontologies: Towards A Unified Methodology. In *Proceedings Expert Systems 96, Cambridge*, 1996.
13. R. Navigli, P. Velardi, and A. Gangemi. Ontology Learning and Its Application to Automated Terminology Translation. *IEEE Intelligent System, vol. 18, nr. 1, pp. 22-31*, 2003.
14. Uschold, M. (2000). Creating, integrating and maintaining local and global ontologies. In. *Proceedings of the First Workshop on Ontology Learning (OL-2000) in conjunction with the 14th European Conference on Artificial Intelligence (ECAI 2000)*. August, Berlin, Germany.
15. Guarino N. (1998). Some Ontological Principles for Designing Upper Level Lexical Resources. *Proc. of the First International Conference on Lexical Resources and Evaluation*, Granada, Spain, 28-30 May 1998.
16. Uschold, M. & Gruninger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2), 93-155.
17. Hwang, C. H. (1999). Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. In. *Proceedings of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99)*, Linköping, Sweden, July 29-30, 1999.
18. Fernandez-Lopez, M. (1999). Overview of methodologies for building ontologies. In., *Proceedings of IJCAI-99 Workshop on Ontologies and Problem-Solving Methods*
19. D. Fensel, *Ontologies: Dynamics Networks of Meaning*, In Proceedings of the 1st Semantic web working symposium, Stanford, CA, USA, July 30th-August 1st, 2001
20. M. Klein. *Change Management for Distributed Ontologies*. PhD thesis, Vrije Universiteit Amsterdam, 2004.
21. N.F. Noy and M.A. Musen. The prompt suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
22. L. Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, Uiversity of Karlsruhe, 2004.
23. RODDICK, J.F., *A Survey of Schema Versioning Issues for Database Systems.* Information and Software Technology, 37(7): pp383-393
24. Flouris, G., Plexousakis, D., & Antoniou, G. (2006). Evolving ontology eolution. *SOFSEM 2006*: Merín, Czech Republic, pp14-29. Available online at: www.ics.forth.gr/isl/publications/paperlink/fgeo_SOFSEM06_IT.pdf
25. Natalya F.Noy and Michel Klein. Ontology evolution : Not the same as schema evolution. *Knowledge and Information Systems*, 5, 2003.
26. G. 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.
27. Klein, M., Noy, N. F.: A Component-based Framework for Ontology Evolution. *Proceedings of the Workshop on Ontologies and Distributed Systems* (IJCAI '03) Acapulco Mexico (2003)
28. Maedche, A., Motik, L., Stojanovic, L., Studer, R., Volz, R.: Ontologies for Enterprise Knowledge Management. IEEE Intelligent System 18(2) (2003) 26-34
29. Peter Plessers, Olga De Troyer, Sven Casteleyn: Event-Based Modeling of Evolution for Semantic- Driven Systems. CAiSE 2005: 63-76
30. J. Hendler, T. Berners-Lee, and E. Miller. The semantic web. Scientific American, May 2001.
31. O'Brien P. (2006): Modeling Intelligent Ontology Evolution Using Biological Evolutionary Processes, *Engineering of Intelligent Systems, IEEE International Conference*, 2006.

6/28/2011