Semantic Versioning of In-Process Scientific Document

Imran Asif, M. Karim

Semantic Versioning of In-Process *Scientific Document*

Imran Asif[1] and M. Shuaib Karim[2]

[1] Department of Computer Sciences, Quaid-i-Azam University, Islamabad, Pakistan
imranasifquaidian@gmail.com
[2] Department of Computer Sciences, Quaid-i-Azam University, Islamabad, Pakistan
skarim@qau.edu.pk

Abstract. The development of scientific documents is an iterative process. Scientific documents go through a continuous informal review phase during writing process and as a result keep changing. The informal review changes are casually recorded. The key issue for maintaining the changes in scientific document is to maintain the review history of individual components within source file at component level. Scientific document is meaningfully organized and it can be easily transformed into an ontology. For this purpose, we use Document Ontology to map each component of the scientific document and manage changes in this ontology by enhancing an already existing technique of semantic repository versioning. In this paper, we explore document change process using semantic versioning and provide the review comments history along each change. In addition, we define a usage scenario to present the viability and benefit of our approach. To achieve this, we developed a prototype system which represents the meaningful track of change in individual components of a scientific document, provides the review comments history along each change and at the end of document writing the author can see the progress report.

Keywords: Scientific Document, Ontology, Semantic Versioning

1 Introduction

Suzanne Briet [1] describes *Document* as an entity that is used to organize the physical evidence and to record the textual representation. It is also defined as a piece of written, printed, or electronic matter that provides information or evidence, or that serves as an official record [2]. For the purpose of our study, a document which represents a scientific discourse is a *Scientific Document*. It has a meaningful structure. There are several types of scientific documents which are used to explain work and preserve information for technical writing. The most common types of scientific documents are research thesis, research papers, articles, manuals, software reports, software requirements specification reports, books, research magazines, journals, and many others. Mostly, these scientific documents follow the same document structure, i.e., having keywords, sentences,

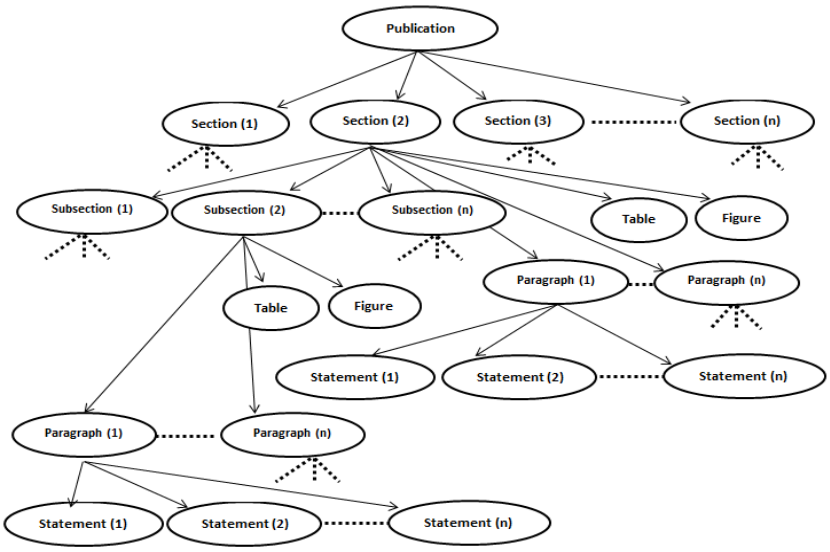

Fig. 1. Structure of *Scientific Document*

paragraphs, sections, headings, subheadings, references, tables and figures (see Figure 1). Due to meaningful structure of scientific document, we can easily map document components to ontological concepts. The term ontology means a specification of the shared conceptualization of a domain [3]. Ontology represents an extensive collection of formal representation, including taxonomies, hierarchical terminology, vocabularies, or describes logical theories of a domain [4]. It is also described as a collection of concepts and describes relationships between concepts that represent the meaning of the domain [5].
Due to iterative development process of *Scientific Document*, handling the change in document and maintaining the review comments history along each change is a difficult issue to handle. There is a need to maintain and keep track of each change in a meaningful way. Semantic versioning is used to maintain the ontology evolution and ontological differences and make each version of ontology compatible with each other [6]. Semantic versioning is exploited in real world application such as for e-Government [7]. Similarly, it can be used in case of research documents.

1.1 Scenario: Tracking changes and maintaining review comments in *Scientific Documents*

> "An author is writing a *Scientific Document*. He needs to consult his supervisor to complete this work. He is refining the document using the review comments that are given by the supervisor. He manages the reviewing comments for further use and also manages the document change history

> manually. He makes use of different folders to save the modified document or rename the modified document and makes several copies. The supervisor gives the comments. Those are managed by student through writing different notes using some software (Notepad, Word files, html editors) or in hard form (diaries, registers). In subsequent meetings supervisor may look for modifications of his previous remarks too. Student often forget to take into account all the remarks or sometimes supervisor modifies his previous remarks. As a result document is not finalized in-time."

In this paper, we propose a meaningful change process of the *Scientific Document* with review comments history. Collectively all this information helps the author to complete document authoring within time while bearing minimum overhead of offline tracking and storage of review comments.

2 Background & Literature Review

In the past few years, useful models have been proposed for scientific document representation which aim to express the rhetoric and argumentation within publications [8]. Harmsze's model [9] is one of the first inclusive models for providing the rhetorical structure of scientific information in electronic articles. The ABCDE Format [10] categorizes papers by five types of rhetorical blocks: Annotation, Background, Contribution, Discussion, and Entities. SALT (Semantically Annotated LaTeX) [11] is created by three ontologies (Document Ontology, Rhetorical Ontology, and Annotation Ontology). The Scientific Knowledge Object (SKO) [12] proposed patterns for scientific document representation model particularly for knowledge management in the evolving social web and semantic web. It has strong capabilities of semantic annotations, semantic search and strategic authoring grounded on logical reasoning (i.e. deduction, induction, and abduction) and focused on section level representation of the scientific document.
Traditionally version control systems (VCS) are used to keep track of changes in a document. There are two types of version control systems, Centralized Version Control Systems (CVCS) and Distributed/Decentralized Version Control Systems (DVCS) [13]. The CVCS contains CVS and SVN[3], and DVCS contains Git[4], Bazar[5] and Mercurial[6] software tools.
While Writing the scientific document, some authors use versioning system like SVN or Microsoft word to manage versions. But these versioning systems do not keep track of component level changes along with review comments history. Microsoft Word enables for writing the comments, perform operations like addition/deletion with each change. But it does not identify the complex operations, such as, displacement of text or block of text within a document. For example

[3] http://subversion.apache.org/
[4] http://git-scm.com/
[5] http://bazaar.canonical.com/en/
[6] http://mercurial-scm.org/

the paragraph in the section of a scientific document was moved to a new section. There are several techniques of semantic versioning such as PROMPTDIFF [14], SEMVERSION [15] and RDF(S) Repository Technique [16].
The RDF(S) Repository Versioning Technique works on RDF triple format of the OWL [17] ontology. It has two variables i.e., Update Counter (UC) and Update Identifier (UI). The UC is an integer variable and its value increases when the repository is updated. Each value of UC is identified by Update Identifier. UI represents the state of the repository. This technique performs two basic operations that are add and remove, which together represents the lifetime of the statement. The complete history of the repository is presented via add/remove operations. Versioning information is stored in ontology. Each version transaction has the format: UID:nn add—remove ⟨subj, pred, obj⟩ [16].

3 Initial Survey based upon usage scenario

We conducted an initial survey to assess the viability of the identified problem. The survey consists of questions that are asked from the research students. This questionnaire represents the personal and versioning information, how they manage their documents during write-up and also asked about the benefit of review comments history, separation of document's components and component level versioning within document. Our target audience for this survey is academic researchers who know about versioning process and in-process *Research Thesis* written in LaTeX and MS-Word. We selected 40 researchers (Faculty, doctorate student and post doctorate students) out of 50 from different departments of a University[7], and conducted the survey. The actual population statistics are shown in Table 1.
Figure 2 shows that, 100% faculty *mostly demanded* the previous review com-

Table 1. Qualifying Candidates for our study

Characteristic	Value	Number
Academic Position	Faculty	2
	Doctorate student	8
	Post Graduate Student	30
Department	Computer Sciences	6
	Mathematics	19
	Economics	15
Versioning Information	Use Versioning System	10
	Know about versioning	30

ments. 25% of the doctorate students gave response that their supervisors *mostly demanded* the previous review comments, while 12% said supervisors *always demanded* previous comments. The remaining doctorate students gave response

[7] Quaid-i-Azam University, Islamabad, Pakistan

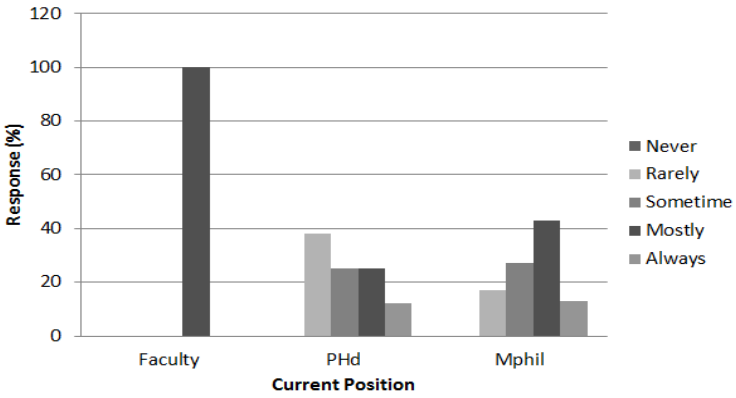

Fig. 2. Supervisor Demand previous Review Comments

that supervisor *sometimes or rarely demanded* previous review comments. 43% of MPhil students gave response that *mostly* supervisor demanded the previous review comments and 13% shows that they *always demanded* the review comments. Collectively all this information represents that review comments history is very important to save.

Figure 3 shows the graph which represents that the review comments are benefi-

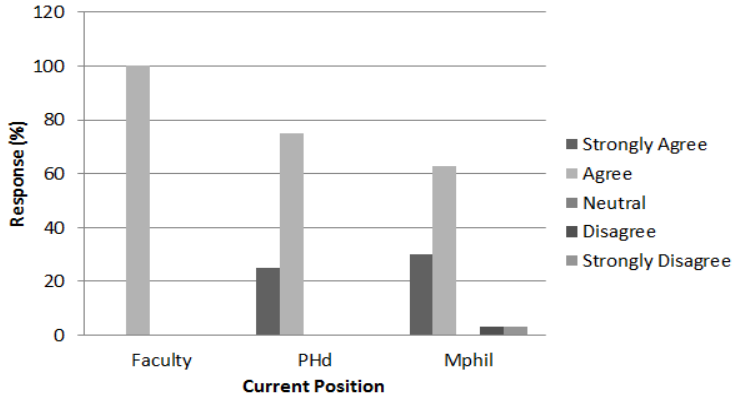

Fig. 3. Qualifying Candidates Response about benefit of Review Comments

cial for the research students. In this Graph, 100% faculty *agreed* to give response that review comments are beneficial in write-up process. 25% and 75% doctorate students *strongly agreed* and *agree* respectively, that they have interest about review comments. Similarly 30% and 63% post-doctorate students *strongly agree* and *agree* about review comments and remaining 3% each *disagree* and *strongly*

disagree about review comments usefulness.
Figure 4 shows the usefulness of different components of the document if dis-

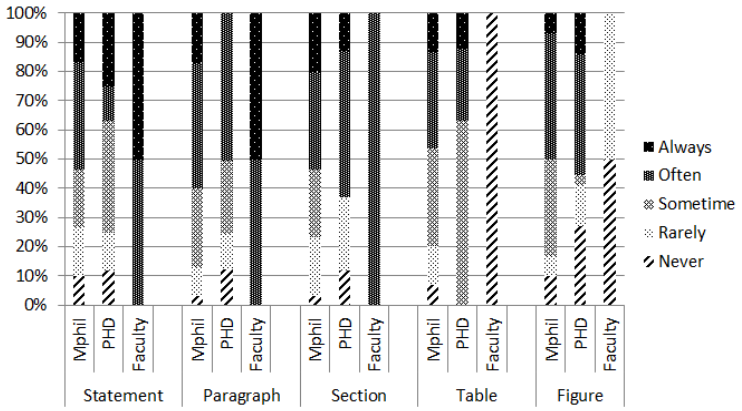

Fig. 4. Qualifying Candidates Response about Separate display of document Components

played separately. Mostly, response from the MPhil students, doctorate students and faculty members is that the sections, paragraphs and sentences are *always* to be displayed separately. Some of them gave response that figures and tables are *rarely* needed to be displayed separately. Collectively all this information represents that specific components of the document are necessary to be displayed separately. Based upon the finding in literature review and survey in local context a technique is proposed.

4 Proposed Work

Our technique is able to specify more meanings like each section has some paragraphs, subsections, tables and figures. Similarly each paragraph has some sentences. Existing versioning systems do not represent that which section has maximum change count, positioning of the component after change, and maintaining review comments history along each component of the document. This gap is also covered in our work.
For this purpose, we developed the prototype application which provides a meaningful change track along with review comments history in the *Research Thesis*. We enhanced the RDF(S) Repository technique, because it is extendable upto content level which is helpful in case of textual reports using the *Scientific Document* Ontology (see in Figure 5).

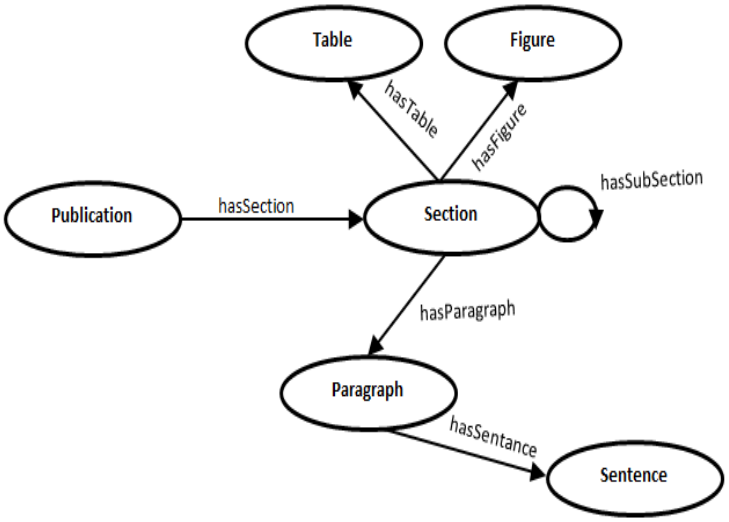

Fig. 5. *Scientific Document* Ontology

4.1 Enhanced RDF Repository Technique

In existing technique there is no way to store the review comments history along each change. So we enhanced the existing RDF Repository technique according to our research problem. Along with each update counter and update identifier we used RCID that represents Review Comment Identifier. The RCID is used for storing additional information along with each component of the document. So versioning history of the research thesis in RDF Repository can be represented as UID: nn, RCID: mm add — delete ⟨subject, predicate, object⟩.
During first time population of RDF(S) repository, only addition operation is performed and RCID is set to 0 for all the statements. e.g.,

- UID: 1, RCID:0 add ⟨A, r1, B⟩.
- UID: 2, RCID:0 add ⟨E, r1, D⟩.
- UID: 3, RCID:0 add ⟨E, r3, B⟩.
- UID: n, RCID:0 add ⟨D, rn, A⟩.

If the contents change in the document then repository is also changed to new state. If document is modified then two operations are performed (add, remove) along with updated RCID value. The RDF(S) Repository contains all operations from start, so we can easily find that which statements are added, removed or updated.

In our prototype system, author provides the LaTeX source file as input. Then it is automatically converted to document ontology and system shows the tree structure of the *Scientific Document*. It helps author to navigate to different components within the document. For each change, author can easily add, delete or

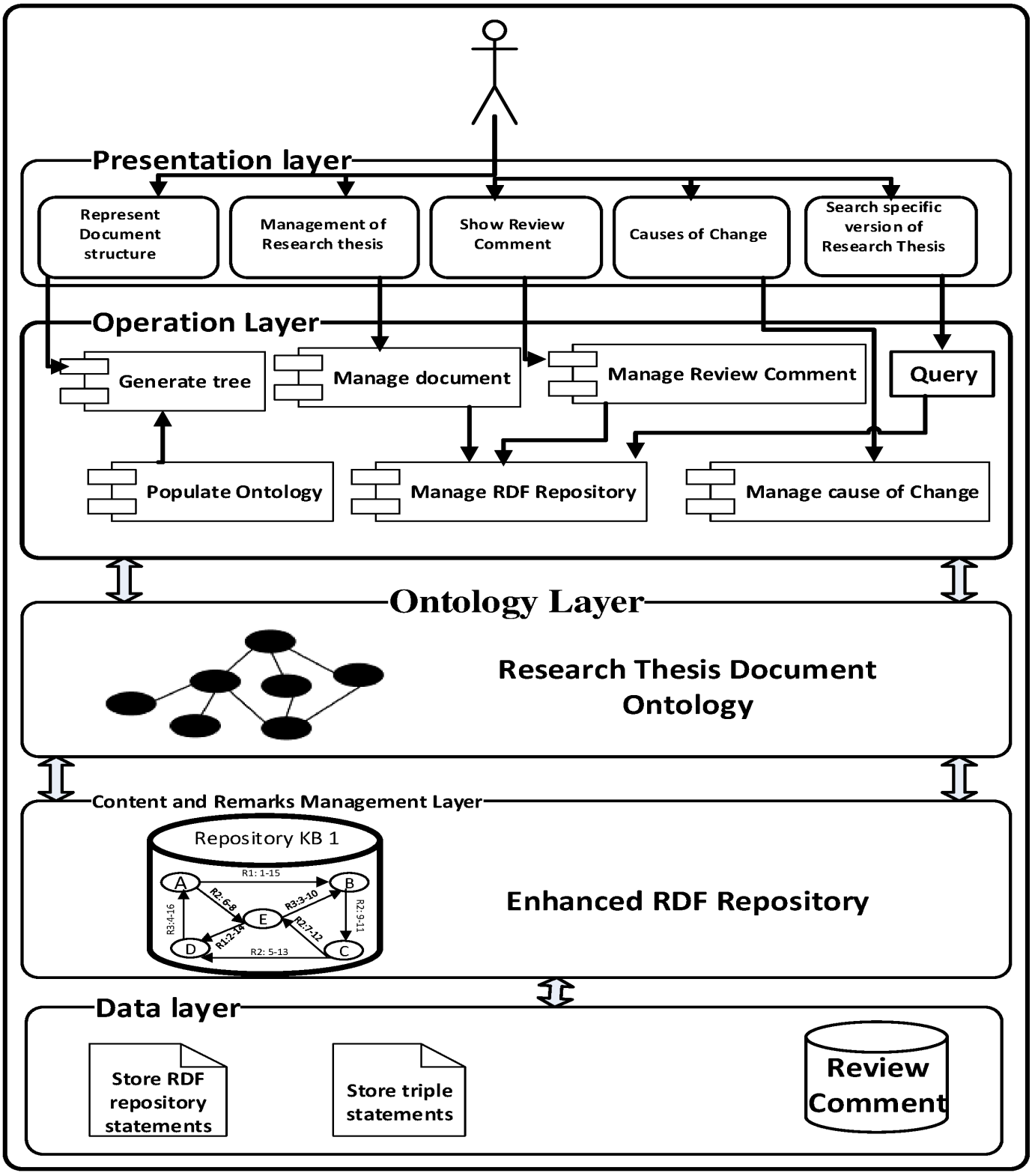

Fig. 6. Architectural Design for Proposed Framework

update section, subsection, paragraph, statement. Our prototype system consists of three component i.e., tree structure of the scientific document, contents of the document and review comment history along each component of the scientific document. Our system also provides the weekly or monthly progress report. It helps the author that at which stage he could not work properly.
Prototype system is used to help the author to maintain each change and view the previous changes in a meaningful way and also reduces the communication gap between the supervisor and the author. One of aims of this study is to reduce the authors effort which causes a lot of their energy to cope with review comments of each change in the scientific document.

4.2 Implementation

We have developed a prototype (see System Architecture in Figure 4.1) for experimentation and testing. We used java language to develop the system. Jena API[8] is used to create the ontology of the *Scientific Document*. RDF Repository Technique is used to maintain each change in the document along with review comments. The source code of our prototype can be downloaded from SourceForge[9].

5 Conclusion and Future work

In this paper, we have highlighted the need for tracking the changes along review comments in a document. Our survey shows that author has to face a lot of difficulties to save review comments along each change and there is no usable way of displaying a meaningful track of each change in the document. So our proposed work is used to help the author to maintain each change and view the previous change in a meaningful way and thus reduces the communication gap between the supervisor and the author. Our proposed prototype also shows document components in tree format to provide ease to the author.
Our in-process work, is to integrate proposed work with the LaTeX software. This will help authors to manage their changes and keep track of review comments of each change in a meaningful way. Behind each change there is a cause. The cause represents why the document is changed. So we will also explore the *Change Causality Model* for *Scientific Documents*.

Acknowledgements

The authors would like to give special thanks to Higher Education Commission (HEC), Pakistan, for providing travel support for presenting this work in a conference.

[8] http://jena.apache.org/
[9] http://sourceforge.net/projects/nbiaak/

References

1. Suzanne Briet, Laurent Martinet, Ronald E Day, and Hermina GB Anghelescu. *What is documentation?: English translation of the classic French text*. Scarecrow Press, 2006.
2. Michael K Buckland. What is a "document"? *JASIS*, 48(9):804–809, 1997.
3. Thomas R Gruber et al. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.
4. Natalya F Noy and Michel Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and information systems*, 6(4):428–440, 2004.
5. Yaozhong Liang, Harith Alani, and Nigel Shadbolt. Ontology versioning and evolution for semantic web-based applications. 9-month progress report. 2005.
6. Yaozhong Liang, Harith Alani, and Nigel Shadbolt. Change management: The core task of ontology versioning and evolution. 2005.
7. Heru Agus Santoso, Ziyad T Abdul-Mehdi, and Su-Cheng Haw. Semantic enhancement framework for e-government using ontology versioning approach. In *Proceeding of The6'th Conference on Information Technology and Application (ICITA 2009), Hanoi*, pages 296–301, 2009.
8. Tudor Groza, Siegfried Handschuh, Tim Clark, S Buckingham Shum, and Anita de Waard. A short survey of discourse representation models. 2009.
9. Frédérique-Anne Pacifique Harmsze. *A modular structure for scientific articles in an electronic environment*. 2000.
10. Anita de Waard and Gerard Tel. The abcde format enabling semantic conference proceedings. In *SemWiki*, 2006.
11. Tudor Groza, Siegfried Handschuh, Knud Möller, and Stefan Decker. Salt-semantically annotated\ mbox {\ LaTeX} for scientific publications. In *The Semantic Web: Research and Applications*, pages 518–532. Springer, 2007.
12. Fausto Giunchiglia, Hao Xu, Aliaksandr Birukou, and Ronald Chenu. Scientific knowledge object patterns. In *Proceedings of the 15th European Conference on Pattern Languages of Programs*, page 15. ACM, 2010.
13. Eric Sink. *Version control by example*. Pyrenean Gold Press, 2011.
14. Natalya Fridman Noy and Mark A Musen. Promptdiff: A fixed-point algorithm for comparing ontology versions. *AAAI/IAAI*, 2002:744–750, 2002.
15. Max Völkel and Tudor Groza. Semversion: An rdf-based ontology versioning system. In *Proceedings of the IADIS international conference WWW/Internet*, volume 2006, page 44. Citeseer, 2006.
16. Damyan Ognyanov and Atanas Kiryakov. Tracking changes in rdf (s) repositories. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pages 373–378. Springer, 2002.
17. Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(2004-03):10, 2004.