

# A Comparative Study: Change Detection and Querying Dynamic XML Documents

Vijay Sonawane, D.R. Rao

Department of Computer Engineering, K.L.University, Green Fields, Guntur, Andhra Pradesh

---

## Article Info

### Article history:

Received Feb 22, 2015

Revised May 14, 2015

Accepted May 29, 2015

### Keyword:

Change Detection

Delta

Dynamic

Querying

XML

---

## ABSTRACT

The efficient management of the dynamic XML documents is a complex area of research. The changes and size of the XML documents throughout its lifetime are limitless. The increasing applications that use and exchange XML data is creating a demand for multi version support and store XML documents for future references. Change detection is an important part of version management to identify difference between successive versions of a document. Document content is continuously evolving. Users wanted to be able to query previous versions, query changes in documents, as well as to retrieve a particular document version efficiently. In this paper we provide comprehensive comparative analysis of various control schemes for change detection and querying dynamic XML documents.

Copyright © 2015 Institute of Advanced Engineering and Science.

All rights reserved.

---

## Corresponding Author:

Vijay Sonawane

Department of Computer Engineering, K.L.University, Green Fields, Guntur, Andhra Pradesh, 522 502

Email: vijaysonawane11@gmail.com

---

## 1. INTRODUCTION

In today's information era XML has rapidly become the language of choice for data representation, storage, and exchange over the web. Information available online is not only useful for individual user but also to business organizations mostly for decision making purpose. XML offers many features of business functions including content integration and intelligence. So it is very important to maintain those documents properly and know how to use efficiently information in it.

### 1.1. Background

Web browsers use Hyper Text Markup Language (HTML) to compose textual, graphical, audible, and visual material in the form of Web pages. HTML has some limitations of fixed set of elements for representing the Web content and its representation. Therefore, to overcome those limitations W3C designed XML (Extensible Markup Language) which is simpler and has self describing nature. Its format is same as the HTML but more flexible and also used for data exchange and support external data confirmation. The XML Schema or Document Type Description (DTD) defines a structure of XML documents and helps applications to authenticate imported data. Several standards are available for data sharing and management in XML such as Extensible Stylesheet Language (XSL) to transform an XML document and Document Object Model (DOM) to allow programs to dynamically access or update an XML document[1]. XML and database technology are more corresponding. The task of managing loads of XML data has increased the demand to discover the integration of XML with database technology. XML databases are equally beneficial for content-centric and change-centric applications. The change-centric applications such as software configuration and cooperative work may exploit dynamic XML documents since the content and structure of a dynamic XML document changes continuously depending on the underlying application, such as the content of a website of an e-commerce web site. In contrast the XML documents whose structure remains unchanged are called static XML document. Whenever the XML document content changes then mainly two things happen: a) Document size grows b) New version of the same document is created. A version control Scheme may use object-referencing or deltas to detect and store the changes between multiple versions. Most of the time only small changes in the document may create the new version.

---

Authors in [2, 3, 4, 5] have been written on growing semi-structured or XML data and schema. Overview of the schema evolution support in several commercial tools is provided in [6].

## 1.2. Motivation

An organization can publish its products information online and allow customers to provide reviews and suggestions by browsing and querying the catalogue. XML assures to achieve such features of business functions including content integration, intelligence and salvage. So maintenance and utilization of information in efficient way contained in XML document is very important. XML is often called as self-describing data it is designed in a way that the schema is repeated for each record in the document. XML is used to create vocabularies by storing and exchanging metadata for information exchange on the internet. Demand to store XML documents for future references is increased as the applications that use and exchange XML data is also increased. Importantly XML documents are dynamic in nature as its content is application specific and change across the time.

## 1.3. The Problem

Dynamic XML documents are applicable in many fields of information management and create the demand that it should support multi-version documents. Therefore it is necessary to store different versions of XML documents with time. Storage of all the versions of an XML document is not effective solution as it increases the duplication and makes searching and querying harder on growing document. So it is essential to find techniques to store and detects the changes in multiversion XML documents. Its also important to find schemes to efficiently execute the cross version queries over dynamic of XML document. In this paper, we discuss the two main issues related to dynamic XML documents: (i) *Change detection in dynamic XML document* (ii) *Querying dynamic XML documents*

## 2. DYNAMIC XML DOCUMENTS

XML documents are mainly static XML document and Dynamic XML document. In real world application changes from one version of an XML document to another cannot be predicted as those changes are applications specific. Version control is required to manage and query the semi-structured nature of continuously changing XML documents. Many researches so far proposed various schemes for version control based on various approaches.

- Delta Based Approach [7, 8, 9, 10, 11, 12, 13, 14, 15].
- Object Referencing [16, 17, 18, 19].

Delta document records the changes between two consecutive versions, using delta one can easily locate its previous version so it has been considered that delta approach is superior to the object referencing approach. Web management system records multi-version XML documents with changing time. Multiple effort have been taken for the effective management of growing XML documents [20, 21, 22].

## 3. CHANGE DETECTION

Change detection is a process of identifying differences between successive versions of a document. The concept of change detection was first applied for relational database in distributed environment [23]. An algorithm was proposed by [24] to detect all changes of the base table in distributed relational database system. Other than relational data, success of XML as the standard for data exchange has paved way for a number of change detection techniques and attracted research attention. Change detection helps to reduce the storage space of historical data by avoiding redundancy and in its ability to support temporal queries. Temporal queries are time specific and changes with creation, modification, and deletion of data. So change detection is important to mine historical data of the documents to get detailed information on the changes made since its beginning. Most of the on-line data change rate is higher and that change needs to be monitored effectively. Change detection also helps to improve incremental query evaluation wherein there are continuous queries that monitors a particular data source and updates and to reduce cost of query evaluation. Rather than evaluating query on the entire data it is beneficial to combine the results of query on changed data with previous query [25].

Fig. 1 shows three versions of an XML document. In the version 1 and version 2 underlined parts reflects the change in both versions, these two versions are alike. Now, version 3 shows an alternative version of the same XML document with even more considerable change. This different version also has the same information but they are just arranged in a different schema. Instinctively, due to that structural changes version 3 requires more number of addition and deletion operations even though they have the same information as its ancestor version.

Version 1	Version 2	Version 3
<pre>&lt;bib&gt; &lt;author&gt; &lt;name&gt;A1&lt;/name&gt; &lt;book&gt; &lt;title&gt;B1&lt;/title&gt; &lt;publisher&gt;C1&lt;/publisher&gt; &lt;/book&gt; &lt;/author&gt; &lt;author&gt; &lt;name&gt;A2&lt;/name&gt; &lt;book&gt; &lt;title&gt;B2&lt;/title&gt; &lt;publisher&gt;C2&lt;/publisher&gt; &lt;/book&gt; &lt;book&gt; &lt;title&gt;B1&lt;/title&gt; &lt;publisher&gt;C1&lt;/publisher&gt; &lt;/book&gt; &lt;/author&gt; &lt;/bib&gt;</pre>	<pre>&lt;bib&gt; &lt;author&gt; &lt;name&gt;A1&lt;/name&gt; &lt;book&gt; &lt;title&gt;B1&lt;/title&gt; &lt;publisher&gt;C1&lt;/publisher&gt; &lt;/book&gt; &lt;/author&gt; &lt;author&gt; &lt;name&gt;n2&lt;/name&gt; &lt;book&gt; &lt;book-title&gt;B2&lt;/book-title&gt; &lt;publisher&gt;C2&lt;/publisher&gt; &lt;/book&gt; &lt;book&gt; &lt;title&gt;B1&lt;/title&gt; &lt;publisher&gt;C1&lt;/publisher&gt; &lt;/book&gt; &lt;/author&gt; &lt;/bib&gt;</pre>	<pre>&lt;bib&gt; &lt;author&gt; &lt;name&gt;A1&lt;/name&gt; &lt;book&gt; &lt;title&gt;B1&lt;/title&gt; &lt;publisher&gt;C1&lt;/publisher&gt; &lt;/book&gt; &lt;/author&gt; &lt;author&gt; &lt;name&gt;A2&lt;/name&gt; &lt;book&gt; &lt;book-title&gt;B2&lt;/book-title&gt; &lt;publisher&gt;C2&lt;/publisher&gt; &lt;/book&gt; &lt;book&gt; &lt;title&gt;B1&lt;/title&gt; &lt;publisher&gt;C1&lt;/publisher&gt; &lt;/book&gt; &lt;/author&gt; &lt;/bib&gt;</pre>

Figure 1. Consecutive versions of an XML document

#### 4. DETECTING CHANGES IN DYNAMIC XML DOCUMENTS

Version detection has many applications such as web document ranking, duplicate code identification, plagiarism detection, and on-line shop management. Similarities between various documents are considered to detect the versions, for this purpose selection of similarity threshold value and function is an important issue. Content based and structure based similarity and also application specific requirement are useful in selection of similarity function for version discovery. Table1 shows various proposals for discovering changes in dynamic XML document.

Table 1. Various schemes for changes discovery in dynamic XML documents

Approach	Proposed by author
Diff-based schemes	COB 02, RUS 08, SAC 07, WAN 03, WON 02, WON 03
Changes in documents textual content	BAE 11
Changes in document structure	FLE 06
Key based	VIY 09
Document classification	PON 07 , WAN 03

The diff-based schemes may use a delta document or a consolidated delta document for version management. A delta document records only the changes between two versions of an XML document whereas a consolidated delta document contains the data of the original version as well as all deltas. Moreover delta document can be categorized as forward and backward delta. Author in [26] provide more efficient consolidated delta as it provide mapping between forward and backward delta and reduce the storage space needed to store intermediate deltas. Consolidated delta is only the one document which stores all the changes between all versions of XML documents rather than storing intermediate deltas. If document is having multiple versions at different time, then any version T can be obtained by any time moment using intermediate delta or consolidated delta.

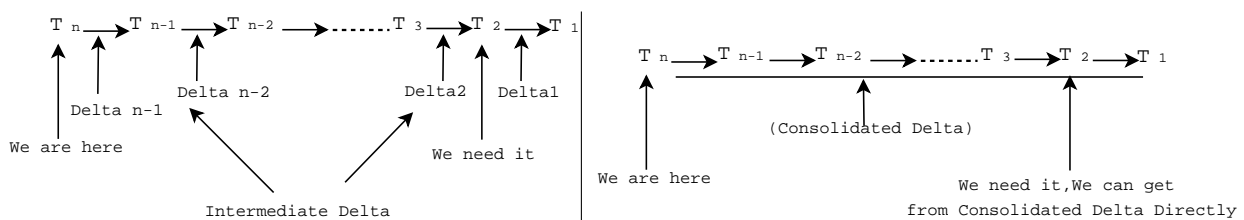


Figure 2. Intermediate delta and Consolidated delta

Consolidated delta approach has advantages over intermediate delta with all three drawbacks because consolidated delta stores only changed part of each version rather than storing each version every time. If document is

unchanged then it is not treated as version. It has a drawback of time overhead to query backward the unchanged parts of a document.

The storage of intermediate complete versions of XML documents improves the efficiency and space complexity as the required version can be created by using the appropriate intermediate complete version instead of the current version only. Query processing becomes efficient while a system stores the intermediate complete versions because there is no need to reconstruct the intermediate versions. The schema-unaware approaches use a fixed schema to store the logical order of an XML document and provide two benefits (1) Handle XML schema evolution and (2) change detection does not require any modification in SQL queries. The reference-based schemes use object references and preserve the logical order of dynamic XML documents. A version is represented by using objects, which can (1) reference records representing unchanged parts between two versions and (2) be the actual document objects records.

Well formed XML document gratifies some constraints. As an XML documents updates are application specific then updated part need to be tracked. Thus validation of the time-varying multi-version XML documents is important research area [27]. The storage of all the versions of XML documents with more similarity introduces redundancy. The objective is to reduce the redundancies without loss of significant information. The storage redundancy can be avoided by storing only once the elements which remain unchanged between multiple versions.

In table 2 we have shown the comparative study of change management schemes by considering various parameters including change discovery between two versions of an XML document, scalability, affordable file size, use of relational approach (schema aware/unaware), delta approach, object-referencing approach for change discovery and Redundancy. Each parameter has its own importance.

Authors in literature [8, 10, 11, 12] presented various schemes which can explicitly show changed parts of the document to user. Schemes presented by authors [28, 29, 30, 31, 32, 33, 34] do not show explicitly the changed part of the document to the user.

It is beneficial to store an intermediate complete XML documents version which improves the efficiency and space complexity. Also the required version can be created by using the appropriate intermediate complete version instead of the current version only. Storing intermediate complete versions also improves querying speed as there is reconstruction of intermediate version is not needed. In schema-aware approaches fixed schema are used to store the logical order of an XML document. This gives the profit of handling XML schema evolution and change discovery without any modification in SQL queries. The object-referencing approach for change discovery uses object references and maintains the logical order of multi-version XML documents. Objects are used to represents versions which represents unchanged parts between two versions.

## 5. QUERY EXECUTION ON DYNAMIC XML DOCUMENT

Queries associated with dynamic XML document are two types - change centric and content centric. Change centric query execution may cover a single version or multiple versions of document. Content centric query execution cover single version using querying edit operations performed over a single document version. Queries covering multiple document versions are executed over multiple deltas and their results are combined to find the final answer set. The queries can be categorized in to different types as follows:

1. Structural Projection Queries – Like project the documents 15th version between the 5th and the 10th chapter require a continuous portion of the document from a given version.
2. Path-Expression Queries – Like for 15th version, find all table elements under chapter 6 require a version and a regular path expression to output the elements that satisfy the expression for a specified version.
3. Parent child Expression Queries – Like find all table elements under chapters in 15th version combine the structural joins with a path expression.
4. Content Based Selection Queries – Its output version satisfy the predicates given in the where clause.
5. Versions Interval Queries – These queries are supporting multiple versions extended from single versions queries.
6. Version Aggregation Queries – These queries finds an aggregate values of elements, which are specified by a version-interval query.

Table 2. Various Change management schemes: Comparative Analysis

Proposal	Change detection		Affordable file size	Relational approach (schema aware/unaware)	Change management approach			Ordered XML document	Redundancy
	Change detection	Scalable			Find intermediate deltas	Store consolidated deltas	Store intermediate complete versions		
[AMA 00]	N	na	na	N	N	N	N	N	N
[CHA 01]	Y	Y	-	N	N	N	Y	Y	N
[COB 02]	Y	N	10Mb	N	Y	N	Y	Y	N
[WON 02]	Y	N	-	N	N	N	Y	Y	N
[GER 03]	N	na	na	na	N	N	N	N	N
[WAN 03a]	Y	Y	-	N	N	Y	N	Y	N
[WAN 03]	Y	N	LT 10mb	N	N	N	N	N	N
[WON 03]	Y	N	-	Y	Y	N	N	Y	N
[WUW 04]	Y	N	-	Y	Y	N	N	N	N
[GRA 05]	N	na	na	N	N	N	N	N	N
[LEO 05]	Y	Y	na	aware	Y	N	N	N	N
[LEO 05]	Y	Y	-	unaware	Y	N	N	N	N
[RUS 05]	Y	Y	-	N	N	N	Y	N	N
[GUE 07]	N	na	na	N	N	N	N	N	N
[ROS 07]	N	na	na	N	N	N	N	N	N
[GUE 08]	N	na	na	N	N	N	N	Y	N
[RIZ 08]	N	na	na	na	na	N	N	N	N
[SNO 08]	N	na	na	N	N	N	N	N	N
[WAN 08]	N	na	na	N	N	N	N	N	N
[ZHO 09]	N	na	na	N	N	N	N	N	N
[BRA 10]	N	na	na	N	N	N	N	N	N
[CAV 10]	N	na	na	N	N	N	N	N	N
[CAV 11]	N	na	na	N	N	N	Y	Y	N
[FED 13]	N	na	na	N	N	N	N	N	N

Note How to read na and -? If proposal failed in detecting changes then scalable and affordable file size becomes illogical then their values are represented by na. - in affordable file size parameter indicates those proposals does not discuss about affordable file size.

These queries are difficult kind to respond effectively. It requires identification of parent and child relationship between document elements, and it can be answered efficiently by avoiding full visit to all documents sub trees. Methods based on line edit script or object reference schemes are not effective to capture the logical order and parent child relationship of a document in version XML documents. Author [35, 36, 37, 38] proposed numbering schemes those are used to capture the logical order of the documents. Various numbering schemes are available like range based [35, 36], durable node [37], and prefix-based [40, 39].

- Range Based – In this numbering scheme range of number is associated with each node to find its position in the XML tree. It reduces path expression queries to join operations.
- Durable Numbering – With this scheme the insertion and deletion of the element does not affect the numbers associated with elements so it maintains the nodes in the pre order traversal.
- Prefix Based Numbering – In this scheme parent node number is used as a prefix to assign a number to a child node. But as it uses variable length encoding it needs more space than other schemes.

Table3. gives comparative analysis of various schemes that efficiently support queries for XML documents by considering various query type diff, parent child expression, path expression, version interval, version aggregation, structural projection, content based selection. The effectiveness of any scheme is depends on how much it support for powerful queries. Author [33] describe support for powerful temporal queries expressed in XQuery for multi-version XML documents without introducing new constructs in the language. Similarly [31] discusses support for temporal queries expressed in XPath and XSLT for multi-version XML documents without introducing new constructs in the language. Schema evolution is time specific, so even it is difficult to present the correct query. Research in the area of dynamic XML documents does not travel around sovereignty property as these systems must have strong knowledge of schema. Author [41] introduces the advantages of design independence and the potential statistical based methods having such independence property.

## 6. FUTURE RESEARCH DIRECTION

There are still some open research problems for dynamic XML documents. Here are some of these issues we identified:

1. In change detection problem finding similarity functions (formula to measure similarity between files) threshold value is very important. Also while detecting the versions many other features (rather than only content and structure) must be considered together. Moreover similarity function for version detection should not be application specific. Hence defining effective similarity function for detecting the changes between XML documents is challenging research area.
2. Self describing nature of XML introduces the problem of verbosity and results in large document size. XML Delta is document which records changes between two versions of an XML document. It is possible to compress the evolving size of XML documents and find delta without their decompression. Delta document compression may save storage and computing power, so finding compression technique which will support time efficient query processing without delta decompression need added research.

## 7. CONCLUSION

Data available on-line today is in the form of XML documents to achieve maintenance and management of important information. The massive storage of data in XML documents needs the efficient data management schemes. Changes in XML documents are application specific and they evolve over time. Many applications require querying of earlier versions of an XML document. Storing all the versions of documents may increases the redundancy. Unlike static XML documents, capricious and continuous changing nature of dynamic XML documents requires different managing schemes. This paper discusses several issues related to dynamic XML documents including change management between versions of XML documents by using deltas, object references, or relational approaches, version support, and trend analysis queries. Lastly we identified some hot research directions in the area of dynamic XML document.

Table 3. Various querying support schemes: Comparative Analysis

Proposal	Complex										Temporal	
	Query Type					For dynamic XML documents					Supported	For dynamic XML documents
	Diff	Parent child expression	Path expression	Version interval	Version aggregation	Structural projection	Content based selection	Simple	For dynamic XML documents			
[AMA 00]	N							Y		Y		N
[LIQ 01]	N	Y	N	N	N	N	N	N		N		N
[MAR 01]	N											
[CHA 01]	Y	Y	Y	Y	N	Y	N	Y		N		
[BRU 02]	N	Y	Y	N	N	N	N	N		N		
[CHA 02]	Y	N	N	Y	N	Y	Y	Y		N		
[COB 02]	N											
[WON 02]	N							N		Y		Y
[VAG 03]	Y	N	Y	N	N	N	N	Y		N		
[WAN 03]	N	N	N	Y	N	Y	Y	Y		Y		Y
[WUW 04]	N											
[GRA 05]	N											
[ROS 07]	N	N	N	N	Y	Y	N	Y		Y		Y
[WAN 08]	N											
[ZHO 09]	N											
[BRA 10]	N											
[CAV 10]	N											
[CAV 11]	N											

Note How to read the table? 1.[AMA 00,WON02,WAN 08] - Do not provide support for complex queries for either a static XML document or dynamic XML documents. 2.[AMA 00] - Provides support of simple queries and temporal queries. 3[LIQ 01,WAN 08]- Provides support of complex temporal queries for dynamic XML documents. However, it does not provide support of temporal queries for dynamic XML documents. 3.[COB 02,MAR 01,WUW 04,ZHO 09,BRA 10,CAV 10,CAV 11] - Do not provide any type of query support.

## REFERENCES

- [1] [ELH 14] Elham B. F., Hasan K., "Improving semantic clustering using with Ontology and rules", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 4, No. 1, pp. 7-15, Feb. 2014.
- [2] [ALI 09] Ali. K. A, and Pokorny z, Comparison of XML-based temporal models *Advanced Internet Based Systems and Applications, LNCS*, 4879, pp.339-350, 2009.
- [3] [COL 10] Colazzo. D, Guerrini. G, Mesiti. M, Oliboni, B, and Waller. E, Document and schema XML updates, IGI Global. <http://dx.doi.org/10.4018/978-1-61520-727-5.ch016>, 2010.
- [4] [GUE 09] Guerrini, G., and Mesiti, M., XML schema evolution and versioning: Current approaches and future trends *Open and novel issues in XML database applications, future directions and advanced technologies*, IDEA Group, 2009.
- [5] [HAR 11] Hartung, M., Terwilliger, J., and Rahm, E., Recent advances in schema and ontology evolution. Schema matching and mapping *Springer*, pp.149-190, 2011.
- [6] [GUE 08] Guerrini, G., and Mesiti, M. X-Evolution: A comprehensive approach for XML schema evolution *Proceedings of the international workshop on database and expert systems application*, pp.251-255, 2008.
- [7] [CHA 01b] Chien. S. Y, Tsotras. V. J, and Zaniolo. C, Copy-based versus edit-based version management schemes for structured documents, *Proceedings of the eleventh international workshop on research issues in data , engineering* pp. 95-102, 2001.
- [8] [LEO 05] Leonardi. E, and Bhowmick. S. S. Detecting changes on unordered XML documents using relational databases: A schema conscious approach *Proceedings of the ACM conference on information and, knowledge management* pp. 509-516, 2005.
- [9] [MAD 05] Madria. S, Leonardi. E, and Bhowmick, S. S. Xandy: Detecting changes on large unordered XML documents using relational databases. *Proceedings of the international conference on database systems for advanced applications* pp. 711-723, 2005.
- [10] [RUS 05] Rusu, L. I., Rahayu, W., Taniar, D. Maintaining versions of dynamic XML documents *Proceedings of the sixth international conference on web information, systems engineering* pp. 536-543, 2005.
- [11] [WON 02] Wong, R. K., Lam, N. Managing and querying multi-version XML data with update logging *Proceedings of the ACM symposium on document, engineering* pp. 74-81, 2002.
- [12] [WUW 04] Wuwongse, V., Yoshikawa, M., Amagasa, T. Temporal versioning of XML documents *Proceedings of the Seventh International conference on digital libraries: International collaboration and cross-fertilization* pp. 419-428, 2004
- [13] [WAN 03a] Wang, F., Zaniolo, C. Temporal queries in XML document archives and web warehouses *Proceedings of the tenth international symposium on temporal representation and reasoning and fourth international conference on temporal logic* pp. 47-55, 2003(a).
- [14] [MAR 01] Marian, A., Abiteboul, S., Cobena, G., Mignet, L. Change-centric management of versions in an XML warehouse *Proceedings of the twenty seventh international conference on very large data base* pp. 581-590, 2001.
- [15] [WON 03] Wong, R. K., Lam, N., Efficient re-construction of document versions based on adaptive forward and backward change deltas *In Proceedings of the international conference on database and expert systems applications* pp. 266-275, 2003.
- [16] [CHA 01] Chien. S. Y, Tsotras. V. J, Zaniolo. C. and Zhang. D. Storing and querying multiversion XML documents using durable node numbers *Proceedings of the second international conference on web information, systems engineering* pp. 232-241, 2001.
- [17] [CHA 01a] Chien. S. Y, Tsotras. V. J, and Zaniolo. C, Efficient management of multiversion documents by object referencing *Proceedings of the twenty seventh international conference on very large data bases* pp. 291-300, 2001.
- [18] [CHA 01c] Chien. S. Y, Tsotras, V. J, and Zaniolo. C. XML document versioning, *ACM SIGMOD Records*, pp. 46-53, 2001.
- [19] [CHA 02] Chien. S. Y, Tsotras. V. J, and Zaniolo. C. Efficient schemes for managing multiversion XML document. *Very Large Data Bases*, pp.332-353, 2002.
- [20] [GER 03] Gergatsoulis. M, and Stavarakas. Y, Representing changes in XML documents using dimensions, *Proceedings of the international XML database, symposium* pp.208-221, 2003.
- [21] [GRA 05] Grandi. F, Mandreoli. F, and Tiberio, P. Temporal modelling and management of normative documents in XML format, *Data and Knowledge Engineering*, 54(3), pp.327-354, 2005.
- [22] [GRA 03] Grandi, F., Mandreoli, F., Tiberio, P., and Bergonzini, M., A temporal data model and management system for normative texts in xml format, *Proceedings of the international workshop on web information and data management*, pp. 29-36, 2003



- [23] [SAM 12] Samini S., Su-Cheng H., Poo Kuan H., "Bridging XML and Relational Databases: An Effective Mapping Scheme based on Persistent ", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol.2, No.2, pp. 239-246, April 2012.
- [24] [LIN 86] Lindsay, B. G., Haas, L. M., Mohan, C., Pirahesh, H. and Wilms, P. F., A Snapshot Differential Refresh Algorithm Proc. ACM SIGMOD *International Conference on Management of Data, Washington, D.C., U.S.A.*, pp. 53-60 1986.
- [25] [FAB 04] Fabio Grandi. Introducing an Annotated Bibliography on Temporal and Evolution Aspects in the World Wide Web. SIGMOD Record, Volume 33, Number 2, June 2004.
- [26] [RUS 08] Rusu, L. I., Rahayu, W., Taniar, D., Storage techniques for multi-versioned XML documents, *Proceedings of the thirteenth international conference on database systems for advance applications* pp. 538-545, 2008.
- [27] [BRA 10] Brahmia. Z, Bouaziz. R, Grandi, F, and Oliboni, B. Schema versioning in Xschema-based multitemporal XML repositories, Tech. Rep. TR-93, TimeCenter, 2010.
- [28] [CAV 10] Cavalieri F, EXup: an engine for the evolution of XML schemas and associated documents, *Proceedings of the international conference on extending database technology* pp. 1-10, 2010.
- [29] [CAV 11] Cavalieri. F, Guerrini. G, Mesiti. M, and Oliboni. B, On the reduction of sequences of XML document and schema update operations *Proceedings of the IEEE twenty seventh international conference on data engineering workshops* pp. 77-86, 2011.
- [30] [GUE 07] Guerrini. G, Mesiti. M, and Sorrenti. M. A. XML schema evolution: Incremental validation and efficient document adaptation *proceedings of the international symposium on XML, database* pp. 92-106, 2007.
- [31] [ROS 07] Rosado, L. A., Mrquez, A. P., Gil, J. M. Managing branch versioning in versioned/temporal XML documents
- [32] [SNO 08] Snodgrass, R. T., Dyreson, C., Currim, F., Currim, S., Joshi, S. Validating quicksand: Temporal schema versioning in sXSchema *Data and Knowledge Engineering*, 65(2), 223-242, 2008.
- [33] [WAN 08] Wang, F., Zaniolo, C. Temporal queries and version management for XML document archives. *Data and Knowledge Engineering*, 65(2), 304-324, 2008.
- [34] [ZHO 09] Zhuludev, V., Kohlhase, M. TNTBase: A versioned storage for XML. Balisage: The Markup Conference. *Proceedings of the international symposium on XML, database* pp. 107-121, 2009.
- [35] [ABI 01] Abiteboul, S., Kaplan, H., Milo, T. Compact labeling schemes for ancestor queries, *Proceedings of the twelfth ACM-SIAM symposium on discrete algorithms*, pp. 547-556, 2001.
- [36] [BRU 02] Bruno. N, Koudas. N, and Srivastava. D, Holistic twig joins: optimal XML pattern matching, *Proceedings of the ACM SIGMOD international conference on management of data* pp. 948-952, 2002.
- [37] [LIQ 01] Li, Q., Moon, B. Indexing and querying XML data for regular path expressions *Proceedings of the twenty seventh international conference on very large data base* pp. 361-370, 2001.
- [38] [WAN 03] Wang, H., Park, S., Fan, W., Yu, P. ViST: a dynamic index method for querying XML data by tree structures *Proceedings of the ACM SIGMOD international conference on management of data* pp. 110-121, 2003.
- [39] [SAN 08] Sans, V., Laurent, D. Prex based numbering schemes for XML: Techniques, applications and performances *Proceedings of the VLDB Endowment*, 1(2), 1564-1573, 2008.
- [40] [DYE 11] Dyreson. C. E, and Mekala. K. G, Prex-based node numbering for temporal XML, *Proceedings of the international conference on web information, systems engineering* pp. 172-184, 2011.
- [41] [TER 11] Termehchy, A., Winslett, M., Chodpathumwan, Y. How Schema Independent are Schema Free Query Interfaces? *Proceedings of the international conference on data, engineering* pp. 649-660, 2011.

## BIOGRAPHY OF AUTHORS



**Vijay Sonawane** Vijay Sonawane obtained Bachelor Degree in Information Technology from North Maharashtra University (MS), India. Then he earned Master in Technology (Computer science) from Shivaji University, Kolhapur (MS), India. He is currently PhD Research scholar in computer science and engineering in K.L.University, Vijayawada. He is working as Assistant Professor in Sandip Institute of Technology and Research Centre, Nashik. His major area of interest is Data mining, Information retrieval, web information management. He published various papers in Journals and Conferences.