University of Thi-Qar Journal

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023

A Comparison of Pertinence between Containment and Dewey Labelling Schemes to the XML Documents

Samer H. A. Al-khazraji

University of Diyala-College of Education for Pure Science- Computer Sc. Dept.

Emails: samer.alkhazraji@uodiyala.edu.iq

https://doi.org/10.32792/utq/utj/vol18/2/1

Abstract

Researchers are interested in improving the XML database management system using an indexing system by labelling XML nodes. They proposed many labelling schemes to be equivalent to the indexing system of the relational database management system. There are many algorithms that have been suggested for efficiently assigning labels to XML documents in a small amount of storage space. But, the majority of the models concentrated on assessing the effectiveness of proposed labelling schemes. The pertinence of a labelling technology with a certain pattern of XML database has not been adequately inspected. For analysing this point, several experiments were carried out have been executed to label different XML data structures using Containment and Dewey labelling schemes. The requirements of time and space have been taken into account to measure the relevance of the labelling scheme to a specific structure of XML document. It was discovered that both schemes are relatively efficient when allocating labels for a shallow structure of an XML tree. Both schemes are efficient, however, the former for short tree and the latter for wide tree.

Keywords: XML, label, Scheme, database, relevance.

Introduction

Without a doubt, XML has raised a predominant technology for transmitting and representing data on the web (Bertino, E. and Ferrari, E., 2001; Alsayed Algergawy, Richi Nayak, Gunter Saake, 2010; Alsayed Algergawy, Marco Mesiti, Richi Nayak, and Gunter Saake, 2011). Therefore, it is a major demand to develop a labelling scheme to manage XML data storage efficiently and effectively (Almelibari, 2015).

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023

XML databases are categorised into two types (Kurt, A. and Atay, M., 2002; Bellahs`ene, 2003): XML-enabled databases are traditional database management systems that store XML data in tables, for instance, TL/SQL and PL/SQL (Kurt, A. and Atay, M., 2002; Win, K.M., Ng, W.K. and Lim, E.P., 2003). To store XML data in tables, it requires an expensive mapping operation to map the data from the tree structure into tables. Another type, known as Native-XML databases (NXD), is a data model that keeps the tree shape of the XML document while eliminating the mapping operation (Kurt, A. and Atay, M., 2002; Win, K.M., Ng, W.K. and Lim, E.P., 2003), and this dataset is the core of this study.

The traditional databases store data in tables made up of columns and rows, and an indexing system is employed to manage the stored data. The employed indexing system is not appropriate to query information organised in a hierarchical pattern as XML documents (Almelibari, 2015) and demonstrated in Figure (1).

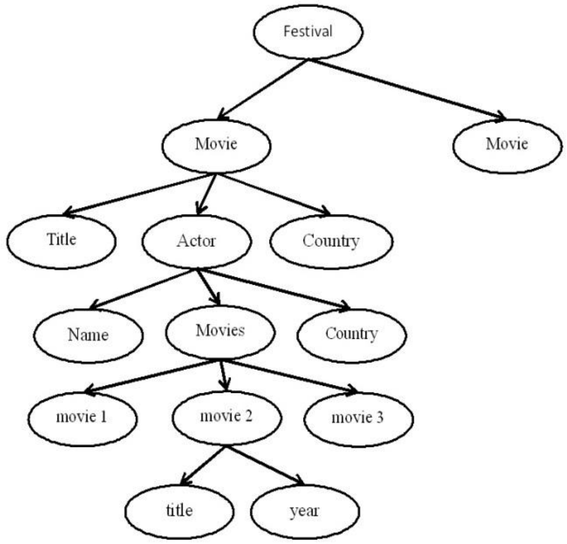
The information in XML databases are correlated by a variety of hierarchical relationships, including parent-child P-C, ancestor-descendant A-D, and siblings relationships (Alsayed Algergawy, Marco Mesiti, Richi Nayak, and Gunter Saake, 2011; Wilde, 2012; Ebtesam Taktek and Dhavalkumar Thakker, 2020; Amjad Qtaish and Jalawi Alshudukhi, 2022). An indexing system is required that can represent the node's location in XML databases and effectively and efficiently direct the user query to the intended node (Yun, J.H. and Chung, C.W., 2008). Labelling schemes were exploited as indexing models that generates a single number for each tag in XML data. The node label describes the node path from the root as well as node's relationships in the tree (Mlynková, 2008). So, labels can improve the XML query performance through matching the patterns of the query with XML database. Since, the patterns of user query and database are similar (Yu, J.X., Luo, D., Meng, X. et al, 2005; G. Wang and M. Liu, 2003; Farag Azzedin, Salahadin Mohammed, Mustafa Ghaleb, Jaweed Yazdani, and Adel Ahmed, 2020).

The time consumption for labelling XML documents has been demonstrated to be determined by the numbers XML elements and the document size (L. Xu, Z. Bao, and T. W. Ling, 2007). Despite that, they rarely took into account the architecture of XML data. The tree shape in Figure (1b) is shallower than the tree in Figure (1a), and the former tree is wider than the latter. Researchers in the XML domain use time and storage space to assess the performance of the proposed labelling schemes. But, they did not measure the effectiveness of the novel models with variant structures of XML data.

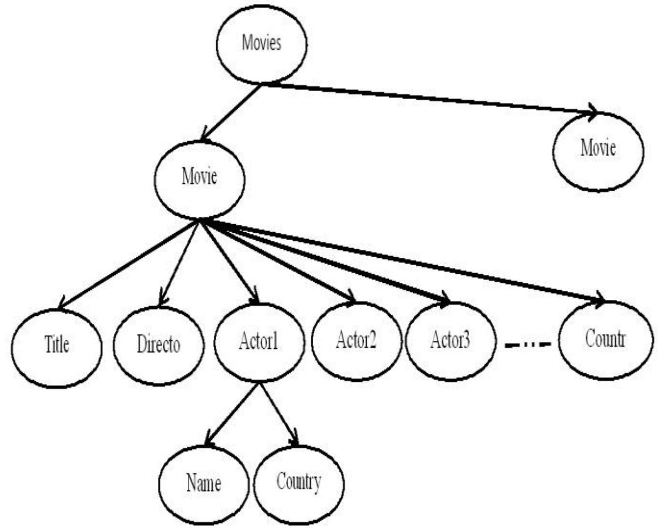
University of Thi-Qar Journal

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023



a: Deep pattern of XML tree



b: Wide pattern of XML tree

Figure (1): Varies structures of XML documents.

XML labelling schemes was classified into; Interval-Based labelling schemes, Prefix-Based labelling schemes, Multiplicative labelling schemes, and Hybrid labelling schemes (Subramaniam, Samini and Haw, Su-Cheng, 2014). This research compares the performance of Containment which, is a class of Interval-Based labelling schemes, and Dewey, which belongs to Prefix-Based labelling schemes.

The remainder of the study is structured as follows: Partition (2) examines a group of existing works. Partition (3) analyses the usefulness of XML schemes. Partition (4) involves discussion of the study findings, and Part (5) concludes the paper.

2. The Related Works

XML data have been employed in a variety of fields for data representation, exchange, and data warehousing (S. S. Chawathe, A. Rajaraman, H. Garcia-Molina, and J. Widom, 1996; S. S. Chawathe et al., 1999; G. Cobena, S. Abiteboul, and A. Marian, 2002), mathematics (Mathematics Markup Language (MathML)) (I. P. M. P. Carlisle, D., 2014), and healthcare (P. T. T. Thuy, Y.-K. Lee, and S. Lee, 2013). The utilisation of XML data has led to an increase in data production, and the demand for a technology able to organise the increased data is necessary (B. G. Assefa and B. Ergenc, 2012; M. Duong and Y. Zhang, 2008; L. F u and X. Meng, 2013). The XML labelling scheme is the required technology that produces a unique identifier called a

University of Thi-Qar Journal

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023

label that describes the node's relationships (B. G. Assefa and B. Ergenc, 2012; T. A. Ghaleb and S. Mohammed, 2015; Ebtesam Taktek and Dhavalkumar Thakker, 2020).

The XML labelling schemes were developed to enhance the processing of user queries. There are many XML query languages, such as XPath and XQeury. The structures of the languages are homogeneous with the structures of the XML dataset. The XML labelling scheme plays an important role in XML database management by comparing the nodes' relationships in XML databases and user queries (Yun, J.H. and Chung, C.W., 2008). In other words, the performance of query processing relies on the performance of a scheme. It can reduce the comparison time by generating a short label.

The following section will explain two popular labelling schemes, namely Containment and Dewey that are based on Interval and Prefix techniques, respectively. These schemes are considered the fundamental techniques for many XML labelling schemes.

2.1. Interval-Based Labelling Schemes

The first approach for encoding XML elements in this category was designed by (Dietz, 1982). It used the interval between numbers in the nodes' labels to define the relationships of nodes (X. Wu, M.-L. Lee, and W. Hsu, 2004) as cited by (Almelibari, 2015) . The model (Dietz, 1982) produces node labels during the '*Preorder'* and *'postorder'* traversals of XML tree. Each label consists of two digits that represent the location of node in the XML document (L. Xu, T. W. Ling, and H. Wu, 2012). For instance, the node Conference in Figure (2) is the parent of node Paper-1 because 1 is less than 2 in preorder and 3 is less than 7 in postorder. However, the labels of these nodes (i.e., Conference and Paper-1) do not give clear evidence that they have parental relationships.

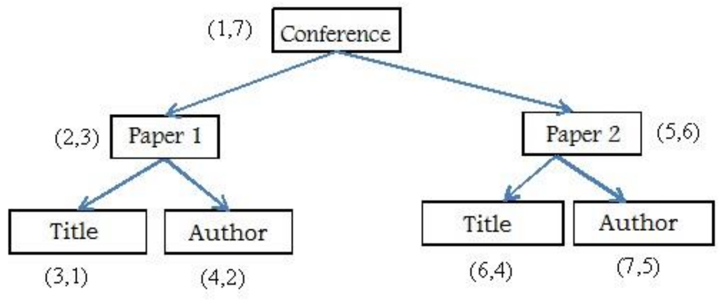


Figure (2): Pre-order / Post-order labelling scheme.

University of Thi-Qar Journal

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023

Another team of researchers (C. Zhang, J. Naughton, D. DeWitt, Q. Luo, and G. Lohman, 2001) proposed an extension to this scheme to address this shortcoming. A label in the suggested scheme consists of the following elements: *'Start'*, *'End'*, and *'Position'*. Where Start and End represent the scope of the descendant label in the XML document. The Position element represents the node level in the XML tree, which explains the node path from the root. The parent child P-C relationship can be noticed in this scheme because the child's position is one higher level than that of the parent (C. Zhuang and S. Feng, 2012; Subramaniam, Samini and Haw, Su-Cheng, 2014). For example, it can be noticed that the Conference node is parent of the Paper-1 node because the latter level increased by one of the former level.

$$Level_{Paper\text{-}1} = Level_{Conference} + 1$$

The property of including the children labels within the span of their ancestor label known as *Containment* (L. Xu, Z. Bao, and T. W. Ling, 2007).

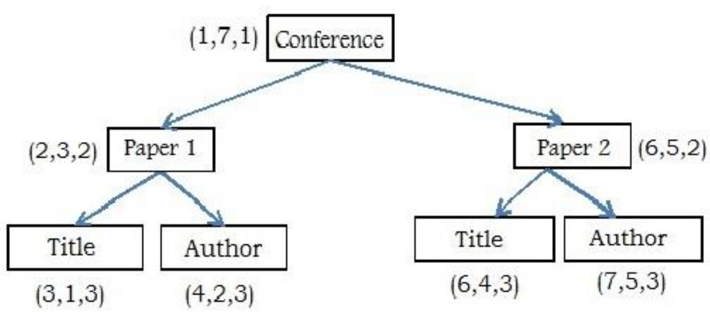


Figure (3): Containment labelling scheme.

The approach of (C. Zhang, J. Naughton, D. DeWitt, Q. Luo, and G. Lohman, 2001) focused on the representation of the node's relationships in Interval-Based labelling schemes. However, more studies (Subramaniam, Samini and Haw, Su-Cheng, 2014) researched the ease of generating labels in this class of labelling scheme.

They simplified the process of creating labels of the (Dietz, 1982) model by designing a novel labelling scheme. As shown in Figure (4), the algorithm of (Subramaniam, Samini and Haw, Su-Cheng, 2014) generates a single label for each node including (*level*, *ordinal*, *rID*).

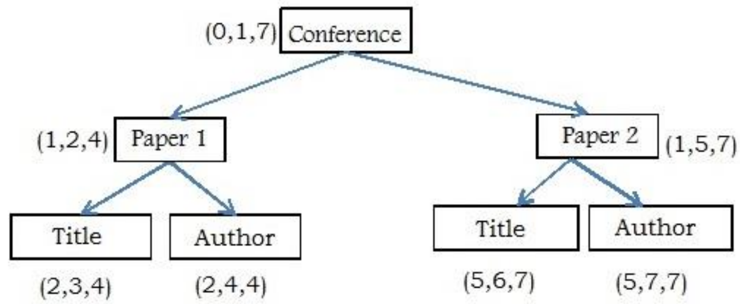


Figure (4): ReLab model for labelling XML data.

Where level is the position of XML element beginning at level 0, which is the root level. The ordinal vector is the ordinal of the most right sibling in a subtree. The vector rID is the distinct digit assigned to the node during preorder tree traversal.

Interval labelling schemes make two visits to XML data to allocate labels for the XML elements. This technique is costly in terms of both storage space and time, with the latter increasing exponentially as the tree grows. A method for producing labels in linear time was required (V. Sans and D. Laurent, 2008) as will be discussed in the following subsection.

2.2. Prefix-Based Labelling Schemes

This class of schemes is identical to the *Dewey Decimal Coding* technique used by librarians (V. Sans and D. Laurent, 2008). According to a group of researchers (B. G. Assefa and B. Ergenc, 2012), this type of scheme could represent different types of hierarchical relationships among the XML tree elements. The labels of Prefix schemes consist of two sections separated by a delimiter, either ',' or '.', and are produced by the technique of depth first search. The prefix section is the parent label and, the second section of the label is the child (B. G. Assefa and B. Ergenc, 2012; V. Sans and D. Laurent, 2008; I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, 2002). *Dewey Encoding* is a popular Prefix labelling scheme proposed by (I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, 2002).

The labelling scheme in (I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, 2002) was designed to process queries of order-sensitive such as (I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, 2002; X. Wu, M.-L. Lee, and W. Hsu, 2004): *Following*, *Preceding*, *Following-Sibling*, *Preceding-sibling*, and *Position = n*. The first and second types ignore the context tag's ancestor or descendant in favour of focusing on whether the tag comes after or before it. The third and fourth types will return the siblings of the following and preceding children. The final query type will simply retrieve information for the specified node.

The *Dewey Order* labelling scheme was designed by (I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, 2002) to be compatible with order-sensitive queries by merging a couple of numbering methods dubbed Global Order and Local Order. The first method allocates a single number for every element in the XML dataset based on its global order in the dataset, as illustrated in Figure (5). The second method assigns a digit to each tag depending on its position among its siblings, as clarified in Figure (6). Figure (7) demonstrates a combination of these numbering models and forms the *Dewey Order* labelling scheme. The label of the resulting labelling scheme represents the node's context in the XML tree.

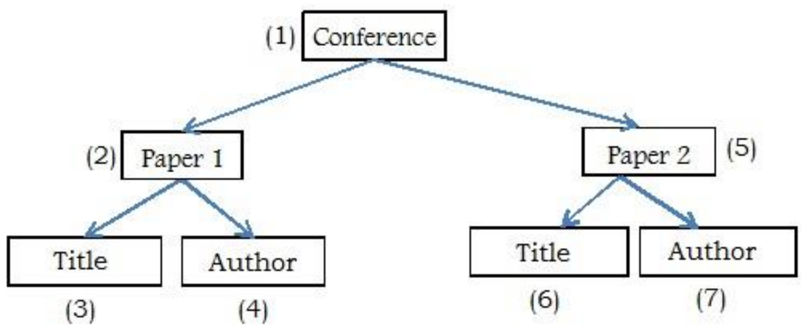


Figure (5): Global order model.

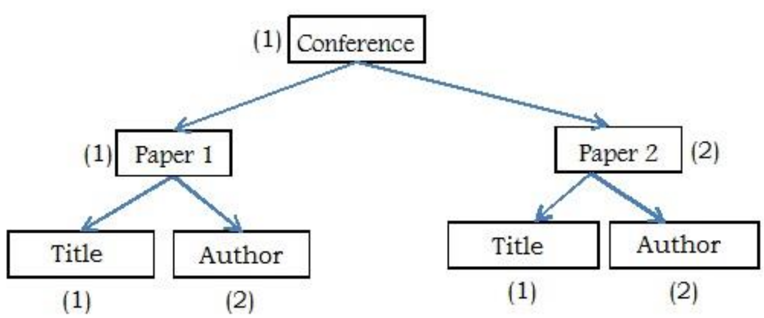


Figure (6): Local order model.

University of Thi-Qar Journal

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023

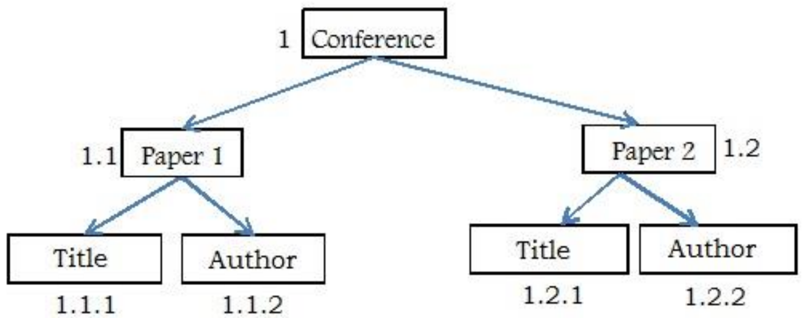


Figure (7): Dewey method.

The Dewey method easily explains the node's relationship in XML documents. According to Figure (7), the elements Paper-1 and Paper-2 are siblings because their prefix part has the same father number (i.e., 1) and their labels are consecutive (i.e., 1 and 2). Furthermore, it can be seen that the Conference element is a grandfather of the tag Author because the label of Author starts the label of Conference, as demonstrated in Figure (7).

There are several schemes that adopted the *Dewey Encoding* to design their schemes, such as ORDPATH (P. O'Neil, E. O'Neil, S. Pal, I. Cseri, G. Schaller, and N. Westbury, 2004), *Dynamic Float-Point Dewey* 'DFPD' (J. Liu, Z. Ma, and L. Yan, 2013), *Labelling Scheme for Dynamic XML data* LSDX (M. Duong and Y. Zhang, 2005), *Compressed Dynamic Labelling Scheme* Com-D (M. Duong and Y. Zhang, 2008), OrderedBased (B. G. Assefa and B. Ergenc, 2012), etc. These labelling schemes were suggested for labelling dynamic XML trees without a relabeling process. These labelling schemes are outside the context of the article and will not be addressed.

3. A Pertinence Comparison Investigation of XML Labelling Schemes

XML labelling models were used to describe the structural relationship of XML elements through labels (Mlynková, 2008; Ebtesam Taktek and Dhavalkumar Thakker, 2020). As a result, query processing efficiency can be improved using the labels instead of the actual document (Yu, J.X., Luo, D., Meng, X. et al, 2005; G. Wang and M. Liu, 2003).

Many XML labelling schemes have been suggested in order to determine the best method that generates labels efficiently and effectively employing a small amount of memory. The majority of studies have been devoted to defeat the weaknesses of preceding schemes by proposing a new algorithms. However, researchers did not examine their work against various XML database structures.

Two common labelling schemes were used to analyse the relevance between the type of schemes and XML tree structure: Dewey Encoding (Dewey in short) and Containment. A series of investigations were carried out to assess the production of these schemes in terms of pace and memory space. A number of real XML documents were used: nasa, dblp, and Treebank-e which have different structures and are available for research purposes on the Washington University website (Xml data repository).

The first set of experiments was executed to determine the time consumed for labelling these datasets. The scheme that allocates labels for a specific structure of an XML database in a short time indicates its relevance for that database.

More analyses were conducted to determine the suitability of the scheme with XML data by measuring the label size. Small label sizes can enhance the performance of query processing. A short label length reduces the time required to compare query structures and node labels (Yun, J.H. and Chung, C.W., 2008).

4. Experiments and Results Analyses

4.1. System Setup

Several tests were carried out using Eclipse 'version 4.4.0RC1 32-bit', which is a programming platform to run Java code on a computer equipped with an Intel (R) Core (TM) i7-8550U CPU 1.80GHz 1.99 GHz, RAM 8 GB of RAM, and Win.10 Pro 64-bit. Furthermore, SPSS 23, a popular statistical application, was used to analyse the outcome. The article evaluated the productivity of the two most known algorithms for labelling XML data; Dewey and Containment. These schemes are employed to label a group of XML datasets, which are listed in Table (1) with their specifications.

Table (1): XML datasets and their specifications.

University of Thi-Qar Journal

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023

XML database	No. of elements	Max Depth (Level)	File Size
Nasa	476646	8	23MB
Dblp	3332130	6	127MB
Treebank-e	2437666	36	82MB

4.2. Discussion

The information in Table (2) is depicted in Figure (8). The type of XML labelling scheme in the graph is represented by the vertical-axis, and the horizontal-axis represents the mean time for producing labels.

Table (2): The time required for numbering tags of the XML datasets.

Scheme Category	nasa		dblp		Treebank-e	
	Mean	STD	Mean	STD	Mean	STD
Dewey	376.48	9.865	2614.48	84.568	1638.78	29.708
Containment	392.12	11.739	2804.33	99.889	1696.47	60.742

It can be noticed that in Table (2) and illustrated in Figure (8) the meantime of Dewey 2614.78 for encoding dblp dataset which shorter than that of Containment 2804.33 for encoding the same dataset

As stated in (L. Xu, T. W. Ling, and H. Wu, 2012) Containment explores an XML database between 1 and 2n times, where ‘n’ is the number of the database elements. To label the elements, therefore, it took longer to label dblp nodes and other databases, as illustrated in Table (2).

University of Thi-Qar Journal

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023

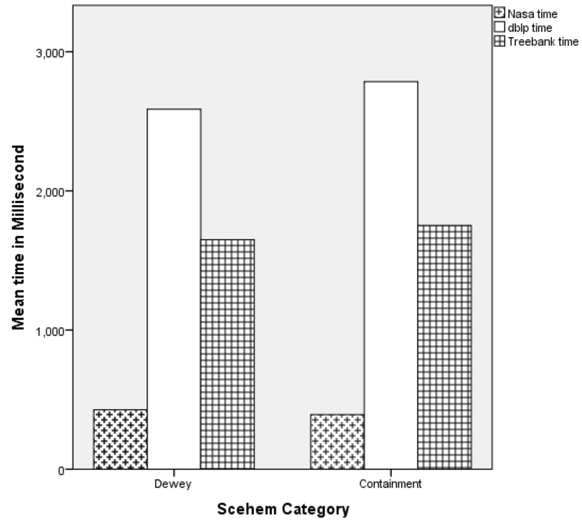


Figure (8): The time needed for labelling XML datasets.

Another inspection was conducted to assess the storage space needed by Dewey and Containment to label the same set of XML databases.

Prefix generates labels in a sequential manner, and the length of the label is determined by the node position in the XML tree. The level of the deepest leaf in the Treebank-e dataset is 36 because the depth of the dataset is 36. Thus, Dewey allocates 48,921.83 KB in the memory to store labels of Treebank-e dataset, as clarified in Figure (3). As mentioned previously, the Dewey labelling scheme generates labels sequentially that consist of sections. The number of label sections depends on the node's location in the XML data (I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, 2002). Moreover, the element number of dblp databases is the largest among other databases in Table (1). So, as can be seen in Figure (9), the Containment consumed 59,858.03 KB in memory to store labels of dblp nodes which, is the largest storage space.

In order to verify our findings, we compared them to the results reported in (L. Xu, T. W. Ling, and H. Wu, 2012). Our findings were found to match the published findings.

Table (3): The size of memory required to allocate labels for XML database.

Scheme Category	nasa Space	dblp Space	Treebank-e Space
Dewey	7119.90	37664.14	48921.83
Containment	7754.21	59858.03	44187.48

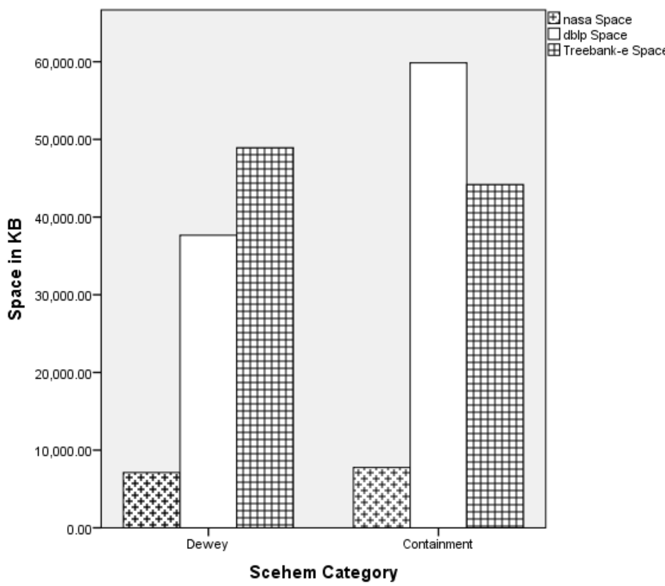


Figure (9): Storage space allocated to store labels of XML datasets.

5. Conclusion

The study analysed the issue of the relevance of the XML labelling scheme type in relation to the structure of XML data. This problem has received insufficient attention in the XML literature. It has the potential to reduce the effort required to propose a new approaches by exposing the shortcomings of other schemes. This makes it easier to design and optimise novel schemes. To accomplish this, three different structures of real XML databases (nasa, dblp, and Treebanck-e) were used. The nodes of these databases were labelled by the well-known XML labelling schemes (Dewey and Containment). Time and space constraints were employed in a series of examinations to measure the best performance of the labelling scheme for a distinct XML dataset. In terms of time, it was discovered that Dewey consumed less time than Containment to label the three databases. From a space perspective, Dewey is suited for shallow structures of XML documents, and Containment is suited for deep documents. The applicability of more schemes to these various XML database structures should be investigated in future work.

ACKNOWLEDGMENT

This research was supported in part by the Iraq Ministry of Higher Education and Scientific Research and the University of Diyala.

References

Almelibari, A. (2015). *Labelling dynamic xml documents: A groupbased approach.* Sheffield: Ph.D. dissertation, University of Sheffield.

Alsayed Algergawy, Marco Mesiti, Richi Nayak, and Gunter Saake. (2011). XML data clustering: An overview. *ACM Computing Surveys (CSUR), 43*(4), 41.

Alsayed Algergawy, Richi Nayak, Gunter Saake. (2010). Element similarity measures in XML schema matching. *Information Sciences, 180*(24), 4975-4998.

Amjad Qtaish and Jalawi Alshudukhi. (2022). An Efficient Prefix-Based Labeling Scheme for XML Dynamic Updates Using Hexagonal Pattern. *IEEE Access, 10*, 57107 - 57123.

B. G. Assefa and B. Ergenc. (2012). Orderbased labeling scheme for dynamic xml query processing. *Multidisciplinary Research and Practice for Information Systems.Springer*, 287–301.

Bellahs`ene, Z. (2003). Database and XML Technologies. Berlin, Germany: Lecture Notes in Computer Science. Springer.

Bertino, E. and Ferrari, E. (2001). Xml and data integration. *IEEE internet computing, 5*(6), 75–76.

C. Zhang, J. Naughton, D. DeWitt, Q. Luo, and G. Lohman. (2001). On supporting containment queries in relational database management systems. *ACM SIGMOD Record, 30*(2), 425–436.

C. Zhuang and S. Feng. (2012). Full tree-based encoding technique for dynamic xml labeling schemes. International Conference on Database and Expert Systems Applications. Springer.

Dietz, P. F. (1982). Maintaining order in a linked list. P. F. Dietz, “Maintaining order in a linked list,” in Proceedings of the fourteenth annual ACM symposium on Theory of computing.

Ebtesam Taktek and Dhavalkumar Thakker. (2020). Pentagonal scheme for dynamic XML prefix labelling. *Knowledge-Based Systems, 209*, 106446.

Farag Azzedin, Salahadin Mohammed, Mustafa Ghaleb, Jaweed Yazdani, and Adel Ahmed. (2020). Systematic Partitioning and Labeling XML Subtrees for Efficient Processing of XML Queries in IoT Environments. *IEEE Xplore, 8*, 61817 - 61833.

G. Cobena, S. Abiteboul, and A. Marian. (2002). Detecting changes in xml documents. Data Engineering, 2002. Proceedings. 18th International Conference on. IEEE 2002 .

G. Wang and M. Liu. (2003). Query processing and optimization for regular path expressions. International Conference on Advanced Information Systems Engineering, Springer.

I. P. M. P. Carlisle, D. (2014). *W3c, mathematical markup language (mathml) version 3.0.* Retrieved 4 15, 2023, from https://www.w3.org/TR/MathML/chapter1.html#intro.example

I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang. (2002). Storing and querying ordered XML using a relational database system. SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data.

J. Liu and X. Zhang. (2016). Dynamic labeling scheme for XML updates. *Knowledge-Based Systems, 106*, 135-149.

J. Liu, Z. Ma, and L. Yan. (2013). Efficient labeling scheme for dynamic XML trees. *Information Sciences, 221*, 338-354.

Kurt, A. and Atay, M. (2002). Kurt A, Atay M. An experimental study on query processing efficiency of native-XML and XML-enabled database systems. InInternational Workshop on Databases in Networked Information Systems. *International Workshop on Databases in Networked Information Systems. Springer*, 268–284.

L. F u and X. Meng. (2013). Triple code: An efficient labeling scheme for query answering in xml data. Web Information System and Application Conference (WISA), 2013 10th. IEEE.

L. Xu, T. W. Ling, and H. Wu. (2012). Labeling dynamic xml documents: an order-centric approach. *Knowledge and Data Engineering, IEEE Transactions, 24*(1\), 100–113.

L. Xu, Z. Bao, and T. W. Ling. (2007). A dynamic labeling scheme using vectors. International Conference on Database and Expert Systems Applications. Springer.

M. Duong and Y. Zhang. (2005). Lsdx: a new labelling scheme for dynamically updating xml data. Proceedings of the 16th Australasian database conference-Volume 39. Australian Computer Society, Inc., 2005.

M. Duong and Y. Zhang. (2008). Dynamic labelling scheme for xml data processing. *On the Move to Meaningful Internet Systems: OTM 2008. Springer*, 1183–1199.

Mlynková, I. (2008). An analysis of approaches to xml schema inference. Mlynková, Irena. "An analysis ofIEEE International Conference on Signal Image Technology and Internet Based Systems.

P. O'Neil, E. O'Neil, S. Pal, I. Cseri, G. Schaller, and N. Westbury. (2004). ORDPATHs: insert-friendly XML node labels. SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, June 2004.

P. T. T. Thuy, Y.-K. Lee, and S. Lee. (2013). Semantic and structural similarities between xml schemas for integration of ubiquitous healthcare data. *Personal and ubiquitous computing, 17*(7), 1331–1339.

S. S. Chawathe et al. (1999). Comparing hierarchical data in external memory. *VLDB, 99*, 90–101.

S. S. Chawathe, A. Rajaraman, H. Garcia-Molina, and J. Widom. (1996). Change detection in hierarchically structured information. *ACM SIGMOD Record, 25*(2), 493–504.

Subramaniam, Samini and Haw, Su-Cheng. (2014). ME labeling: A robust hybrid scheme for dynamic update in XML databases. Telecommunication Technologies (ISTT), 2014 IEEE 2nd International Symposium.

T. A. Ghaleb and S. Mohammed. (2015). A Dynamic Labeling Scheme Based on Logical Operators: A Support for Order-Sensitive XML Updates. *Procedia Computer Science, 57*, 1211-1218.

V. Sans and D. Laurent. (2008). Prefix based numbering schemes for XML: techniques, applications and performances. *Proceedings of the VLDB Endowment, 1*(2), 1564–1573.

V. Sans and D. Laurent. (2008). Prefix based numbering schemes for XML: techniques, applications and performances. *Proceedings of the VLDB Endowment, 1*(2), 1564–1573.

University of Thi-Qar Journal

ISSN (print): 2706- 6908, ISSN (online): 2706-6894

Vol.18 No.2 June 2023

Wilde, E. (2012). *Wilde's WWW: Technical Foundations of the World Wide Web*. Springer Science & Business Media.

Win, K.M., Ng, W.K. and Lim, E.P. (2003). ENAXS: efficient native XML storage system. Berlin Heidelberg.: Asia-Pacific Web Conference, Springer.

X. Wu, M.-L. Lee, and W. Hsu. (2004). A prime number labeling scheme for dynamic ordered XML trees. Boston, MA, USA: Proceedings. 20th International Conference on Data Engineering, IEEE.

Xml data repository. (n.d.). Retrieved 3 15, 2023, from https://aiweb.cs.washington.edu/research/projects/xmltk/xmldata/

Yu, J.X., Luo, D., Meng, X. et al. (2005). Dynamically Updating XML Data: Numbering Scheme Revisited. *World Wide Web, 8*, 5-26.

Yun, J.H. and Chung, C.W. (2008). Dynamic interval-based labeling scheme for efficient xml query and update processing. *Journal of Systems and Software, 81*(1), 56–70,.