The International Journal of Engineering and Information Technology

journal homepage:www.ijeit.misuratau.edu.ly

A NEW EFFECTIVE LABELLING SCHEME For EFFICIENT QUERYING And UPDATING XML DOCUMENTS

Alhadi A. Klaib
Department of Software Engineering
Elmergib University
alhadi.klaib@elmergib.edu.ly

Abstract— **Nowadays Extensible Mark-up Language (XML) is a dominant technology for formatting and exchanging data across the Internet world. Updating and retrieving a massive amount of XML data is an interesting and active research area. In addition, indexing XML data is a significant task to improve the efficiency of XML queries. Labelling nodes is the used technique for indexing XML data efficiently. There are many labelling schemes that have been proposed. However, these schemes have many limitations and shortcomings. Therefore, this paper aims to propose a new XML labelling scheme that addresses the issue of efficiency of XML query performance. Thus, this paper developed a new XML labelling scheme. Consequently, four experiments were designed in order to evaluate this. The results of these experiments suggest that the proposed scheme achieved the target results and showed an improvement in the performance and the efficiency of labelling XML documents.**

***Index Terms*— XML data, XML labelling scheme, Data retrieval, Querying XML data, Information retrieval.**

I. INTRODUCTION

Indexing XML is a very important factor used to enhance XML data queries. In general, the efficiency of the performance of any query in a database is based on the indexing [1, 2]. labelling XML data is the technique used to index XML data efficiently and robustly. Labelling XML data is performed by assigning labels to all nodes in that XML document. Every node is provided with a unique label that can be used to build the relationship among nodes in that XML tree [3, 4]. Initially, the main focus of the XML research was about handling static documents in terms of data retrieval and navigation. As a result, many labelling schemes have been proposed [5-9]. However, and the key problem is that none of these schemes suit all users' requirements, Thus they are only appropriate for certain circumstances. The main challenges are with dynamic XML data since static XML data has been efficiently processed by proposing many of successful schemes such as the Dewey scheme and containment scheme [10, 11]. The case with dynamic XML data is different as the XML databases still struggle to manage large numbers of relabelling cases as dynamic XML data. Furthermore, many labelling schemes have been proposed for dynamic XML data as well [8, 12-16]. Any efficient dynamic XML data labelling scheme should provide effective query performance, labelling XML data efficiently, reducing the required relabelling cases, and determining the relationships.

Therefore, this paper addressed the lack of efficiency labelling scheme and came up with a proposed solution for this problem as a new XML labelling scheme. It is a hybrid scheme for labelling XML data named Clustering-based Labelling Scheme (CLS). This scheme is intended to address some limitations and challenges of indexing XML data. Theoretically, the proposed scheme is based on dividing the nodes of an XML document into clusters. Two existing labelling schemes, which are the Dewey and LLS labelling schemes, were selected for labelling these clusters and their nodes. Furthermore, the proposed scheme was designed and developed. Four experiments were designed in order to evaluate the proposed scheme. This proposed scheme was developed and evaluated to provide an improvement to the query processing, reduce the relabelling to the lowest possible level, and update efficiency. The results of these experiments suggest that the proposed scheme achieved the target results.

The remainder of the paper is structured as follows: section two discusses the background and previous work. Section three illustrates and describes the research methodology. Section four demonstrates the design and implementation. Section five illustrates the results. Section six demonstrates the discussion and main findings. Finally, the conclusion was explained in section seven.

Received 29 Apr , 2022; revised 31 Aug, 2022; accepted 3 Aug, 2023.

Available online 18 Dec, 2023.

II. BACKGROUND AND PREVIOUS WORK

Due to the increase in the importance of XML data management, many researches focus on this area. In addition, labelling schemes are an interesting topic for XML data developers and researchers as this area plays a key role in improving query performance. In order to query XML data, an efficient XML labelling scheme is needed [17]. However, even though heavy research on labelling schemes and techniques has been carried out, there is as yet no appropriate labelling scheme for all users' requirements [18]. Thus, this research focuses on this area to develop a new approach that resolves some of the challenges that these existing labelling schemes face. In an earlier investigation, we performed a preliminary study on the indexing XML techniques. We conducted further study that proposed an initial idea for a new labelling scheme. Consequently, this study extended the previous research and implemented this proposed scheme. An implementation of the Dewey and LLS labelling schemes were also carried out for the testing purpose. This section also presents the significance of the labelling schemes.

A. Level-based Labelling Scheme

This is one of the two labelling schemes that used in the proposed scheme, and thus, this section discusses this scheme. The LLS labelling scheme was developed to combine the advantages of both the interval and prefix labelling schemes, and to avoid their disadvantages. Therefore, this labelling scheme supports the processing of simple queries and twig queries. The LLS is based on the levels of the nodes in XML trees and the summary of an XML tree. The element labels and values are firmly fixed with a structural summary so the method provides efficient query processing. This labelling scheme shows high performance in comparison to existing labelling schemes. The LLS has shown some advantages such as: the LLS maintains the best features and advantages of interval labelling and prefix labelling schemes. Second, the LLS provides a constant size of the label s regardless of the data-tree depth. Path information is easily available since the root path of an element can be produced from the labels. Third, LLS is based on the levels of the tree. Knowing the level at which a node is located can speed up query processing by improving the search space at an early evaluation stage.

B. Dewey Labelling Scheme

This scheme is also one of the the two labelling schemes that used in the proposed scheme, and thus, this section discusses this scheme. The Dewey labelling scheme is also called Dewey code labelling. It is one of the prefix labelling schemes. This labelling scheme was introduced for general knowledge classification. Tatarinov and colleagues [10] first used this labelling scheme for XML tree-shaped data. Each node is related with a vector of numbers that reflect the node-ID path from the root to the designated node. Moreover, this labelling scheme is classified as a node index and a path index since each node is represented as a complete path from the root to the indexed node [17].

A prefix matching operation on the index string is carried out in order to determine whether a relationship of a parent–child or an ancestor–descendent exists. In a certain data-tree, node x is an ancestor of node y if the label of node x is a substring of the label of node y.

C. Significance of XML Labelling Schemes

The rapid growth in XML data has led to this high demand for querying this data. Furthermore, indexing is one of the techniques used to achieve high query performance and to retrieve data very fast. One of the main shortcomings of indexing XML data is that there is always a trade-off between the size and efficiency of the index. In other words, indexes can be large in order to provide high performance; or small size with weak performance. Another common problem with indexing XML data is that the update operations are usually expensive [17]. Therefore, this research concentrates on how we can improve the performance of query processing by enhancing the updates operations.

III. METHODOLOGY

This paper came up with an idea of developing a new approach. The idea of this approach is to develop a hybrid labelling scheme. The proposed scheme is based on clustering nodes in order to ease the determination of the child–parent and sibling relationships as the child–parent relationship is available in each XML document. Moreover, these relationships also ease the process of inserting new nodes. The parent–child clustering-based technique helps in dealing with a small tree rather than the entire XML tree. It was also found that the parent–child clustering technique supports the labelling process, and is more efficient than the simple tree in terms of the accuracy and spent time for query processing [17].

One of the features of a clustering-based technique is that it uses two labels for every node as this idea eases the procedure of labelling nodes that share the same cluster; whereas the label of the cluster is used to link that cluster with the entire tree. This feature assists in determining the relationships among nodes that form different clusters. It was stated that a scheme is developed to provide fast identification of relationships, as this feature helps in the optimising of query processing. Therefore, these advantages support the proposed scheme, which consequently helps in enhancing the query processing and other targeted features of the proposed scheme such as improving the process of labelling XML documents.

A. The proposed Approach

This section starts by explaining the approach of the proposed labelling scheme such as the idea of this scheme and the mechanism for implementing it, as well as the data model. It continues by describing how the update cost should be affected in this proposed scheme; subsequently, the design of the proposed scheme is illustrated including the implementation of the LLS and the Dewey labelling schemes for evaluation purposes. The experimental setup is discussed in this section, as well as the objectives and types of the experiments. Theoretically, the idea of the proposed labelling scheme is to divide the whole data tree into small groups (clusters). This mechanism makes each cluster itself a

sub-data tree. The advantage of this mechanism is to reduce the number of required re-labelling cases to the lowest possible level, and as a result improve the efficiency of the query performance processing. Subsequently, two XML labelling schemes were used to label the nodes and their clusters of a data tree. These two schemes are the Dewey labelling scheme which was used to label the clusters, and the LLS labelling scheme which was used to label the nodes of the data tree. Figure 1 shows the proposed labelling scheme.

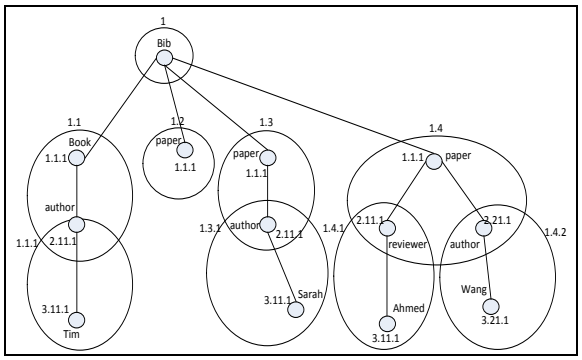

Figure 1. The proposed labelling scheme for an XML document

B. *The Implementation Plan of the LLS and the Dewey Labelling Schemes:*

The evaluation of the proposed scheme is based on testing the proposed scheme against the LLS scheme and the Dewey labelling scheme. Thus, the implementation of both the LLS and Dewey labelling schemes was required as they are not available either as open source code or on the shelf software. Thus, a new implementation of the LLS scheme was carried out. The Dewey labelling scheme is not available as open source code since it is a theory that other researchers have used as a base to implement their interpretations of it in software. Since the Dewey labelling scheme is not available as open source code, it was implemented using Java as a coding language. The implementation was according to the data model [10].

C. *Testing Approach*

In general, the purpose of testing the proposed scheme is to ensure it achieved objectives. The experiments were executed twenty times for each query and then the average was taken in order to gain as accurate results as possible. This number of executions for each experiment was chosen as a fair number to get an accurate result by calculating the average of these twenty executions. Other researchers used the same number for testing similar labelling schemes.

Regarding the dataset for the testing stage, an investigation into the available XML datasets and benchmarks was carried out. Consequently, the XMark benchmark was selected for testing the testing experiments. The XMark benchmark helps both implementers and users to obtain insights into the XML storage. XMark was chosen to test the proposed scheme for the following reasons: first and most importantly, the XMark was used to evaluate the performance of the LLS scheme by the founder. Thus, it would be appropriate to use the same dataset to evaluate the proposed scheme and compare the results. Secondly, this benchmark is widely used to test XML queries and XML database performance. Moreover, XMark is a good choice since it has many features such as providing a document generator to create documents in different sizes. Thus, users can generate datasets that are appropriate for their requirements. Also, this benchmark provides a binary version of the XMark that can be run as an independent platform on any operating system. XMark provides a broad range of queries – twenty-one in total. These queries are designed to evaluate different aspects of the datasets. They are divided into groups based on their goals and purposes.

D. *Design and Implementation*

The first stage of the design is to parse the XML document. The DOM was selected for this platform because it is the most suitable option as it is a commonly used parser and straightforward to apply. The XML DOM is a standard means for accessing and manipulating XML documents. It has also already been employed by researchers for designing a labelling scheme. Java 1.8 was used for implementing the algorithms for the proposed scheme and the LLS and Dewey labelling schemes. MySQL Workbench was used for building the databases and as a database management system to save the data for all schemes. The second stage is to label each node and any linked cluster. The first step is to read the XML document and save all nodes in a nodelist. Inserting a new node has different types of cases such as the node is a first child for its parent, or if the node is inserted between two existing nodes and so on.

IV. RESULTS

The items that need to be tested are: the proposed scheme, the LLS labelling scheme, and the Dewey labelling scheme. Furthermore, there are certain features that need to be tested, namely the query performance, efficiency of labelling XML documents, efficiency of scalability, and functionality of the proposed scheme. The testing technique is based on running nineteen queries of the XMark dataset to test the target schemes in order to compare the results and ascertain the achievement of the proposed scheme. The experiments were carried out to test both static and dynamic documents separately as follows:

A. *Experiments for Static Documents:*

- **Labelling XML documents:**
  The idea of this experiment is to compute the time needed for assigning labels to the XML data nodes. The proposed labelling scheme shows difference in spent time for labelling XML documents from the LLS and Dewey schemes. A comparison of the results shows that the proposed labelling scheme achieved better results than the LLS and the Dewey labelling schemes.
- **Determining Different Relationships:** this experiment computed the calculation time of the relationships between two nodes using the labels between these nodes. Five aspects

and relationships were measured in this experiment, namely: sibling, parent/child, ancestor/descendant, order, and level. A comparison of the results shows that the proposed labelling scheme achieved better results than the LLS and the Dewey labelling schemes in four relationships, namely, order, level, sibling, and parent/child; whereas the Dewey scheme achieved best result in the ancestor/descendant relationships.

- **Query Efficiency Measurement:** this experiment was carried out to measure query efficiency and performance before and after insertions. The proposed scheme achieved the best results in sixteen queries out of nineteen.

B. *Experiments for Dynamic Documents*

Since developing a labelling scheme that handles the dynamic documents is one of the goals of the proposed scheme, these experiments were intended to measure any updates on the dynamic XML documents. The experiments were executed to assess the proposed scheme's ability to deal with different kinds of insertion. Thus, the experiments were divided into three types of groups as follows; experiments for inserting new nodes, experiments for determining different relationships, and experiments for query performance.

- **Inserting New Nodes:** Labelling schemes should be able to handle diversity of insertions as a significant aspect. Thus, different types of insertions need to be executed and tested. Uniform is an important type of insertion that needs to be performed. The time spent and size of the labels is measured in this insertion. Ordered skewed is another type of insertion. It is used to insert new nodes before and after a specific node times [17]. Random skewed is another kind of insertion. New nodes are inserted between other nodes randomly. This type of insertion tests the flexibility of the scheme to deal with random insertions as some schemes have difficulty in doing so.

  Regarding the ordered Skewed, the XMARK1 file was used to test this experiment to measure the time spent. The number of insertions was used to monitor the changes. These numbers start from one thousand up to ten thousand.

  Regarding the size of the labels, the proposed scheme and LLS scheme achieved the same results and better than the Dewey scheme in one testing (7 thousand). However, the proposed scheme achieved best results in nine other experiments.

  Regarding the random skewed, again the XMRK1 file was used to test this experiment to measure the time spent, and the number of insertions was used to monitor the changes. The proposed scheme achieved better results in nine out of ten (2, 3, 4, 5, 6, 7, 8, 9 and 10 thousand) experiments. However, the LLS scheme achieved better results in one experiment (1 thousand).

  With respect to the size of the labels, again, the XMRK1 file was used for this experiment. However, the proposed scheme has not achieved better results in all experiments. The LLS scheme has achieved best results in 1, 2, 5, 6, 7, 8, 9, and 10; whereas, the Dewey scheme was the best one in queries 3 and 4.

- **Determining Relationships:** these experiments test the scheme in terms of determining relationships. Therefore, this experiment observes the time spent for determining relationships after inserting new nodes as follows:
  - **Determining Relationships after Uniform Insertions:** the proposed scheme shows the best results in determining relationships after inserting new nodes between two nodes. Figure 2 shows the results.

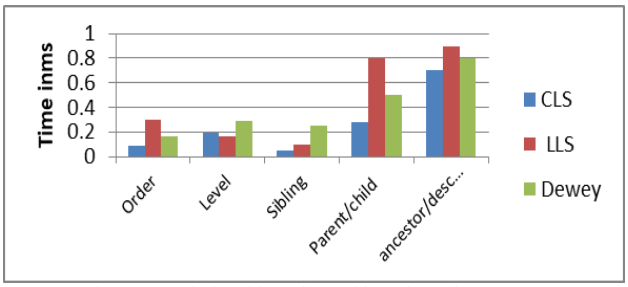

Figure 2. Relationships after uniform insertions

  - **Determining Relationships after Ordered Skewed Insertions:** the proposed scheme achieved better results in all experiments except the ancestor/descendant in which the Dewey scheme shows better results. Figure 3 shows the results.

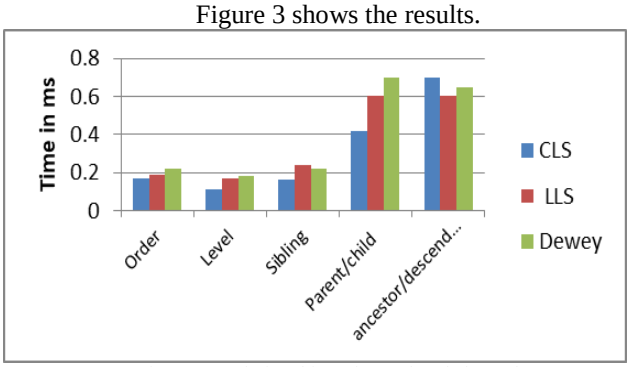

Figure 3. Relationships after ordered skewed

  - **Determining Relationships after Random Skewed:** this experiment is to insert new nodes between two other nodes randomly. The proposed scheme achieved better results in all experiments as can be seen in figure 4.

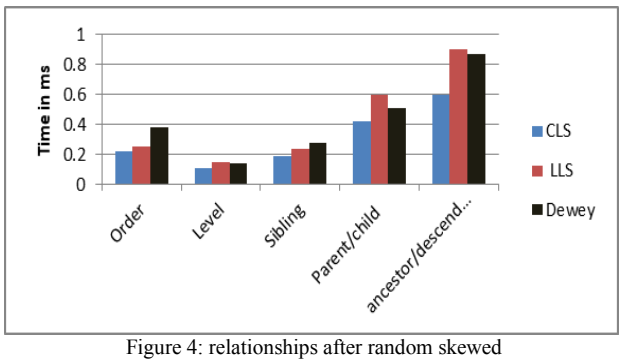

Figure 4: relationships after random skewed

- **Query Performance for Dynamic Documents:** this experiment was carried out to assess the same query performance that was used for the static document. However, the assessment this time is after insertions. Figure 5 illustrates the results.

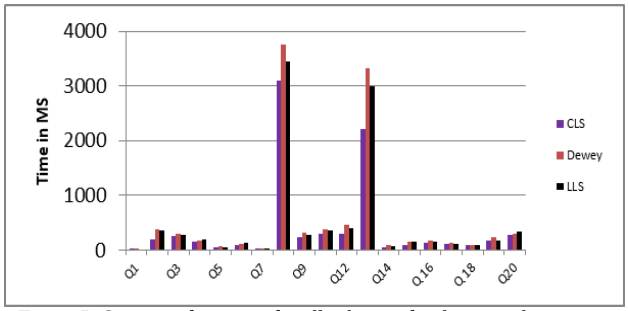

Figure 5. Query performance for all schemes for dynamic documents

The proposed scheme presented best spent time and response times in eighteen queries out of twenty. The LLS scheme achieved best result in queries (1, 5). The query performance on dynamic documents presented a better spent and response time compared to the results of the static documents. However, few exceptions were occurred. Therefore, the proposed scheme demonstrated more efficiency with dynamic documents than the LLS scheme.

V. DISCUSSION AND MAIN FINDINGS

The objective of this section is to discuss the results and the key findings of the testing experiments. The section is also intended to identify the effective characteristics of the proposed scheme as well as any limitations. This task is significant as it can be used as a basis for considering the future work for this research.

The proposed scheme and other schemes were tested to evaluate certain features such as labelling XML documents, determining the relationships, and querying performance for both static and dynamic XML documents. Discussion and assessment of the results of these experiments are as follows:

Regarding the assessment of the labelling XML documents, the experiment's results of spent time in the previous section show that the proposed scheme achieved better performance over the other schemes. Second, with regard to measuring the size of the labels, the proposed scheme was not able to achieve better results than the other schemes. The justification for this failure is due to the method that the proposed scheme used to label the nodes and clusters which allocate two labels for each node. Furthermore, there is always a trade-off between the query performance and the size of the indexes.

With respect to assessment of the determination of the relationships, this experiment was intended to check the determination speed of the relationships. The achieved results from these experiments met the expected goals. In terms of the static XML documents, the results the proposed scheme was faster than the others in all results. In addition, the results of the dynamic documents are also faster than the others with a few exceptions in which the proposed scheme did not achieve better results.

With regard to the assessment of query efficiency, the results as follows; regarding the static XML documents, the results in general show that the proposed scheme achieved the expected results, with a few exceptions. These exceptions are three queries, namely number 7, 17, and 19. Regarding the dynamic documents, all the results met the expectations except query number 19 where LLS achieved a better result than the proposed scheme. To summarise, the proposed scheme has shown an improvement over the other schemes in terms of the efficiency of the query performance. As far as the query efficiency of other labelling schemes is concerned, prefix schemes such as Dewey, LSDX, and ORDPATH support query processing, however, some researchers have already claimed that prefix labelling schemes need more improvement in terms of query processing. Moreover, graph schemes handle the path query efficiently as they all return precise and complete answers.

Concerning the assessment of inserting new nodes, the goal of this experiment is to assess the capability of the proposed scheme to handle a variety of insertions. This experiment was only intended for dynamic documents. The measure was the insertion time and the size of the labels after the insertions. There were three different types of insertions used, namely, uniform insertion, ordered skewed, and random skewed. Regarding the time for inserting new nodes, the proposed scheme has achieved better results than the other schemes in all times of intended insertions with a few exceptions. Concerning the size of the labels, the proposed scheme has beaten the other schemes in uniform and ordered skewed insertions with a few exceptions; however, the proposed scheme failed to beat the others in random skewed insertions.

VI. CONCLUSION

The idea of this kind of schemes is to save values that represent the locations of the nodes in the XML tree structure. These values are used to determine the relationships of a node such as parent, child, sibling, ancestor and descendent. Thus, this research argued that using certain existing labelling schemes to label nodes, and using clustering-based techniques can improve the query and labelling of nodes efficiency. The clustering-based technique is also used in this proposed scheme to improve the functionalities and reduce the problematic issues to the lowest possible level. The clustering-based technique was used due to the great advantages of this technique. It reduces the required relabelling cases and as a consequence enhances the inserting new nodes processes, with easy determination of the relationships. Thus, all these issues mentioned above were intended to

present into the proposed scheme of this research.The mechanism of the proposed scheme is based on clustering nodes in order to facilitate the determination of the child–parent and sibling relationships. Furthermore, the evaluation of the proposed scheme is based on testing it against the LLS scheme and the Dewey labelling scheme. Subsequently, this proposal was achieved through the implementation of this scheme. Experiments were designed and implemented to test the proposed scheme based on the objectives of this research. The evaluation was also through testing the proposed scheme against the other two schemes used in building this scheme. A data analysis was carried out to determine the main findings.

To conclude, this research developed an XML labelling scheme called Clustering-based Labelling scheme. The aim of this scheme is to improve certain functions that many existing labelling schemes suffer from. Such functions are query processing, update processing, and labelling MXL data. This proposed scheme has been tested and evaluated successfully. Thus, the proposed scheme considered the reduction of the relabelling cases to the lowest possible level. It has also shown an improvement in terms of query performance and inserting new nodes process. Therefore, the aim of this research was achieved, despite some limitations.

REFERENCES

(Book style)

[1] Elmasri, R. and S. Navathe, Fundamentals of database systems. 2007, Boston, Mass: Pearson Addison-Wesley.

[2] Connolly, T.M. and C.E. Begg, Database systems: a practical approach to design, implementation, and management. 2010, Boston, Mass: Addison-Wesley.

(Published Conference Proceedings style)

[3] Cohen, E., H. Kaplan, and T. Milo, Labeling dynamic XML trees. SIAM Journal on Computing, 2010. **39**(5): p. 2048-2074.

[4] Eda, T., et al. Dynamic range labeling for XML trees. in International Conference on Extending Database Technology. 2004. Springer.

[5] O'Neil, P., et al. ORDPATHs: insert-friendly XML node labels. in Proceedings of the 2004 ACM SIGMOD international conference on Management of data. 2004. ACM.

[6] Wu, X., M.L. Lee, and W. Hsu. A prime number labeling scheme for dynamic ordered XML trees. in Data Engineering, 2004. Proceedings. 20th International Conference on. 2004. IEEE.

[7] Tatarinov, I., et al., Storing and querying ordered XML using a relational database system. Proceedings of the ACM SIGMOD International Conference on Management of Data, 2002: p. 204-215.

[8] Li, C., et al. On reducing redundancy and improving efficiency of XML labeling schemes. in Proceedings of the 14th ACM international conference on Information and knowledge management. 2005. ACM.

[9] Liu, J., Z. Ma, and L. Yan. Efficient processing of twig pattern matching in fuzzy XML. in Proceedings of the 18th ACM conference on Information and knowledge management. 2009. ACM.

[10] Xu, L., et al. DDE: from dewey to a fully dynamic XML labeling scheme. in Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. 2009. ACM.

[11] Xu, L., T.W. Ling, and H. Wu, Labeling dynamic XML documents: an order-centric approach. IEEE transactions on knowledge and data engineering, 2012. **24**(1): p. 100-113.

[12] Piwowarski, B., A. Trotman, and M. Lalmas, Sound and complete relevance assessment for XML retrieval. ACM Transactions on Information Systems (TOIS), 2008. **27**(1): p. 1-37.

[13] Edith, C., K. Haim, and M. Tova, LABELING DYNAMIC XML TREES. SIAM Journal on Computing, 2010. **39**(5): p. 2048.

[14] Catania, B., A. Maddalena, and A. Vakali, XML document indexes: a classification. IEEE Internet Computing, 2005. **9**(5): p. 64-71.

[15] Mohammad, S. and P. Martin, XML structural indexes. 2009, Citeseer.

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

[17] Ali Klaib, A., Clustering-based Labelling Scheme-A Hybrid Approach for Efficient Querying and Updating XML Documents. 2018, University of Huddersfield.

[18] Klaib, A.A., A NEW METHOD FOR QUERYING XML DATA, in ISERD 163rd INTERNATIONAL CONFERENCE. 2019: Mecca, Saudi Arabia p. 7.

BIOGRAPHIES

Alhadi A. Klaib is currently an assistant professor in Software Engineering at the Elmergib University, Libya. He received his PhD from Huddersfield University, UK, and received his Master's degree from Sheffield University, UK. He received his bachelor's degree from Elmergib University, Libya. His research interests include, but are not limited to, software engineering, sustainable software engineering, computer networks, XML technology, Data retrieval, and machine learning. Contact him at alhadi.klaib@elmergib.edu.ly.