

New Hybrid Data Model for XML Document Management in Electronic Commerce

Eun-Young Kim and Se-Hak Chun

Abstract XML has been known as a document standard in representation and exchange of data on the Internet, and is also used as a standard language for the search and reuse of scattered documents on the Internet. The issues related to XML are how to model data on effective and efficient management of semi-structured data and how to actually store the model data when implementing a XML contents management system. This paper considers a relational data system to store XML document efficiently and propose a hybrid data model for XML document management.

Keywords XML document management · Relational database · Objective database · Multi-format information retrieval · Electronic commerce

1 Introduction

Extensible Markup Language (XML) is a simple and flexible markup language. As XML has been used for data transaction in EDI from 1998, it supports all kinds of electronic commerce transaction [1]. It provides a method for finding information which users want and XML's abundant data representative method enables users

E.-Y. Kim

Department of Multimedia Contents, Sin Ansan University,
671 Chosi-dong, Ansan City, Kyunggi-do 425-792, Republic of Korea
e-mail: key@sau.ac.kr

S.-H. Chun (✉)

Department of Business Administration, Seoul National University of Science and
Technology, Kongneung-gil 138 Nowon-gu, Seoul 139-743, Republic of Korea
e-mail: shchun@seoultech.ac.kr

to do business on the web by making intellectual mechanism. Therefore, it has received a lot of attention and it has been utilized on almost all fields on the web such as on-line banking, push technology, search engine, web based control system and agent and so on. In addition, application areas are expanding rapidly, and XML documents are made possible to be reused from returned XML documents through the search engine, another existing XML documents, or transmitted XML documents users can draw out data that they want and process their own data structure and store it. Approaches to manage XML documents data can be classified into using new database system for XML documents and using existing database system such as relational database system. Using new database system for XML documents is based on a new data model that represents semi-structured XML data. Also, when users take the existing relational database system or the object system in consideration, mapping model that fits in a relational database system or an object system is required.

Individual data structure was defined according to the XML documents type in the previous researches in data model of XML documents. This poses some problems because the new data structure should be defined for XML documents that have new structures and data; it is difficult to expand the domain and the data structure that needs to be defined in a different way every time according to the object to apply and top-down access is the only available way to search for information. Both the data and structure view of XML's original documents are lost in the data model to map the existing relational database system or object system, so XML documents cannot be generated from the stored data. The range of the application can be expanded easily when the data model for XML documents has a unique data structure regardless of data and structures of XML documents. Also, in terms of searching for information on XML documents, bottom-up access and left-right access should be possible as well as top-down access. The search should also be possible without previous knowledge on the XML document structure.

In this paper, we suggest a data model that supports all these requirements and is applicable to new database system and existing relational database system for the XML document and represents the data and the structure view of the XML document. The paper is organized as follows. In [Sect. 2](#), we describe research motivation. In [Sect. 3](#), we propose the hybrid data model for XML document management. [Section 4](#) concludes this study.

2 Research Motivation

An XML document describes data and structures but it cannot be regarded as a database system simply because it describes data. Even though it contains data, it is just a general text file that cannot perform any function without additional software that manages data. However, once an XML document, a related XML tool and XML's various functions are united, it can be regarded as a database system because it contains storage, schema, query language, programming

interface (factors of database system) and so forth. But, it cannot support effective storage, index, security, transaction, data perfection, multi-user access, trigger, and query on multiple documents that the existing database system supports. Therefore, if the amount of data is not large and the number of users is small and the circumstances require just general capabilities, the XML document can be used as a database, but if there are many users and the circumstances require data perfection and advanced capabilities, the XML document can be used as a database. In this case, the database that stores the XML data and structure is necessary. At present, there are two ways to store data of the XML document; one is to store it as a new data model in the new database system only for the XML document and another is to store the data by mapping the data of the XML document into an existing relational or object system.

The new database system only for the XML document requires a new model that can store, represent and query data of the XML document effectively, while the data model for the existing relational database system or object system requires a data model that coincides with these database system models naturally.

3 New Proposed Hybrid Data Model

This paper proposes a data model that is needed when a new XML-native database system for only XML is designed and is applicable to an existing relational database system. Based on the basic structure of Lore's XML Data Model and Edge Labeled Graph Model [2, 3], proposed hybrid xml data model is renewed and extended to support requirements shown below.

- (1) Document in global domain is the object.
- (2) Data model doesn't depend on the type of database system chosen.
- (3) Both structure and data of the document are represented.
- (4) Structure of the document and data change is applicable flexibly.
- (5) Mapping document from data model reversely, all element orders of document are preserved.
- (6) Top-down, bottom-up, left-right, right-left search about query should be possible.

3.1 Basic Model of XML

It is the most natural to represent the XML document as a graph because the element structure of the XML document can be represented as a tree and mutual reference among elements is added. When representing a basic structure of the XML document data as a graph, each element of the XML document covers a graph that takes root on the node having its own tag name. Therefore, another element included to element is represented as a sub-graph in the element graph

```

<purchaseOrder no="12087998">
  <customer id="key777"/>Kim eunyoung
  <c_PHONE>0314908123</c_PHONE>
</customer>
<orderList>
  <order no="120806">
    <product>m234-2t</product>
    <quantity>3</quantity>
    <price>50000</price>
  </order>
</orderList>
<shipTo>
  <s_NAME cID="key777"/>
  <s_PHONE>01098762345</s_PHONE>
  <s_ADDR1>seoul,mail box 490</s_ADDR1>
  <s_ADDR2></s_ADDR2>
  <s_MSG>safe&quick delivery!</s_MSG>
</shipTo>
</purchaseOrder>

```

Fig. 1 XML document

containing itself. Also, each node is linked by edges and has level or represents reference meanings.

In the basic model, the XML document is represented as a graph that all nodes and edges are labeled and that is ordered among nodes and edges and that edges are directed on. When the XML document is represented as a graph $G = \{V, E, A\}$ where V is a vertex, i.e. set of nodes, E , a set of edges, and A , a set of attributes defined on start tag of element. Figure 1 shows an example of XML document for electronic commerce and Fig. 2 represents it with graph.

An element in the XML document means all contents from the equivalent start tag to finish tag are related to the start tag, and each element of the XML document is represented as a sub-tree/graph that takes the tag of an equivalent element as a root node. The label of node V is a tag value in the case of a middle node and it corresponds with the value (contains NULL and EMPTY) in the case of a leaf node. The label of the edge E_i from node V_i corresponds with relation between node V_i and arriving child node V_j . If child node V_j is another element, it is labeled as CHILD that indicates child element information of node V_i and if the child node V_j is a value of element, it is labeled as VALUE that manifests V_i value information.

3.2 Data Extended Model Considering DTD

DTD means a mutual agreement on the XML document transmitted when XML documents are exchanged. Attributes referring elements in DTD are presented by defining the type of IDREF or IDREFS. Also the value of attributes of IDREF or

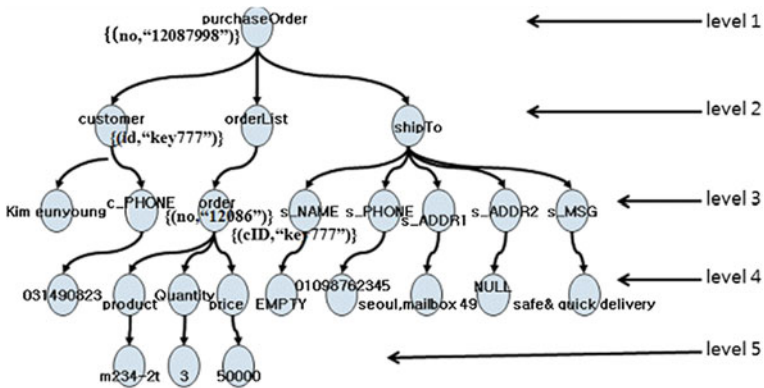


Fig. 2 XML document represented as a graph of the data model

```

<?xml version="1.0" encoding="euc-kr"?>
<!DOCTYPE purchaseOrder [
  <!ELEMENT purchaseOrder(c_INFO,c_orderList,shipTo)>
  <!ATTLIST purchaseOrder no ID #REQUIRED>
  <!ELEMENT customer ANY>
  <!ATTLIST customer id ID #REQUIRED>
  <!ELEMENT c_PHONE (#PCDATA)>
  <!ELEMENT orderList (order)+>
  <!ELEMENT order (product,quantity,price)>
  <!ATTLIST order no ID #REQUIRED>
  <!ELEMENT product (#PCDATA)>
  <!ELEMENT quantity (#PCDATA)>
  <!ELEMENT price (#PCDATA)>
  <!ELEMENT shipInfo (s_NAME,s_PHONE,s_ADDR1,s_ADDR2,s_MSG)>
  <!ELEMENT s_NAME (#PCDATA|EMPTY)>
  <!ATTLIST s_NAME cID IDREF #IMPLIED>
  <!ELEMENT s_TEL (#PCDATA)>
  <!ELEMENT s_PHONE (#PCDATA)>
  <!ELEMENT s_ADDR1 (#PCDATA)>
  <!ELEMENT s_ADDR2 (#PCDATA|EMPTY)>
  <!ELEMENT s_MSG (#PCDATA)>
]

```

Fig. 3 An example of purchase order with IDREF type

IDREFS must be identical to the value of the specified property in ID type of different type. Figure 3 shows an example of purchase order with IDREF type. Figure 4 shows the graph of the XML document with reference information between elements by using DTD.

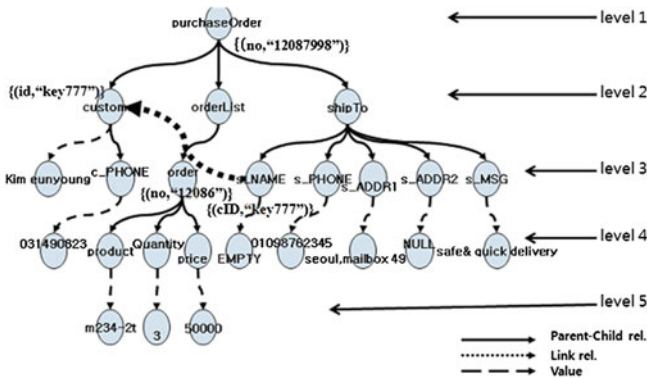


Fig. 4 XML document represented as a graph of the data model with reference information

3.3 Implication

The users see XML documents as a set of one or more objects. These objects can be vertex, edge, attribute, or reference edge objects. Each vertex object has a “label”, which users can look at to perceive the hierarchy structure of XML documents. Each vertex can also contain links (called “Edge Object”) to other attribute object in the system; the relationships between attribute objects can be displayed as connections (called “Reference Edge Object”). Under certain complex conditions, the user can expand and discern a data model for XML documents.

XML has been employed as a critical means for exchange enterprises’ information. The design objectives of XML documents must be clearly and easily understood at the outset and must be defined in terms of the business requirements. For those purposes, the techniques for modeling structures and data of XML documents have been pointed out (e.g., a graph of a data model). The most important reason to build a database of XML documents is to improve the usability of XML data from the user’s perspective.

In spite of this usefulness, the hybrid XML data model has a weakness related to automatic problems. For higher efficiency of the data model, it is necessary to implement the automatic capability for obtaining XML documents in two ways: one is to make an automatic coordination possible with a Computer-Aided Software Engineering (CASE) tool which supports development of the graph of the hybrid XML data model and the other is to make possible automatic storages and reconstruction of XML documents.

4 Conclusion

This paper proposes the new data model for the XML document that is suggested as a document standard to represent the data on the Internet and to exchange data mutually. The hybrid data model in this paper is an applicable model in the case of

designing a new database system for the XML document and in the case of using an existing database system like a relational database system. Because the data view and the structure view of the original XML document are lost in the data model for mapping to the existing relational database system [4–6] or the object-oriented system, not only cannot XML generate again from the stored data, but also XML sub-graph corresponding to element as search result cannot be returned [7].

However, in the proposed model, this problem is solved because the data view and structure view of the original XML document are stored. Therefore, since the proposed model can form a new XML document from the search result of the XML document, it supports reusability as another great feature. The search about the XML document in the proposed model is possible on the condition of all the factors in the XML document, and not only top-down search but also bottom-up search is possible. Besides, as it has the order information between child elements that have same parents, left–right search is possible and the proposed model preserves order of same level as well as hierarchy structure between the elements of the XML document. For a future study, we expect to apply this modeling technique to XML based Request For Proposal (RFP) or Request For Quotation (RFQ) in business to business electronic commerce and other contents management system (CMS).

References

1. Wills, B.: The Business Case for Environmental Sustainability (Green). A 2009 HPS White Paper. (2009). MicroSoft, “XML Scenarios”. <http://msdn.microsoft.com/xml/scenario/inro.asp>
2. Shanmugasundaram, J., Gang, H., Tufte, K., Zhang, C., DeWitt, D.J., Naughton, J.F.: Relational databases for querying XML documents: limitations and opportunities. In: Proceedings of 25th International Conference on Very Large Data Bases, VLDB’99. Edinburgh, Scotland, pp. 302–304 (1999)
3. Florescu, D., Kossman, D.: A performance evaluation of alternative mapping schemes for storing XML data in a relational database. Technical Report 3684, INRIA, March 1999
4. Du, F., Amer-Yahia, S., Freire, J.: ShreX: managing XML documents in relational databases. In: Proceedings of the 30th VLDB Conference, Toronto, Canada (2004)
5. Kappel, G., Kapsammer, E., Retschitzegger, W.: X-ray-towards integrating xml and relational database systems. Technical Report, July 2000
6. Choi, R.H., Wong, R.K.: Efficient date structure for XML keyword search. DASFAA, pp. 549–554 (2009)
7. Atay, M., Chebotko, A., Liu, D., Lu, S., Fotouhi, F.: Efficient schema based XML-to-relational data mapping. *Inf. Syst. (IS)* **32**(3), 458–476 (2007)