International Journal of Fuzzy Logic and Intelligent Systems, vol. 4, no. 2, September 2004 pp. 129-134

A Modeling of XML Document Preserving Object-Oriented Concepts

Chang Suk Kim*, Dae Su Kim, Dong Cheul Son*****

***Dept. of Multimedia Kongju National University, Chungnam, Korea**
****Dept. of Computer Science Hanshin University, Kyunggi, Korea**
*****Dept. of ICE Cheonan University, Chungnam, Korea**

Abstract

XML is the new universal format for structured documents and data on the World Wide Web. As the Web becomes a major means of disseminating and sharing information and as the amount of XML data increases substantially, there are increased needs to manage and design such XML document in a novel yet efficient way. Moreover a demand of XML Schema(W3C XML Schema Spec.) that verifies XML document becomes increasing recently. However, XML Schema has a weak point for design because of its complication despite of various data and abundant expressiveness. Thus, it is difficult to design a complex document reflecting the usability, global and local facility and ability of expansion. This paper shows a simple way of modeling for XML document using a fundamental means for database design, the Entity-Relationship model. The design from the Entity-Relationship model to XML Schema can not be directly on account of discordance between the two models. So we present some algorithms to generate XML Schema from the Entity-Relationship model. The algorithms produce XML Schema codes using a hierarchical view representation. An important objective of this modeling is to preserve XML Schema's object-oriented concepts such as reusability, global and local ability. In addition to, implementation procedure and evaluation of the proposed design method are described.

Key words : XML Schema, Entity-Relationship model, XML document, Reusability, Object-Oriented Concepts

1. Introduction

XML is the new universal format for structured documents and data on the World Wide Web. As the Web becomes a major means of disseminating and sharing information and as the amount of XML data increases substantially, there are increased needs to manage and design such XML document in a novel yet efficient way. XML data or documents have to be modeled using a DTD(Document Type Definitions) which describes document structural constraints. But DTD syntax is different from XML document syntax. So DTD is limited to model and describe a document structure since it does not support XML namespace and various data types. Recently a demand of XML Schema(W3C XML Schema Spec.) that verifies XML document becomes increasing. XML Schema supports various data types which can be represented a complex document. The syntax of XML Schema is the same as XML document against DTD has EBNF type syntax of its own.

However, XML Schema has a weak point for design because of its complication despite of various data and abundant expressiveness. Thus, it is difficult to design a complex document reflecting the usability, global and local facility and ability of expansion[1, 2].

This paper shows a simple way of modeling for XML Schema using a fundamental means for database design, the Entity-Relationship model. The conversion from the Entity-Relationship model to XML Schema can not be directly on account of discordance between the two models. So we present some algorithms to generate XML Schema from the Entity-Relationship model. The algorithms produce XML Schema codes using a hierarchical view representation. An important objective of this automatic generation is to preserve XML Schema's characteristics such as reusability, global and local ability.

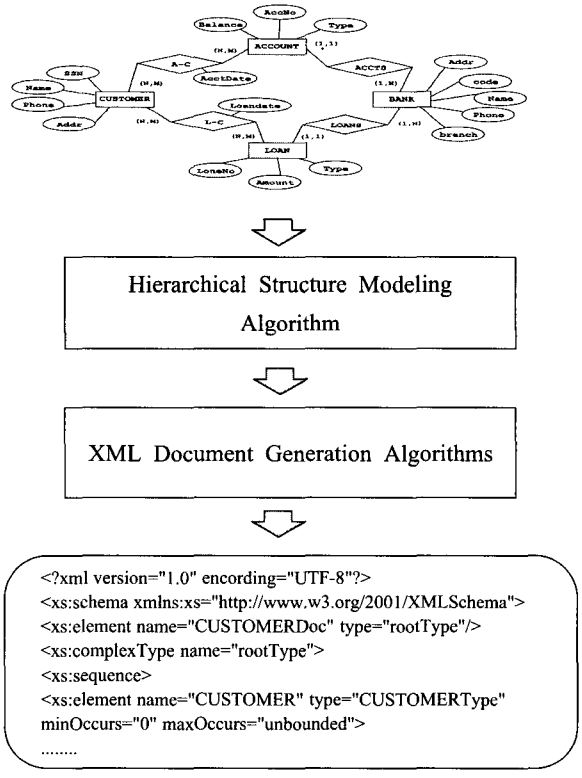

Fig. 1. XML Document Modeling Process

Manuscript received Aug. 23, 2004; revised Sep. 7, 2004.
This work was supported by grant R01-2002-000-00068-0 from the Basic Research Program of the Korea Science and Engineering Foundation in Republic of Korea.

Entity-Relationship model is not accord with XML Schema structure, so the transformation XML Schema from Entity-Relationship model can not be processed directly.

Before to generate XML Schema Entity-Relationship model should be translated hierarchical view. The target XML Schema (xsd file) is generated from the hierarchical view using proposed transformation rule and constraints.

2. Related Works

The researches about generating DTD from Entity-Relationship model are accomplished at UCLA (EXPRESS project), University of Applied Sciences (DB2XML project) and Stanford University (Lore Project) [3]. But it is rare to study generating XML Schema from Entity-Relationship model[4].

Elmasri[5] introduced a design methodology for XML Schema that based on well-understood conceptual model. Elmasri proposed entity migration to transform Entity-Relationship model. So some migrated original entities are disappeared. This cause some problem to preserve XML Schema's characteristics such as reusability and ability of expansion.

Kim [6, 9] proposed the necessity of modeling method to preserving object-oriented properties such as reusability, global and local ability.

3. Transform Hierarchical Structure from E-R Model

A. Transformation Overview

Transformation algorithms about hierarchical view from Entity-Relationship model are described. This algorithm has a characteristic to use reusability of duplicated entities.

Transformation Overview is as follows:

- BFS(Breadth First Search) algorithm searches a root entity
- Transform graph-style E-R representation to hierarchical structure
- Arrange E-R representation

B. BFS Scan

It needs to transform graph-style E-R representation to hierarchical structure. To do this BFS(Breadth First Search) algorithm searches a root entity which is designated by operator(Fig. 2 and Fig. 3).

C. Hierarchical structure model

After to processing BFS scan, we can get a hierarchical structure model as shown Fig. 4. When wescan from ACCOUNT entity to BANK entity, BANK entity is known as duplicated entity. Then the BANK entity generated BANKGroup entity. All the attributes of BANKsuch as branch, phone, name, code, addr are moved into BANKGroup entity(Fig. 5). BANK entity refers BANKGroup entity as shown in Fig. 4.

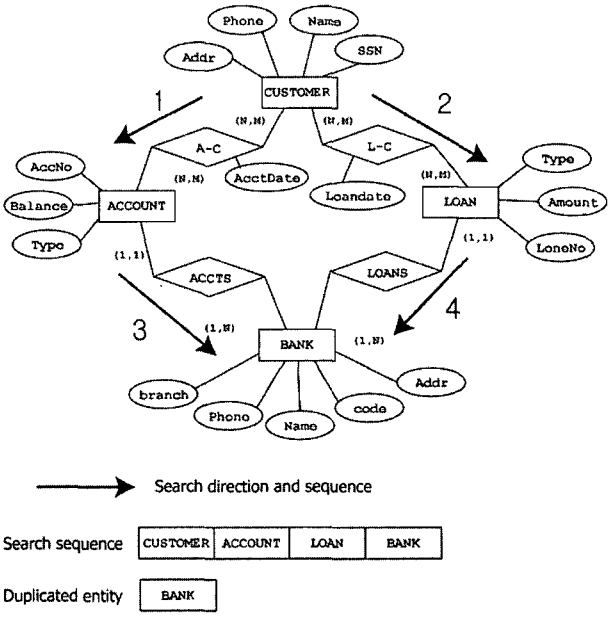

Fig. 2. BFS scan

```
Algorithm BFSScan
Begin search designated root entity
do
{
Search designated root entity
if (duplicated entity){
   Save duplicated entity into duplicated queue
      (duplicateQue);
}else{
Save normal entity into queue (bfsQue);
}
}while(in searching process);
```

Fig. 3. BFS scan algorithm

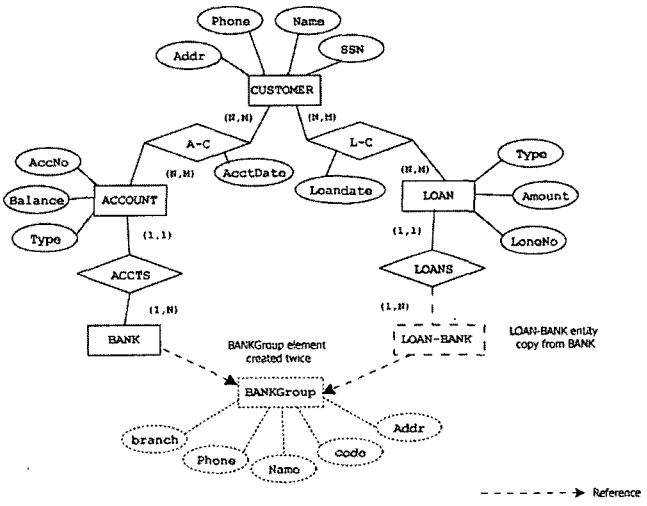

Fig. 4. Hierarchical structure model

```
algorithm GenerateHierView
for(number of entities){
if(child entity of current entity is duplicated){
if(first duplication){
generate new group entity // named as "node-nameGroup"
move to the new generated entity of current child entity
current child entity refer new generated entity
}else{
copy duplicated entity // named as "parent entity-child entity"
duplicated entity refer generated group entity already
}
}
}
```

Fig. 5. Algorithm GenerateHierView

4. Generation XML Schema from Hierarchical Structure

A. Assumptions

This chapter describes XML Schema generation from hierarchical structure model. In general, it ispossible to generate several XML Schemas from the same hierarchical structure model. So we assume some restrictions.

Entity is generated as designated complex type and global style and it is named as 'entity-nameType'.

Attributes are generated as simple type. If current entity has a child entity, it includes the child entity.

If mapping constraints is one-to-one, then the occurrence generated as minOccurs = "1"maxOccurs = "1". Else if mapping constraints is one-to-N, then the occurrence generated as minOccurs = "1" maxOccurs = "unbounded".

If group entity are existed, the group entity should be declared global.

B. XML Schema file and Schema declaration

Following algorithm creates XML Schema file and Schema declaration. All created XML Schema documents have a prefix 'xs:' which differentiate same named entities.

```
Algorithm createXmlSchemaDoc
XML Schema file creation
write("<?xml version="1.0" encording="UTF-8"?>");
write("<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">");
```

Fig. 6. Algorithm createXmlSchemaDoc

Following code is generated by above algorithm create-XmlSchemaDoc.

```
<?xml version="1.0" encording="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
.....
</xs:schema>
```

A. Root entity creation

Root element is defined as designated complex typeand element type is named 'rootType'. The compositor is defined <sequence> and root entity is named 'root-entity-nameType'. Occurrence generated as minOccurs = "1" maxOccurs = "unbounded". In this example, root type's name is CUSTOMER.

B.Group element creation

If hierarchical structure model have group elements, 'group-elementGroup' are generated. The compositor is defined <sequence>, names are defined the same names of attribute and type is generated as 'xs:string' (Fig. 6).

Root element is defined as designated complex type and element type.

```
algorithm createGroup
for(number of group){
write(creation group element 'group-elementGroup');
}
```

Fig. 7. Algorithm createGroup.

```
<xs:group name="BANKGroup">
<xs:sequence>
<xs:element name="Addr" type="xs:string"/>
<xs:element name="code" type="xs:string"/>
<xs:element name="Name" type="xs:string"/>
<xs:element name="Phone" type="xs:string"/>
<xs:element name="Branch" type="xs:string"/>
</xs:sequence>
</xs:group>
```

The resulting codes are generated from Entity-Relationship model in Fig. 2 using proposed algorithms. The CUSTOMER, ACCOUNT,LOAN and BANK elements in Fig. 2 are changed to CUSTOMER, ACCOUNT, LOAN and BANK elements of XML.

```
<?xml version="1.0" encording="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="CUSTOMERDoc" type="rootType"/>
<xs:complexType name="rootType">
<xs:sequence>
<xs:element name="CUSTOMER" type="CUSTOMERType"
minOccurs="0" maxOccurs="unbounded">
</xs:sequence>
</xs:complexType>

<xs:complexType name="CUSTOMERType">
<xs:sequence>
<xs:element name="Ssn" type="xs:string"/>
<xs:element name="Name" type="xs:string"/>
```

```
<xs:element name="Phone" type="xs:string"/>
<xs:element name="Addr" type="xs:string"/>
<xs:element name="ACCOUNT" type="ACCOUNTType"
minOccurs="1"maxOccurs="unbounded"/>
<xs:element name="LOAN" type="LOANType"
minOccurs="1"maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ACCOUNTType">
<xs:sequence>
<xs:element name="Balance" type="xs:string"/>
<xs:element name="AccNo" type="xs:string"/>
<xs:element name="Type" type="xs:string"/>
<xs:element name="AcctDate" type="xs:string"/>
<xs:element name="BANK" type="BANKType"
minOccurs="1"maxOccurs="1"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="LOANType">
<xs:sequence>
<xs:element name="LoanNo" type="xs:string"/>
<xs:element name="Amount" type="xs:string"/>
<xs:element name="Type" type="xs:string"/>
<xs:element name="LoanDate" type="xs:string"/>
<xs:group ref="BANKGroup"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="BANKType">
<xs:sequence>
<xs:group ref="BANKGroup"/>
</xs:sequence>
</xs:complexType>

<xs:group name="BANKGroup">
<xs:sequence>
<xs:element name="Addr" type="xs:string"/>
<xs:element name="code" type="xs:string"/>
<xs:element name="Name" type="xs:string"/>
<xs:element name="Phone" type="xs:string"/>
<xs:element name="Branch" type="xs:string"/>
</xs:sequence>
</xs:group>

</xs:schema>
<xs:element name="Name" type="xs:string"/>
<xs:element name="Phone" type="xs:string"/>
<xs:element name="Branch" type="xs:string"/>
</xs:sequence>
</xs:group>

</xs:schema>
```

5. Implementation and Discussion

The proposed algorithms for transforming hierarchical structure and generating XML Schema are implemented by Java programming language (JDK 1.4.2.). We built some Java classes such as GenenrateXmlSchema, MakeE, MakeR etc. Table 1.

Table 1. Test environment

	Specification
OS	Wondows 2000 Professional
Programming Environments	JDK 1.4.2
CPU	Pentium 1,5 GHz
RAM	512MB

shows implementation and test environment. To model XML schema, we have to model E-R diagram of target documents. The class GenerateXmlSchema receives E-R representation as input, and produces XML Schema sources as output. The class MakeE represents entity and its attributes and the class MakeR expresses relationship and its attributes. The above classes are included in the class GenerateXmlSchema. The Fig. 8 shows that the source codes of the class MakeE, MakeR and GenerateXmlSchema are complied. and were genereated execution file GenerateXmlSchema.

Our approach comparing to previous study has advantages such as reusability and expansion. Fig. 11 show redundant element creation of the existing XML Schema generation method. The bold characters expresses redundant elements of BANK element.

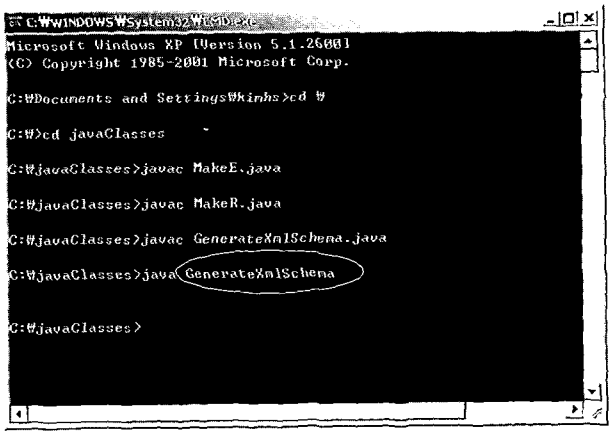

Fig. 8. Execution of GenerationXmlSchema

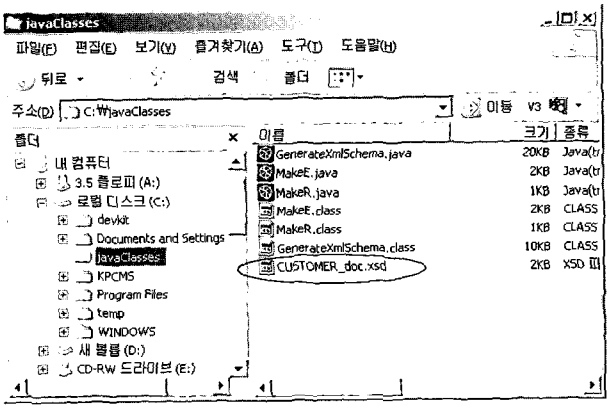

Fig. 9. Genaration of CUSTOMER_doc.xsd

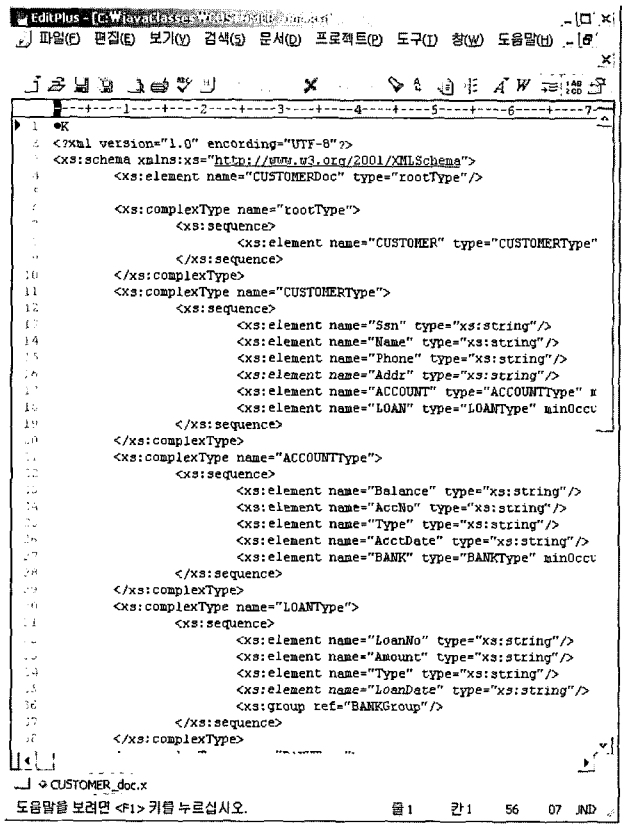

Fig. 10. Content of CUSTOMER_doc.xsd

```
.....
<xs:complexType name="ACCOUNTType">
<xs:sequence>
<xs:element name="BankAddr" type="xs:string"/>
                    <!--elements are duplicated -->
<xs:element name="Bankcode" type="xs:string"/>
<xs:element name="BankName" type="xs:string"/>
<xs:element name="BankPhone" type="xs:string"/>
<xs:element name="Bankbranch" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="LOANType">
<xs:sequence>
<xs:element name="Loandate" type="xs:string"/>
<xs:element name="Amount" type="xs:string"/>
<xs:element name="LoneNo" type="xs:string"/>
<xs:element name="Type" type="xs:string"/>
<xs:element name="BankAddr" type="xs:string"/>
                    <!--elements are duplicated -->
<xs:element name="Bankcode" type="xs:string"/>
<xs:element name="BankName" type="xs:string"/>
<xs:element name="BankPhone" type="xs:string"/>
<xs:element name="Bankbranch" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

Fig. 11. XML Schema code with duplicated elements

6. Concluding Remarks

This paper showed a simple way of design for XML document preserving object-oriented concepts. This design from the Entity-Relationship model to XML Schema can not be directly on account of discordance between the two models. So we presented some algorithms to generate XML Schema from the Entity-Relationship model. The algorithms produce XML Schema codes using a hierarchical view representation. An important objective of this design method is to preserve XML Schema's characteristics such as reusability, global and local ability, ability of expansion and various type changes. We also showed the reusability of the proposed design method compare to existing approach. Finally, implementation and test procedure are described.

References

[1] Jon Duckett, etc., *Professional XML Schemas*, Wrox, 2001.

[2] Kevin Williams, etc., *Professional XML Databases*, Wrox, 2001.

[3] Dongwon Lee, "Schema Conversion Methods between XML and Relational Models", *Knowledge Transformation for the semantic Web*, 2003.

[4] Garsten Kleiner and Udo Lipeck, "Automatic Generation of XML DTDs from Conceptual Database Schemas", *Informatik 2001 - Wirtschaft und Wissenschaft in der Network Economy - Visionen und Wirklichkeit*, 2001.

[5] Ramez Elmasri, "Conceptual Modelingfor Customized XML Schema", *Proceedings of the 21st International Conference on Conceptual Modeling 2002*, page 429-443.

[6] Hyung-Seok Kim, Chang Suk Kim "XML Schema Design Method using EER Diagram", *Proceedings of the KIPS Conference 2003.*

[7] Bo-Jin Hur, Chang Suk Kim, "A Comparative Study on XML schema Conversion Method", *Proceedings of the KIPS Conference 2003.*

[8] Garsten Kleinr an Udo Lipeck, "Automatic Generation of XML DTDs from Conceptual Database Schemas", Informatik 2001 - Wirtschaft und Wissenschaft in der Network Economy - Visionen und Wirklichkeit, 2001.

[9] Chang Suk Kim, "A Generation of XML Schema from E-R Model", *Internation Symposium on Intelligent Systems 2004* (to be published).

Chang Suk Kim

He received the B.S., M.S. and Ph. D. degrees in Computer Engineering from Kyungpook National University, Daegu, Korea, in 1983, 1990 and 1994, respectively. He was a post-doctoral researcher at University of California, San Diego. He worked for ETRI from 1983 to 1994. At present, he is an Associate Professor at the Department of Computer Education, Kongju National University, since 1998. His research interests include intelligent databases, fuzzy theory and XML based information integration.

Phone : 041-850-8822
Fax : 041-850-8165
E-mail : csk@kongju.ac.kr

Dae Su Kim

He received the B. S. degree from Seoul National University, Seoul, Korea in 1977, the M. S. degree in Computer Science from the University of Mississippi, in 1986, and the Ph. D. degree in Computer Science from the University of South Carolina in 1990. He was a researcher at the Intelligent lab. in U. S. A. He worked as a Senior Researcher at the Electronics and Telecommunications Research Institute in Korea from 1991 to 1993. From 1996 to date, he has been a member of the Trustee Board of the Korea Fuzzy Logic and Intelligent Systems Society. He has been an Associate Professor at the Department of Computer Science, Hanshin University, since 1993. His current research interests include Neural Networks, Fuzzy Theory, Artificial Intelligence, Intelligent Systems, Agent Modeling and Evolutionary Computing.

Phone : 032-370-6784
Fax : 032-370-6784
E-mail : daekim@hanshin.ac.kr

Dong Cheul Son

He received the B.S., M.S. degrees in Computer Engineering from Kyungpook National University, Daegu, Korea, in 1983 and 1985, respectively. and Ph.D. degrees in Electrinic Egineering from Chungbuk National University, Chungju, Korea, in 2001. From 1983 to 1998, he worked for ETRI. At present, he is an Associate Professor at the Department of Information and Communication Eng., Cheonan, University, since 2002. His research interests include AI, fuzzy theory and XML based information integration, Intenet engineering.

Phone : +82-41-620-9536
Fax : +82-41-620-9507
E-mail : dcson@cheonan.ac.kr