Journal of Computer Science 3 (4): 212-217, 2007
ISSN 1549-3636

Query Pattern-based Relational Data to XML Data Translation Algorithm

[1]Jinhyung Kim, [2]Dongwon Jeong, [1]Doo-Kwon Baik
[1]Department of Computer Science and Engineering, Korea University
136-713, #225, Asan Building, Anam dong, Sungbuk gu, Seoul, Republic of Korea
[2]Department of Informatics and Statistics, Kunsan National University
573-701, San 68, Miryong dong, Kunsan, Jollabuk do, Republic of Korea

Abstract: This paper proposes a new query pattern-based relational schema-to-XML schema translation (QP-T) algorithm to resolve implicit referential integrity issue. Various translation methods have been introduced on structural aspects and/or semantic aspects. However, most of conventional methods consider only explicit referential integrities specified by relational schema. It causes several problems such as incorrect transformation, abnormal relational model transition and so on. Researches about syntactic/semantic structure extraction also have been executed in reverse-engineering part. Many systems or algorithms suggested in Reverse Engineering researches are so complicate and not proper to RDB-to-XML translation. Some of them just consider syntactic structure extraction and others reflect every structure and constraints. That is, most of methods in reverse engineering part include unnecessary part. The QP-T algorithm analyzes query pattern and extract implicit referential integrities through equi-join between columns. The QP-T algorithm was based on a concept that columns related to equi-join in relational schema can have referential integrity. The most distinct contribution of QP-T algorithm was to enhance extraction of referential integrity relation information for translation. Therefore, the QP-T algorithm reflects not only explicit referential integrities but also implicit referential integrities during RDB-to-XML translation.

Key words: Relational schema model, query-pattern, XML schema model, referential integrity relation information

INTRODUCTION

With XML emerging as the data format of the Internet era, there is a considerable increase in the amount of data encoded in XML[1,2]. However, the majority of data is still stored and maintained in relational database[3]. Therefore, we need to translate such relational data into XML document. In RDB-to-XML translation, there is a problem which is particularly complex when old, ill-designed and poorly documented applications are addressed.

Various translation methods have been developed on structural aspects and/or semantic aspects. Generally, we can classify conventional methods into 3 categories; user-specific translation method, structural translation method and semantic translation method. In case of user-specific translation methods, users must define mapping rules for translation additionally. Typical methods of structural method are a FT (Flat Translation) which maps properties of RDB into XML elements simply and a NeT (Nesting-based Translation) which considers structural relation of RDB. However, FT and NeT cannot reflect referential integrity relation information because they only consider structural part during translation[4]. Representative of semantic algorithm are CoT (Constraints-based Translation) and ConvRel (Relation Conversion to XML nested Structure) which reflect semantic relation such as foreign key constraints[5-7]. The CoT algorithm considers semantic part during translation but it can reflect only referential integrity relation information defined explicitly (RI_{exp}). If implicit referential integrities exist, we cannot guarantee translation accuracy. The ConvRel algorithm cannot execute exact translation if relation information between tables is not defined explicitly. To solve this problem, the implicit referential integrity issue should be considered over the conversion. In this paper, we propose a new RDB-to-XML translation algorithm considering the implicit referential integrity relations. The QP-T algorithm analyzes user query pattern stored in DBMS and extracts implicit referential integrity relations by using equi-join property. By using the QP-T algorithm, we can get better translation accuracy and referential integrity relation information loss ratio than conventional methods.

Backgrounds:

<User-Specific Translation Method> Translation methods need user specifications for RDB-to-XML translation. XML Extender from IBM, XML-DBMS[8], SilkRoute[9], XPERANTO[10] and DB2XML[11] are included in this group. These methods need user specifications about mapping rules for translation. In XML Extender users must define mapping rules by using DAD or XML extender transform language. Template-based mapping language is provided for

Corresponding Author: Doo-Kwon Baik, #225, Asan Building, 1, 5 ga, Anam dong, Subgbuk gu, Seoul, Republic of Korea, Tel: 82-11-1716-3269, Fax: 82-2-921-9137

specification of mapping rule in XML-DBMS. SilkRoute provides declarative query language for description of relational data. XPERANTO uses XML query language for data searching in XML. DB2XML is similar to FT but needs user specification of mapping. Methods of the first group have drawback that user always must provide relation for mapping.

<Structural Translation Method> FT[5] and NeT[5,6] algorithms are included in automatic structural translation methods. The FT algorithm is the simplest method for RDB-to-XML translation and use 1:1 manner. By the FT algorithm, tables in RDB are changed to elements in XML and columns in RDB are changed to attributes in XML. The core idea of the NeT algorithm is to find proper model by using nesting operators such as "*" and "+"[12]. Thus we observe that NeT is useful for decreasing data redundancy and obtaining a "more intuitive" schema by removing redundancies caused by multi-valued dependencies and performing grouping on attributes. However, Demerit of automatic structural translation methods is that these algorithms cannot reflect referential integrity relation during RDB-to-XML translation.

<Semantic translation model> A CoT algorithm[6,7] and the ConvRel algorithm[13] are included in semantic translation methods. The CoT algorithm concerned mostly with the usage of sub-elements and IDREF attribute for translation. In case of two tables (s and t), two columns (α and β, α s, β⊆t) and foreign key constraint {s(α)⊆t(β)}, we can extract semantic information during translation by using translation rules. That is, the CoT algorithm considers not only structural part such as tables and columns but also semantic part such as constraints and referential integrity relation However, the CoT algorithm can only reflect explicit referential integrity relation. If implicit referential integrity relation is exists, the CoT algorithm cannot create exact XML document.

Translation models: Here, we define models to describe translation from a relational schema to a XML schema. Definition 1 shows the model for initial relational database schema.

This relational database schema model is used as the input of the QP-T algorithm. In relational databases, relational schemas are defined with table names, column names, column types and constraints. The constraints include various constraint types (e.q., Unique, Not null, Foreign key, Primary key and so on).

Definition 1 (Initial relational database schema model): A initial relational schema model is denoted by 5-tuple R_{input} = (T, C, P, RI_{exp}, Q_p K), where

T is a finite set of table names
C is a function which represents a set of column names in each table
P is a function which represents properties of each column and the result of P consists of 3-tuples:
T represents data type of column such as integer, string, etc
u represents unique or not of column value by u(unique, ~u(not unique)
n represents nullable or not of values of colunn by n(nullable), !n(not nullable)
RI_{exp} represents explicit referential integrities information
Q_p represents query pattern of users
K is a function which represents primary key information

The output relation schema model is mid-output of QP-T algorithm. This model can be created by adding implicit referential integrity relation information extracted by the QP-T algorithm.

Definition 2 (Output relational schema model): A relational schema model with implicit referential integrity relations are denoted by a 6-tuples R_{output} = (R, C, P, RI_{exp}, RI_{imp-Q}, K), where

RI_{imp-Q} represents implicit referential integrities information extracted by the QP-T algorithm

Translation procedure: Now, we illustrate the QP-T algorithm and translation procedure using the QP-T algorithm. Metadata of relational database includes foreign key constraints information and relational data can be translated into XML data by referring referential integrities relation information. If referential integrity relations are not defined explicitly, we must extract and reflect during RDB-to-XML translation. Therefore, we propose the QP-T algorithm for automatic extraction of implicit referential integrity relation information.

QP-T Algorithm: The QP-T algorithm consists of four steps. The first step is syntactic/semantic error checking step. We check queries received from DBMS by using SQL parsing rules. If there is some error in quires, we can request quires to DBMS again. The second step is new resource generation step. We extract 'where' clause from each query and use these as new resource for query analysis. The third step is query analysis step. We analyze new resources and extract columns related to equi-join. The fourth step is refinement step. Finally, we refine column list extracted at the third step and select implicit referential integrity relation. The overall process of the QP-T algorithm is as Fig. 1.

Translation example: Here, we describe translation procedure from RDB to XML schema model through relational database example. Table 2 shows relational database sample for representation of translation procedure.

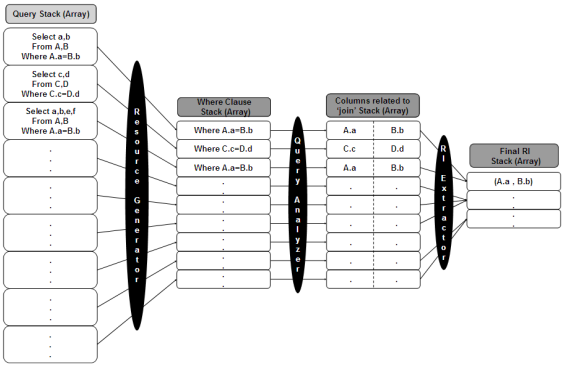

Fig. 1: Overall process of the QP-T algorithm

Table 1. QP-T Algorithm

Input: An array of quires (QueryList). Array of query list is represented as $Q_L[]$. Array of tokenized query list is represented as $Q_T[]$. Array of where clause is represented as $Q_{WH}[]$.
Mid Output: An array of candidate for implicit referential integrities extracted by QP-T algorithm ($RI_{can\text{-}Q}[]$)
Output: An array of implicit referential integrities extracted by QP-T algorithm ($RI_{imp\text{-}Q}[]$)

Procedure:

1. Initialize a=1, b=1, c=1, d=1, g=1, n=1, m=1, j=1
2. Do while $b<b_{(max)}$
3. $Q_L[a]$ = Get QueryList(b)
4. increment a,b
5. For c=1 to $c_{(max)}$
6. For d=1 to $d_{(max)}$
7. $Q_T[c][d]$ = GetToken ($Q_L[c]$, d)
8. Next d
9. Next c
10. For c=1 to $c_{(max)}$
11. For d=1 to $d_{(max)}$
12. if ($Q_T[c][d]$ = 'where')
13. For n=1 to $n_{(max)}$
14. $Q_{WH}[c][n] = Q_T[c][d+n]$
15. Next n
16. Next d
17. Next c
18. For g=1 to $g_{(max)}$
19. For j=1 to $j_{(max)}$
20. if ($Q_{WH}[g][j]$ = '=')
21. For m=1 to $m_{(max)}$
22. $Ri_{can\text{-}Q}[m]$ = ($Q_{WH}[g][j\text{-}3]$. $Q_{WH}[g][j\text{-}1]$,$Q_{WH}[g][j+1]$. $Q_{WH}[g][j+3]$)
23. Next m
24. Next j
25. Next g
26. Initialize k=2, t=1
27. $RI[1] = RI_{can\text{-}Q}[1]$
28. Do while $k<k_{(max)}$
29. For t=1 to $t_{(max)}$
30. if $RI_{imp\text{-}Q}[k] = RI_{can\text{-}Q}[t]$
31. Increment p
32. else $RI[k] = RI_{can\text{-}Q}[t]$
33. End if
34. Increment k
35. Loop

End procedure

This relational database consists of Student (SID, Sname, PID, Cname), Professor (Pname, Office), Class (Cname, Room, Time) and Project (Projname, SID, PID). Each student can take one or more classes and each professor can teach one or more students. The office column of the professor table can null value. Each project is related to one or more students and professors.

Table 2: Relational database sample

< Student >

SID	Sname	PID	Cname
s01	Tom	p01	Database
S02	John	p02	Automata
s03	Cathy	p02	Database
s04	Brown	p03	Simulation
s05	Cabin	p04	Automata
s06	Jorge	P04	Algorithm

< Professor >

PID	Pname	Office
p01	Prof. Kim	#217
p02	Prof. Lee	#613
p03	Prof. Park	#121
p04	Prof. Jean	#222

< Class >

Cname	Room	Time
Database	#701	#2
Automata	#702	#1
Simulation	#703	#2
Algorithm	#704	#3

< Project>

Projname	PID	SID
Data Integration in Sensor Network	p01	s01
Wireless Sensor Network Designation	p02	s02
Ontology System for Data Integration	p03	s01
Integration System based on XML	p01	s03
Simulation Research for Network	p02	s02
e-Government Roadmap Designation	p04	s04
SSL Component implementation	p03	s03

T={Student,Professor,Class,Project}
C(Student)={SID,Sname,PID,Cname}
C(Professor)={PID,Pname,Office}
C(Class)={Cname,Room,Time}
C(Project)={Projname,PID,SID}
K(Student)={SID}
K(Professor)={PID}
K(Class)={Cname}
K(Project)={Projname}
P(SID) = {string,u,!n}
P(Sname) = {string,~u,!n}
P(PID) = {string,~u,!n}
P(Pname) = {string,~u,!n}
P(Office) = {integer,u,n}
P(Cname) = {string,u,!n}
P(Room) = {integer,u,!n}
P(Time) = {integer,~u,n}
P(Projname) = {string,u,!n}
P(PID) = {string,~u,!n}
P(PID) = {string,~u,!n}
RI_{exp} = {(Student.Cname,Class.Cname), (Student.PID, Professor.PID)}
Q_P = ⋈student.SID=Project.SID, ⋈Professor.PID=Project.PID

Fig. 2: Initial relational schema model

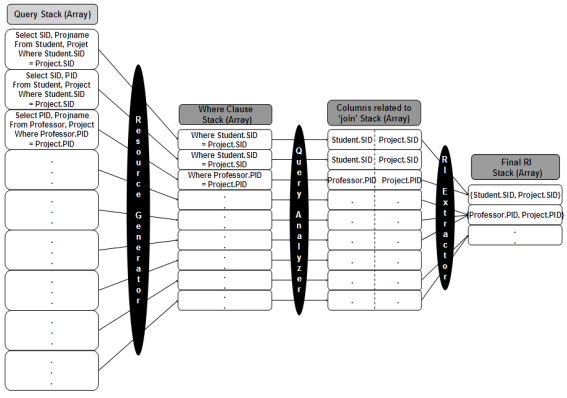

Fig. 3: Extraction procedure of QP-T algorithm

T={Student,Professor,Class,Project}	P(SID) = {string,u,!n}
	P(Sname) = {string,~u,!n}
C(Student)={SID,Sname,PID,Cname}	P(PID) = {string,~u,!n}
C(Professor)={PID,Pname,Office}	P(Pname) = {string,~u,!n}
C(Class)={Cname,Room,Time}	P(Office) = {integer,u,n}
C(Project)={Projname,PID,SID}	P(Cname) = {string,u,!n}
	P(Room) = {integer,u,!n}
K(Student)={SID}	P(Time) = {integer,~u,n}
K(Professor)={PID}	P(Projname) = {string,u,!n}
K(Class)={Cname}	P(PID) = {string,~u,!n}
K(Project)={Projname}	P(PID) = {string,~u,!n}

RI_{exp} = {(Student.Cname,Class.Cname), (Student.PID, Professor.PID)}}
$RI_{imp\text{-}Q}$ = {(Student.SID, Project.SID), (Professor.PID, Project.PID)}

Fig. 4: Output relational schema model

First of all, we translate RDB into the initial relational schema model. Figure 2 shows the initial relational schema model.

Because the initial relational database schema model does not include implicit referential integrity relation information, we must extract implicit referential integrity relation information by QP-T algorithm.

Figure 3 represents extraction procedure of the QP-T algorithm. First, we get query list from the shared SQL area in DBMS and store queries in query stack (array). Second, we extract 'where' clause from queries and store them in where clause stack (array). We create new resource for analysis of user query patter through the second step. Third, we analyzes where clause stack (array) and extract columns related to equi-join. Finally, we select implicit referential integrity relation as pair form from join stack (array). According to general properties of equi-join, if some columns are related to equi-join, those columns have close relationship such as foreign key constraint.

After extraction of implicit referential integrity relation by QP-T algorithm, we can create the output relational schema model by adding implicit referential integrity relation information based on analysis about quires patterns by QP-T algorithm. Figure 4 shows the output relational schema model.

We translate the output relational schema model by referring explicit referential integrity relation information and implicit referential integrity relation information into XML document. We create a XML document by element information of the translation model. The final XML document includes not only explicit referential integrity relation information but also implicit referential integrity relation information. Figure 5 shows the XML document as a final result of translation.

Architecture of QP-T translator: Architecture of QP-T translator is as Fig. 6. The QP-T processor consists of four components. A syntactic & semantic checker parses query received from DBMS and checks whether there is any syntactic or semantic error in queries. A resource generator separates 'where' clause from original quires and creates new resource for analysis of query pattern. A query analyzer analyzes query pattern of queries which have equi-join and extracts column pairs related to equi-join as implicit integrity relation as candidates. A RI extractor refines implicit integrity relation candidates and decides final implicit integrity relation. The QP-T processor receives query list of users from share SQL area in DBMS instance.

The extraction of implicit referential integrity relations by QP-T processor is executed semi-automatically. Generation of new resource, analysis of queries and etc. are performed automatically. However, efforts of designer or expert can be added at the final refinement of implicit referential integrity relation step.

```
<!ELEMENT Student (SID, Sname)>
<!ATTLIST Student ID_Student ID>
<!ATTLIST Student Ref_Class IDREF>
<!ELEMENT Professor (PID, Pname, Office?, Student*,Project*)>
<!ELEMENT Class (Cname, Room, Time)>
<!ATTLIST Class ID_Class ID>
<!ELEMENT Project (Projname)>
<!ATTLIST Project Ref_Student IDREF>
```

Fig. 5: XML document

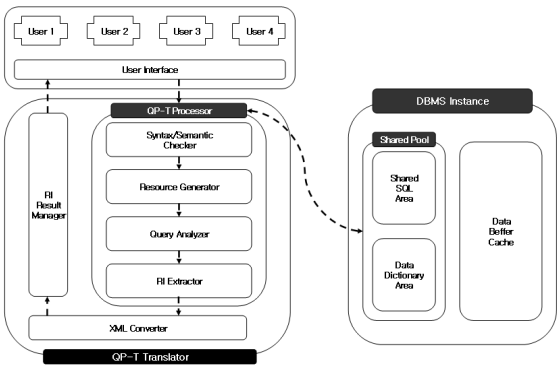

Fig. 6: QP-T translator

We can extract implicit referential integrity relation by the QP-T processor. The QP-T processor analyzes query pattern and extract implicit referential integrity relation based on analysis results.

Comparison evaluation: Here, we compare between the QP-T algorithm and conventional algorithms (FT, NeT, CoT, ConvRel). We execute comparison evaluation through translated XML documents converted by FT, NeT, CoT and ConvRel. Finally, we can extract common properties and difference between the QP-T algorithm and conventional algorithms.

Translated XML documents: The NeT algorithm can remove redundancy by using nesting operators such as

'*', '+'. However, the NeT algorithm does not consider referential integrity relation and the translated XML document by the NeT algorithm cannot reflect semantic information of initial relational database exactly. The CoT algorithm can reflect explicit referential integrity relation information. The translated XML document by the CoT algorithm only considers referential integrity relation information defined at the RI_{exp}. Therefore, the CoT algorithm cannot guarantee referential integrity relation information defined implicitly. The translated XML document by the QP-T algorithm reflects not only explicit referential integrity relation information but also implicit referential integrity relation information. Because we can extract referential integrity relation information defined implicitly by analysis of query patterns, we can reflect all information of initial relational database to XML document during RDB-to-XML translation. Translated XML documents by NeT, CoT and the QP-T algorithm are as in Fig. 7.

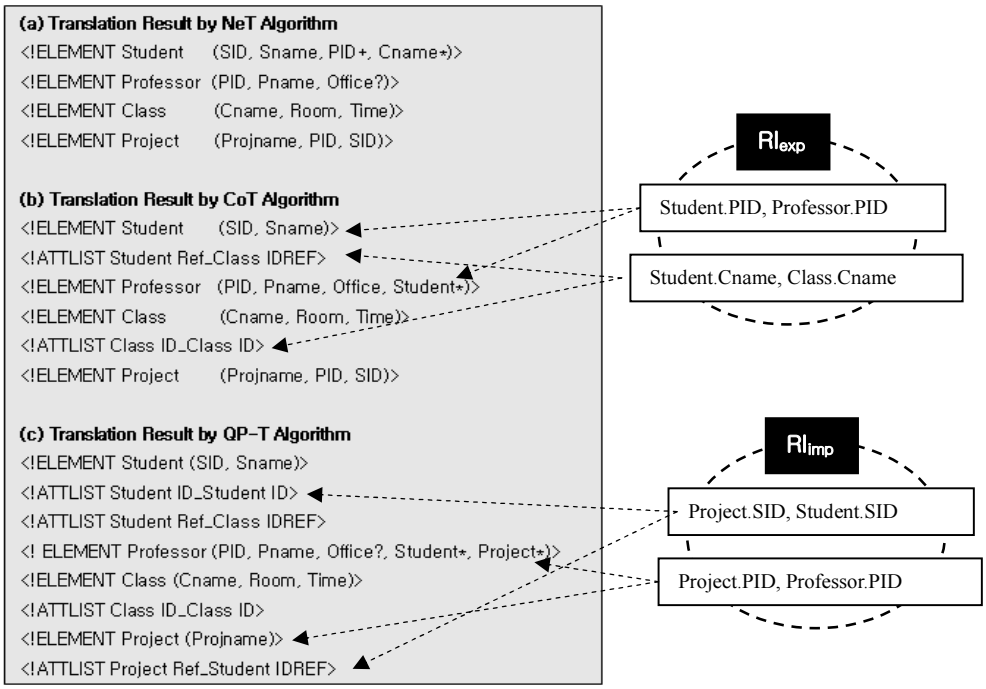

Fig. 7: Translated XML documents by various algorithms

Table 3: Itemize comparison

	FT	NeT	CoT	ConvRel	QP-T
Structural Translation	Partially Support	Fully Support	Fully Support	Fully Support	Fully Support
Explicit RI Extraction (Part of Semantic Translation)	Not support	Not support	Support	Support	Support
Implicit RI Extraction (Part of Semantic Translation)	Not support	Not support	Not support	Not support	Support
Translation Accuracy	Low	Low	Medium	Medium	High
RI Loss Ratio	High	High	Medium	Medium	Low

Table 3 summarizes comparison results between the QP-T algorithm and conventional algorithms. The FT algorithm support structural RDB-to-XML translation partially but FT does not support semantic translation. The NeT algorithm support better structural translation than the FT algorithm but NeT also does not support semantic translation. Therefore, FT and NeT cannot extract implicit referential integrity relation information. In addition to, translation accuracy of FT and NeT is low and referential integrity relation information loss ratio of them is high because they cannot reflect implicit referential integrity relation information. The CoT algorithm and the ConvRel algorithm can support semantic translation partially. Therefore, translation accuracy of CoT and ConvRel is higher than FT or NeT and referential integrity relation information loss ratio of them is lower than FT and NeT. However, RDB-to-XML translation of CoT and ConvRel is not perfect because they only consider explicit referential integrity relation information during RDB-to-XML translation. The QP-T algorithm support not only structural translation but also semantic translation. Because the QP-T algorithm can extract and reflect implicit referential integrity relation information, the QP-T algorithm shows higher translation accuracy and lower referential integrity relation information loss ratio compared to conventional algorithm.

CONCLUSION

In this paper, we have defined translation models and proposed the QP-T algorithm for automatic extraction of implicit referential integrity relation information. We defined the initial relational schema model as input of QP-T algorithm, the output relational schema model as mid-output. As a final result of QP-T algorithm, we can get translated XML document. The QP-T algorithm get user query lists from DBMS through JDBC interface and analyze query pattern and extract implicit referential integrity relation information. By using the QP-T algorithm, we can get more exact XML documents and execute more effective translation. We also can avoid the insertion and deletion errors by using the conventional algorithms.

For future works, we must research the case that there is not enough user query list in DBMS. In this paper, we assume that we can get enough user query lists from DBMS for analysis of query pattern. That is, if user query lists is not enough, the QP-T algorithm cannot extract implicit referential integrity relation information. Therefore, we need to research to solve this problem.

REFERENCES

1. Bray, T., J. Paoli and M. Cavary, 2000. Extensible Markup Language (XML) 1.0, 2000. 2nd Edn. W3C Recommendation.
2. ISO/IEC JTC 1 SC 34, ISO/IEC 8839: 1986. Information processing--Text and office systems--Standard Generalized Markup Language (SGML), 2001.
3. Elmasri., R. and S. Navathe, 2003. Fundamental of Database Systems. 4th Edn., Addison-Wesley.
4. Fernandez, M., W. Tan and D. Suciu, 2000. SilkRoute: Trading between Relations and XML. Intl. World Wide Web Conf. (WWW), Amsterdam, Netherlands.
5. Carey, M., D. Floirescu, Z. Ives, Y. Lu, J. Shanmugasundaram, E. Shekita and S. Subramanian, 2000. XPERANTO: Publishing Object-Relational Data as XML. Intl. Workshop on the Web and Databases (WebDB), Dallas, TX.
6. Goodson, J., 2002. Using XML with Existing Data Access Standards. Enterprise Application Integration Knowledge Base (EAI) J., pp: 43-45.
7. Widom, J., 1999. Data management for XML: Research directions. IEEE Data Engg. Bull., pp: 44-52.
8. Turau, V., 1999. Making legacy data accessible for XML applications. http:// www.informatik.fh wiesbaden.de/~tarau/veroeff.html.
9. Kurt, C. and H. David, 2001. Beginning XML. 2nd Edn., John Wiley & Sons Inc.
10. Jaeschke, G. and H.J. Schek, 1982. Remarks on the Algebra of Non First Normal Form Relations. ACM PODS, Los Angeles, CA, 12: 124-138, Los Angeles.
11. Lee, D., M. Mani, F. Chit and W.W. Chu, 2002. Effective schema conversions between XML and relational models. Eur. Conf. Artificial Intelligence (ECAI), Knowledge Transformation Workshop (ECAI-OT), Lyon, France.
12. Lee, D., M. Mani, F. Chiu and W.W. Chu, 2002. NeT&CoT: Translating relational schemas to XML schemas using semantic constraints. 11th ACM Intl. Conf. Information and Knowledge Management (CIKM). McLean, VA, USA.
13. Duta, A.C., K. Barker and R. Alhajj, 2004. ConvRel: Relationship conversion to XML nested structures. SAC '04, Nicosia, Cyprus.