



UNIVERSITI PUTRA MALAYSIA

**GENERATING NESTED XML DOCUMENT WITH DTD FROM
RELATIONAL VIEWS**

MOHAMMED NASSER AHMED

FSKTM 2008 12



**GENERATING NESTED XML DOCUMENT WITH DTD FROM RELATIONAL
VIEWS**

By

MOHAMMED NASSER AHMED

**DOCTOR OF PHILOSOPHY
UNIVERSITI PUTRA MALYSIA**

2008



**GENERATING NESTED XML DOCUMENTS WITH DTD FROM
RELATIONAL VIEWS**

By

MOHAMMED NASSER AHMED

**Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in
Fulfilment of the Requirements for the Degree of Doctor of Philosophy**

September 2008



*Dedicated to my Parents; Nasser and Mokhlas,
to my wife and
my kids; Zinab, Nada, Aidah , Yasir, and Sarah
to my family.*



Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Doctor of Philosophy

**GENERATING NESTED XML DOCUMENTS WITH DTD FROM
RELATIONAL VIEWS**

By

MOHAMMED NASSER AHMED

September 2008

Chairman : Hamidah Ibrahim, PhD

Faculty : Computer Science and Information Technology

Converting relational database into XML is increasing daily for publishing and exchanging data on the web. Most of the current approaches and tools for generating XML documents from relational database generate flat XML documents that contain data redundancy which leads to produce a massive data on the web. Other approaches assume that the relational database for generating nested XML documents is normalized. In addition, these approaches have problem that lies in the difficult of how to specify the parent elements from the children elements in the nested XML document. Moreover, most of the current approaches and tools do not generate nested XML documents automatically. They require the user to specify the constraints and the schema of the target document.

This research proposes an approach to automatically generate nested XML documents from flat relational database views that are unnormalized. The research aims to reduce



data redundancy and storage sizes for the generated XML documents. The proposed approach consists of three steps. The first step is converting flat relational view into nested relational view. The second is generating DTD from the nested relational view. The third is generating nested XML document from the nested relational view.

The proposed approach is evaluated and compared to other approaches such as NeT, CoT, and Cost-Based and tools such as Allora, Altova, and DbToXml with respect to two measurements: data redundancy and storage size of the document. The first measurement includes several parameters that are number of data values, elements, attributes, and tags.

Based on the results of comparing the proposed approach to several other approaches and tools, the proposed approach is more efficient for reducing data redundancy and storage size of XML documents. It can reduce data redundancy and storage size by approximately 50% and 55%, respectively.



Abstrak tesis dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah

MENJANA DOKUMEN BERSARANG XML DENGAN DTD DARIPADA PANGKALAN

Oleh

MOHAMMAD NASSER AHMED

Jun 2008

Pengerusi : Hamidah Ibrahim, PhD

Fakulti : Sains Komputer dan Teknologi Maklumat

Penukaran pangkalan data hubungan kepada XML meningkat setiap hari untuk penerbitan dan penukaran data dalam web. Kebanyakan pendekatan dan produk semasa untuk menjana dokumen XML daripada pangkalan data hubungan, menjana dokumen XML tidak bersarang yang mengandungi lewahan data yang membawa kepada penghasilan sejumlah data yang banyak di web. Pendekatan lain mengandai bahawa pangkalan data hubungan telah dinormalkan. Tambahan pula, pendekatan ini mempunyai masalah dalam menentukan bagaimana menentukan elemen induk daripada elemen anak? Seterusnya, kebanyakan pendekatan dan produk semasa tidak menjana dokumen XML secara automatik. Mereka memerlukan pengguna untuk menentukan kengkangan dan skema bagi target dokumen.

Penyelidikan ini mencadangkan satu pendekatan untuk menjana secara automatik dokumen XML bersarang daripada gambaran pangkalan data hubungan yang tidak bersarang yang belum dinormalkan. Matlamat penyelidikan adalah untuk mengurangkan lewahan data dan saiz storan bagi dokumen XML yang dijana. Pendekatan yang



dicadangkan ini terdiri daripada tiga langkah. Langkah pertama adalah menukar pangkalan data hubungan yang tidak bersarang kepada gambaran hubungan bersarang. Kedua adalah menjana DTD daripada pangkalan data hubungan bersarang. Ketiga, menjana dokumen bersarang XML daripada pangkalan data hubungan bersarang.

Pendekatan yang dicadangkan dinilai dan dibandingkan dengan pendekatan lain seperti NeT, CoT dan Cost-Based dan alat seperti Allora, Altova dan DbToXml berdasarkan dua ukuran: lewahan data dan saiz storan dokumen. Ukuran pertama melibatkan beberapa parameter iaitu bilangan nilai data, elemen, atribut, dan tag. Berdasarkan kepada keputusan membandingkan pendekatan yang dicadangkan ini dengan beberapa pendekatan yang lain dan alat pendekatan yang dicadangkan lebih efisien bagi mengurangkan lewahan data dan saiz storan bagi dokumen XML. Ia boleh mengurangkan lewahan data dan saiz storan masing-masing hampir 50% dan 55%.

ACKNOWLEDGEMENTS

In the name of ALLAH, the Beneficent, the Compassionate and who gave me strength, patience, and motivation to complete this research work. I would like to express my deep appreciation to my advisor, Associate Professor Dr. Hamidah Ibrahim. She guided me not only on numerous research and technique problems, but also on how to do research. Her profound knowledge, her dedication to research and science, her kindness and optimistic attitude toward the world, will always inspire me. I am also very thankful to my committee members: Associate Professor Dr. Ali Mamat and Associate Professor Dr. Md Nasir Sulaiman continuous help on my research and taking the time to guide me through my thesis. I would like to thank all my colleagues especially Moneer Al-qbatee, Ali alsharafee, Abdurahman A.Mothana, Yahya Al-wadhaf, and Mohammed Radee for helping me. Finally, I am grateful to my family from the bottom of my heart for their unconditional support. Especially, I would like to thank my father, my mother, my wife and my daughters for the love and encouragement.

A great thanks to the Faculty of Computer Science and Information Technology, the university library and Universiti Putra Malaysia that provided the working environment for performing this work.



I certify that an Examination Committee has met on 2007 to conduct the final examination of Mohammed Nasser Ahmed on his Doctor of Philosophy thesis entitles "Generating Nested XML Document with DTD from Relational View" in accordance with Universiti Pertanian Malaysia (Higher Degree) Act 1980 and Universiti Pertanian Malaysia (Higher Degree) Regulations 1981. The Committee recommends that the candidate be awarded the relevant degree. Members of the Examination Committee are as follows:

Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Professor
(External Examiner)

HASANAH MOHD. GHAZALI, PhD
Professor / Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:



This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfillment of the requirement for the degree Doctor of Philosophy. The members of the Supervisory Committee were as follows:

Hamidah Ibrahim, PhD

Associate Professor

Faculty of Computer Science and Information Technology

Universiti Putra Malaysia

(Chairman)

Ali Mamat, PhD

Associate Professor

Faculty of Computer Science and Information Technology

Universiti Putra Malaysia

(Member)

Md. Nasir Sulaiman, PhD

Associate Professor

Faculty of Computer Science and Information Technology

Universiti Putra Malaysia

(Member)

AINI IDERIS, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 13 November 2008



DECLARATION

I declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously and is not concurrently submitted for any other degree at UPM or at any other institution.

MOHAMMED NASSER AHMED

Date:



TABLE OF CONTENTS

	Page
DEDICATION	ii
ABSTRACT	iii
ABSTRAK	v
ACKNOWLEDGEMENTS	vii
APPROVAL	viii
DECLARATION	x
LIST OF TABLES	xv
LIST OF FIGURES	xvii
LIST OF ABBREVIATIONS	xxi
CHAPTER	
1 INTRODUCTION	1
1.1 Overview	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Motivations	4
1.5 Scope of the Research	5
1.5.1 Relational Database Model	6
1.5.2 Relational Database Views	6
1.5.3 Functional and Multi-valued Dependency Constraints	7
1.5.4 Document Type Definition (DTD)	7
1.6 The Contributions of the Thesis	8
1.7 Thesis Organization	9
2 DATABASE AND XML MODELS	11
2.1 Introduction	11
2.2 Relational Database Model	11
2.2.1 Relational Database Constraints	12
2.2.2 Relational Database Views	15
2.2.3 Data Redundancy	21
2.2.4 Database Normalization	21
2.3 XML Model	22
2.3.1 XML and HTML	22
2.3.2 XML Declaration	23
2.3.3 XML Schema Languages	25
2.3.4 Valid and Well-Formed XML Documents	29
2.3.5 Technologies	30
2.3.6 XML Database	33
2.3.7 Motivations of Publishing Relational Database into XML	34
2.4 Summary	35



3	APPROACHES AND TOOLS FOR GENERATING XML DOCUMENTS FROM RELATIONAL DATABASE	36
3.1	Introduction	36
3.2	Model-based Translation	37
3.3	Instance-based translation	41
3.3.1	Flat-based Translation	42
3.3.2	Nesting-based Translation	45
3.3.3	Middleware Language-based Translation	51
3.3.4	Constraints-based Translation	60
3.4	Tools for Generating XML Documents from Relational Database	66
3.4.1	SQLXML	67
3.4.2	DB/XML	68
3.4.3	OracleXML Developer's Kit(XDK)	69
3.4.4	sqlToXML	69
3.4.5	Allora	70
3.4.6	Altova MapForce	71
3.4.7	DbToXML	71
3.5	Reduction of Data Redundancy for XML Documents	72
4.6	Summary	74
4	METHODOLOGY	77
4.1	Introduction	77
4.2	Methodology of the Research	77
4.3	The Proposed Approach for Generating Nested XML Document	79
4.3.1	Generating Nested Relational Views from Flat Relational Views	80
4.3.2	Generating DTD from Nested Relational View	82
4.3.3	Generating Nested XML Document from Nested View	87
4.4	Data Preparation	89
4.5	The Assumption of the Research	90
4.6	Tools for The Proposed Approach	90
4.7	Measurements of the Generating DTD and XML Documents	92
4.7.1	Measurements for Evaluating the Approaches for XML	93
4.7.2	Measurements of the Proposed Approach	95
4.8	Summary	102
5	PROPOSED APPROACH TO GENERATE NESTED XML DOCUMENT	104
5.1	Introduction	104
5.2	The Proposed Approach for Generating Nested XML Document from Flat Relational View	104
5.2.1	Converting Flat Relational View into Nested Relational View	105
5.2.2	Generating DTD from Nested Relational View	112

5.2.3	Generating Nested XML Document from Nested Relational View	119
5.3	Summary	129
6	RESULTS AND DISCUSSION	130
6.1	Introduction	130
6.2	Comparing the Proposed Approach with Tools and Other Approaches	131
6.3	Results for DTD	133
6.4	Experimental Results for XML Documents	135
6.4.1	Data Redundancy	136
6.4.2	Storage Size of XML Document	153
6.5	Summary	166
7	CONCLUSION AND FUTURE WORKS	168
7.1	Introduction	168
7.2	Conclusion	168
7.3	Future Works	170
	BIBLIOGRAPHY	171
	APPENDICES	
	Appendix I: Sample of the results by the proposed approach	177
	Appendix II: Sample of the results by the proposed approach and three tools	189
	BIODATA OF THE STUDENT	215
	LIST OF PUBLICATIONS	216



LIST OF TABLES

Table	Page
2.1 The <i>Employees</i> table	11
2.2 The <i>Course</i> relation	14
2.3 The <i>CourseInstructors</i> relation	14
2.4 The <i>Suppliers</i> relation	17
2.5 The <i>Parts</i> relation	17
2.6 The <i>Capabilities</i> relation	18
2.7 View of <i>Suppliers</i>	18
2.8 View of equi-join of <i>Suppliers</i> on S# with <i>Capabilities</i> on S#	19
3.1 A part of the result of query on <i>Customers</i> and <i>Orders</i> tables	43
3.2 The snapshot history of <i>employees</i>	50
3.3 Summary of approaches for generating XML document	75
4.1 Flat relational view of <i>Courses</i>	81
5.1 A part of flat relational view for <i>Customers</i> view	107
5.2 Frequency of the first column data values	107
5.3 The number of data values for each column in the nested relational view	116
6.1 DTDs lengths by Allora, Altova, DbToXml, and PA in bytes	133
6.2 Storage sizes of DTDs by Allora, Altova, DbToXml, and PA in bytes	134
6.3 PA compared to Allora, Altova, and DbToXml for generating DTD	135
6.4 The number of values and the number of data values for flat XML documents	137
6.5 The number of values and the number of data values for nested XML documents	138
6.6 The number of values for flat and nested XML documents	140
6.7 The number of data values for flat and nested XML documents	141
6.8 The averages of reducing data values by the proposed approach	142
6.9 The number of tags for flat and nested XML documents	144



6.10	The number of elements for flat XML documents	146
6.11	The number of values and the number of elements for nested XML documents	147
6.12	The number of elements for flat and nested XML documents	148
6.13	Percentages of reducing data redundancy by the proposed approach	150
6.14	Percentages of reducing the number of data values	151
6.15	The lengths of XML documents for 100 tuples in bytes	153
6.16	The lengths of XML documents for 200 tuples in bytes	154
6.17	The lengths of XML documents for 400 tuples in bytes	155
6.18	The lengths of XML documents for 800 tuples in bytes	156
6.19	Averages of the lengths for the generated XML documents in bytes	157
6.20	Ratios of the length for the generated XML documents	158
6.21	Storage sizes of XML documents for 100 tuples in kilobyte	160
6.22	Storage sizes of XML documents for 200 tuples in kilobyte	160
6.23	Storage sizes of XML documents for 400 tuples in kilobyte	161
6.24	Storage sizes of XML documents for 800 tuples in kilobyte	162
6.25	The summary results for storage sizes in kilobyte	162
6.26	Ratios of the storage sizes for the generated XML documents	163



LIST OF FIGURES

Figure		Page
2.1	Example of XML document	24
2.2	DTD for <i>Book</i> XML document	27
2.3	XML Schema for <i>Book</i> DTD	28
3.1	A part of XML document for <i>Customers</i> and <i>Orders</i> view	44
3.2	XML DTD of <i>Orders</i> schema using NeT approach	46
3.3	<i>CONSTRUCTOR</i> clause	48
3.4	Example of XML document	49
3.5	Temporally grouped history of <i>employees</i>	50
3.6	The history of <i>employees</i> table as XML document	51
3.7	<i>Customers</i> relational schema	53
3.8	SQL query to construct XML documents	54
3.9	Definition of an XML constructor	54
3.10	An XML document describing a <i>Customer</i>	55
3.11	A fragment of the <i>supplier's</i> public view in Xquery	56
3.12	SQL query	57
3.13	XML template	57
3.14	Tree of the generated XML document for a <i>Customer</i>	59
3.15	The generated SQL query for each element in DTD	60
3.16	Inclusion dependency with one foreign key	61
3.17	Inclusion dependency with two foreign keys	62
3.18	Translating relational schema to XML document	64



4.1	Steps of the research methodology	78
4.2	The steps of generating a nested relational view	80
4.3	Nested relational view of <i>Courses</i>	82
4.4	Regular expressions for DTD	84
4.5	Attributes of \mathcal{SV}_i	84
4.6	DTD for <i>Courses</i>	85
4.7	The steps of generating a nested XML document from a nested relational view	88
4.8	Tags of the <i>name</i> attribute	97
4.9	A part of XML document for <i>Employees</i>	98
4.10	A fragment of the nested XML document for a <i>Student</i>	101
5.1	Flat relational view with blocks	108
5.2	Nested relational view using FD rule	109
5.3	Nested relational view using MVD rule	110
5.4	The algorithm for generating nested relational view using FD rule	111
5.5	The algorithm for generating nested relational view using MVD rule	112
5.6	A sample of nested relational view	115
5.7	Schema of the nested relational view for <i>Customers</i>	116
5.8	Nested elements with their attributes	117



5.9	DTD for the nested relational view of <i>Customers</i>	118
5.10	The algorithm for generating DTD from nested relational view	119
5.11	Tree of the XML document with n fragments	121
5.12	A part of the nested relational view for <i>Customers</i>	122
5.13	Blocks and groups for the nested relational view	123
5.14	Tree of the generated XML document	124
5.15	The first level of the XML fragment	125
5.16	Elements of the second and the third levels	125
5.17	The elements of levels for the XML fragment	126
5.18	Tree of the XML fragment for a <i>Customer</i>	126
5.19	The algorithm for converting nested relational view into nested XML document	127
5.20	A part of the nested XML document for block 1	128
6.1	The framework of the proposed approach	131
6.2	The lengths of DTDs for different views in bytes	134
6.3	Storage sizes of DTDs in bytes	134
6.4	The number of values and the number of data values for flat XML documents	137
6.5	The number of values and the number of data values for nested XML documents	139
6.6	The number of values for flat and nested XML documents	140
6.7	The number of data values for flat and nested XML documents	142



6.9	The number of tags for flat and nested XML documents	145
6.10	The number of elements compared to the number of values for nested XML documents	147
6.11	The number of elements for flat and nested XML documents	149
6.12	The percentages of reducing data redundancy by the proposed approach	150
6.13	The percentages of reducing the number of data values	152
6.14	The lengths of the XML documents for 100 tuples in bytes	154
6.15	The lengths of the XML documents for 200 tuples in bytes	155
6.16	The lengths of the XML documents for 400 tuples in bytes	156
6.17	Averages of the lengths of XML documents in bytes	157
6.18	Ratios of the lengths of XML documents	158
6.19	Percentages of the lengths of the generated XML documents	159
6.20	The storage sizes of XML documents for 100 tuples in kilobyte	160
6.21	The storage sizes of XML documents for 200 tuples in kilobyte	161
6.22	The storage sizes of XML documents for 400 tuples in kilobyte	161
6.23	The storage sizes of XML documents for 800 tuples in kilobyte	162
6.24	Averages of the storage sizes of XML documents in kilobyte	163
6.25	Ratios of the storage size of XML documents	164
6.26	Percentages of reducing the storage size by NeT, CoT, and PA	165



LIST OF ABBREVIATIONS

ATGs	Attribute Translation Grammars
ConvRel	Conversion to XML nested Structure
CoT	Constraints-based Translation
DOM	Data Object Model
DTD	Document Type Definition
FD	Functional Dependency
FRV	Flat Relational View
FT	Flat-based Translation
HTML	Hiper Text Markup Langauge
IND	Inclusion Dependency
MVD	Multi-valued Dependency
NeT	Nesting-based Translation
NPJ	Nest Project Join
NRV	Nested Relational View
ODBC	Open Database Connectivity
PA	The proposed Approach
ROX	Relation Over XML
RXL	Relational to XML transformation Language
SGML	Standard Generlization Markup Language
W3C	World Wide Web Constrium
XML	eXtensible Markup Language
XNF	XML Normal Form
XSD	XML Schema Definition



CHAPTER 1

INTRODUCTION

1.1 Overview

The relational database model and eXtensible Markup Language (XML) model are closely related in most web applications. However, these two models have different structure of schema. The relational database model is based on a two dimensional table that has neither hierarchy nor significant order. XML is based on trees in which order is significant. The hierarchy and sequence features are not used to model information in relational database model but for XML, these features are the main ways to represent information in XML.

XML is useful because of its flexible structure where it closely matches the structure used to display the same information in HTML. Most of data on the web comes from relational database that needs to be converted to XML. Converting relational databases into XML documents includes translating relational schema into XML schemas (DTD or XML Schema) or translating query results (views) with schema into XML documents. Translating query results (views) into XML documents has many problems as follows: view is considered as single relation, non-normalized, and it may contain over millions records that may have duplicates values. If the relational view is converted into a flat XML document, then there will be an amount of useless space because of data



redundancy. To reduce the huge space and data redundancy of XML documents, the view is normalized by the nested relational model. There is a significant advantage in nesting a relation formed by using functional and multi-valued dependencies. The nested relational view can give a great benefit for reducing the data redundancy when it contains at least two groups of attributes from tables that are connected using inclusion dependencies (Lawrence and Ramon, 2005).

1.2 Problem Statement

Although XML has become the prime standard for data exchange on the web, and is increasingly used to represent data currently resides in databases, XML documents have a little semantic and take a large space to store data. Mapping relational database views into XML documents occurs frequently. Thus, there will be massive data on the web because of data redundancy that takes up unnecessary storage, inflate data transfer cost, and lead to update anomalies (Cong and H. Jagadish, 2006). Furthermore, such data redundancies can lead to rendering the database inconsistent (Cong and H. Jagadish, 2008).

Most of the current approaches and tools for generating XML documents from relational database such as Allora (EBIZQ, 2005 and Ronald, 2007), Altova (ALTOVA, 2007 and Ronald, 2007) and DB2XML (Ronald, 2007) generate flat XML documents. The flat XML document contains data redundancy that leads to increase in the cost of storage size. Unlike, Nested XML document has less data redundancy and small storage size. Furthermore, it represents the nature of XML model which is hierarchy. There are some



approaches such as CoT (Dongwon and Murali, 2002), ConvRel (Angela C. et al, 2004), ROX (Alan H. et al, 2004), and Cost-Based (Lawrence and Ramon, 2005) which assume that the relational database tables for generating nested XML documents are normalized with appropriate degree (typically the third degree). However, these approaches required the user to specify the constraints and define the target XML document manually that leads to consume time more than the automatic mapping. Furthermore, the approaches that deal with normalized relational database tables as input to generate nested XML documents have problem to identify and determine which elements are the parents and which elements are the children in the nested XML document. The posed question is how nested XML documents with their schemas can be automatically generated from unnormalized flat relational views with redundancy reduced and less storage size?

1.3 Objectives

The main objective of this research is to propose an automatic approach for generating nested XML documents from relational database views (unnormalized) with less data redundancy and less storage size. There are several sub-objectives derived from the main objective as follows:

1. To propose an approach for converting relational database view into nested relational view.
2. To propose an algorithm for automatically generating nested document type definition (DTD) from nested relational view.

