Automatic Mapping of MPEG-7 Descriptions to Relational Database

Ala'a Al-zoubi[1] and Mohammad Al-zoubi[2]
[1]Department of Computer Information Systems, Irbid National University, Jordan
[2]Department of Computer Graphics, Princess Sumaya University for Technology, Jordan

Abstract: *MPEG-7 is considered an important standard for the description of multimedia content. It is expected that variety of applications based on MPEG-7 media descriptions will be set up in the near future. Basically, MPEG-7 media descriptions are XML documents following media description schemes defined with a variant of XML Schema. Therefore, efficient storage mechanisms for large amount of MPEG-7 descriptions are required. Because of many advantages, the relational DBMS is the best choice for storing such XML documents. However, the existing RDBMS-based XML storage solutions cannot fulfill all the requirements for MPEG-7 descriptions management. In this paper, we present a new automatic approach for mapping MPEG-7 media descriptions to relational database, called RelMPEG-7. RelMPEG-7 automatically generates the relational database tables along with the appropriate columns datatypes and constraints, this is performed by automatic analysis of the XML Schema to extract necessary information from it. This approach is intended to meet almost all the essential requirements to best store the MPEG-7 documents in a relational database, and as a complementary to the existing solutions.*

Keywords: *Mpeg-7, multimedia database, mapping XML to RDB schema.*

Received October 24, 2010; accepted January 3, 2011

1. Introduction

1.1. The MPEG-7 Standard

MPEG-7, formally known as multimedia content description interface, is an ISO/IEC standard developed by the Moving Picture Experts Group (MPEG), following the successful development of the MPEG-1, MPEG-2, and MPEG-4 standards [14]. While the prior standards focus on the coding and the formatting of audiovisual content, MPEG-7 focuses on description and representation of multimedia content. MPEG-7 complements the existing MPEG standard suite and aims to be applicable to many existing formats, which include non-MPEG formats and non-compressed formats as well [19]. The main notion of MPEG-7 is that there are high level and low level descriptors. The low level descriptors are called Descriptors (Ds), they are directly derived from the data and they are instantiated as descriptor values. The high level descriptors are called Description Schemes (DSs). MPEG-7 is not concerned on how the descriptors are extracted, but only on how they are recorded. For this part, the Description Definition Language (DDL) is introduced [13].

The elements that MPEG-7 standardizes will support a broad a range of applications (for example, multimedia digital libraries, broadcast media selection, multimedia editing, home entertainment devices, etc.,). And, in internet multimedia applications, such as video-on-demand, video conferencing, multimedia retrieval services, etc., [1, 10]. This would apply especially to large content archives. Additionally, MPEG-7 descriptions will allow fast and cost-effective usage of the underlying data, by enabling semi-automatic multimedia presentation and editing [19]. According to this diversity of applications and widespread use in a broad spectrum of applications, the amount of MPEG-7 descriptions is increasing dramatically. Therefore, there is a need for adequate means for the management of large numbers of MPEG-7 media descriptions [27]. MPEG-7 descriptions are expressed using the eXtensible Markup Language (XML) [6]. Also, MPEG committee has decided to adopt the XML schema language [11] as the MPEG-7 DDL. Because XML schema language has not been designed specifically for audiovisual content, certain extensions have been proposed as necessary in order for it to meet the MPEG-7 DDL requirements [13]. Since MPEG-7 media descriptions are XML documents which conform to a media description scheme expressed in MPEG-7 DDL, there will certainly be a need for adequate database support for the management of larger numbers of MPEG-7 descriptions [9, 27].

1.2. Database Requirements

To best store and manage the MPEG-7 descriptions (XML documents) in the database, the database solution should fulfill a number of requirements. The analysis made in [27], presented important requirements for the management of MPEG-7 media descriptions and discussed current state-of-the-art database solutions for XML documents to be suitable for the management of MPEG-7 descriptions. MPEG-7 storage solution should satisfy several critical requirements. The first requirement states that the XML database solution should support the use of MPEG-7 DDL (XML Schema) for appropriate management of MPEG-7 media descriptions. This is important for many reasons [8]: First, the information available in the schema definitions is required to ensure the consistency of the database contents by validating whether an XML document is a correct media description with respect to a given media description scheme or not. Second, processing of type information contained in schema definitions is necessary to best represent these types in the DBMS. Finally, XML schema defines the allowable structure of media descriptions.

The second requirement states that suitable representation of description content types (i.e., typed representation) should be presented by the database solution. Basically, much of the information encoded in MPEG-7 media descriptions consists of non-textual data like numbers and rather complex structures like lists [9]. Since MPEG-7 media descriptions are XML documents, and hence, form text documents, these data are encoded as text. The final requirement states that the XML database solution should be efficient in its mapping methodology by means of preserving the structure of the MPEG-7 document, keeping the database size within a reasonable boundary, and the support for complex queries.

1.3. XML Database Solutions

XML database solutions for storing and managing XML documents are categorized by [27] into Native database solutions and database extensions. Native database solutions are DBMSs designed especially for XML documents storage. Traditional Native XML databases stores the XML document as text format. Recent research works on native XML database have proposed the powerful XML storage schemas to support appropriate access to non-textual data. However, according to [9, 27], majority of these solutions are not extensible with index structures to support the index on individual items in the array/matrix datatypes. Also, it is costly to build in-memory trees of very large documents and then query those trees. Furthermore, it is difficult for the Native XML database systems to create a flexible and extensible index structure on the data with various datatypes and query multimedia information across multiple MPEG-7 documents efficiently. More details, analysis, and examples about the native database solutions can be found in [27]. In database extensions, the XML document is stored in relational DBMS. This is due to the various advantages of relational databases such as: widespread usage, proven underlying mathematical theory, query optimization techniques, and advanced processing mechanism.

In [28], the authors categorize how to map and store XML document to RDBMS into two categories: model mapping and structure mapping. In model-mapping, a fixed database schema is used to store the structure and data of any XML document without the assistance of a Document Type Definition (DTD) [6] or XML Schema. The complete structure of an XML document is stored to support complex XPath-based queries and make it easy to reconstruct the data back into XML format. In structure-mapping, the design of the database schema is based on the understanding of XML Schema or DTDs. Majority of the underlying systems define a relation for each element type and uses primary-key and foreign-key to describe the parent-child relationship between the elements in the XML documents. One advantage of this approach is that it supports querying documents efficiently because the mapped data can be indexed easily by built-in database indexes [7].

However, as stated in [7, 8, 9] neither existing structure-mapping approaches nor model-mapping methods can achieve all the MPEG-7 storage requirements due to their intrinsic drawbacks. In this paper, we will present a new automatic approach for mapping MPEG-7 media descriptions to relational database. This approach is intended to meet all mentioned requirements. The remainder of the paper is organized as follows: section 2 presents some related work. In section 3 we present our new mapping approach, followed by analysis and comparison in section 4. Finally, conclusions and future work are given in section 5.

2. Related Work

Some typical examples of model-mapping approach include the edge approach [12], monet [20], Xrel [28], and XParent [15]. The edge approach stores XML data as a directed graph in a table named edge. Monet partitions the Edge table according to all possible label-paths. For each unique path, monet creates a table. Unlike Monet, XRel explicitly keeps all unique path expression as tuples in a table. XRel records elements relationships using the notion of region. The region of an element or text node is a pair of numbers that represents the start and end positions, respectively, of the node in an XML document. XParent is a four table database schema and has been shown to perform better than other model-mapping approaches. Examples of structure-mapping approach can be found in shared and hybrid inlining technique [22], X-Ray [17], a cost-based approach [4], and XML-DBMS [5]. The Inlining technique is an early proposal of structure-mapping approach. However, mapping process with this technique is manual. X-Ray and XML-DBMS define a mapping schema between XML schema and relational schema, and preserve the mapping process autonomy. A fixed mapping is unlikely to work well as different applications may present different access patterns.

In the following subsections, we will discuss the XParent as an example of model-mapping, the XML-DBMS as an example of structure-mapping, and the SM3 [14], which is a hybrid mapping approach of both model-mapping and structure-mapping techniques. In order to illustrate each method with respect to the storage of MPEG-7 descriptions, the MPEG-7 Dominant Color Descriptor (DCD) which is one of the visual color descriptors of MPEG-7 standard [21] listed in Figure 1 is used as an example. (All examples and tables in this section are obtained from [8]).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Mpeg7 xmlns="http://www.mpeg7.org/2001/MPEG-7_Schema"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance">
 <DescriptionUnit xsi:type="DescriptorCollectionType">
  <Descriptor size="5" xsi:type="DominantColorType">
   <ColorSpace type="HSV" colorReferenceFlag="false"/>
   <SpatialCoherency>0</SpatialCoherency>
    <Values>
     <Percentage>2</Percentage>
     <Index>10 6 0</ Index >
    </Values>
    <Values>
     <Percentage>15</Percentage>
     < Index >6 16 9</ Index >
    </Values>
    <Values>
     <Percentage>3</Percentage>
     < Index >7 18 4</ Index >
    </Values>
   </Descriptor>
  </DescriptionUnit>
</Mpeg7>
```

Figure 1. MPEG-7 DCD.

2.1. XParent

XParent [9] is an XML document management system built on top of RDBMS that uses model mapping-based approach without assistant of DTD or XML Schema. XParent uses a fixed relational schema to store the XML documents. This schema consists of four relational tables, which are:

LabelPath(ID, Len, Path)
DataPath(Pid, Cid)
Element(PathID, Did, Ordinal)
Data(PathID, Did, Ordinal, Value)

The *LabelPath* table stores XPath expressions [2] for all elements and attributes in the XML document. Each path is identified by a unique ID stored in the Path column. The number of edges in each XPath expression is stored in the Len column. The *DataPath* table preserves the parent-node id and child-node id of an edge in the data-path by using *Pid* and *Cid* columns. In the Element and Data tables, *PathID* is a foreign key of the ID in the *LabelPath* table. The *Did* is a node identifier which also serves as unique data-path identifier for the data-path ended at the node itself. The value column in the Data table stores all the value content of the XML document. Tables 1-4 show how the MPEG-7 document in Figure 1 will be stored in relational database using XParent.

Table 1. Label path table.

DID	PathID	Ordinal
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1
11	11	1
12	12	1
13	13	1
14	14	1
15	15	1
16	13	2
17	1	1
18	14	1
19	13	3
20	14	1
21	15	1

Table 2. DataPath table.

CID	PID	CID	PID
1	0	15	13
2	1	16	6
3	1	17	16
4	1	18	16
5	4	19	6
6	4	20	19
7	6	21	19
8	6	22	6
9	6	23	22
10	9	24	22
11	9	25	6
12	6	26	25
13	6	27	25
14	13		

Table 3. Element table.

ID	Len	Path
1	1	./Mpeg7
2	2	./Mpeg7./@xmlns
3	2	./Mpeg7./@xmlns:xsi
4	2	./Mpeg7./DescriptionUnit
5	3	./Mpeg7./DescriptionUnit./@xsi:type
6	3	./Mpeg7./DescriptionUnit./Descriptor
7	4	./Mpeg7./DescriptionUnit./Descriptor./@size
8	4	./Mpeg7./DescriptionUnit./Descriptor./@xsi:type
9	4	./Mpeg7./DescriptionUnit./Descriptor./ColorSpace
10	5	./Mpeg7./DescriptionUnit./Descriptor./ColorSpace./
11	5	./Mpeg7./DescriptionUnit./Descriptor./ColorSpace./@type
12	4	./Mpeg7./DescriptionUnit./Descriptor./SpatialCoherency
13	4	./Mpeg7./DescriptionUnit./Descriptor./Values
14	5	./Mpeg7./DescriptionUnit./Descriptor./Values./Percentage
15	5	./Mpeg7./DescriptionUnit./Descriptor./Values./

Table 4. Data table.

Did	PathID	Ordinal	Value
2	2	1	http://www.mpeg7.org/2001/ MPEG-7_Schema
3	3	1	http://www.w3.org/2000/10/ XMLSchema-instance
5	5	1	DescriptorCollectionType
7	7	1	5
8	8	1	DominantColorType
10	10	1	FALSE
11	11	1	HSV
12	12	1	0
14	14	1	2
15	15	1	10 6 0
17	14	1	15
18	15	1	6 16 9
20	14	1	3
21	15	1	7 18 4

2.2. XML-DBMS

XML-DBMS [13] provides an XML-based mapping language that describes both how to construct an object view for an XML document and how to map this view to a relational schema. In XML-DBMS mapping language, element types with element content are usually viewed as classes and mapped to a table. Single-valued attributes and element types with PCDATA (Parsed Character Data) [6] content are viewed as properties and mapped to columns. The relationship of an element type with its parent element type must be stated in the mapping process of the parent class. XML-DBMS creates the following tables for the example in Figure 1.

Mpeg7 (Mpeg7PK, xmlns:xsi, xmlns)
Values (DescriptorFK, ColorValueIndex, Percentage)
Descriptor(DescriptorPK, DescriptionUnitFK, SpatialCoherency, xsi:type, size)
DescriptionUnit (DescriptionUnitPK, Mpeg7FK, xsi:type)
ColorQuantization (ColorQuantizationPK, DescriptorFK)
ColorSpace(DescriptorFK, type, colorReferenceFlag)

2.3. SM3

SM3 [8] depends on the fact that an XML document can be viewed as a tree graph. In XML tree, the internal nodes correspond to the element types with element content in XML document, while the leaf (or evaluated) nodes correspond to the single-valued attributes and element types content in XML document. The idea of SM3 is to use model-mapping approach to map all internal nodes and use structure-mapping approach to map all leaf nodes. SM3 uses the following four table schemas to store internal nodes Tables 5-8.

xpath (xpathid, length, xpathexp)
internalnode (uid, gid, lid, oid, xpathid, nodename, numofchild, tablename, rtablename)
parent (uid, parentid)
ancestor (uid, level, ancestorid)

The Xpath Table 5, stores the XPath information of XML tree in the xpathexp column. The xpathid column represents an identifier for the XPath expression. The number of edges of the XPath expression is stored in the length column. The internalnode table represents the information of each internal node. uid is used to identify the internal node, while gid is the depth-first order of internal node and represents the position of internal node in the XML tree. The tablename column stores the name of the table in which the node is stored. Table 6 depicts the internalnode table that is applicable for the document in Figure 1.

Table 5. XPath table.

Xpathid	Length	Xpathexp
1	1	#/Mpeg7
2	2	#/Mpeg7#/DescriptionUnit
3	3	#/Mpeg7#/DescriptionUnit#/Descriptor
4	4	#/Mpeg7#/DescriptionUnit#/Descriptor#/ColorSpace
5	4	#/Mpeg7#/DescriptionUnit#/Descriptor#/Values

Table 6. InternalNode table.

Uid	Gid	Lid	Oid	Xpa Thid	Node Name	No. of Child	Table Name
1	1	1	1	1	Mpeg7	1	
2	2	1	1	2	Description Unit	1	Descriptor collection
3	3	1	1	3	Descriptor	7	Dominant color
4	4	1	1	4	Color Space	0	Color Space
5	5	3	1	5	Values	2	values
6	6	4	2	5	Values	2	values
7	7	5	3	5	Values	2	values

Table 7. Parent table.

Uid	Parentid
1	0
2	1
3	2
4	3
5	3
6	3
7	3

Table 8. Ancestor table.

Uid	Ancestorid	Level
2	1	1
3	2	1
3	1	2
4	3	1
4	2	2
4	1	3
5	3	1
5	2	2
5	1	3
6	3	1
6	2	2
6	1	3
7	3	1
7	2	2
7	1	3

The parent and ancestor tables maintain parent-child and ancestor-descendant relationship, and maintain parent-child and ancestor-descendant relationship. In order to identify the data type of each evaluated node and map them into database by using structure-mapping approach, a mapping schema is created for representing how to map them into database schema. The mapping schema is defined via Mapping Processing Definition (MPD) file, which is also an XML file. SM3 uses MPD file to map these objects to a relational database. MPD file defines a class for each internal node and indicates which table is used for storing its attributes and children nodes with text content. The evaluated node is usually viewed as property and mapped to column. In MPD file, the corresponding column and appropriate data type is defined for each evaluated node. The database of the leaf nodes in the XML document of Figure 1 is shown in Tables 9-12.

Table 9. Descriptor collection table.

Uid	Xsitype
2	DescriptorCollectionType

Table 10. Dominan color table.

Uid	Size	Xsitype	Spatial Coherency
3	5	DominantColorType	0

Table 11. Color space table.

Uid	Percentage	Colorvalueindex
5	2	10 6 0
6	15	6 16 9
7	3	7 18 4

Table 12. Color space table.

Uid	Type	Colorreferenceflag
4	HSV	False

3. RelMPEG-7 Mapping Approach

RelMPEG-7 is a new especially-designed system for the management of the MPEG-7 descriptions. The main design principle of RelMPEG-7 relies completely on a comprehensive and automatic analysis of the MPEG-7 schema. Basically, RelMPEG-7 is interested in mapping three main components of the XML schema to relational database; Elements and attributes [6], data types [3], and constraints [11]. To explain our method, consider Figure 2 that represents a portion of MPEG-7 DDL for the DCD in Figure 1.

```
1. <complexType name="DominantColorType" final="#all">
2. <element name="ColorSpace" type="mpeg7:ColorSpaceType"
   minOccurs="0"/>
3. <element name="Value" minOccurs="1" maxOccurs="8">
4. <complexType>
5. <sequence>
6. <element name="Percentage" type="mpeg7:unsigned5" />
7. <element name="Index">
8. <simpleType>
9. <restriction>
10. <simpleType>
11. <list itemType="mpeg7:unsigned12" />
12. </simpleType>
13. <length value="3" />
14. </complexType>
15. <complexType name="ColorSpaceType" final="#all">
16. <attribute name="colorReferenceFlag" type="boolean"
    use="optional" default="false"/>
17. <attribute name="type" use="required">
18. <simpleType>
19. <restriction base="string">
20. <enumeration value="RGB"/>
21. <enumeration value="YCbCr"/>
22. <enumeration value="HSV"/>
23. <enumeration value="HMMD"/>
24. <enumeration value="LinearMatrix"/>
25. <enumeration value="Monochrome"/>
26. </simpleType>
27. </attribute>
28. </complexType>
29. <simpleType name="unsigned5">
30. <restriction base="nonNegativeInteger">
31. <minInclusive value="0" />
32. <maxInclusive value="31" />
33. </restriction>
34. </simpleType>
35. <simpleType name="unsigned12">
36. <restriction base="nonNegativeInteger">
37. <minInclusive value="0" />
38. <maxInclusive value="4095" />
39. </restriction>
40. </simpleType>
```

Figure 2. Portion of MPEG-7 DDL for the DCD in Figure 1.

3.1. Mapping Elements and Attributes

The key mapping strategy for elements and attributes of an XML document in RelMPEG-7 is performed in three steps:

- *Step 1:* An XPath expression is generated for every element and attribute defined in the XML schema.
- *Step 2:* From the generated XPath expressions, RelMPEG-7 identifies the items (elements or attributes) that do not have further items.
- *Step 3:* A relational table is created for all items that have the same prior item in the XPath expressions generated in step 1.

For example, consider the schema definition of the *colorSpace* element (lines 2 and 15-28) of Figure 2. Part of the generated XPath expressions [24] will look like:

/Mpeg7/Description/MultimediaContent/Image/VisualDescriptor
/Mpeg7/Description/MultimediaContent/Image/VisualDescriptor/ColorSpace
/Mpeg7/Description/MultimediaContent/Image/VisualDescriptor/ColorSpace/@colorReferenceFlag
/Mpeg7/Description/MultimediaContent/Image/VisualDescriptor/ColorSpace/@type

From this fragment, the *colorReferenceFlag* and *type* do not have further items and both belong to *colorSpace*. Then, the system will generate (automatically) the *colorSpace* table with the corresponding attributes by using the SQL CREATE TABLE statement [25]. This step is explained in Figure 3 (lines 1-4).

3.2. Mapping Data Types

XML Schema has provided three sets of data types [16], these sets are:

1. Built-in primitive data types.
2. Built-in derived data types.
3. Customized derived data types; including the list and matrix data types.

For the first two sets, the RelMPEG-7 selects the nearest data type from the DBMS that is (almost) equivalent to the original data type. But, some types need the SQL CHECK constraints [26] to restrict their values domain. RelMPEG-7 relational tables are built using SQL server 2005 [24, 25, 26]. So, data types are chosen from this DBMS. Table 13 lists some primitive and derived types and their relevant in SQL Server 2005. For example, consider the Schema definition of the *colorReferenceFlag* attribute in Figure 2 (line 16). The original data type for this attribute is *boolean*. Since there is no *boolean* data type in our DBMS, the system selects *varchar(50)* as basic type and assigns the appropriate CHECK constraint. This is illustrated in Figure 3 (lines 3, 11-12).

Table 13. Some primitive and derived types and their relevant in SQL server 2005.

XML Schema Built-in Types	SQL Server 2005 Types	CHECK Constraints
Float	Float	-
Double	Real	-
Decimal	Decimal	-
Binary	Binary	-
QName, string,	Varchar(8000)	-
NMTOKEN, anyURI	Varchar(8000)	-
Integer	int	-
NonPositiveInteger	int	CHECK (someItem <= 0)
NegativeInteger	int	CHECK (someItem < 0)
NonnegativeInteger	int	CHECK (someItem >= 0)
PositiveInteger	int	CHECK (someItem > 0)
Boolean	varchar(50)	CHECK (someItem IN ('true', 'false'))

Customized derived data types (list and matrix) are considered as special cases because there is no data type in SQL Server 2005 (or in the other leading DBMSs) that can directly store this form of values in one single table cell [9]. To solve this problem, the RelMPEG-7 system creates a new table to store the list values in multiple rows in one column. For example, the basic type of Index elements in Figure 2 is list (lines 7-13, 35-40). Each item is of type *nonNegativeInteger*. To handle this type a new table is created that has a column with the appropriate data type and constraints to store each value of the list in a separate row. This is explained in Figure 3 (lines 25-35). For matrix datatype, the same methodology will be followed except that there will be more than one column in the generated table depending on the width of that matrix.

3.3. Mapping Constraints

XML Schema also defines data integrity constraints to insure the valid values for media contents [11]. In order to map these constraints using RelMPEG-7, we classified these constraints into two categories: value (domain) constraints and occurrence constraints. The mapping technique will run as follows:

1. *Value Constraints:* Are used to enforce domain integrity of data contents of the MPEG-7 documents. By restricting the values allowed in a column to specific values. In The MPEG-7 DDL, this is done by using *facets* [20]. Within the scope of this paper, five facets are addressed: *minInclusive, maxInclusive, minExclusive, maxExclusive* and *Enumeration*. In our system, we transformed (mapped) these facets and their functionality into convenience SQL CHECK constraints. Consider the facets definitions of the unsigned 5 type in Figure 2 (lines 29 and 32), this facet is mapped to a SQL CHECK constraint as shown in Figure 3 (lines 20-24).
2. *Occurrence Constraints:* Determine the minimum number of times an element may appear in an XML document. It also, specifies whether an attribute may or may not appear. In our system, we have used these constraints in defining NULL and NOT NULL constrains in SQL [23, 26] to allow or prevent null values from being inserted into a specific column.

For elements, occurrences are specified by the *minOccurs* attributes declared in the Schema. If the *minOccurs* value is 0, this means that this element may or may not appear in the XML document [11], and hence, the column that will hold the value of that element is allowed to store NULL values.

If the *minOccurs* value is 1 or more, this means that this element is required to appear in the XML document. And hence, the column that will hold the value of that element is not allowed to store NULL values. On the other hand, attributes have different syntax in specifying their occurrences in an XML document. In particular, attributes are declared with a *use* attribute to indicate whether the attribute may or may not have null values by using *optional (use="optional")* or *required (use="required")*, respectively [11].

```
1.  Create Table [ColorSpace]
2.  (
3.  [colorReferenceFlag] varchar(50) ,
4.  [type] varchar(8000) Not Null,
5.  [PID] int   References XPath(Pathid),
6.  [DID] int   References Document(DID),
7.  Order int   Not Null,
8.  Constraint Enumeration1
9.  Check ( type In (  'RGB' , 'YCbCr' , 'HSV' ,
10. 'HMMD' , 'LinearMatrix' , 'Monochrome'  ) ),
11. Constraint Boolean1
12. Check ( colorReferenceFlag In ('true', 'false') )
13. )
14. Create Table [Value]
15. (
16. [Percentage] int  Not Null ,
17. [PID] int   References XPath (Pathid),
18. [DID] int   References Document(DID),
19. ORDER int Not Null,
20. Constraint MinVal3
21. Check ( Percentage >= 0 ),
22. Constraint MaxVal3
23. Check( Percentage <= 31 )
24. )
25. Create Table [Index]
26. (
27. [Index1] int  Not Null ,
28. [PID] int   References XPath(PATHID),
29. [DID]int    References Document(DID),
30. Order int   Not Null,
31. Constraint MinVal4
32. Check (Index1>= 0 ),
33. Constraint MaxVal4
34. Check(Index1<= 4095 )
35. )
```

Figure 3. The generated SQL CREATE TABLE statements from the XML schema portion in Figure 2.

3.4. Preserving the Structure of the MPEG-7 Document

Although data in XML documents has been stored in relational database, users or applications still view it as XML to issue queries against XML data [8].

Furthermore, the general structure of the original XML document must be preserved to reconstruct data stored in the relational database back to the XML document format [27]. RelMPEG-7 supports this requirement by automatically generating two extra tables. The first table is called Document table, and since no element or attribute in the XML document can uniquely identify that document, RelMPEG-7 generates this table to hold an identifier for that document in the *DID* column. Table 14 shows the document table for a single document. The second generated table is called the Path table. This table consists of three columns; *XPath, PathID,* and *TableName*. The *XPath* column stores the *XPath* expressions of the direct parent items that have no further items, each expression is uniquely identified by a *PathID.* The third column is *TableName* column, which specifies the table in which the values are stored. For example, Table 15 stores XPath expressions for the XML Schema portion in Figure 2. (note: there will be extra XPath expressions for the complete schema).

Table 14. Document table.

DID
1

Table 15. Path table.

Pathid	Xpath	Tablename
1	/Mpeg7/Description/MultimediaContent/ Image/VisualDescriptor/Value	Value
2	/Mpeg7/Description/MultimediaContent/ Image/VisualDescriptor/ColorSpace	ColorSpace
3	/Mpeg7/Description/MultimediaContent/ Image/VisualDescriptor/Value	Index

Finally, RelMPEG-7 adds three extra columns to the tables that store the values of the MPEG-7 document. These columns are used to connect the generated tables via Primary key–Foreign Key relationship; the first column is *DID*, which is used as a foreign key to reference the Document table. The second column is *PID*, which is used as a foreign key to reference the Path table. The third column is the order column. This column is used to identify the order in which the elements and attributes appear in the XML document. After implementing the mapping steps illustrated earlier, the XML document of Figure 1 (that corresponds to the schema portion in Figure 2) will be stored in relational tables similar to Tables 16 through 18 below.

Table 16. ColorSpace table.

ColorReferenceflag	Type	Order	DID	PID
false	RGB	1	1	2

Table 17. Value table.

Percentage	Order	DID	PID
6	1	1	1
4	2	1	1
4	3	1	1

Table 18. Index table.

Index1	DID	PID	Order
10	1	3	1
6	1	3	1
0	1	3	1
6	1	3	2
16	1	3	2
9	1	3	2
7	1	3	3
18	1	3	3
4	1	3	3

4. Analysis and Comparison

To measure the extent to which RelMPEG-7 is efficient in fulfilling the requirements for mapping and storing MPEG-7 documents in relational database, in this section, we analyze and compare the RelMPEG-7 with some of the related work of model-mapping and structure-mapping approaches discussed earlier. The comparisons are based on three basic assessment metrics in section 1.3.

4.1. RelMPEG-7 vs. XParent

XParent and the other model-mapping approaches as in [28, 12, 20] have some deficiencies that can affect negatively in storing and managing the MPEG-7 documents. First, these approaches are not built on the basis of XML schema. In this case, the validation of XML documents is not considered. So, inconsistent or invalid data can be imported during the insert or update operations. Second, XParent stores all values of XML documents as text. Large portions of the information contained in MPEG-7 media descriptions typically consist of non-textual data. Storing these non-textual values as a string will be cumbersome to handle and more complicated to manipulate [27]. For example, in DCD, the value of the Index element, which has list data type, would be something like “255 173 37”. XParent would store this data as one string value in a single column. Such storage format makes the query inefficient when the users want to issue the query on each item in the list value. The application (or user) needs first to split these values and assign each number to a numeric variable, and then the query will be processed. So, the total number of conversions necessary during the processing of media descriptions might increase a performance overhead [8].

RelMPEG-7 solved these problems by depending completely on the MPEG-7 DDL (XML Schema). So, MPEG-7 documents validated to check their correctness and integrity. Also, the data types are well represented in the RDBMS as explained earlier.

4.2. RelMPEG-7 vs. XML-DBMS

The major disadvantage of XML-DBMS approach is that it does not keep information on the complete structure of the original XML documents. This drawback makes it impossible to support complex XPath-based query and reconstruct data from DBMS into the original XML document. Such reconstruction function is very important for MPEG-7 applications since MPEG-7 documents are primarily for multimedia information exchange between systems. Another limitation of XML-DBMS comes from its use of DTD. DTDs have weak support for different data types since it was originally designed for use with text [17, 23]. This means that typed presentation of media content is not supported in XML-DBMS. The other structure-mapping techniques discussed in section 2 use fixed mapping as in [22], which is unlikely to work well as different applications may present different access patterns. Or the mapping process performed manually as in [17].

In RelMPEG-7, the complete structure is stored within the relational tables that guarantee reconstructing the original XML document. Also, RelMPEG-7 depends on the XML schema in specifying suitable data types (and data structures) to store the values of MPEG-7 descriptions.

4.3. RelMPEG-7 vs. SM3

SM3 and its modified versions [7, 9] present an interesting mapping approach for MPEG-7 descriptions. However, it also has some shortcomings.

First, SM3 depends on model-mapping in defining the structure and XPath expressions of the XML document. So, this step in performed without the assistance of a DTD or XML schema [28]. The problem comes from the fact that the allowable structure of the MPEG-7 document may vary from a document to another, i.e., some elements and attributes may or may not appear in the XML document (according to the occurrence constraints in schema). Therefore, some elements or attributes will not be stored. For instance, the following portion of the DCD may appear in some MPEG-7 Descriptions:

```
<Mpeg7>
  <Image>
   <MediaLocator>
    <MediaUri>
     E:\Corel_1k\38.jpg
    </MediaUri>
   </MediaLocator>
   <TextAnnotation>
    <FreeTextAnnotation>nice image</FreeTextAnnotation>
   </TextAnnotation>
  <Descriptor size="5" xsi:type="DominantColorType">
   <ColorSpace type="RGB" colorReferenceFlag="false"/>
   <Value>
    <Percentage>9</Percentage>
    <Index>0 0 0</Index>
    <ColorVariance>1 1 1</ ColorVariance >
   </Value>
</Mpeg7>
```

According to the relational schema of the SM3 illustrated above, *MediaUri*, *FreeTextAnnotation*, and *ColorVariance* elements have no columns to store their values because these items were not present in the document the SM3 depended on in creating the tables and generating the XPath expressions. The second disadvantage of SM3 is that datatypes mapping is defined via MPD file, which is also an XML file. MPD file provides a map of any XML data that is to be stored in the database. However, this MPD file is created *manually* [9]. In contrast, RelMPEG-7 depends on the XML schema to create the relational tables and to generate the XPath expressions to store data of the XML documents. So, all the variations of the XML document that are valid against one Schema can be stored effectively in the relational database. Also, the datatypes are mapped automatically, so, there is no need for manual steps in mapping datatypes.

5. Conclusions and Future Work

In this paper, we presented RelMPEG-7; a new approach for mapping, storing, and querying MPEG-7 media descriptions in relational database. RelMPEG-7 is intended to meet all critical requirements for storing XML documents in relational database, which are: support the use of XML Schema, present suitable representation for content datatypes, and present efficient mapping mechanism by means of preserving the structure of the document. Comparisons with previous work have shown that RelMPEG-7 has fulfilled these essential requirements. Measuring the efficiency of our work according to the database size and the mapping performance will be accomplished in future work. Furthermore, we plan to use Semantic Web technologies to semantically enquiry the descriptions, and hence, overcome the semantic gap of the MPEG-7 low-level and high level features.

References

[1] Al-Hamami A. and Al-Rashdan H., "Improving the Effectiveness of the Color Coherence Vector," *The International Arab Journal of Information Technology,* vol. 7, no. 3, pp. 324-332, 2010.

[2] Berglund A., Boag S., Chamberlin D., Fernández M., Kay M., Robie J., and Siméon J., "XML Path Language (XPath) 2.0," *W3C Recommendation*, available at: http://www.w3.org/TR/2007/REC-xpath20-20070123/, last visited 2007.

[3] Biron P. and Mahotra A., "XML Schema Part 2: Datatypes," *W3C Recommendation*, available at: http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/, last visited 2001.

[4] Bohannon P., Freire J., Roy P., and Siméon J., "From XML Schema to Relations:A Cost-Based Approach to XML Storage," *in Proceedings of the* 18^{th} *International Conference on Data Engineering*, USA, pp. 64-75, 2002.

[5] Bourret R., Bornhövd C., and Buchmann A., "A Generic Load/Extract Utility for Data Transfer

Between XML Documents and Relational Databases," *in Proceedings of 2nd International Workshop on Advanced Issues of EC and Web-Based Information Systems*, California, pp. 134-143, 2000.

[6] Bray T., Paoli J., Sperberg-McQueen C., Maler E., and Yergeau F., "Extensible Markup Language (XML) 1.0 (Fifth Edition)," *W3C Recommendation*, available at: http://www.w3.org/TR/REC-xml/, last visited 2008.

[7] Chu Y., Chia L., and Sourav S., "SM3+: An XML Database Solution for the Management of MPEG-7 Descriptions," *in Proceedings of the 16th International Conference on Database and Expert Systems Applications*, pp. 134-144, 2005.

[8] Chu Y., Chia L., and Sourav S., "Looking at Mapping, Indexing and Querying MPEG-7 Descriptors with RDBMS with SM3," *in Proceedings of the 2nd ACM International Workshop on Multimedia Database*, USA, pp. 55-64, 2004.

[9] Chu Y., Chia L., and Bhowmick S., "Mapping, Indexing and Querying of MPEG-7 Descriptors in RDBMS with IXMDB," *Journal Data and Knowledge Engineering*, vol. 63, no. 2, pp. 224-225, 2007.

[10] Doller M., "MPEG-7 Meets Multimedia Database Systems," *Journal of Universal Knowledge Management*, vol. 1, no. 1, pp. 18-25, 2006.

[11] Fallside D. and Walmsley P., "XML Schema Part 0: Primer (Second Edition)," *Technical Document*, *W3C Recommendation*, pp. 1-89, 2004.

[12] Florescu D. and Kossmann D., "A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database," *IEEE Data Engineering Bulletin*, vol. 22, no. 3, 1999.

[13] Hunter J., "An Overview of the MPEG-7 Description Definition Language (DDL)," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 6, no. 11, pp. 765-772, 2001.

[14] ISO/IECJTC1/SC29/WG11/N3751, *Introduction to MPEG-7*, (V 2.0), 2000.

[15] Jiang H., Lu H., Wang W., and Xu J., "XParent: An Efficient RDBMS-Based XML Database System," *in Proceedings of the 18th International Conference on Data Engineering*, USA, pp. 335, 2002.

[16] Kappel G., Kapsammer E., and Retschitzegger W., "XML and Relational Database Systems - a Comparison of Concepts," *in Proceedings of the 2nd International Conference on Internet Computing*, USA, pp. 1-7, 2001.

[17] Kappel G., Kapsammer E., and Retschitzegger W., "X-Ray-Towards Integrating XML and Relational Database Systems," *in Proceedings of Conference on Conceptual Modelling*, *Lecture Notes in Computer Science*, Berlin, pp. 339-353, 2000.

[18] Kosek J. and Nalevka P., "Relaxed-on the Way Towards True Validation of Compound Documents," *in Proceedings of 15th International World Wide Web Conference*, USA, pp. 427-436, 2006.

[19] Manjunath B., Salembier P., and Sikora T., *Introduction to MPEG-7: Multimedia Content Description Standard*, Wiley, 2001.

[20] Schmidt A., Kersten M., Windhouwer M., "Efficient Relational Storage and Retrieval of XML Documents," *in Proceedings of the 3ed International Workshop on the Web and Databases*, Texas, pp. 47-52, 2000.

[21] Shanmugasundaram J., Tufte K., He G., Zhang C., DeWitt D., and Naughton J., "Relational Databases for Querying XML Documents: Limitations and Opportunities," *in Proceedings of the 25th VLDB Conference*, Scotland, pp. 302-314, 1999.

[22] Sikora T., "The MPEG-7 Visual Standard for Content Description-An Overview," *IEEE Transaction Circuits System Video Technology*, vol. 11, no. 6, pp. 696-702, 2001.

[23] Terzi E., Vakali A., Fan J., and Hacid M., "The MPEG-7 Multimedia Content Description Standard and the XML Schema Language," *in Proceedings of the 7th International Conference on Distributed Multimedia Systems*, pp. 1-8, 2001.

[24] The Microsoft Developer Network (MSDN), available at: http://msdn.microsoft.com/en-us/library/ms174979.aspx, last visited 2012.

[25] The Microsoft Developer Network (MSDN), available at: http://msdn.microsoft.com/en-us/library/ms187752.aspx, last visited 2012.

[26] The Microsoft Developer Network (MSDN), available at: http://msdn.microsoft.com/en-us/library/ms188258.aspx, last visited 2008.

[27] Westermann U. and Klas W., "An Analysis of XML Database Solutions for the Management of MPEG-7 Media Descriptions," *ACM Computing Surveys*, vol. 35, no. 4, pp. 331-373, 2003.

[28] Yoshikawa M. and Amagasa T., "XRel: Apath-Based Approach to Storage and Retrieval of XML Documents using Relational Databases," *ACM Transactions on Internet Technology*, vol. 1, no. 1, pp. 110-141, 2001.

Ala'a Al-zoubi received his BSc in computer information systems from Jordan University of Science and Technology in 2004, and MSc in computer science from Jordan University of Science and Technology in 2007. Currently, he is working as a lecturer at Irbid National University. His research interests include multimedia databases and data mining.

Mohammed Al-zoubi received his BSc and MSc in electrical engineering from Jordan University of Science and Technology in 1990 and 1994, respectively. In 2002, he received his PhD in computer science from University Science Malaysia, specializing in multimedia systems. He joined Princess Sumaya University for Technology in 2003 and was chairman of the Computer Graphics Department for the period 2009-2010. He has several publications in cloud computing, e-learning, and image coding.