XQuery adaptation for multimodal retrieval of multimedia documents

Mohamed KHARRAT, Anis JEDIDI, Faiez GARGOURI
Multimedia, InfoRmation systems and Advanced Computing Laboratory (MIR@CL)
University of Sfax ISIMS
Bp 242, Cp 3021, Sfax
Tunisia
{med_khr@yahoo.fr, anis.jedidi@isimsf.rnu.tn, faiez.gargouri@fsegs.rnu.tn}

ABSTRACT: *Recent years witness a phenomenal growth of multimedia data in various modalities, such as image, video, audio, and graphic, which poses a challenge of finding an efficient information retrieval technology. Rather than monotonous, single-modal information, users would like to have a multimodal system to query multimedia documents. In this paper, we present our propositions in two parts, the first part consist on a conceptual model to define semantic relations between multimedia documents. The second part defines an extension of XQuery language to support multimodal querying.*

Keywords: Component; XQuery, Multimedia, Multimodal, Retrieval, XML

Received: 28 January 2011, Revised 1 March 2011, Accepted 7 March 2011

1. Introduction

Advanced web-based applications require dealing with diverse types of multimedia documents which are available in digital format. However, extraction of useful information from multimedia data and manipulation of these information in practical systems are still open problems.

The multimedia challenge querying, consist on extending Databases and resources by homogeneous description and afterwards integrate them.

XML is rapidly accepted as a standard data exchange format on the web, and becoming the emerging standard to describe multimedia content, due to its flexibility and simplicity. The main deficiency of XML for multimedia data is the lack of managing relationships among objects.

In order to allow an efficient exploitation of the existing huge collections of multimedia documents, it is necessary to design tools to manage access to these documents. Especially the semantics of medias which are often derived from the interaction between resources in the same collection.

The object of this work is to help the users to query multimedia data composed of different types of news documents (video, audio, image, text) through a multimodal retrieval approach. Our multimedia collection documents are described in NewsML which defines an XML based language for expressing the structure of news and associated metadata.

Our first contribution in this paper is to formulate a conceptual model to formally specify relationships between multimedia resources in our collection and illustrate their use in semantic query processing. The goal of this model is to help the users to query multimedia data through their semantic relationships.

The W3C community has proposed the XML Query Language XQuery. This language has a huge expressive power as it encompasses features belonging both to query and functional languages.

Our second contribution consists on a local extension of XQuery by adapting it to query multimedia documents. So we define new functions and semantic operators to support new relationships defined above.

This paper is organized as follows. In section 2 we present some approaches and works for modeling multimedia data. Section 3 deals with our proposition of creating links between media using contextual model. In section 4 we explain in detail our extension of XQuery. We conclude by listing some future works and perspectives.

2. Related Works

Several researches works in the literature suggest extending XQuery or define a query language for multimedia data, we briefly describe them below and emphasize differences between them and our work.

Reference [2] describes in detail an extension of XQuery with an inflationary fixpoint operator. The authors describe the operator and its semantics. The new construct is a highlevel template which allows expressing most common forms of recursive queries. They show also, how to implement the operator on top of a relational back-end.

Reference [4] proposes to integrate textual, spatial and temporal meta-data in order to find and present document suitable to the user’s needs. We also propose the extension of the XQuery by new operators in.

These works focus mainly on queries over the structure or for retrieval of full-text documents. Contrariwise, the core of our work is retrieving multimedia documents using structure and content. The way we use, is by extending of XQuery with new operators and functions in XQuery grammar to facilitate the use of the meta-documents. We define our contextual model, which allows users to express their need.

In [5] VeXQuery language has been proposed as an extension to XQuery, to resolve the problem of vector-based feature query in MPEG-7 retrieval. To fulfill the vector-based feature query, VeXQuery has given a set of vector similarity measurement expressions and defines the formal semantics of VeXQuery in terms of XQuery itself.

VeXQuery can be integrated seamlessly into XQuery via defining the formal semantics in terms of XQuery itself.

Reference [10] proposes an adaptation algorithm which is domain independent. This algorithm consists in enriching the user query by user profile parameters in order to adapt the results to him. The adaptation can be applied to the content or/and navigation, in order to answer respectively the problems of cognitive overload and disorientation.

In fact, our work also deals with the problem of multimodality, rather than with our global system which proposes different modalities for querying multimedia documents. We will show in section 4, how we could express multimodality over XQuery. In follow, we present some works which mainly focus on multimodality.

Some works combine approaches for context-based and content-base for data retrieval. In [11] author combines visual and textual search to retrieve images. In [12] [13] authors fuse results from two types of queries. They proved that is more successful than using uni-modal retrieval technique.

2M2Net system in [1] represents a novel framework of multi-modality data retrieval in digital libraries. As its specific approaches, the learning-from-elements strategy is devised for interactive propagation of keyword descriptions, and the cross media search mechanism with relevance feedback. Experiments conducted on a digital encyclopedia manifest the efficiency of this approach.

In [3], authors propose multimodal queries through XML fragments query language that was originally designed as a Query by example for full-text XML collections. They introduce a multimedia features into these fragments to query MPEG-7 XML documents.

The aim of [6] is to find a named person in a video, which can be achieved by exploiting the multi-modal information in videos, including transcript, video structure, and visual features. Authors propose a comprehensive approach for finding specific persons in broadcast news videos by exploring multimodal content in videos, such as names occurred in the transcript, face information, anchor scenes, and most importantly, the timing pattern between names and people.

3. Contextual Graph

To search in multimedia collection such as image, video, audio and text, we require integrating heterogeneous multimedia data from different sources into a single set. We propose a novel contextual model for integrating relations between multimedia resources.

We design a common contextual model based on XML Schema. What XML Schemas do, is provide an Object Oriented approach to defining the format of an XML document.

Schemas are more powerful and flexible than DTDs. XML schema has also been used for the XQuery data model.

A number of researchers have attracted much attention to model video content. Some of these proposals are based on describing physical objects and their spatial relationships [7]. Other works depend on the semantic classification of video content, which allow hierarchical abstraction of video expressions representing scenes and events which provides indexing and content-based retrieval mechanisms [8].

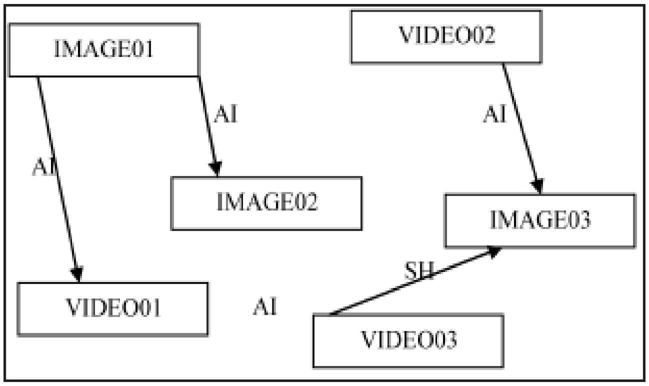

Figure 1. XSD of our contextual model

Many relationships exist between different multimedia resources. These relations could be spatial, temporal or semantic, and could be efficient for multimedia retrieval because any existing systems cannot return all relevant results. The aim of using these relations is the increasing of efficiency of our system. Many ways to raise and describe these relationships in a multimedia context, to facilitate querying by providing a semantic dimension.

Our contextual model c.f Figure1 contains only relationships between documents or parts of documents. It is very simple to extend this model, to add a spatio-temporal relations or a weighting element for example, due to the flexibility of the XML. First relationships would be explicit, and secondly semantic relations may exist implicitly between heterogonous data.

This contextual model contains three types of data: the identifier “ID” of media resources, “type” and the “nature” of relationships, which are given (Talk, Talk About, Appear In, Speak About, Speak, Show) a resource can be seen, can be heard, can be talked of. Note that for resource which represent a part of a full document, the model contain a parent element of this resource. This element is necessary to access to the rest of data related to this resource.

In fact, we can ask directly about object or event in the resources. For example: The president X appear in an interview. This resource might be generated to answer a query like “find all image and video that show president X”. Or find all video which show president X through photo P.

We can ask too referring to the new relationship elicited for the resources. For example: “Show me the picture of catastrophe

X” or “I want to know what Y has said about X”. The second query would look in the audio resources in which Y is the speaker and X is mentioned in the keywords or descriptions of this resource.

Using these capabilities, gives more expressiveness and efficiency by providing more relevant results.

To use this model for querying multimedia documents, we propose a new XQuery operator to retrieve new results based on relations defined above. Since every resource has an identifier “ID” so that facilitate the access to multimedia documents Over the “IDs” of resources.

We plan to extend our model to support spatio-temporal relationships, and to create an inference mechanism to generate new relations based on the old one. A simple example c.f Figure2, of our model is show in follow.

```
 <?xml version=”1.0"?>
 <xsd:schema
 xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>
     <xsd:element name=”RESOURCES”>
     <xsd:complexType>
             <xsd:sequence>
              <xsd:element name=”resource”>
              <xsd:complexType>
                <xsd :element name= “Link”>
                 <xsd:complexType >
                      <xsd:sequence>
                     <xsd:element name=”resource”
                    type=”xsd:IDREF”/>
                     </xsd:sequence>
                    <xsd:attribute name=”name”
                    type=”xsd:STRING” use=”required”/>
                </xsd:complexType >
                </xsd :element>
       </xsd:complexType>
       </xsd:element>
       <xsd:element name=”parent” minOccurs=”0">
       <xsd:complexType>
                <xsd:element name=”id” type=”xsd:ID” />
                 <xsd:element name=”type”
                type=”STRING”/>
      </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
        <xsd:attribute name=”id” type=”xsd:ID”
         use=”required”/>
        <xsd:attribute name=”Type” type=”xsd:STRING”
         use=”required”/>
   </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

(a)

```
<?xml version=”1.0"?>
<resources>
<resource id=”IMAGE01" type=”image”>
<link name= “AI”>
<resource id=”IMAGE02" type=”image”></resource>
<resource id=”VIDEO01" type=”video”></resource>
</link>
</resource>
<resource id=”VIDEO07" type=”video”>
<link name=”SH”>
<resource id=”IMAGE03" type=”image”></resource>
</link>
<parent>
<id>VIDEO011</id>
<type>VIDEO</type>
</parent>
</resource>
<resource id=”VIDEO02" type=”video”>
<link name=”TA”>
<resource id=”IMAGE03" type=”image”></resource>
</link>
</resource>
<resource id=”VIDEO01" type=”video”>
<link name=”TA”>
<resource id=”VIDEO03" type=”video”></resource>
</link>
</resource>
</resources>
```

(b)

Figure 2. Example XML Document (a) and XML Graph (b)

4. Xquery Extension

The XQuery formalism supports user-defined functions. Our proposal consists in extending XQuery with a set of semantic operators to support multimodal querying. The goal is to have a powerful extension that is appropriate for all use cases, including the full-text querying. A lot of languages were designed to be easily understood by humans, but, the number of their experts is still limited.

Our extension will be implemented through graphical interface to allow users expressing their queries over graphical symbols.

As we have mentioned above, this extension will provide another retrieval method of multi-modality data, added to the classic methods combined in our main system.

XQuery is a strongly typed language. Each expression has a type which is inferred during the static analysis and the dynamic evaluation phases of the processing query. Inferring the static type of an expression is based on a set of rules which are a part of the formal semantics of XQuery. This set of rules must be extended to take into account the semantic multimedia operators that we propose. In this paper, we do not deal with this problem, which is certainly not obvious [9]. We use the common XQuery expression's syntax, to define new operators.

4.1 XQuery extended with Functions

We describe here new defined functions including the formal definition.

```
declare function local:FModel($X as xs:string*,$Y as
xs:string*)
{
 for $x2 in doc('model.xml')//resources/resource
 where $x2/@id=$X
  and
 $x2/link/@name=$Y
 return $x2/link/resource
};
```

$X represents one of relation types defined above.
$Y represents a resource identifier.

This function returns additional result to the main query. In fact, these additional results are based on our contextual model. This is an optional function that could provide results which would not appear previously. It takes a resource and a relation type, in the input and returns the set of the resource which is related to the input resource over this relation.

```
declare function local:fn_nature($R as xs:string*,$N as
xs:string*) as xs:integer
{
if($N='All' or $R=$N) then
 1
 else 0
};
declare function local:keyword_Spec ($X as xs:string*,$N as
xs:string*,$G as xs:string*,$EX as xs:string*)
{
let $S1:=doc($X)//newsItem//genre/name/text()
let $R:=doc($X)//itemMeta/itemClass/@qcode
let $E:=doc($X)//remoteContent/@contenttype
where
$E=$EX
and
(local:fn_nature($R,$N)=1)
and ($S1=$G)
    return 1
};
```

This function has the role to verify if the queried file matches the required specification of the user to find multimedia resources. It contains four parameters: $X: represents a resource's identifier
$N: Type of searched file
$G: Genre of searched file

$EX: Extension of searched file

The result of this function is Boolean. If the resource matches user’s criteria, the function will return value 1. The Keyword function needs to use another user defined function ‘fn_nature’ which is used to verify if the user searches a specific type of media or any type.

```
declare function local:Match($X as xs:string*,$KS as
xs:string*) as xs:integer
{
if (some $KS1 in tokenize($KS, “\s”),$SU in $X satisfies
contains($SU,$KS1)) then
1 else 0
};
declare function local:keyword($X as xs:string*,$KS as
xs:string*)
{
let $SUBJECT:=doc($X)//newsItem//subject/name/text()
let $DESC:=doc($X)//newsItem//description/text()
let $HLINE:=doc($X)//newsItem//headline/text()
where
local:Match($X,$KS)
or
(some $KS1 in tokenize($KS, “\s”) satisfies
contains($HLINE,$KS1))
or
(every $KS1 in tokenize($KS, “\s”) satisfies
contains($DESC,$KS1))
return 1
};
```

This function has the role to verify if the queried file matches the keywords entered by the user.

$KX: represents keywords
$X: represents a resource

It requires the use of another user defined function ‘Match’ which is used to verify the matching between keywords and subjects of resource.

```
declare function local:EQUAL($X as xs:string*,$Y as
xs:string*) as xs:integer
{
let $Y1:=doc($Y)//newsItem/contentMeta/subject
return
if (some $S in $Y1 satisfies
$S=$X ) then 1 else 0
};
declare function local:PSIMILAR($X as xs:string*,$Y as
xs:string*)
{
let $SUBJECT :=
doc(‘v2.xml’)//newsItem/contentMeta/subject
let
$G:=doc(‘v2.xml’)//newsItem/contentMeta/genre/name/text()
let $SUBJECT1 :=
doc(‘v3.xml’)//newsItem/contentMeta/subject
let
$G1:=doc(‘v3.xml’)//newsItem/contentMeta/genre/name/text()
return
if (every $F in $SUBJECT satisfies
local:EQUAL($F,$Y)=1
and $G=$G1
)
then
1 else 0
};
```

This function ‘PSIMILAR’ verifies if two resources are similar. As we have indicated above, our system is multimodal. The user could supply a multimedia resource as input in addition to keywords. This function will return 1 only if there is almost total matching between the input and the searched resource. This function could be extended with additional criteria or transformed for flexible use. It uses another user defined function EQUAL to verify the similarity between subjects of two resources.

```
declare function local:VCONTENT($X as xs:string*,$Y as
xs:string*)as xs:integer
{
if (some $x in tokenize($X,”\s”) satisfies
some $y in tokenize($Y, “\s”) satisfies $x = $y)
then 1 else 0
};
declare function local:EQUAL2($X as node()* ,$Y as
node()*) as xs:integer
{if (some $S in $Y satisfies
$S=$X ) then 1 else 0
};
declare function local:SIMILAR($X as xs:string*,$Y as
xs:string*)
{
let $SUBJECT := doc($X)//newsItem/contentMeta/subject
let $SUBJECT2 := doc($Y)//newsItem/contentMeta/subject
let $TITLE := doc($X)//newsItem/itemMeta/title/text()
let $HEADLINE:=
doc($X)//newsItem/contentMeta/headline/text()
let
$DESC:=doc($X)//newsItem/contentMeta/description/text()
let $SUBJECT1 := doc($Y)//newsItem/contentMeta/subject
let $HEADLINE1:=
doc($Y)//newsItem/contentMeta/headline/text()
let $DESC1:=
doc($Y)//newsItem/contentMeta/description/text()
let $TITLE1 := doc($Y)//newsItem/itemMeta/title/text()
let $GENRE :=doc($X)//newsItem/contentMeta/genre/text()
let $GENRE1:=doc($Y)//newsItem/contentMeta/genre/text()
return
if (some $F in $SUBJECT satisfies
local:EQUAL2($F,$SUBJECT2)=1
and $GENRE=$GENRE1
and
local:VCONTENT($HEADLINE,$HEADLINE1)=1
and
local:VCONTENT($DESC,$DESC1)=1
and
local:VCONTENT($TITLE,$TITLE1)=1
)
then 1 else 0
};
```

There is a difference between this function and the previous one. ‘SIMILAR’ function returns true if there is a matching between two resources and returns false otherwise. In addition, it could generate an additional result in the main query. It has fewer restrictions compared to the previous. This function could be extended with additional criteria. We could for example define a weighting constant to fix threshold. The weight is then calculated through a given rule **R.** Since some researches [14] have shown that document weighting as well as query term weighting are necessary tools for effective retrieval in textual documents, we expect the same thing with multimedia documents. This function uses two user defined functions EQUAL2 and VCONTENT, like PSIMILAR, it takes two resources as input.

4.2 XQuery extended with Operators

Operator 1: Sem_Operator

The operator **Sem_operator** that we propose here is applied to our conceptual model and our collection. It will filter the result of queries. Actually it allows users to specify their requirement. This operator is not limited to a particular media but it allows the user to make a search in all multimedia documents. It has the following syntax:

This operator is used within a “Where clause” expression.

whereClause ::= “where”(**Expr**| **OperatorExpr***)

The “ | “ operator builds the union of two operators. The wildcard “*” in OperatorExpr expression means that we can take zero or more sequences of semantic operators as input.

The OperatorExpr has the following syntax:

OperatorExpr ::= “(“Expr <**SOperator**> Expr “)”
SOperator ::= **Talk About** |**Appear In** |**Talk**| **Show**| **Speak**|**Speak About**

TalkAbout ::= “TA”

Talk ::= “T”

AppearIn ::= “AI”

Show:= “SH”

Speak:=”S”

Speak About:=”SA”

This operator represents the semantic relationship between Parts of documents and between whole ones. SOperator is a list of semantic elementary operators: Talk About and Talk are used for relation in video resources Speak and Speak About are used for relation in audio resources. A simple example is show in follow.

```
<result>
Let $Y:=doc(“F1.xml”)
For $X in collection()//
Where $X2=”WAR”
And
$X AI $Y
Return
{<id>$X/ID</id>
}
</result>
```

Operator2: SIMILAR

The second operator we propose here is called SIMILAR, this operator is used within a “Where clause” expression:

whereClause ::= “where”(**Expr**| **OperatorExpr***)

The “ | “ operator builds the union of two operators. The wildcard “*” in OperatorExpr expression means that we can take zero or more sequences of semantic operators as input.

The OperatorExpr has the following syntax:

OperatorExpr ::= “(“Expr <**SOperator**> Expr “)”
SOperator ::= **SIMILAR**

This operator returns “True” if there is similarity between two resources. It matches between resources given by the user, In fact, it is a conditional clause that filters the result more. In other words this operator lets us derive to the concept of multimodality, since we will have four modalities for querying in our system, one of them is by defining a resource for matching resources which provide more relevant results. An example of multimodality is the fact that the user uses two types of media resources in query at the same time. An example is given below:

```
<result>
Let $Y:=doc(“F1.xml”)
For $X in collection()//
Let $X2:=$Y/keywords
Where $X2=”WAR”
And
$X SIMILAR $Y
Return
{<id>$X/ID</id>
}
</result>
```

Where:
$Y: media resource
$X2: variable
$X: specific collection of documents

This query return documents which contains the word “WAR” as keyword and must be similar to $Y.

5. Conclusion

The main contribution of our work is to provide a framework for retrieval of multi-modality data instead of any specific type of media.

In this paper, we have presented a contextual model for multimedia XML document based on the XML schema. It consists in enclosing semantic relations between multimedia resources, to be used over XQuery to enhance relevance in multimedia retrieval.

We have also proposed a set of semantic operators and functions to extend the XQuery. In addition, we have an example to express multimodal query over XQuery. This contribution may improve multimedia retrieval.

Our future work research deals with enhance the language extension and the mapping of proposed graphical expressions to a standard formalism.

References

[1] Yang, J., Zhuang, Y., Li, Q.(2001). Search for multimodality data in digital libraries, *In*: Proc. 2nd IEEE Pacific-Rim Conference on Multimedia (PCM 2001), p. 482-489, Beijing, China, Oct.
[2] Afanasiev, L., Grust, T., Marx, M., Rittinge, J., Teubner, J. (2008). An Inflationary Fixed Point Operator in XQuery, *In*: 24th International Conference on Data Engineering ICDE’08. p.1504~1506.
[3]Mamou, J., Mass, Y., Shmueli-Scheuer, M., Sznajder, B., (2007). A Query Language for Multimedia Content, *In*: Multimedia Information Retrieval workshop, SIGIR.
[4] Ammar, S., Amous, I., Gargouri, F. (2005).Contribution to graphical querying language for XML semi-structured data, *In*: International conference on signal-image technology & internet– based systems. Yaoundé, Cameroon .

[5] Xue, L., Li, C., Wu, Y., Xiong, Z. (2006). VeXQuery: An XQuery Extension for MPEG-7 Vector-Based Feature Query, *In:* Advanced Internet Based Systems and Applications: Second International Conference on Signal-Image Technology and Internet-Based Systems, SITIS.

[6] Yang, J., Chen, M., Hauptmann, A., (2004). Finding Person X: Correlating Names with Visual Appearances, *In:* Int'l Conf. on Image and Video Retrieval, Dublin City, July 21-23.

[7] Berretti, S., Del Bimbo, A., Vicario, E., (2001). Efficient Matching and Indexing of Graph Models in Content-Based Retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (10) 1089-1105, October.

[8] Zhu, X., Fan, J., Elmagarmid, A.K., Wu, X., (2003). Hierarchical video content description and summarization using unified semantic and visual similarity, *In:* Multimedia Systems, 9 (1) 31-53, July .

[9] Bruno, E., Le Maitre, J., Murisasco, E. (2003). Extending XQuery with Transformation Operators, *In*: Proc. ACM symposium on Document engineering France.

[10] Zayani, C., Péninou, A., Canut, M.F., Sèdes, F. (2006). An adaptation approach: query enrichment by user profile in Signal-Image Technology & Internet—Based Systems, SITIS .

[11] Mori, Y., Takahashi, H., Oka, R.(1999). Image-to-word transformation based on dividing and vector quantizing images with words, *In:* International Workshop on Multimedia Intelligent Storage and Retrieval Management.

[12] Jones, G. J. F., Burke, M., Judge, J., Khasin, A., Lamadesina, A. M., Wagner, J. (2005). Dublin City University at CLEF 2004 Experiments in Monolingual, Bilingual and Multilingual Retrieval, *In*: Proc. CLEF 2004: Workshop on Cross-Language Information Retrieval and Evaluation, Bath, U.K., p. 207-220 .

[13] Torjmen, M., Pinel-Sauvagnat, K., Boughanem, M. (2008). Methods for combining content-based and textual-based approaches in medical image retrieval in Evaluating Systems for Multilingual and Multimodal Information Access, 9th Workshop of the Cross-Language Evaluation Forum, Bibliographie 183 CLEF 2008, Aarhus, Denmark, September 17-19, Revised Selected Papers, p. 691–695.

[14] Wang, J.Z., Du, Y. (2001). RF*IPF: a weighting scheme for multimedia information retrieval, *In:* 11th International Conference on Image Analysis and Processing (ICIAP'01) Palermo, Italy.