https://doi.org/10.48009/1_iis_2010_483-490

A SMART WEB IMAGE RETRIEVAL SYSTEM BASED ON SEMANTIC MARKUP FOR INTELLIGENT E-BUSINESS

Seong-Yong Hong, Korea Advanced Institute of Science and Technology (KAIST), gosyhong@kaist.ac.kr

ABSTRACT

Many companies are rushing into e-business and adopting dangerous strategies by applying old business models to e-business.

Therefore, with the rapid development of internet technology, the number of internet users and the amount of web image information on the internet is ever increasing. Recently, the web sites, such as e-business sites and shopping mall sites, deal with lots of image information. As a result, it is required to support semantic markup-based image retrieval efficiently on such web image data.

This paper proposes a Smart Web Image retrieval System (SWIS), which adopts semantic markup technology. To support object-based content retrieval on product catalog web images containing multiple objects, we describe a multi-level metadata structures representing the local features, global features, and semantics of image information. To enable semantic-based and content-based retrieval on such image data, we design a XML-Schema for the proposed metadata. We also describe how to automatically transform the retrieval results into the forms suitable for the various user environments, such as web browser or mobile browser, using XSLT. The proposed scheme can be utilized to enable efficient web image metadata sharing between systems, and it will contribute in improving the retrieval correctness and the user's satisfaction on semantic markup-based web e-catalog image retrieval.

Keywords: Semantic Markup, XML, Metadata, XML-Schema, Web Image Retrieval, Intelligent E-Business.

INTRODUCTION AND BACKGROUND

Recently, the usage of web image information is ever increasing with the rapid development of multimedia technology. Multimedia data is commonly used on wired and wireless internet in various industrial areas including information technology area. A number of research groups are exploring different approaches to provide and transform image information on various client devices, such as web browsers, PDAs, iPhone, smart phone, iPad and cellular phones. There are various types of multimedia data in various industrial areas and it becomes quite essential to provide standard format of multimedia information to allow data exchange and sharing thought internet. One of the well-known standards for the description of multimedia information is MPEG-7. There are efforts to represent MPEG-7 based multimedia information in XML format, but they are still in an early stage [1].

Among multimedia data, images are widely used in various applications. Retrieving images attracts high and increasing interest from the wide range of applications, such as medical information systems, biological information systems, electronic museums, e-commerce electronic catalogs, etc. There have been lots of research works to support content-based retrieval on images. But, we found that most of the previous research was focusing on content-based retrieval of images based on image features in global level.

This paper is an effort to make web e-catalog images better utilized by adopting semantic markup technology. To support object-based content retrieval on product catalog images containing multiple objects, we describe a multi-level metadata structures representing the local features, global features, and semantics of image data. To enable semantic-based and content-based retrieval on such image data, we design a XML-Schema for the proposed metadata and show how to represent such metadata using XML documents. We also describe how to automatically transform the retrieval results into the forms suitable for the various user environments, such as web browser or mobile browser, using XSLT. The proposed scheme can be easily implemented on any commercial platforms supporting XML technology. We strongly believe that the proposed schemes can be utilized to enable efficient image metadata sharing between systems. Also, it will contribute in improving the retrieval correctness and the user's satisfaction on semantic markup-based web image retrieval.

The remainder of this paper is organized as follows. Section 2 describes overview on previous image retrieval systems and XML based on semantic markup related technologies. Section 3 explains the proposed SWIS architecture. In section 4, we show a multi-level image metadata modeling and represent our modeling results using XML-schema. In section 5, we explain the prototype implementation of smart web image retrieval system by using a dynamic XML document search techniques. Finally, section 6 summarizes the paper.

RELATED WORK

The previous image searching techniques can be classified into three branches. First approaches are simple search techniques based on keywords, which describes subjects or titles. Formatted data on images and image file names are stored in databases, while images are stored in external files. Searching operations are usually performed by using descriptive keywords. Automatic extraction of image related keywords from HTML web pages are possible, but unrelated or unimportant words can be selected deteriorating retrieval performance.

Second approaches are content-based image retrieval techniques using feature vectors. Multidimensional vectors representing color, texture, and shape features of images are commonly used. Image metadata are added within databases or file systems to support similarity-based searching on image data [2,3].

Third approaches are hybrid style of above approaches. Feature vectors are extracted by image preprocessing and meta information, such as keywords, semantic information, and visual information, are manually or automatically added.

Semantic Markup

Semantic markup adds meaning to a document. The markup looks physically similar to structure markup. Instead of describing physical parts of the document, it applies meaning to the text contained inside the body it is tagging. It is completely abstract from the text and implies no styling whatsoever. Because we earlier moved all our styling into a separate stylesheet, we are now able to apply that styling in a more logical way. Because a fragment of text can be labeled as a product name or price code, we can apply style to objects of a similar semantic type. Sales price and discount amount, for instance, can be distinguished from each other now and styled accordingly. This approach is used in syndication schemes like Atom, which marks a new item as having been updated. On its own, that means nothing and is invisible to the user. If the RSS browser chooses to, it can highlight any updated items by applying a style directly to those objects and only those objects. If we can distinguish between semantics for applying style and appearance control, then we can leverage those semantics to pass meaningful documents from one application to another. This is the core of way XML is so powerful. If there were one single reason to use XML, this would be it. A variant of semantic markup seeks to describe document content in ways other than by a strictly hierarchical tree structure. Trees are useful, but they are not the best way to represent all content. The important thing is to indicate the semantics. Possibly not maintain a tree- like structure to the layout. Internally, even documents that don't look tree-structured really are for purpose of XSLT transforms. The tree still has a top node but builds a flatter structure with fewer levels underneath [4].

XML Techniques

XML is a metalanguage, a language that can be used to describe other languages. Is covers a huge range of file-based storage structures. If an application is expecting to import a MathML file and you give it a NewsML file instead, the interchange won't work. The Document Type Definition (DTD) differences might be handled quite gracefully and some systems may be able to import freeform XML data of any kind without being told what kind of data it is. XML dose provide supporting mechanisms that would allow you to embed a MathML formula inside a NewsML story, and the name-spacing mechanism and DTD support would ensure that both could coexist and their intent and meaning could be understood unambiguously by the receiving application. Table 1 lists some XML-based language schemas.

Table 1. Example of XML Schemas

XML Schema	Explain
Atom	Atom Syndication Format
ChemML	Chemical modeling
DocBook	Book manuscripts
GML	Geography Markup Language
MathML	Mathematical formulae
MusicML	Music scores
NewsML	Syndicated news stories
RDF	Resource Description Framework
RSS	Really Simple Syndication
SVG	Vector graphic diagrams
XUL	XML User Interface Language
XSPF	XML Shareable Playlist Formal

XML is a standard markup language proposed for data exchanges on the web. XML is proposed by W3C to describe next generation web pages. In database field, there have been lots of research efforts to store, index and retrieve XML documents in database systems [5-8]. Some systems are stand-alone, while others are built on top of relational databases or object-relational databases. Another related research trend is to transform normal data stored in databases into XML documents for efficient integration of heterogeneous information resources. New query languages, such Xpath, XML-QL, XQL, Quilt and Xquery, are proposed to support structure-based and content-based retrieval of XML documents [9-13].

One of the interesting characteristics of XML is document conversion technology. XSLT is used to transform a XML document into a document having different format [14]. Figure 1 shows a situation where XML documents are transformed into WML (Wireless Markup Language) documents by using XSLT for wireless internet users.

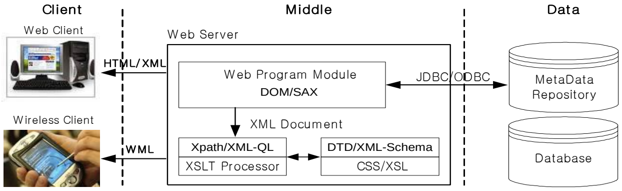


Figure 1. XML-based web service architecture

Image Retrieval System

Some systems support keyword only retrievals and others support content-based image retrievals. In the latter approach, image retrievals based on feature information, such as average colors, color histograms, texture patterns, and shape objects, are supported. Most of them are developed for image database applications. Representative examples are QBIC, Safe, VisualSEEK, Phontobook, WBIIS, Chabot, and Blobworld. One of the most recent research work has been done by the SIMPLIcity system which supports content-based image retrieval based on the color, texture, shape features, while increasing matching correctness by utilizing local features on regions. In the medical domain, the KMeD(Knowledge-Based Medical Databse) system utilizes semantic modeling focusing on object shapes and spatial relationships between them [15-22].

SWIS ARCHITECTURE

This paper proposes a system, called SWIS(**S**mart **W**eb **I**mage retrieval **S**ystem), that can support intelligent image retrieval by utilizing dynamic XML documents. We utilize commercial database systems as storage systems to make our system stable and cost-effective. The system architecture is shown in figure 2. Data extractors preprocess images to extract image feature (global feature) and object feature (local feature). Object features are used to provide content-based retrieval from the viewpoint of component objects contained in images. Image features, object features, semantic metadata, and other meaningful image metadata are stored in databases. Semantic markup-based queries are submitted through query interface and processed by feature analyzer to extract major features of query objects. The matched results are transformed into HTML or WML documents according to the client platforms.

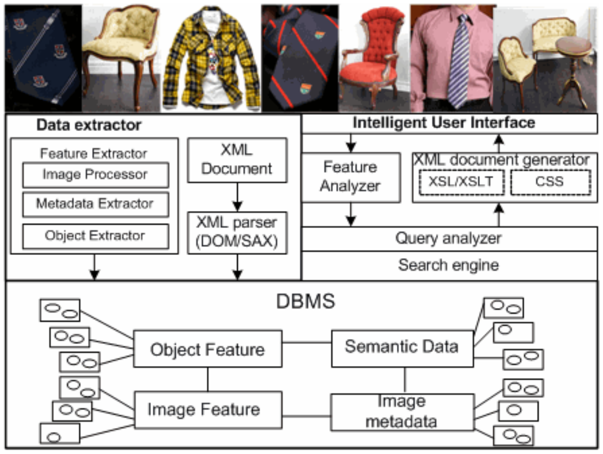


Figure 2. System architecture of the SWIS

The storage structures are designed by considering mapping relationships with XML documents. Our metadata structure contains global feature, local feature, semantic metadata, and major element of MPEG-7 standard. Figure 3 shows the proposed metadata schema in E-R diagram. Feature vectors for objects in images are stored in Color, Shape, Texture, and Spatial tables. Semantic information and other meaning metadata are stored in SemanticDS and Metadata InformaionDS, respectively. One or more images can share metadata of contained objects by referencing OIDs (Object IDs).

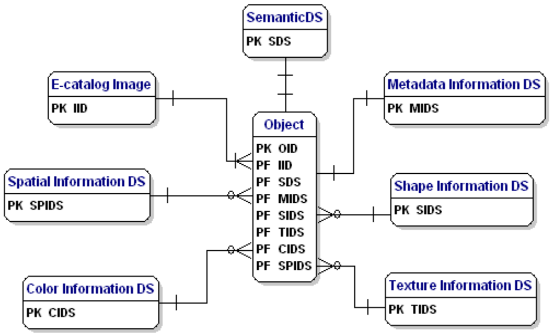


Figure 3. Metadata Schema for Web Images

MULTI-LAYER IMAGE METADATA MODELING

In this section, we propose multilevel metadata structures for smart web image retrieval. Image metadata is data for image data. There are two types of basic metadata for images.

■ **registration metadata:** Image resolution(width, height), color map, compression ratio, etc are typical examples of registration metadata. This metadata is required to display and manipulate images. In image files, this information is usually hidden within image headers.

■ **description metadata:** Image title, caption, keywords, natural language descriptions, and image file names are typical examples of descriptive metadata. This metadata is used to search images, when content-based retrieval operations are not supported.

Registration metadata or description metadata are not enough for semantic markup-based searching. We can further define a multi-layered metadata structure on top of raw image data.

■ **global feature metadata:** Average values or multi-dimensional vectors representing color, texture, and shape of image are examples of global feature metadata. This metadata is heavily used in current content-based retrieval system.

■ **local feature metadata:** Average values or multi-dimensional vectors representing color, texture, and shape of each objects or regions belonging to a given image are examples of local feature metadata.

■ **semantic contents metadata:** Subjective feelings and knowledge's on images, such as concepts, meaning, category, spatial relationships, or other useful interpretations are examples of semantic metadata.

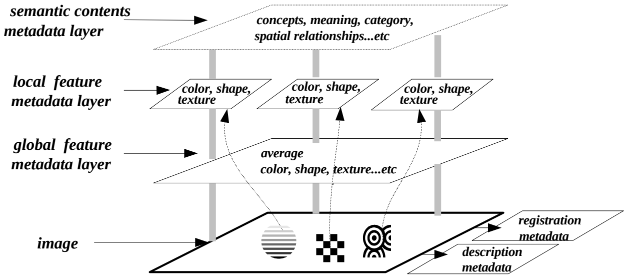


Figure 4. Multi-layer Metadata Modeling

Figure 4 explains a processing model of multi-layered metadata from an image. The bottom layer shows examples of objects contained in images. Global metadata layer contains global features, such as global average color, and it also represents semantic content or knowledge content of a total image. Local metadata layer represents feature vectors of each objects belonging to a given image. Semantic content metadata layer represents semantics or knowledge content of each object.

For example, a web e-catalog image usually has multiple objects in an image. A feature vector assigned to each object plays an important role in image searching process. In addition, it can be utilized to improve the efficiency of searching process. There has been a limitation on user's satisfaction in existing search engines, because they mainly rely on global features only.

Figure 5 shows an example of a web e-catalog image containing multiple objects. The first image of figure 5 has multiple central objects "chair" and another central objects "table". The third image has central objects "necktie" and "T-shirt". The fifth image has central object "T-shirt". But, it is easily shown that each has different color. Like this situation, central objects can have different feature vectors, even though their meanings are the same. Vice versa, they can have same feature vectors, while their meanings are different.

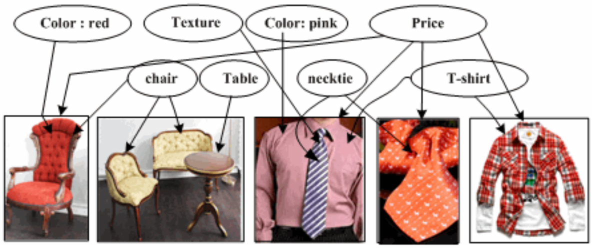


Figure 5. Examples of web e-catalog images

XML Representation of Image Metadata

The DTD structure to represent metadata of an image is shown in figure 6. This DTD structured is focused on the object viewpoint as well as the global image viewpoint.

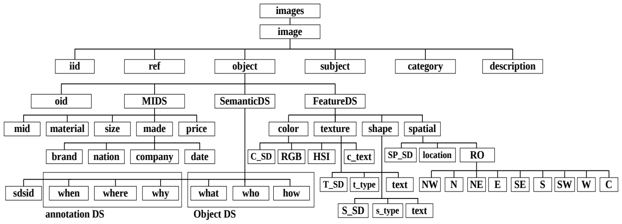


Figure 6. DTD Structure for metadata

In this design, we also adopted major elements of MPEG-7 standard. Figure 7 shows an example of XML document to describe metadata of an image containing multiple objects.

```
<?xml version='1.0' encoding='euc-kr' ?>
<images xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <image>
    <iid>i00001</iid>
    <ref> image01.jpg</ref>
    <subject> man shirt </subject>
    <category> fashion </category>
    <description/>
  <object> <oid>o00001</oid>
    <MIDS>
      <mid>m0001</mid>
      <material> cotton 100% </material>
      <size> large </size>
      <made> <company> samsung </company>
             <nation> korea </nation>
             <brand>SMC</brand>
          <date> 2002-10-15</date> </made>
      <price>33,000</price>
    </MIDS>
    <semanticDS> <sdsid>sd0001</sdsid>
      <when>2002-08-28</when>
      <where/>
      <why/>
      <what> shirt</what>
      <who> man </who>
      <how/>
    </semanticDS>
    <featureDS>
      <color> <cid>c0001</cid>
         <r>255</r><g>0</g><b>0</b><h>0</h><s>0</s><i
>100</i>
         <color_SD> warm</color_SD>
        <color_text>red</color_text>
      </color>
      <texture><tid>t0001</tid>
        <texture_type>0001000000000000</texture_type>
        <texture_SD>coarse</texture_SD>
          <texture_text>stripes</texture_text>
      </texture>
      <shape><sid>s0001</sid>
        <shape_type>0100000000000000</shape_type>
        <shape_SD>simple</shape_SD>
        <shape_text>rectangle</shape_text>
      </shape>
      <spatial> <spid>sp0001</spid>
        <location>NE</location>
        <spatial_SD>enough</spatial_SD>
        <RO> <NW/> <N/> <NE/> <E/> <SE/> <S/> <SW/>
<W/> </RO>
      </spatial>
    </featureDS>
   </object>
 </image>
</images>
```

Figure 7. An example of XML document to describe metadata of web image

We use XML-Schema to define the form of XML documents for image metadata. We designed a XML-Schema to represent image metadata containing multiple objects. Figure 8 shows a metadata structure of such an image from the viewpoint of elements. 'object-1' references all the elements, while 'object-2' references part of its elements. The proposed XML-Schema model allows reuse of element definition. It also allows definition of patterns or models within elements.

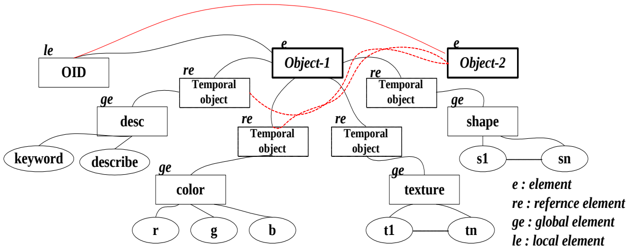
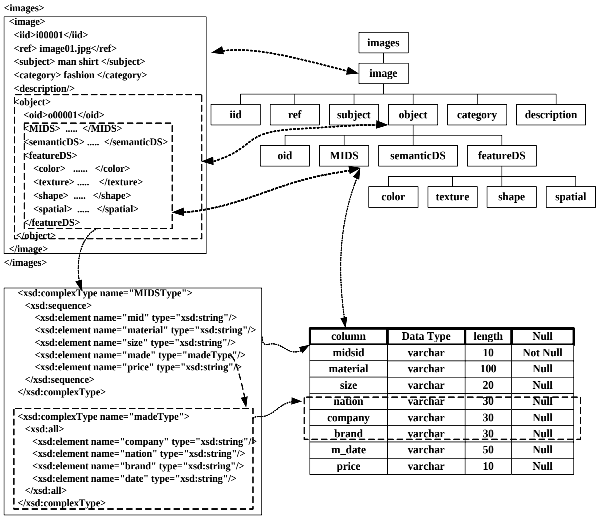


Figure 8. XML-Schema modeling for multiple objects

```
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:complexType name="imagesType">
    <xsd:sequence>
    <xsd:element name="image" type="imageType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="imageType">
    <xsd:sequence>
    <xsd:element name="iid" type="xsd:string"/>
    <xsd:element name="ref" type="xsd:string"/>
    <xsd:element name="subject" type="xsd:string"/>
    <xsd:element name="category" type="xsd:string"/>
    <xsd:element name="description" type="xsd:string"/>
    <xsd:element name="object" type="objectType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="objectType">
  <xsd:sequence>
<xsd:element name="oid" type="xsd:string"/>
<xsd:element name="MIDS" type="MIDSType"/>
<xsd:element name="semanticDS" type="semanticDSType"/>
<xsd:element name="featureDS" type="featureDSType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="MIDSType">
    <xsd:sequence>
    <xsd:element name="mid" type="xsd:string"/>
    <xsd:element name="material" type="xsd:string"/>
    <xsd:element name="size" type="xsd:string"/>
    <xsd:element name="made" type="madeType"/>
   <xsd:element name="price" type="xsd:string"/>
</xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="madeType">
    <xsd:all>
    <xsd:element name="company" type="xsd:string"/>
    <xsd:element name="nation" type="xsd:string"/>
    <xsd:element name="brand" type="xsd:string"/>
    <xsd:element name="date" type="xsd:string"/>
</xsd:all>
  </xsd:complexType>

…….

<xsd:simpleType name="bitType">
    <xsd:restriction base="xsd:string">
            <xsd:length value="16"/>
            <xsd:pattern value="[0,1]*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="spatialType">
    <xsd:all>
    <xsd:element name="spid" type="xsd:string"/>
    <xsd:element name="location" type="xsd:string"/>
    <xsd:element name="spatial_SD" type="xsd:string"/>
    <xsd:element name="RO" type="ROType"/>
    </xsd:all>
  </xsd:complexType>

  <xsd:complexType name="ROType">
    <xsd:sequence>
    <xsd:element name="NW" type="xsd:string"/>
    <xsd:element name="N" type="xsd:string"/>
    <xsd:element name="NE" type="xsd:string"/>
    <xsd:element name="E" type="xsd:string"/>
    <xsd:element name="SE" type="xsd:string"/>
    <xsd:element name="S" type="xsd:string"/>
    <xsd:element name="SW" type="xsd:string"/>
    <xsd:element name="W" type="xsd:string"/>
</xsd:sequence>
  </xsd:complexType>
     <xsd:element name="images" type="imagesType" />
</xsd:schema>
```

Figure 9. An example of XML-Schema

Figure 9 shows a document about XML-Schema in figure 8. In this way, XML-Schema documents verify effectiveness XML documents for multi-object images. Figure 10 shows the section, which defines image metadata information in an XML-Schema document. It also explains a mapping scheme of image metadata part in XML-Schema document into relational databases. A user define type called “madeType” is used in order to define a child of an element in an XML-Schema document. Definition of such parent elements and child elements are mapped to table schemes in relational databases.



```
<images>
 <image>
  <iid>i00001</iid>
  <ref> image01.jpg</ref>
  <subject> man shirt </subject>
  <category> fashion </category>
  <description/>
  <object>
    <oid>o00001</oid>
    <MIDS>  .....  </MIDS>
    <semanticDS> .....  </semanticDS>
    <featureDS>
      <color>   ......   </color>
      <texture> .....    </texture>
      <shape>  .....   </shape>
      <spatial>  .....   </spatial>
    </featureDS>
  </object>
 </image>
</images>
```

```
  <xsd:complexType name="MIDSType">
    <xsd:sequence>
      <xsd:element name="mid" type="xsd:string"/>
      <xsd:element name="material" type="xsd:string"/>
      <xsd:element name="size" type="xsd:string"/>
      <xsd:element name="made" type="madeType"/>
      <xsd:element name="price" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="madeType">
    <xsd:all>
      <xsd:element name="company" type="xsd:string"/>
      <xsd:element name="nation" type="xsd:string"/>
      <xsd:element name="brand" type="xsd:string"/>
      <xsd:element name="date" type="xsd:string"/>
    </xsd:all>
  </xsd:complexType>
```

column	Data Type	length	Null
midsid	varchar	10	Not Null
material	varchar	100	Null
size	varchar	20	Null
nation	varchar	30	Null
company	varchar	30	Null
brand	varchar	30	Null
m_date	varchar	50	Null
price	varchar	10	Null

Figure 10. Mapping XML schema to RDB tables

PROTOTYPE IMPLEMENTATION

In this section, we describe implementation aspects of intelligent image retrieval system proposed in this paper. The prototype system is implemented on the Windows XP with MS-SQL SERVER 2005. Web client platform is Microsoft Internet Explorer 7.0 and Openwave SDK 5.1 by Openwave. We used languages, such as DOM(Document Object Model), JAVA and JSP(Java Server Page) based on SAX(Simple API for XML). Web server is IIS virtual directory server that is supported by SQL server.

Storing XML Documents in RDB System

XML document can be used to provide the platform transparency hiding the difference between systems. The proposed system provides such transparency. We keep XML documents in disk databases. Data extraction procedure from XML documents is explained in Figure 11.

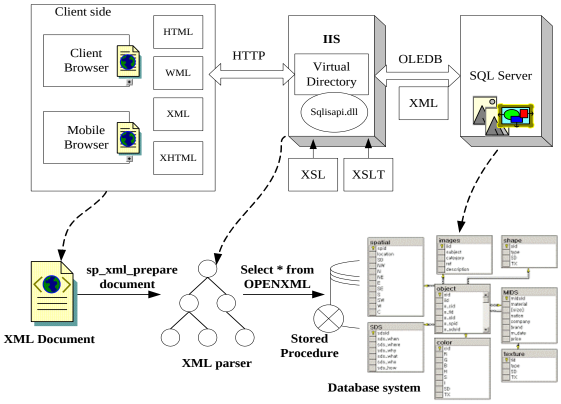


Figure 11. Process to store XML documents

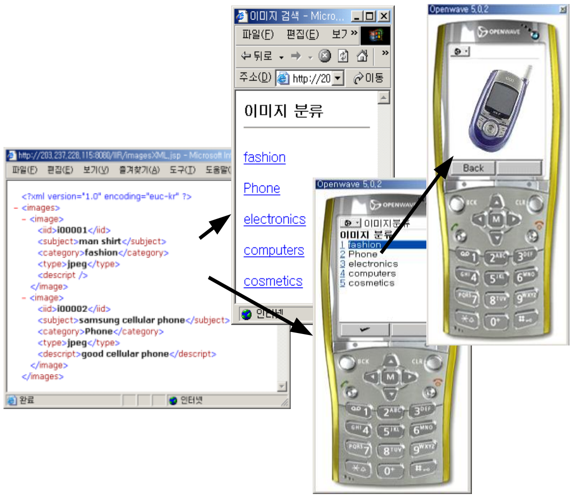
To store data, XML documents are parsed and node structure of XML documents is mapped into tree structure before OpenXML function is called. The stored procedure *sp_xml_preparedocument* validates effectiveness of XML document. After validation, node tree handle, that can extract data from attribute and element, is returned by this stored procedure. After node tree creation is finalized, data is stored in the table to return low set data of XML document.

Figure 12 explains the procedure to store portion of XML document describing image metadata into database tables.

```
CREATE PROCEDURE INSERTXMLDOC
AS
   DECLARE @xmlDATA VARCHAR(2000)
   SET @xmlDATA = '
          XML Document Structure
'

DECLARE @iTree INTEGER
EXEC  sp_xml_preparedocument  @iTree  OUTPUT,
@xmlDATA

      INSERT images (Attributes)
      SELECT   DB Schema
                  FROM
                  OPENXML(@iTree, 'images/image', 1)
                  WITH ( XML Tags)
```

Figure 12. Procedure to store XML documents

By applying XSL stylesheet into XML data, we can transmit retrieved data in different forms, such as WML documents, to web browser-based clients, such as mobile phones using WAP. Before web browser accesses web server, a HTTP header, which contains the required pages, the machine type used by current web user and various information about web browser, is sent to the web server. A HTTP header in HTTP request packet also contains information on user IP, OS, type of browser, type of document that can be processed in web browser and cookie used by login information. We can submit the retrieval results to the mobile device after translating them by using XSLT. For this process, HTTP header information is used to identify the type of connected machine. GetHeader method of request object is used to search the header information and *"ACCEPT"* attribute is used to identify the type of connected machine. To identify connected browser, *"USER-AGENT"* attribute is used. Figure 13 shows how image searching is done using XSLT transform process.



```
<?xml version="1.0" encoding="euc-kr" ?>
- <images>
  - <image>
      <iid>i00001</iid>
      <subject>man shirt</subject>
      <category>fashion</category>
      <type>jpeg</type>
      <descript />
    </image>
  - <image>
      <iid>i00002</iid>
      <subject>samsung cellular phone</subject>
      <category>Phone</category>
      <type>jpeg</type>
      <descript>good cellular phone</descript>
    </image>
  </images>
```

Figure 13. An example of image searching in SWIS

CONCLUSIONS

XML related technologies are evolving rapidly in internet environments. Many application systems that utilize merits of XML technologies are recently developed. However, there have been little efforts in adopting XML techniques to realize information retrieval and efficient exchange of multimedia data.

In this paper, we presented an intelligent image retrieval system, which adopts XML technology. To support object-based content retrieval on product catalog images containing multiple objects, we described a multi-level metadata structures which represent the local features, global features, and semantics of image data. To enable semantic-based and content-based retrieval on such image data, we designed a XML-Schema for the proposed metadata and showed how to represent such metadata using XML documents. We also described how to automatically transform the retrieval results into the forms suitable for the various user environments, such as web browser or mobile browser, using XSLT. The proposed scheme can be easily implemented on any commercial platforms supporting XML technology.

We are planning to develop an automatic conversion module of multimedia data such as image or video, according to the client platform of end users. There should be further researches to improve user's satisfaction by developing various contents transformation XSLT.

ACKNOWLEDGE

This work was supported by the Korean Government IT R&D program of MKE/KEIT. [10035166, Development of Intelligent Tutoring System for Nursing Creative HR]

REFERENCES

1. Neil Day.(2001). Introduction to MPEG-7 http://www.w3.org/2001/05/mpeg7/w4032.doc
2. H. V. Jagadish. (1991). A retrieval technique for similar shapes, Proceedings of the 1991 ACM SIGMOD international conference on Management of data, pp.208-217.
3. D. Androutsos, K. N. Plataniotis and A. N. Venetsanopoulos. (1998). Image Retrieval Using the Directional Detail Histogram, Storage and Retrieval for Image and Video Database SPIE, pp.129-139.
4. Berners-Lee, T.; Hendler, J.; and Fensel, D. (2001). The Semantic Web. *Scientific American* 78(3):20–88.
5. Alin Deutsch, Mary Fernandez, and Dan Suciu. (1999). Storing Semistructured Data with STORED, SIGMOD.
6. Daniela Florescu and Donald Kossmann. (1999). Storing and Querying XML Data using an RDBMS, Data Engineering Bulletin,22(3).
7. Jayavel Shanmugasundaram, Kristin Tufte, Gang He, Chun Zhang, David DeWitt, and Jeffrey Naughton. (1999). Relational Databases for Querying XML Documents: Limitations and Opportunities, VLDB.
8. Takeyuki Shimura, Masatoshi Yoshikawa, and Shunsuke Uemura. (1999). Storage and Retrieval of XML Documents using Object-Relational Databases, DEXA.
9. W3 Consortium. (2001). http://www.w3.org/XML
10. http://www.almaden.ibm.com/cs/people/chamberlin/quilt.html
11. http://www.ibiblio.org/xql/
12. http://www.w3.org/TR/NOTE-xml-ql/
13. http://www.w3.org/TR/xquery/
14. http://www.w3.org/TR/xslt20/
15. Chu, W.W., leong, I.T., and Taira, R.K. (1994). A Semantic Modeling Approach for Image Retrieval by Content, VLDB Journal, 3, pp.445-477.
16. Flickner, M. et al. (1995). Query by Image and Video Content: The QBIC System, IEEE Computer, pp.23-32.
17. Ogle, V.E. and Stonebraker, M. (1995). Chabot: Retrieval from a Relational Database of Images, IEEE Computer, pp.40-48.
18. John R. Smith and Shin-Fu Chang. (1996). VisualSEEk : a fully automated content-based image query system, ACM Multimedia 96, pp.87-98.
19. Alex Pentland, Rosalind Picard, and Stan Sclaroff, “Photobook: Tools for Content-Based Manipulation of Image Databases," SPIE PAPER 2185-05 Storage and Retrieval of Image and Video Databases II, San Jose, CA. February 6-10, 1994.
20. Wang, J.Z., Li, J., and Wiederhold, G. (2001). SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries, IEEE TKDE. (http://www-db.stanford.edu/IMAGE/)
21. Jia Li, James Z. Wang, Gio Wiederhold. (2000). IRM: Integrated region matching for image retrieval, Proc. ACM Multimedia, pp.147-156, Los Angeles, ACM.
22. Jia Li, James Z. Wang, Gio Wiederhold. (2000). Classification of textured and non-textured images using region segmentation, Proc. IEEE International Conference on Image Processing (ICIP), Vancouver, BC, Canada, pp.754-757.