A Study of MPEG-7 Document Management System Based Native Embedded Database

Byeongtae Ahn*

Liberal & Arts College, Anyang University, Anyang-shi, South Korea. Email: ahnbt@anyang.ac.kr

Abstract. The embedded database technology can be used for the management of MPEG-7 data under the limited resources of the mobile environment. In this case, XML document clustering methods can be used, but in order to increase efficiency, a clustering method for efficiently storing MPEG-7 documents is required. In this paper, an MPEG-7 document management system that can efficiently store MPEG-7 documents in a mobile terminal is designed. This system used Berkeley DB XML, a dedicated embedded XML database system based on MPEG-7 data clustering method.

Key Words *: Multimedia; Embedded DB;Mpeg-7;Mobile;XML*

1. Introduction

In order to effectively manage large-capacity multimedia content in a mobile environment, it is essential to manage metadata for multimedia content under limited resources. Recently, MPEG-7 has been adopted as an international standard for a method of describing multimedia contents in order to handle multimedia more effectively. MPEG-7 is a standard that defines a method for describing multimedia contents, that is, metadata for multimedia in XML format [1]. MPEG-7 targets the expression method of the contents of multimedia data. Therefore, it makes it possible to extract features from multimedia data or to use the data through a search engine [2]. However, in order to effectively manage MPEG-7 documents in a mobile terminal, a pure embedded XML database technology can be used as an efficient way to store MPEG-7 documents [3]. In this case, it is necessary to support a clustering method suitable for the characteristics of MPEG-7 data.

In this paper, the MPEG-7 document management system (MDMS:MPEG-7 Document Management System) is designed, clustered and saved. MPEG-7 database management can utilize XML document database management technology. For this reason, recent studies on MPEG-7 database management are mainly conducted by applying XML document database management technology [5]. Accordingly, research on the MPEG-7 database management system is largely conducted in two ways. That is, an approach that extends the existing database management system (mainly relational and object-relational database systems) with XML document management functions to support MPEG-7 characteristics [6,7] and an approach that extends the native XML database management system There is [8,9].

However, in general, pure XML database systems define a logical model for XML documents, store and extract XML documents according to the model, and are mainly used for document-oriented applications. These pure XML database systems can be largely divided into text-based or model-based types according to the type of storing XML documents [10]. Text-based pure XML database systems store the entire XML document in text form and provide various database functions for accessing this document. Model-based pure XML database systems store and handle binary models of XML documents like a DOM tree. However, almost all of these are not applied to embedded systems considering the mobile environment. In general, embedded systems have very limited hardware support. Therefore, in mobile devices such as PDA, in order to handle multimedia, a method for managing only necessary multimedia information and metadata information from the server should be considered in consideration of the limited hardware support of the embedded system. Recently, as the need for XML document database management in mobile devices has increased, research on built-in XML DBMS has been active. Accordingly, various types of embedded XML DBMS are being developed. Typical built-in XML DBMS include Tamino [8] and InfonyteDB [11]. Most of these XML documents enable data update at the element level, which is a small unit. However, it does not provide the core functions of traditional DBMS such as transaction management, concurrency control, and recovery management. However, Berkeley DB XML, which supports XML as an embedded database system, was recently developed by Sleepycat [4]. As open software, it was built on Berkeley DB [12], which has already been recognized as a basic function of a database system as an enterprise product. Above all, it supports to have a built-in DBMS by freely selecting functions for various devices. It is provided as a library and directly connected to the client application, so communication between processes or systems is greatly reduced and excellent performance is provided. We developed an MPEG-7 document management system that adopts MPEG-7 data clustering technique to utilize this Berkeley DB XML for MPEG-7 database management.

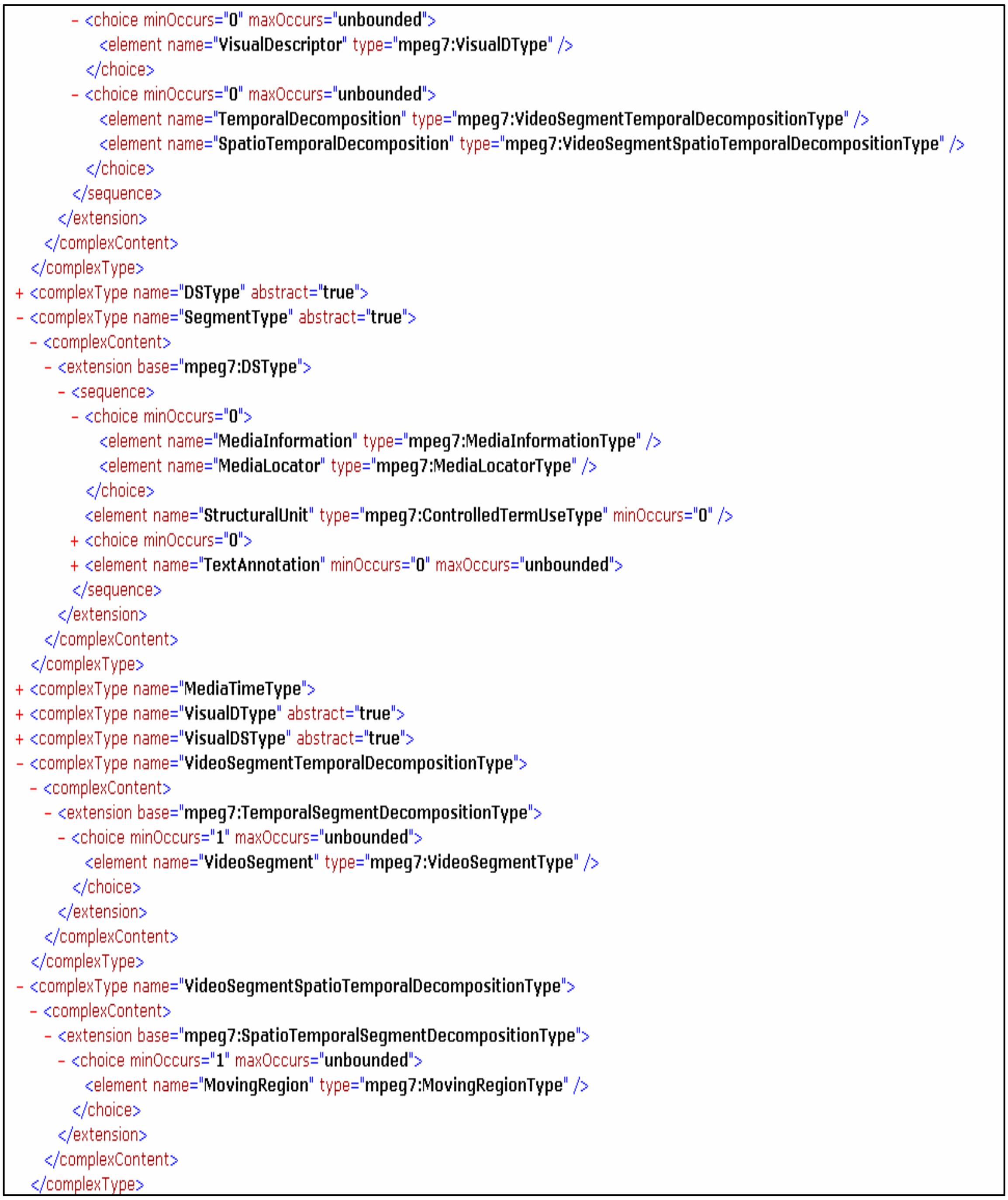
The structure of the thesis is as follows. First, the proposed clustering method is explained in Chapter 2, and the MPEG-7 document management system is designed in Chapter 3. Finally, Chapter 4 presents the conclusion and future tasks.

2. MPEG-7 Data Clustering

The aforementioned Berkeley DB XML was developed as a database system for managing general XML documents. Therefore, in order to utilize this for MPEG-7 document management, although it is an XML document format, a new management method that reflects the unique characteristics of MPEG-7 is required. We proposed a new data clustering method applicable to MPEG-7 documents in [13]. In this paper, this method was adopted for the development of MPEG-7 document management system, and this chapter briefly describes it.

For the storage of MPEG-7 documents, an XML database system is used because MPEG-7 documents are in XML format. In this case, various XML document clustering methods proposed in [14] can be applied. However, since MPEG-7 documents deal with multimedia characteristics rather than general XML documents, a clustering policy that fully considers this should be applied.

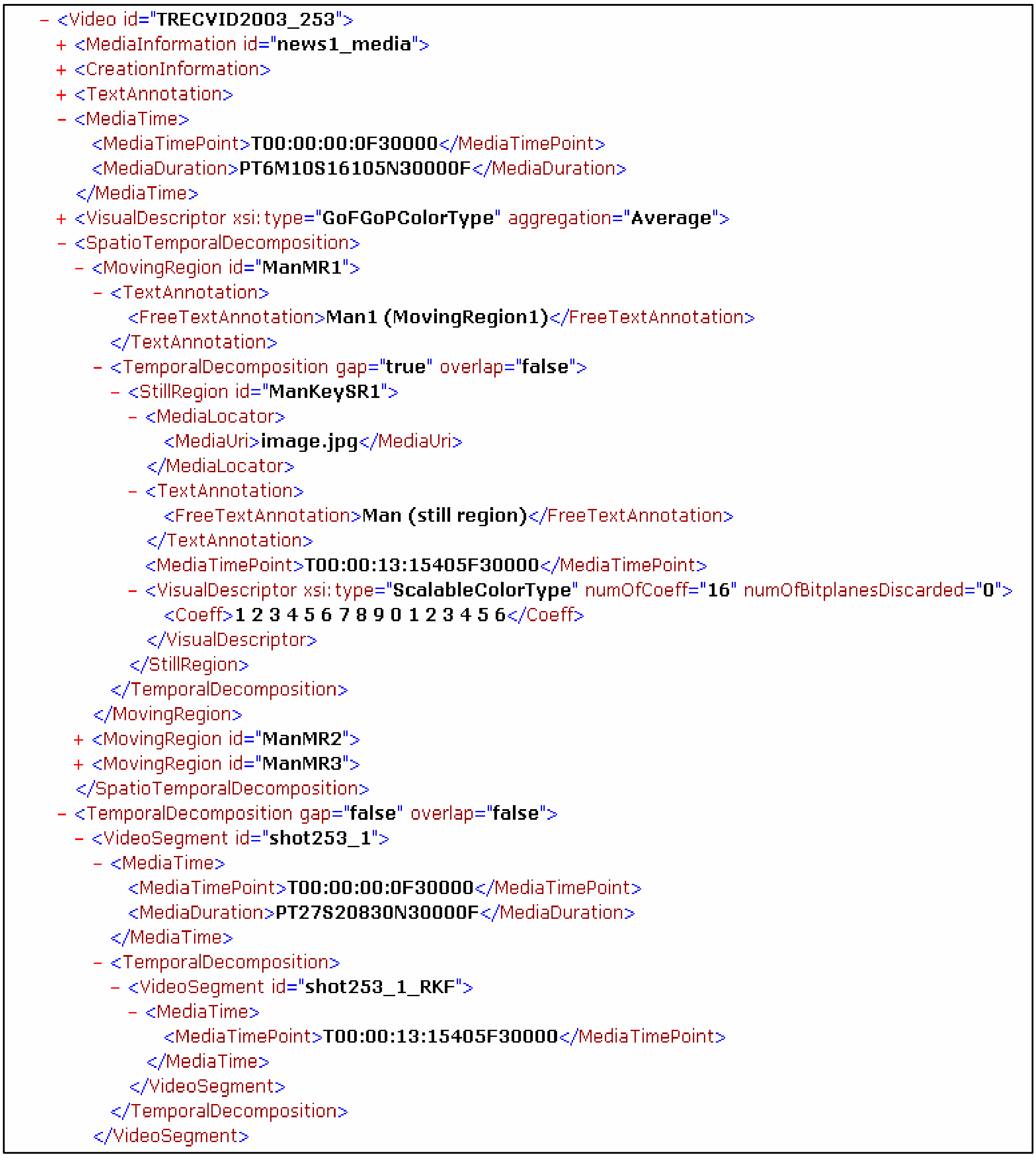
For explanation, we will use the MPEG-7 schema of an actual video-related application (VideoAnnEx) and an example of the MPEG-7 document generated accordingly.



```
        - <choice minOccurs="0" maxOccurs="unbounded">
            <element name="VisualDescriptor" type="mpeg7:VisualDType" />
          </choice>
        - <choice minOccurs="0" maxOccurs="unbounded">
            <element name="TemporalDecomposition" type="mpeg7:VideoSegmentTemporalDecompositionType" />
            <element name="SpatioTemporalDecomposition" type="mpeg7:VideoSegmentSpatioTemporalDecompositionType" />
          </choice>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
+ <complexType name="DSType" abstract="true">
- <complexType name="SegmentType" abstract="true">
  - <complexContent>
    - <extension base="mpeg7:DSType">
      - <sequence>
        - <choice minOccurs="0">
            <element name="MediaInformation" type="mpeg7:MediaInformationType" />
            <element name="MediaLocator" type="mpeg7:MediaLocatorType" />
          </choice>
          <element name="StructuralUnit" type="mpeg7:ControlledTermUseType" minOccurs="0" />
        + <choice minOccurs="0">
        + <element name="TextAnnotation" minOccurs="0" maxOccurs="unbounded">
        </sequence>
      </extension>
    </complexContent>
  </complexType>
+ <complexType name="MediaTimeType">
+ <complexType name="VisualDType" abstract="true">
+ <complexType name="VisualDSType" abstract="true">
- <complexType name="VideoSegmentTemporalDecompositionType">
  - <complexContent>
    - <extension base="mpeg7:TemporalSegmentDecompositionType">
      - <choice minOccurs="1" maxOccurs="unbounded">
          <element name="VideoSegment" type="mpeg7:VideoSegmentType" />
        </choice>
      </extension>
    </complexContent>
  </complexType>
- <complexType name="VideoSegmentSpatioTemporalDecompositionType">
  - <complexContent>
    - <extension base="mpeg7:SpatioTemporalSegmentDecompositionType">
      - <choice minOccurs="1" maxOccurs="unbounded">
          <element name="MovingRegion" type="mpeg7:MovingRegionType" />
        </choice>
      </extension>
    </complexContent>
  </complexType>
```

Figure 1. MPEG-7 Schema

Figure 1 shows the MPEG-7 schema for this. It consists of various description schemes (DS: Description Scheme), descriptors (D: Descriptor), and data types defined in MPEG-7. In other words, MovingRegionType and StillRegionType, which describe each key image, focusing on the VideoSegment type for describing the metadata about video, and MediaInformationType, CreationInformationType, TextAnnotationType, etc., which indicate other video information, were used.



```
- <Video id="TRECVID2003_253">
  + <MediaInformation id="news1_media">
  + <CreationInformation>
  + <TextAnnotation>
  - <MediaTime>
      <MediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
      <MediaDuration>PT6M10S16105N30000F</MediaDuration>
    </MediaTime>
  + <VisualDescriptor xsi:type="GoFGoPColorType" aggregation="Average">
  - <SpatioTemporalDecomposition>
    - <MovingRegion id="ManMR1">
      - <TextAnnotation>
          <FreeTextAnnotation>Man1 (MovingRegion1)</FreeTextAnnotation>
        </TextAnnotation>
      - <TemporalDecomposition gap="true" overlap="false">
        - <StillRegion id="ManKeySR1">
          - <MediaLocator>
              <MediaUri>image.jpg</MediaUri>
            </MediaLocator>
          - <TextAnnotation>
              <FreeTextAnnotation>Man (still region)</FreeTextAnnotation>
            </TextAnnotation>
            <MediaTimePoint>T00:00:13:15405F30000</MediaTimePoint>
          - <VisualDescriptor xsi:type="ScalableColorType" numOfCoeff="16" numOfBitplanesDiscarded="0">
              <Coeff>1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6</Coeff>
            </VisualDescriptor>
          </StillRegion>
        </TemporalDecomposition>
      </MovingRegion>
    + <MovingRegion id="ManMR2">
    + <MovingRegion id="ManMR3">
    </SpatioTemporalDecomposition>
  - <TemporalDecomposition gap="false" overlap="false">
    - <VideoSegment id="shot253_1">
      - <MediaTime>
          <MediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
          <MediaDuration>PT27S20830N30000F</MediaDuration>
        </MediaTime>
      - <TemporalDecomposition>
        - <VideoSegment id="shot253_1_RKF">
          - <MediaTime>
              <MediaTimePoint>T00:00:13:15405F30000</MediaTimePoint>
            </MediaTime>
          </VideoSegment>
        </TemporalDecomposition>
      </VideoSegment>
```

Figure 2. MPEG-7 Instance Document

Figure 2 is an example of an instance document generated according to this MPEG-7 schema. It is largely divided into three video segments, and each segment has a key image. In the MPEG-7 data clustering method, first, a relationship indicating that there is a close relationship between each element element is expressed as a level numeric value from 0 to 99. This was named R-CT (Relationship-ClusTer) attribute, and R-CT attribute was added to the existing MPEG-7 schema for its display. Various MPEG-7 applications were analyzed for level extraction of this R-CT attribute. And based on these analyzes, three major criteria were established. First, when the parent element is the same, the same level is set, and the R-CT attribute value is set to 1. Also, when the parent element is different, it is set to a different level, and the R-CT attribute values are set from 2 to 9. Finally, when the user creates MPEG-7 schema, the user defined level is defined as the user-defined level, and R-CT attribute values are specified from 10 to 99. The redefined MPEG-7 schema constitutes a semantic block that binds elements that are analyzed by the system and determined to be semantically handled together. There are two rules for constructing a semantic block; (Rule S1) When an element has an R-CT attribute value and has child nodes, they are grouped into the same semantic block. (Rule S2) When an element has cardinality and child nodes, they are grouped into the same semantic block.

First, rule S1 is applied and each element is analyzed. Next, rule S2 is applied to find an element having cardinality. For the previous example, R-CT attribute values range from 1 to 3. Figure 3 shows the final form of dividing the actual MPEG-7 schema tree into semantic blocks using semantic block construction rules. Eight semantic blocks were largely composed. And among them, according to the R-CT value, C and D, E and F, and G and H are treated as unified semantic blocks.

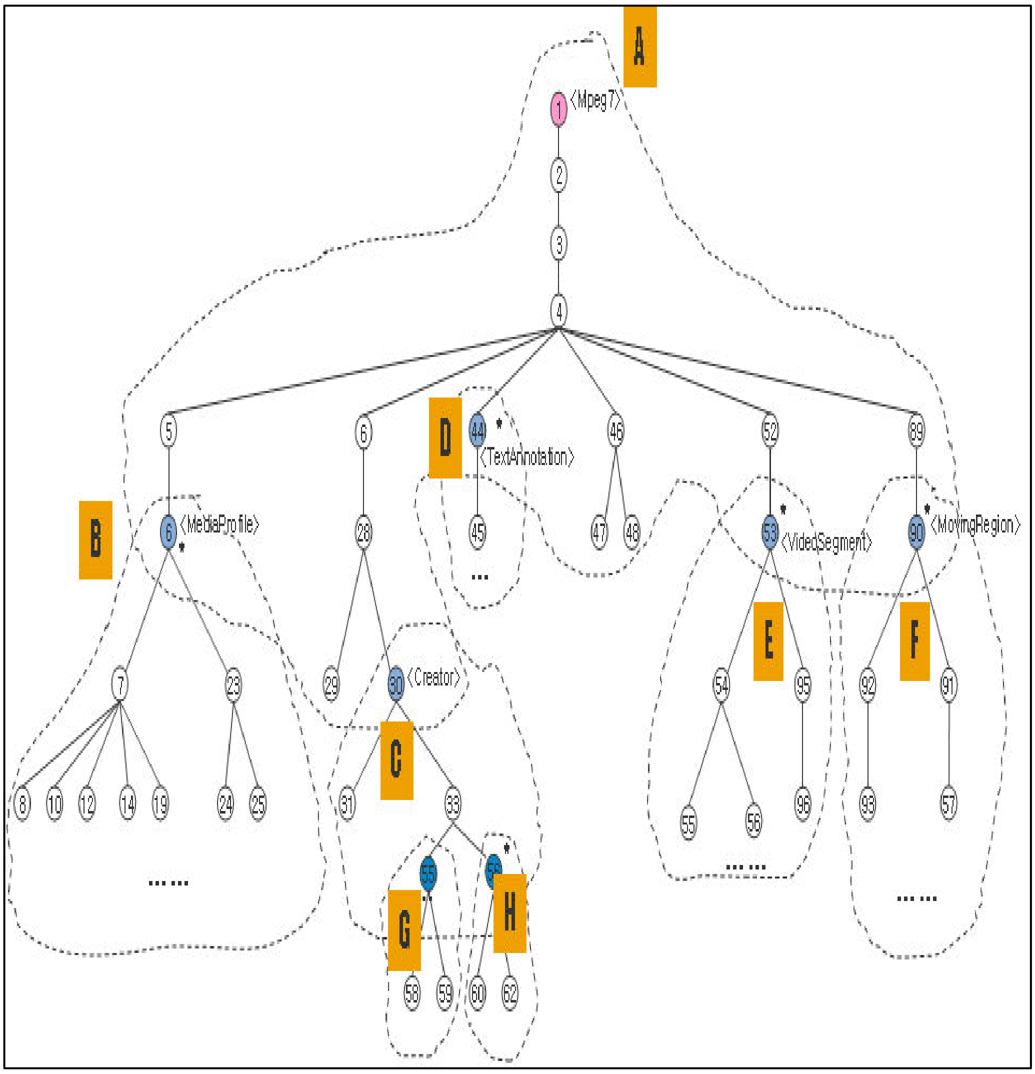


Figure 3. Graph of Semantic Block Schema

Finally, this semantic block tree is traversed by depth-first traversal (DFS) to store document instances generated from the same semantic block so that they are physically located close together. That is, first, it finds the root node and visits its child nodes one by one in a depth-first traversal manner to check whether the corresponding child node is storable within the specified block, so that it is stored in the same block as much as possible. Figure 4 shows the storage format for each final block applied to the previous example.

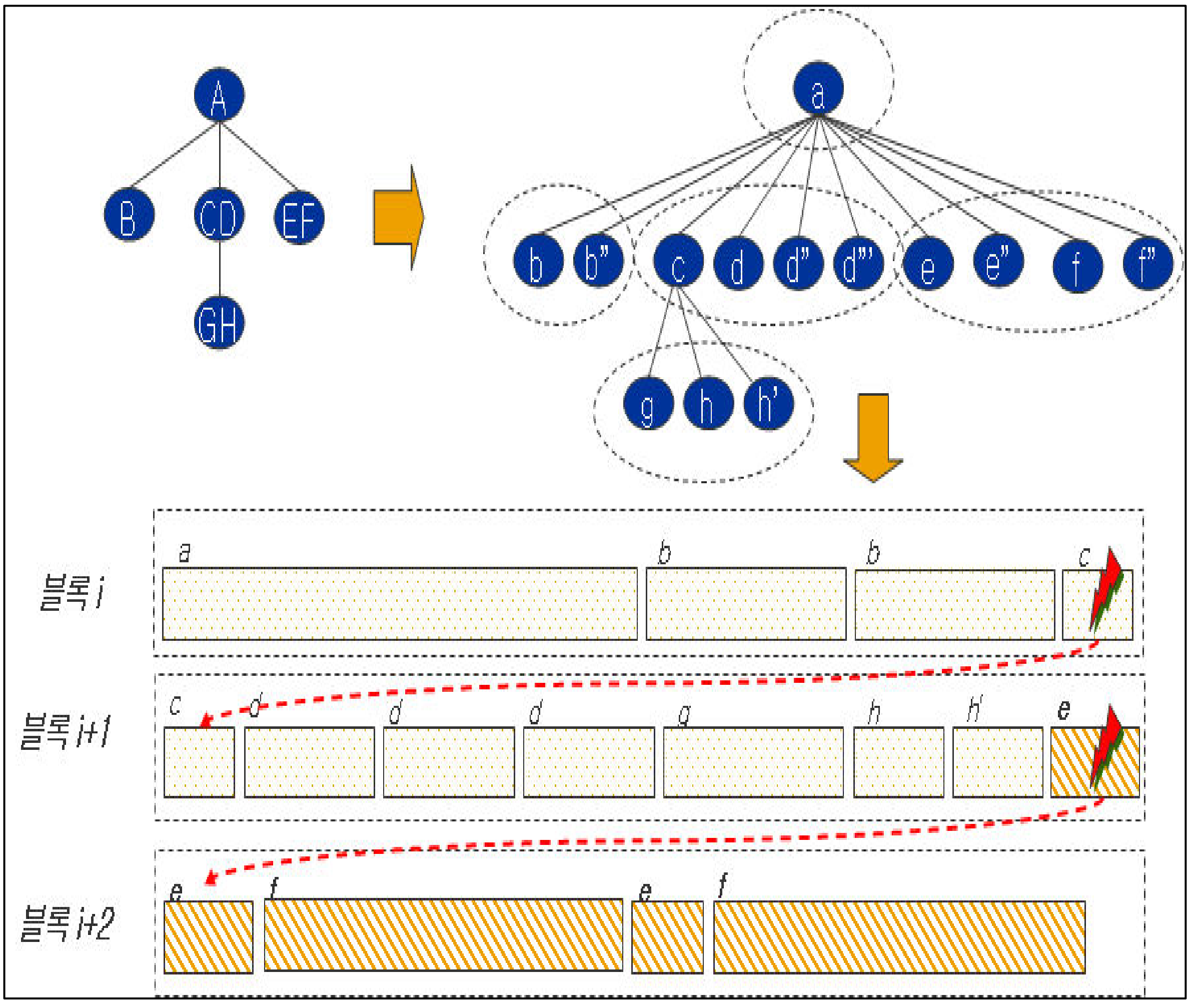


Figure 4. Block Storage Type

Since the MPEG-7 data clustering storage method considers the relationship between MPEG-7 elements, it reduces the storage space in a limited system compared to before clustering and configures the most appropriate semantic block for the application. However, prior work is required to redefine the R-CT value in advance by analyzing various applications.

3. Design of MDMS

Adopting the MPEG-7 clustering method presented in Chapter 2, we developed the MPEG-7 Document Management System (MDMS). In this chapter, we will look at the design of this MDMS.

3.1 The Overall Structure of MDMS

MPEG-7 document management system is mainly composed of two subsystems. That is, the schema management subsystem (SMS: Schema Management Subsystem) that generates the semantic block tree based on the redefined MPEG-7 schema document and the generated semantic block tree are divided into semantic block instances based on the actual MPEG-7 document. It is a document management subsystem (DMS) that stores in Berkeley DB XML through MPEG-7 document clustering method. The schema management subsystem analyzes the MPEG-7 schema, assigns R-CT attribute values, and creates a semantic block tree based on this. The document management subsystem divides the generated semantic block tree into MPEG-7 document instances and stores them in the embedded XML database system.

3.2 Schema Management Subsystem

The schema management subsystem (SMS) is mainly composed of two modules. A schema analyzer (SAM: Schema Analysis Module) that analyzes the schema and analyzes the R-CT attribute value of each element, and a semantic block generator (SGM: SemanticBlock Generation Module) that creates a semantic block based on the analyzed schema and expresses it as a tree)am.

1) Schema Analyzer

The schema analyzer (SAM) consists of two submodules. These are the Schema graph Presentation SubModule (SPSM) and the Element Analysis SubModule (EASM). First, SPSM draws a schema graph by accepting MPEG-7 schema as an input value. In addition, each element of this schema graph assigns R-CT attribute values to interrelated elements in the schema graph through the element analysis submodule and analyzes the cardinality of each element.

2) semantic block generator

The semantic block generator (SGM) consists of two submodules. Semantic Block Generation submodule (SBGM) and Semantic Block Tree Presentation subModule (STPM). The semantic block generation submodule divides the schema graph into semantic blocks between related elements by using the semantic block generation rule through the analysis of the schema graph. And the semantic block divided in this way is made into a semantic block tree between interrelated elements using the semantic block tree expression submodule. Figure 8 shows the processing process of the semantic block generator.

5. Conclusions and Future Research

In this paper, we designed and implemented MDMS (MPEG-7 Document Management System) that can efficiently store MPEG-7 documents even on mobile devices. MDMS is a system that adopts MPEG-7 data clustering method based on built-in XML DB.

The advantages of the MPEG-7 document management system proposed in this paper are as follows. First, it improves query processing speed by supporting clustering based on MPEG-7 schema.Second, by generating the most suitable semantic block for MPEG-7 application, it can be used appropriately for various applications.Third, by using the built-in XML database system, MPEG-7 documents can be efficiently managed even in a mobile terminal with insufficient resources.

The current MPEG-7 document management system was developed as a prototype to support various MPEG-7 applications in a mobile environment. Based on this, it is necessary to add functions and improve the performance of the system so that it is more suitable for the application through the development of various MPEG-7 applications.

References

1. F. Nack and A. Lindsay, “Everything You Want to Know About MPEG-7: Part 1 and Part 2,” IEEE Multimedia, Vol. 6, No. 3, pp. 65-77, July 1999.
2. P. Beek, A. Benitez, J. Heuer, J. Martinez, P. Salembier, J. Smith, and T. Walker, “MPEG-7: Multimedia Description Schemes, ISO/IEC FDIS 15938-5:2001,” International Standard Document, 2001.
3. T. Fiebig, et al., “Anatomy of a Native XML Base Management System,” VLDB, Vol. 11, No. 4, pp. 292-314, 2002.
4. Sleepycat, “BerklyDB XML, Berkeley DB,” http://www.sleepycat.com
5. U. Westermann and W. Klas, "An Analysis of XML Database Solutions for the Management of MPEG-7 Media Descriptions," ACM Computing Surveys, Vol. 35, No. 4, pp. 331-373, December 2003.
6. MicroSoft, " XML for SQL Server 2000," http://www.microsoft.com/technet/prodtechnol/sql/2000/evaluate/xmlsql.mspx
7. Oracle, “Oracle 9i,” http://www.oracle.com
8. Software AG, “Tamino,” http://www1.softwareag.com/corporate/products/tamino/
9. X-Hive, “X-Hive/DB,” http://www.x-hive.com/products/db/index.html
10. K. Eunjung, "The Design of Effective Storage Structure and Index Model in XML Document" KISC, Vol. 29, No. 02, pp. 0110-0118, 2001.
11. Infonyte, “Infonyte DB 3.0,” http://www.infonyte.com/
12. Sleepycat, “Berkeley DB," http://www.sleepycat.com/products/bdb.html
13. A. Guillaume and F. Murtagh, “Clustering of XML Document,” Computer Physics Communications, Vol. 127, pp. 215-227, May 2000.
14. AlphaWorks, "IBM MPEG-7 Annotation Tool," http://www.alphaworks.ibm.com/ tech/videoannex