

MPEG-7 데이터의 효율적인 관리를 위한 클러스터링 방법

안병태[†], 강병수^{**}, 조건화^{***}, 강현석^{****}

요 약

모바일 환경의 제한된 자원 하에서 멀티미디어 데이터를 이용하기 위해서는 XML로 표현되는 MPEG-7 문서의 효율적인 관리 방법이 필요하다. 이 때 XML 문서 클러스터링 방법들을 이용할 수 있겠으나 보다 효율성을 높이기 위해 MPEG-7 문서의 특성을 반영한 새로운 클러스터링 방법이 요구된다. 새로운 클러스터링은 모바일 환경에서 멀티미디어 검색시 질의 처리 속도가 향상되며 다양한 응용에 적합한 문서 저장이 가능하다. 본 논문에서는 대용량의 멀티미디어 데이터를 MPEG-7 문서로 효율적으로 관리하기 위해 MPEG-7 문서의 엘리먼트들 사이에 나타나는 의미 관련성을 이용해 저장할 수 있는 새로운 클러스터링 방법을 제안하고, 이를 기존 방법들과 비교하였다.

Clustering of MPEG-7 Data for Efficient Management

Byeong-Tae Ahn[†], Byeong-Shoo Kang^{**}, Jianhua Diao^{***}, Hyun-Syug Kang^{****}

ABSTRACT

To use multimedia data in restricted resources of mobile environment, any management method of MPEG-7 documents is needed. At this time, some XML clustering methods can be used. But, to improve the performance efficiency better, a new clustering method which uses the characteristics of MPEG-7 documents is needed. A new clustering improved query processing speed at multimedia search and it possible document storage about various application suitably. In this paper, we suggest a new clustering method of MPEG-7 documents for effective management in multimedia data of large capacity, which uses some semantic relationships among elements of MPEG-7 documents. And also we compared it to the existed clustering methods.

Key words: MPEG-7, XML, Data Clustering(데이터 클러스터링), Semantic Block(의미 블록)

1. 서 론

최근 들어 모바일 기기에서 MP3, 동영상 메일, 디지털 멀티미디어 방송(DMB: Digital Multimedia Broadcast) 등 다양한 멀티미디어 응용들이 나타나면서 제한된 자원 하에서 멀티미디어 데이터의 관리는 이제 매우 중요한 연구 주제가 되었다. 그런데 제

한된 자원 하에서 대용량의 멀티미디어 데이터를 효과적으로 관리하기 위해서는 멀티미디어 데이터에 대한 메타데이터의 관리가 필수적이다.

최근에 멀티미디어를 효과적으로 다룰 수 있도록 하기 위해 멀티미디어 데이터 기술 방법에 대한 국제 표준으로 MPEG-7이 채택되었다[1]. MPEG-7은 멀티미디어 데이터에 대한 기술 방법 즉, 멀티미디어에

※ 교신저자(Corresponding Author) : 강현석, 주소 : 경남 진주시 가좌동 900번지(660-701), 전화 : 055)751-5998, FAX : 055)762-1944, E-mail : hskang@gnu.ac.kr
접수일 : 2006년 7월 14일, 완료일 : 2006년 11월 3일

[†] 정회원, 유한대학 경영정보과 교수

E-mail : ahnbt@yuhan.ac.kr

^{**} 준회원, 포항공과대학교 SBD-NCRC 연구원

E-mail : shoo99@empal.com

^{***} 경상대학교 컴퓨터과학부 박사과정

E-mail : diaoxiaoxiao@hotmail.com

^{****} 종신회원, 경상대학교 컴퓨터과학부 교수

※본 연구는 2005 정보통신 기초 기술연구지원 사업의 지원으로 수행되었음

대한 메타데이터를 XML 형태로 정의하는 표준이다 [2]. 따라서 이러한 MPEG-7을 이용하면 멀티미디어 데이터를 효과적으로 다룰 수 있다[3,4].

그런데 이러한 MPEG-7 데이터를 제한된 자원 하에서 보다 효율적으로 다루기 위해서는 적절한 클러스터링 방법이 요구되며 MPEG-7 문서는 XML 형태로 나타낼 수 있다. 따라서 MPEG-7 문서 관리에 기존의 XML 클러스터링 방법들을 활용할 수 있다. 다행히 최근에 XML 문서에 대한 다양한 클러스터링 방법들이 제안되었다[5-8]. 하지만 MPEG-7 문서는 일반 XML 문서와는 다른 특징을 가지고 있다. 즉, MPEG-7 문서의 엘리먼트들 사이에는 다양한 의미 관련성을 가지고 있다. 따라서 MPEG-7 문서를 보다 효율적으로 관리하기 위해서는 이를 반영한 새로운 클러스터링 방법이 요구된다.

XML 클러스터링에 대한 기존 연구들은 크게 문서를 대상으로 한 방법과 스키마를 대상으로 한 방법으로 나눌 수 있다. 여기서는 본 논문과 같이 XML 스키마를 대상으로 하는 것은 물론이고 XML 문서를 대상으로 하는 클러스터링 방법들의 특징 및 문제점을 간략하게 알아본다. 문서 기반 클러스터링은 XML 스키마에 대한 정보 없이 XML 문서만을 입력으로 받아 효과적인 클러스터링 정책을 결정하는 방법이다. 이 방법은 주로 XML 문서를 트리로 인식하여 다룬다.

Guillaume와 Murtagh[5]는 XML 문서 클러스터링 방법을 제안하였다. 이 논문에서는 XML 문서를 클러스터링 하는 문제를 그래프 이론을 이용하여 최적의 파티션(Partition)을 찾는 문제로 간주한다. 즉, 데이터베이스 내의 문서를 각각 하나의 노드로 모델링하고 문서 사이의 링크를 노드 간의 가중치 간선으로 모델링한 그래프에서 최적의 파티션을 찾아냄으로서 데이터베이스 내의 문서들을 여러 개의 클러스터로 나눌 수 있게 한다. 또한 동일한 키워드를 공유하는 문서 간에 키워드 링크를 추가함으로써 클러스터링의 정확도를 높이고 있다. 하지만 엘리먼트들 사이의 다른 의미적 관련성은 사용하지 않고 있다.

Francesca 등[6]은 XML 문서를 나타내는 두 트리의 엘리먼트들 사이에 일치하는(Matching) 공통 구조를 추출하는 방법을 제안하였다. 이들은 이를 통해 트리간의 병합과 단순화 과정을 통해 클러스터의 대표 구조를 추출한다. 그리고 공통 구조의 크기로서

정의된 가중치 값에 기반하여 계층적 클러스터링 알고리즘을 적용한다. 하지만 복잡한 트리의 경우 병합 시간이 많이 걸리는 문제점을 가진다.

스키마 기반 클러스터링은 XML 문서의 틀을 나타내는 스키마를 알 수 있을 때 사용하는 방법이다. 일반적으로 잘 정의된 많은 응용들은 대부분 다루는 XML 문서에 대해 스키마를 갖는다. MPEG-7 역시 잘 정의된 스키마를 가지고 있다[2]. 이러한 스키마는 XML 문서들의 구조를 미리 결정하게 된다. 따라서 이를 이용하면 보다 효과적인 클러스터링 정책을 고안할 수 있다.

Lee 등[7]이 제안한 XClust는 스키마를 나타내는 두 DTD간의 유사도를 측정하여 그 유사도를 기준으로 같은 도메인의 DTD들을 계층적 기법으로 클러스터링 한다. 이 방법에서는 DTD를 트리 형태로 모델링 한 후, DTD를 구성하는 각 노드의 이름, 속성, 그리고 뿌리 노드(root node)로 가는 경로를 비교하여 언어적 및 구조적 유사도를 계산한다. 또한 직계 자손 및 단말 노드끼리의 유사도를 통해 문맥적 유사성을 결정한다. 이렇게 구해진 노드 간 유사도를 통해 DTD 사이의 유사도를 행렬로 계산하고 이를 이용하여 유사도가 높은 DTD 쌍으로부터 점진적으로 DTD 인스턴스 문서를 클러스터링 한다. 이 방식은 각 DTD의 노드 사이의 문맥적인 요소까지 고려하여 높은 정확도를 갖지만 시간이 오래 걸리고 서로 크기가 다른 DTD들 사이에서는 정확한 유사도를 구할 수 없는 문제점을 갖는다.

Xiaofeng Meng 등[8]이 제안한 OrientX는 스키마를 이용하여 다양한 클러스터링을 적용한다. 이는 XML 문서를 서브트리로 나누고 연관성을 고려하여 논리적인 페이지 단위로 묶어 하나의 레코드에 저장하는 방식으로 XML 문서의 서브트리를 단순히 물리적인 페이지 단위로 나누어 저장하는 방식을 개선한 방법이다. 먼저 스키마를 분석해서 의미 블록(Semantic Block)을 형성한다. 이 의미 블록은 관련성 있는 엘리먼트들의 그룹으로서 저장을 위한 논리적 단위가 된다. OrientX에서는 이러한 의미 블록을 얻기 위해서 경험적인 방법을 이용하고 있다. 하지만 단순히 스키마의 문법적인 요소만(예, 카디널리티)을 고려하였기 때문에 복잡한 스키마 구조에서 의미 블록을 구성하기 위한 시간이 많이 소요되고 각 엘리먼트들 간의 문맥적인 유사도를 지원하지 않는다. 본

논문에서는 이 방법을 개선하여 MPEG-7 문서를 효율적으로 관리할 수 있는 새로운 클러스터링 방법을 제안한다. 논문의 구성은 다음과 같다. 2장에서는 제안하는 MPEG-7 클러스터링 방법을 분석하고 3장에서는 제안하는 클러스터링 방법을 기존의 방법들과 비교한다. 끝으로 4장에서는 결론 및 향후 과제를 제안한다.

2. MPEG-7 문서 클러스터링 방법

이 장은 앞 장의 연구들에 대한 분석을 바탕으로 MPEG-7 문서의 관리에 적용할 수 있는 새로운 클러스터링 방법을 제안한다.

2.1 제안의 동기

앞 장에서 소개한 OrientX에서는 클러스터링을 위해 스키마 분석을 한다. 스키마를 통해서 스키마 그래프를 그리는데, 이때 스키마 그래프의 어떤 노드는 의미 블록의 루트(root)가 된다. 이 때 의미 블록의 루트가 되는 경우는 두 가지로 다음과 같다. 첫째 스키마 그래프의 루트일 경우이다. 둘째, 카디널리티로 *나+를 가지면서, 자식 노드를 가질 경우이다. 이 방법에서는 이런 의미 블록의 인스턴스를 하나의 논리적인 레코드로 보고 같은 의미 블록에 속할 경우에 근접시켜(클러스터링하여) 저장한다. [그림 1]은 스키마와 XML 문서의 한 예(Sample)를 나타낸다. [그림 2]는 이에 대해 OrientX의 클러스터링 저장 정책을 적용한 결과를 나타낸다.

우리는 XML 문서 관리에 적용된 이러한 OrientX의 클러스터링 방법을 MPEG-7 문서 관리에 적용하

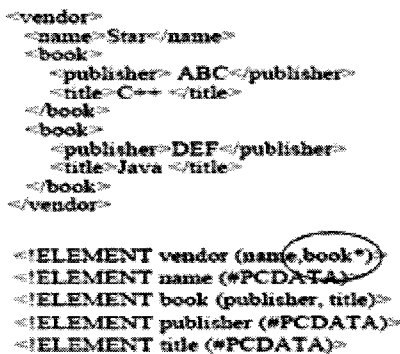


그림 1. 스키마와 XML 문서 예

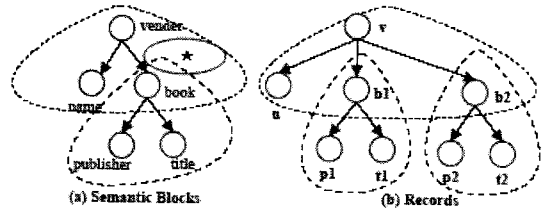


그림 2. OrientX 클러스터링 저장 정책 적용

고자 한다. MPEG-7 문서 또한 XML 문서이므로 OrientX 방법을 적용하여 저장할 수 있기 때문이다. 하지만 OrientX 방법만으로 MPEG-7 문서를 저장했을 때 크게 두 가지 문제점이 발생한다. 첫 번째, MPEG-7 스키마는 그 구조가 매우 복잡하기 때문에 단순히 문법적인 의미만으로 의미 블록을 구성하기 힘들다. 두 번째, MPEG-7 엘리먼트들 간의 연관성을 무시한 저장이다.

이에 본 논문에서는 두 가지 해결책을 제시한다. 첫째로 각 엘리먼트 사이의 유기적인 연관성을 나타내는 R-CT 속성을 도입한다. 둘째로 복잡한 MPEG-7 스키마를 고려해 다양한 의미 블록 생성 규칙을 사용한다. 자세한 방법은 다음 절에서 알아본다.

2.2 클러스터링 적용 절차

우리는 MPEG-7 문서 클러스터링을 위해 3단계 클러스터링 적용 절차를 제안한다.

2.2.1 3단계 적용 절차

우리가 제안하는 클러스터링 방법의 적용은 크게 3단계로 이루어진다. 먼저 1단계는 기존의 MPEG-7 스키마에 관련성 있는 엘리먼트들을 나타내기 위해서 R-CT 속성 값을 추가하여 스키마를 재정의 한다. 2단계는 재정의된 스키마를 바탕으로 의미 블록을 구성한다. 마지막 3단계는 구성된 의미 블록을 통해 클러스터링 알고리즘을 이용하여 클러스터링을 진행한다. [그림 3]은 이러한 3단계 클러스터링 적용 절차를 보여준다. 이에 대한 상세한 설명은 이후의 절들에서 이루어진다.

2.2.2 사용 샘플

제안하는 클러스터링 방법을 설명하기 위해서 실제 비디오 관련 응용의 MPEG-7 스키마와 발생된 MPEG-7 문서를 사용한다. [그림 4]는 사용될 샘플

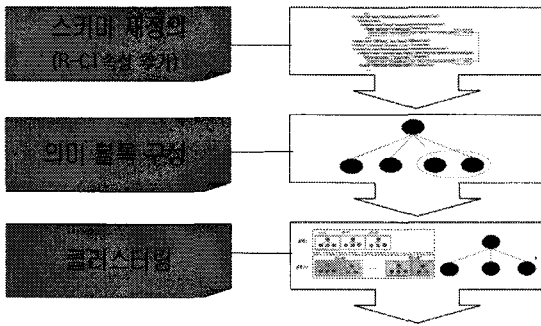


그림 3. MPEG-7 문서 클러스터링 처리 절차

로 비디오 세그먼트 별로 나누어진 간단한 동영상이다. 이 샘플을 위해 IBM사에서 개발된 VideoAnnEx[9] 시스템을 이용하여 비디오를 순차 구조의 세그먼트들로 분류하고 각각의 세그먼트에 대해서 주석을 입력하였다. 뿐만 아니라 자동으로 키 이미지를 추출하여 주석 및 기타 정보를 나타내었다.

[그림 5]는 이에 대한 MPEG-7 스키마이다. 이는 MPEG-7에서 정의한 다양한 기술자(D: Descriptor)와 기술 스킴(DS: Description Scheme), 데이터타입

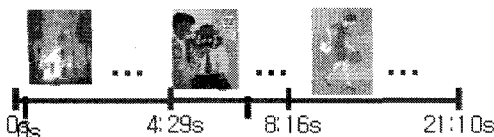


그림 4. 세그먼트별 동영상 샘플

```

<?xml:namespace prefix="mpeg7" base="http://www.itd.wpi.edu/schemas/mpeg7" />
<video id="TRECVID2003_253">
  <creationInformation id="news1_media" />
  <textAnnotations />
  <mediaTime />
  <mediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
  <mediaDuration>PT6M10S1610S</MediaDuration>
  </MediaTime />
  <visualDescriptor xsi:type="ColorColorType" aggregation="Average">
  <spatioTemporalDecomposition />
  <movingRegion id="ManMR1">
    <textAnnotations>
      <FreeTextAnnotations>Man1 (MovingRegion1)</FreeTextAnnotation>
    </TextAnnotations />
    <temporalDecomposition gaps="true" overlap="false">
      <stillRegion id="ManKeySR1">
        <mediaUri>image.jpg</MediaUri />
        <mediaLocator />
        <textAnnotations>
          <FreeTextAnnotations>Man (still region)</FreeTextAnnotation>
        </TextAnnotations />
        <mediaTimePoint>T00:00:13:1540F30000</MediaTimePoint>
        <visualDescriptor xsi:type="ScalableColorType" numOfCoeffs="16" numOfBiplanesDiscarded="0">
          <Coeff>1 2 3 4 5 6 7 8 9 1 2 3 4 5 6</Coeff />
        </VisualDescriptor />
      </StillRegion />
    </TemporalDecomposition />
  </MovingRegion />
  <movingRegion id="ManMR2">
  </MovingRegion id="ManMR2">
  </SpatioTemporalDecomposition />
  <temporalDecomposition gaps="false" overlap="false">
  </VideoSegment id="shot253_1">
  </MediaTime />
  <mediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
  <mediaDuration>PT27S28630N30000F</MediaDuration />
  </MediaTime />
  <temporalDecomposition />
  <videoSegment id="shot253_1_RKF">
  </VideoSegment id="shot253_1_RKF">
  </MediaTime />
  <mediaTimePoint>T00:00:13:1540F30000</MediaTimePoint />
  </MediaTime />
  </VideoSegment />
  </TemporalDecomposition />
  </VideoSegment />
</video>

```

그림 5. MPEG-7 스키마

으로 구성되었다. 즉, 비디오에 대한 메타데이터를 기술하기 위한 VideoSegment 타입을 중심으로 각 키 이미지를 설명하는 MovingRegionType과 StillRegionType, 기타 비디오에 대한 정보를 나타내는 MediaInformationType, CreationInformationType, TextAnnotationType 등이 사용되었다.

[그림 6]은 이 MPEG-7 스키마에 맞게 발생한 인스턴스 문서의 예이다. 크게 3개의 비디오 세그먼트로 나누어지고 각각 키 이미지를 가지고 있다.

2.3 스키마 재정의

제안하는 클러스터링 방법의 첫 단계인 스키마 재정의 단계에서는 응용의 관점에서 문서 구성 요소 엘리먼트들 사이에 밀접한 관계가 있음을 나타내는 관련성을 표시한다. 우리는 이를 R-CT(Relationship-Cluster) 속성이라 명명하였다. 그리고 R-CT 속성의 중요성을 구분하는 레벨 추출을 위해서 대표적인 MPEG-7 응용들인 VideoAnnEx[9]와 TV-Anytime[10]을 분석하였다. 이 MPEG-7 응용들을 분석한 결과, 일반적으로 MPEG-7 응용에서는 서로 관련성이 높은 엘리먼트가 있다는 것을 알 수 있었다. 우리는 이러한 정보를 클러스터링에 활용하기 위해 MPEG-7 문서 클러스터링을 위한 MPEG-7 표준

```

<?xml:namespace prefix="mpeg7" base="http://www.itd.wpi.edu/schemas/mpeg7" />
<video id="TRECVID2003_253">
  <creationInformation id="news1_media" />
  <textAnnotations />
  <mediaTime />
  <mediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
  <mediaDuration>PT6M10S1610S</MediaDuration>
  </MediaTime />
  <visualDescriptor xsi:type="ColorColorType" aggregation="Average">
  <spatioTemporalDecomposition />
  <movingRegion id="ManMR1">
    <textAnnotations>
      <FreeTextAnnotations>Man1 (MovingRegion1)</FreeTextAnnotation>
    </TextAnnotations />
    <temporalDecomposition gaps="true" overlap="false">
      <stillRegion id="ManKeySR1">
        <mediaUri>image.jpg</MediaUri />
        <mediaLocator />
        <textAnnotations>
          <FreeTextAnnotations>Man (still region)</FreeTextAnnotation>
        </TextAnnotations />
        <mediaTimePoint>T00:00:13:1540F30000</MediaTimePoint>
        <visualDescriptor xsi:type="ScalableColorType" numOfCoeffs="16" numOfBiplanesDiscarded="0">
          <Coeff>1 2 3 4 5 6 7 8 9 1 2 3 4 5 6</Coeff />
        </VisualDescriptor />
      </StillRegion />
    </TemporalDecomposition />
  </MovingRegion />
  <movingRegion id="ManMR2">
  </MovingRegion id="ManMR2">
  </SpatioTemporalDecomposition />
  <temporalDecomposition gaps="false" overlap="false">
  </VideoSegment id="shot253_1">
  </MediaTime />
  <mediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
  <mediaDuration>PT27S28630N30000F</MediaDuration />
  </MediaTime />
  <temporalDecomposition />
  <videoSegment id="shot253_1_RKF">
  </VideoSegment id="shot253_1_RKF">
  </MediaTime />
  <mediaTimePoint>T00:00:13:1540F30000</MediaTimePoint />
  </MediaTime />
  </VideoSegment />
  </TemporalDecomposition />
  </VideoSegment />
</video>

```

그림 6. MPEG-7 인스턴스 문서

스키마에 R-CT 속성을 다양한 레벨로 추가한다. 이는 [표 1]과 같이 그 특성에 따라 크게 3가지 레벨로 분류될 수 있다.

[그림 7]은 [그림 5]의 일부에 이러한 R-CT 속성을 추가해서 재정의한 MPEG-7 스키마 문서의 한 예이다.

2.4 의미 블록 구성

두 번째 단계인 의미 블록 구성에서는 재정의된 MPEG-7 스키마를 분석하여 의미 블록을 구성한다. 먼저 스키마 그래프를 그리고, 이를 의미 블록 구성 규칙을 통해 분석한 후 의미 블록을 구성한다.

2.4.1 의미 블록 구성 규칙

MPEG-7 스키마를 바탕으로 그래프를 그릴 수 있

다. 이 그래프를 이용하여 의미 블록 생성 규칙을 적용한다. 본 연구에서는 [표 2]와 같이 두 가지 의미 블록 구성 규칙을 사용한다.

2.4.2 스키마 그래프 분석

앞의 두 가지 규칙을 이용하여 우리는 스키마 그래프를 분석한다. [알고리즘 1]은 이를 수행하는 것으로 스키마 그래프의 각 엘리먼트를 분석해서 R-CT 속성 값과 자식 노드를 찾아 의미 블록의 구성을 위한 트리를 생성한다.

[그림 8]은 각 규칙들이 해당 엘리먼트에서 적용되는 것을 나타내고 있다. 먼저 규칙 S1을 적용해 각 엘리먼트를 분석한다. 앞의 샘플에 대한 일부인 아래 예제에서는 R-CT 속성 값이 1~3 까지를 가지고 있음을 나타낸다. 다음으로 규칙 S2를 적용하여 카디

표 1. 레벨에 따른 R-CT 속성값

레벨	의미	R-CT 속성값
같은 레벨	부모 엘리먼트가 같은 경우	1
다른 레벨	부모 엘리먼트가 다른 경우	2 - 9
사용자 정의 레벨	기타 사용자가 MPEG-7 스키마를 생성할 때 별도로 정의한 경우	10 - 99

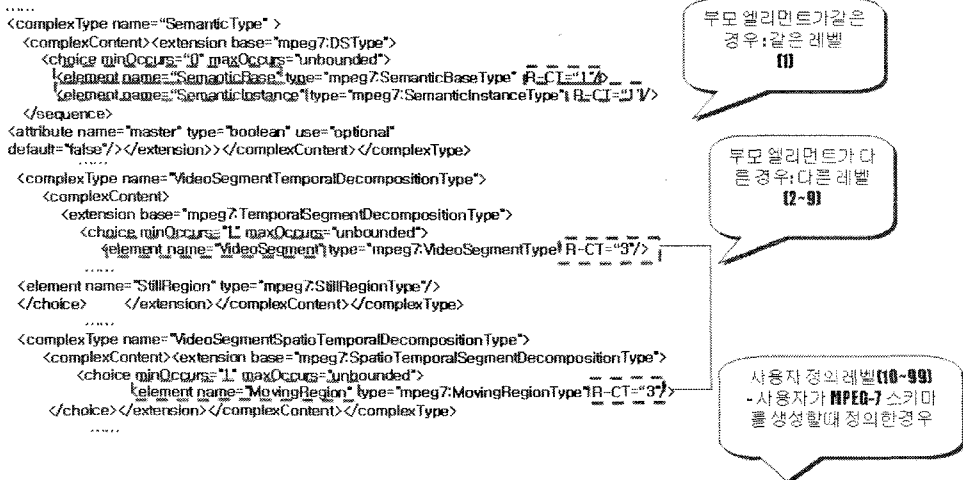


그림 7. R-CT 속성이 추가된 재정의 MPEG-7 스키마

표 2. 의미 블록 구성 규칙

규칙	의미	비고
S1	엘리먼트가 R-CT 속성 값을 가지고, 자식 노드를 가질 때	R-CT 값이 작을수록 우선순위가 높음
S2	엘리먼트가 카디널리티 "*" 또는 "+" 을 가지고, 자식 노드를 가질 때	S1 규칙 적용 후, S2 규칙 적용 가능

알고리즘 1. 스키마 그래프 분석 알고리즘

```

// 스키마 파일을 분석한다.
protected Document parse(String schemaFile) throws Exception {
    SAXReader reader = new SAXReader();
    reader.setXMLFilter(new TrimXMLFilter());
    return reader.read(schemaFile);
}

public Element haveRCT(Node node){ // R-CT 속성값의 분석
    HashMap h = new HashMap();
    int curRCTPoint;
    if (RCTPoint!=null){
        haveChild(node);
    }else{
        h.put(curRCTPoint,curElement);
    }
}

public Element haveChild(Node node){ // 자식 노드의 존재 여부
    if (havechildPoint!=null){
        createSemanticBlock(node);
    }
}

public void createSemanticBlock(Node node){ // 스키마 그래프로부터
    try{
        // 의미 블록 생성
        XmlDocument doc = new XmlDocument();
        Node virRoot =doc.createElement(node+1);
        doc.appendChild(virRoot); //의미 블록의 가상 루트
        Node subChild = doc.createElement(childNode);
        return doc;
    } catch (Exception ex){
        ...
    }
}
    
```

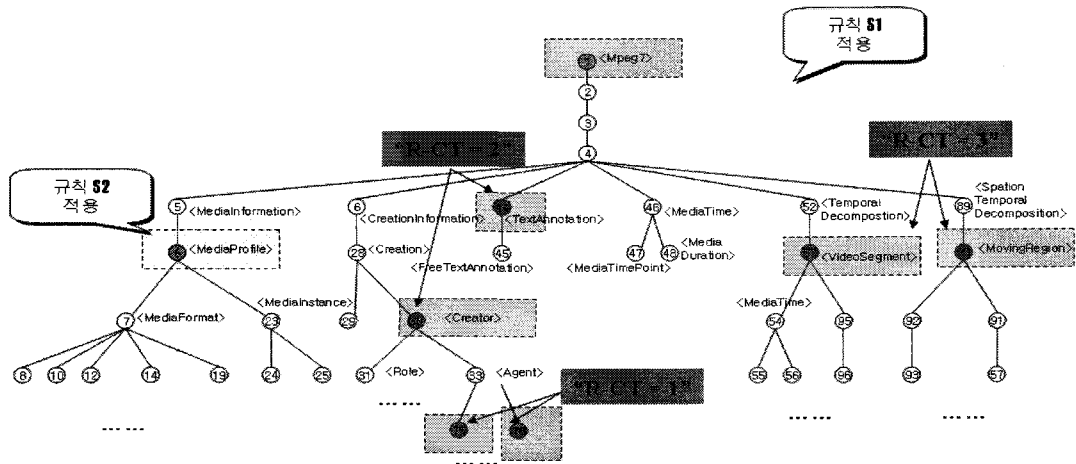


그림 8. 규칙을 적용한 스키마 그래프

널리티를 가지는 엘리먼트를 찾는다. 이 때 규칙 S1을 적용한 엘리먼트 중 카디널리티를 가지는 것은 S2가 적용된 것을 볼 수 있다.

2.4.3 의미 블록 구성 결과

[그림 9]는 의미 블록 구성 규칙들을 이용하여 실제 MPEG-7 스키마 그래프를 의미 블록들로 나눈

최종 형태이다. 크게 8개의 의미 블록이 구성되었다.

2.4.4 의미 블록 트리 간소화

다음은 구성된 의미 블록을 간소화시키는 작업을 통해서 간소화시킨다. 즉, 같은 의미 블록에 소속된 노드들을 결합하여 하나의 노드 형태로 나타낸다. [그림 10]은 의미 블록 트리의 간소화된 형태이다.

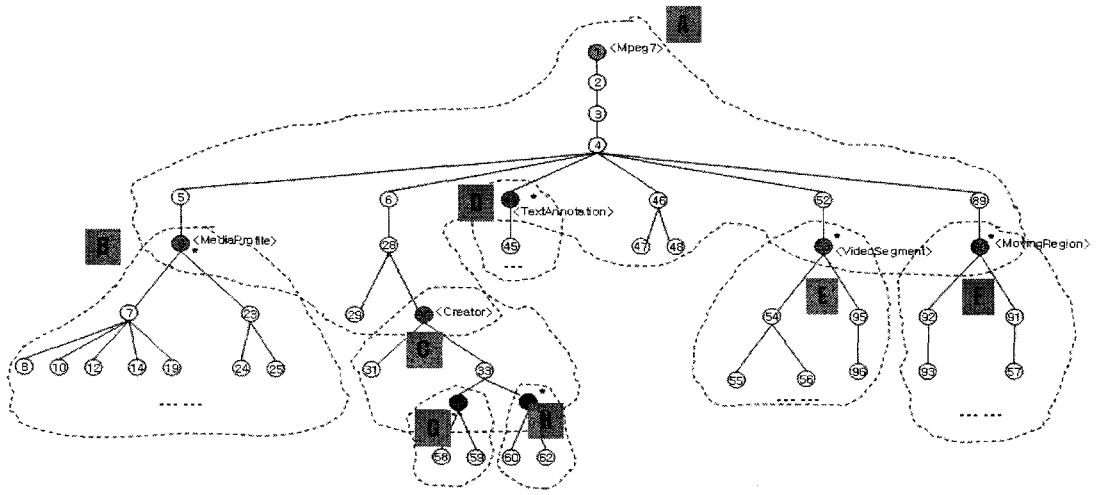


그림 9. 의미 블록으로 나누어진 스키마 그래프 결과

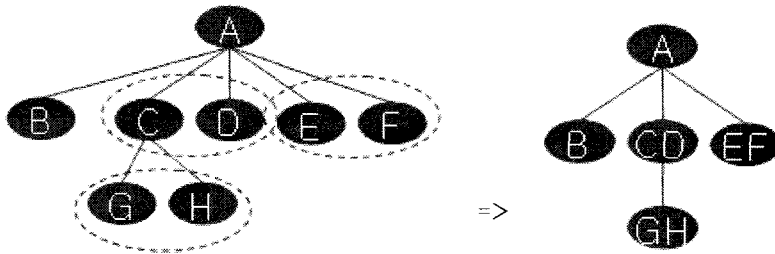


그림 10. 의미 블록 트리 간소화

2.5 클러스터링 기법 적용

앞 단계에서 간소화되어 넘어온 의미 블록 트리에 대해 마지막 세 번째 단계에서는 OrientX에서와 유사한 방법으로 클러스터링 기법을 적용한다.

2.5.1 적용 알고리즘

의미 블록 트리에 대해 DFS(깊이 우선 순회) 순회를 기반으로 의미 블록에 의해 발생된 문서 인스턴스를 같은 블록 내에 가깝게 위치시켜 저장한다. [알고리즘 2]는 이를 수행하는 것으로 먼저 루트 노드를 찾고 그에 따르는 자식 노드를 DFS 순회 방식으로 하나씩 방문하여 해당 자식 노드가 지정된 블록 내에 저장 가능한지 확인한다.

2.5.2 블록 저장 형태

앞의 클러스터링 알고리즘을 실행했을 때 실제 블록의 할당은 블록 크기에 따라 결정된다. 즉, 간소화

된 의미 블록 트리를 바탕으로 생성된 인스턴스들이 최대한 같은 블록 내에 저장될 수 있도록 한다. [그림 11]은 앞의 예에 대해 적용된 최종 블록별 저장 형태를 보이고 있다.

2.6 구현 및 처리과정

앞에서 제시한 알고리즘을 기반으로 실제 시스템 상에서의 구현 및 결과를 나타낸다. [그림 12]는 MPEG-7 클러스터링 기법을 적용한 실제 구현 중 일부를 나타낸 것이다. 인터페이스는 비디오에 대한 특징을 추출하고, 메타데이터를 입력하기 위한 인터페이스이다. 멀티미디어 데이터를 순차 구조의 세그먼트로 분리하고 각 세그먼트마다 주석 정보를 입력하는 주석 인터페이스이다. 세그먼트 분리의 효율성을 위해서 앞뒤 3초 간격으로 비디오 탐색이 가능하며 정확한 키 이미지 추출을 위해 1/24초 단위로 비디오 탐색이 가능하도록 하였다.

알고리즘 2. 클러스터링 알고리즘

```

//깊이 우선 탐색 알고리즘을 사용
//의미 블록에 의해 발생된 인스턴스를 같은 블록 내에 가깝게 위치시킨다.
//루트 노드를 찾는다.
Node subRoot = SEDoc.getRootElement;
Public void visitChild(Node subRoot){ // 자식 노드들을 하나씩 방문한다.
    If (subRoot!=null){
        BlockInset (subRoot.displayNode());
        visitChild (subRoot.leftChild);
        visitChild (subRoot.rightChild);
    }
}
Public void BlockInsert(Node curNode) {
    //블록 사이트 지정
    blockSite = 4K; // 이는 변경될 수 있다.
    curBlock = TotalBlock.getCurBlock; // 해당 노드를 블록에 넣는다.
    while (curBlock.emptySize > 0){
        if (curNode.getSize < curBlock.emptySize) {
            insertBlock(curNode, curBlock.blockNumber);
        } else {
            curBlock.blockNumber ++ // 다음 블록에 저장하기 위해서
        }
    }
}
    
```

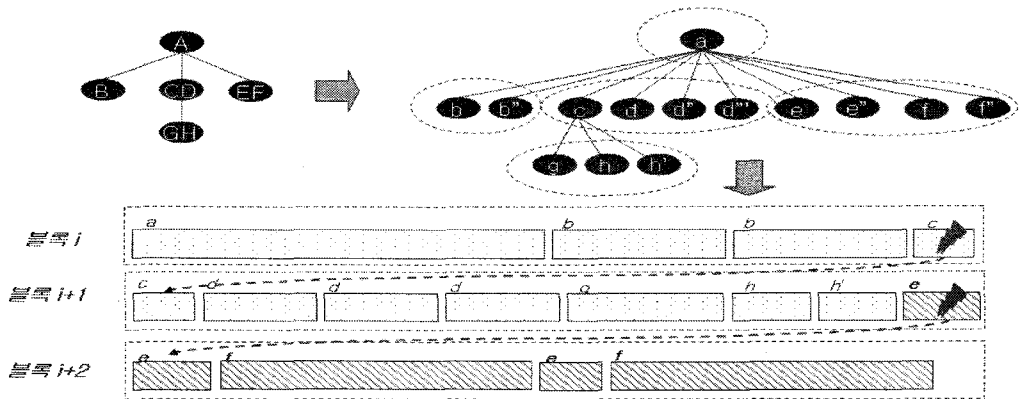


그림 11. 블록별 저장 형태

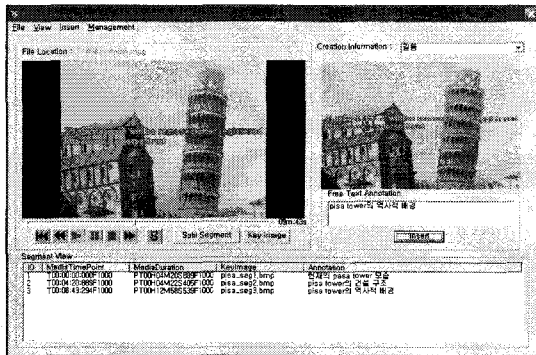


그림 12. Annotation 인터페이스

[그림 13]은 의미 인터페이스를 나타낸 것이다. 화면에서 Semantic Type List에는 비디오에서 사용할 SemanticTime, SemanticPlace, Event, Object, Agent 등의 데이터를 입력할 수 있으며, 리스트의 데이터를 더블클릭하면 해당 데이터들이 Semantic View에 입력된다. 이를 통해서 비디오의 세그먼트에 대한 의미 정보를 추가할 수 있게 된다. [그림 14]는 제시한 알고리즘을 기반으로 실제 시스템 상에서의 과정을 나타내고 있다. 기본적인 과정은 ① XQuery 생성, ② 분류 콘텐츠 너 검색, ③ MPEG-7 문서 검색, ④ 키 이미지 정보

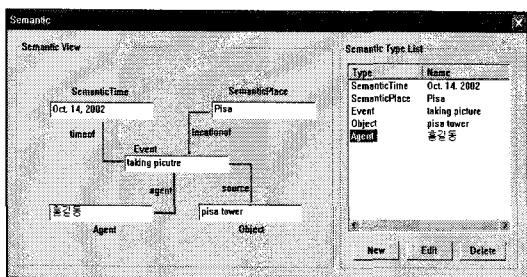


그림 13. 의미 인터페이스

검색, ⑤ 결과 전송의 5단계 과정을 거쳐 처리된다. 각각에 대해 알아본다.

(1) XQuery 생성

검색 요청이 서버 측에 들어오게 되면 XQuery 생성 모듈을 통해서 검색 요청 데이터를 이용하여 XQuery를 생성한다.

(2) 분류 컨테이너 검색

검색 요청 데이터 중 분류 정보가 있다면 버클리

DB XML의 분류 컨테이너를 검색한다. 분류 컨테이너의 최상위 노드의 XML 문서를 이용하여 하위 노드의 XML 문서를 검색한다. XML 문서의 <subset> 요소의 id 속성값이 'grouping'이면, <location> 요소가 가리키는 XML 문서를 계속 검색한다. 그리고 id 속성값이 'MPEG'이면, <location> 요소가 가리키는 컨테이너의 위치를 반환한다.

(3) MPEG-7 문서 검색

[그림 12]에서와 같이 검색된 컨테이너에 XQuery를 실행하여 조건을 만족하는 MPEG-7 문서를 찾는다.

(4) 키 이미지 정보 검색

검색 조건을 만족하는 MPEG-7 문서를 발견하면 <Segment> 요소의 id 속성값을 이용하여 버클리 DB에 저장되어 있는 키 이미지의 위치 정보를 검색한다.

(5) 결과 전송

XQuery의 결과와 키 이미지를 클라이언트 측으로 전송한다. 비디오 데이터는 사용자가 검색 결과를

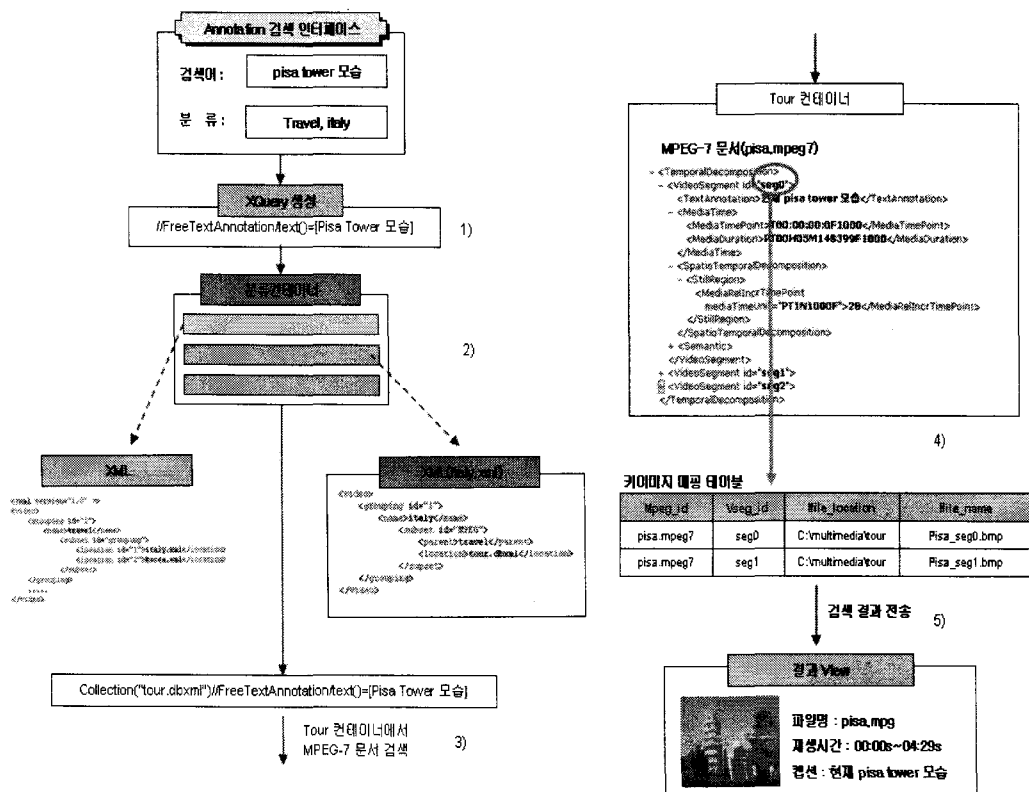


그림 14. Video searching process

확인한 후, 전송을 요청하게 되면 서버에서 클라이언트 측으로 전송한다.

3. 비교 분석

이 장에서는 제안하는 MPEG-7 문서 클러스터링 저장 방법을 기존 방법들과 비교 분석한다. 우리는 별도의 클러스터링 정책을 사용하지 않는 Berkeley DB XML의 저장 방법과 OrientX 클러스터링 저장 방법을 비교한다.

샘플로 비디오 관련 응용의 MPEG-7 XML 문서를 이용하였다. 비교 대상은 원하는 결과의 동영상 메타데이터를 검색하여 그 결과를 얻는데 걸리는 시간을 비교하였다. 스키마 정보는 미리 알고 있으며, 질의 또한 유효하다고 가정한다. 비교 질의는 3가지로 다음과 같다.

- 1) Q1 : 해당 비디오의 제목을 검색할 경우(point query)

```
for $a in doc("video@MPEG7_Video.xml")//Title
return <Video_Title>
    { $a }
</Video_Title>
```

- 2) Q2 : 해당 비디오의 요약 정보를 검색할 경우(join)

```
for $a in doc("video@MPEG7_Video.xml")//SpatioTemporalDecomposition/MovingRegion/TemporalDecomposition/StillRegion
return <summary>
    { $a/MediaTimePoint }
    (for $b in $a/TextAnnotation
    return <KeyImage>
        { $b }
    </KeyImage> }
</summary>
```

- 3) Q3 : 검색된 요약 정보를 바탕으로 원하는 비디오 세그먼트 메타정보를 검색할 경우(ordered

```
access query)
for $b in doc("video@MPEG7_Video.xml")//SpatioTemporalDecomposition/MovingRegion/TemporalDecomposition/StillRegion/
return <Summary>
    <title> { $b/TextAnnotation } </title>
    (for $a in doc("video@MPEG7_Video.xml")//VideoSegment/MediaTime[MediaTimePoint = $b/MediaTimePoint]
    return <SegmentInfo>
        { $a/price }
    </SegmentInfo> }
</Summary>
```

[그림 15]는 이 3가지 종류의 질의를 처리하는 평균 처리시간을 비교한 것이다. 특정 레코드를 찾아오는 경우에는 비슷하지만(Q1), 비디오에 대한 요약 정보(Join연산)라든지(Q2) 요약 정보를 바탕으로 비디오 세그먼트를 찾는 경우에는(Q3) 처리 시간에 큰 차이를 보이고 있다.

제안하는 MPEG-7 문서 클러스터링 저장 방법은 재정의된 스키마를 그래프로 나타내고 이것을 관련 엘리먼트를 중심으로 의미 블록 트리로 구성하여 저장하는 방법이다. 따라서 최초 저장 시에 의미 블록 트리를 구성하기 위한 처리 시간이 추가로 필요하며, 이는 문서가 복잡할수록 느려진다. 하지만 저장된 후 해당 멀티미디어에 대한 MPEG-7 문서의 검색에는 빠른 속도의 질의 처리가 가능하다. 또한 동일한 질의

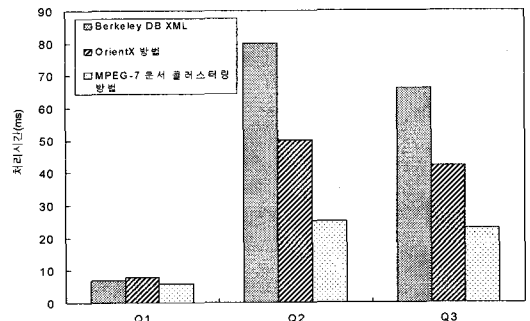


그림 15. 질의 처리 시간 비교

터미디어에 대한 잦은 질의 처리 요구에도 빠르게 대응할 수 있다.

4. 결론 및 향후 과제

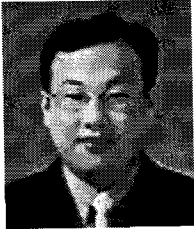
본 논문에서는 MPEG-7 문서의 효율적인 저장을 위해 XML 스키마 기반의 새로운 클러스터링 방법을 제안하였다.

본 논문에서 제안하는 클러스터링 방법의 이점은 다음과 같다. 첫째, MPEG-7 스키마를 기반으로 클러스터링을 지원함으로써 질의 처리 속도를 향상 시킨다. 둘째, MPEG-7 응용에 가장 적합한 의미 블록을 생성함으로써 다양한 응용에 적합한 MPEG-7 문서 저장 시스템의 구현에 이용할 수 있다. 이들을 바탕으로 기존의 시스템들과 달리 모바일 단말기에서 멀티미디어를 효과적으로 검색하고 관리하는 것이 가능하다. 그리고 많은 양의 MPEG-7 문서 검색에 대해 일관성 있는 처리가 가능하다.

본 논문에서는 다양한 MPEG-7 응용을 바탕으로 생성된 의미 블록 생성 규칙 두 가지를 제안하였다. 하지만 이 부분에서 실제 의미에 보다 적합한 생성 규칙을 더 필요로 한다. 따라서 완전한 의미 블록 생성을 지원하기 위해서는 다양한 R-CT 속성의 설정이 가능하도록 추가적인 연구가 필요하다. 이때 점점 복잡해지고 다양해지는 MPEG-7 응용들에 맞출 필요가 있다.

참 고 문 헌

- [1] F. Nack and A. Lindsay, "Everything You Want to Know About MPEG-7: Part1 and Part2," *IEEE Multimedia*, Vol. 6, No. 3, pp. 65-77, July 1999.
- [2] J. Martinez, "Overview of the MPEG-7 Standard, ISO/IEC JTC1/SC29/WG11," N4980 (Klagenfurt Meeting), July 2002, <http://www.chiariglione.org/mpeg/>
- [3] H. Kosch, "Distributed Multimedia Database Technologies Supported by MPEG-7 and MPEG-21," *Auerbach Publications*, 2004.
- [4] 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.
- [5] A. Guillaume and F. Murtagh, "Clustering of XML Document," *Computer Physics Communications*, Vol. 127, Issue 2-3, pp. 215-227, May 2000.
- [6] F. Francesca, G. Gordano, R. Ortale, and A. Tagarelli, "Distance-based Clustering of XML Documents," *Proc. First Int'l Workshop on Mining Graphs, Tree and Sequence*, pp. 75-78, Sept. 2003.
- [7] M. Lee, L. Yang, W. Hsu, and X. Yang, "XClust: Clustering XML Schemas for Effective Integration," *Proc. 11th Int'l conf. Information and Knowledge Management*, pp. 151-158, 2001.
- [8] X. Meng, Y. Wang, D. Luo, S. Lu, J. An, Y. Chen, Y. Jiang, and J. Ou, "OrientX: A Native XML Database System," *National Database Conference*, Oct. 2003.
- [9] AlphaWorks, "IBM MPEG-7 Annotation Tool," <http://www.alphaworks.ibm.com/tech/video-annex>
- [10] TV-Anytime Forum, "TV-Anytime," <http://www.tv-anytime.org>.



안 병 태

1999년 2월 국민대학교 컴퓨터과
학부 졸업
2001년 2월 경남대학교 컴퓨터공
학부 석사
2006년 8월 국립경상대학교 컴퓨
터과학부 박사
2006년 9월~현재 유한대학 경영

정보과 교수

관심분야 : Embedded DB, XML, MPEG-7, Mobile



조 건 화

2004년 7월 안산과학대학 컴퓨터
과학과 졸업
2006년 8월 경상대학교 컴퓨터과
학부 석사
2006년 3월~현재 경상대학교 컴
퓨터과학부 박사과정
관심분야 : Embedded DB, XML,

MPEG-7, Mobile



강 병 수

2006년 2월 경상대학교 컴퓨터
과학부 졸업
2006년~현재 포항공과대학교
SBD-NCRC 연구원
관심분야 : Embedded XML DB,
데이터마이닝



강 현 석

1981년 2월 동국대학교 전자계산
학과 졸업
1983년 2월 서울대학교 전산학과
석사
1986년 2월 서울대학교 전산학과
박사
1993년~현재 경상대학교 컴퓨
터과학부 교수

관심분야 : Embedded DB, XML, MPEG-7