

XML을 이용한 지능형 이미지 검색 시스템

홍성용[†], 나연묵^{**}

요 약

인터넷 기술의 급속한 발전으로 인하여 인터넷 사용자의 수와 인터넷상의 멀티미디어 정보의 양이 계속 증가하고 있다. 최근의 e-비즈니스나 쇼핑몰 사이트에서는 많은 양의 이미지 정보를 취급하고 있으며, 이로 인하여 이미지에 대한 효율적인 내용 검색의 필요성이 대두되고 있다. 본 논문에서는 XML 기술을 이용하여 웹상의 이미지를 지능적으로 검색할 수 있는 시스템을 제안한다. 상품 카탈로그와 같은 복잡하고 다중 객체를 보유하고 있는 이미지에 대하여 객체 기반 내용 검색을 수행할 수 있도록 지역 특징, 전역 특징, 의미 등의 메타데이터를 표현하는 다계층 메타데이터 구조를 제안한다. 또한, 이미지에 대한 의미 기반 검색 및 내용 기반 검색을 수행할 수 있도록 이러한 메타데이터를 저장하기 위한 XML-Schema를 설계하고 각 메타데이터를 XML 문서 형태로 표현하는 방법을 보인다. 또한, XSLT를 이용하여 이미지에 대한 검색 결과를 웹 브라우저나 모바일 브라우저와 같은 다양한 사용자 환경에 보여줄 수 있도록 자동 변환하는 방법을 제시한다. 본 논문에서 제시한 방법은 이미지에 대한 메타데이터를 XML 형태로 표현하므로 XML을 지원하는 상용 시스템을 이용하여 용이하게 시스템을 구현할 수 있으며, 이미지 메타데이터의 시스템간 공유도, 검색 질의에 대한 정확성, 사용자의 검색 만족도를 증가시킬 수 있다.

An Intelligent Image Retrieval System using XML

Sungyong Hong[†], Yunmook Nah^{**}

ABSTRACT

With the rapid development of internet technology, the number of internet users and the amount of multimedia 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 content-based image retrieval efficiently on such image data. This paper proposes an intelligent image retrieval system, which adopts XML technology. To support object-based content retrieval on product catalog images containing multiple objects, we describe a multi-level metadata structure which represents 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. It can be utilized to enable efficient image metadata sharing between systems, and it will contribute in improving the retrieval correctness and the user's satisfaction on content-based e-catalog image retrieval.

Key words: Multimedia Information(멀티미디어 정보), Image Retrieval(영상 검색), Metadata(메타데이터), XML, XML-Schema, XSLT

※ 교신저자(Corresponding Author): 홍성용, 주소: 서울 용산구 한남동 산8번지(140-714), 전화: (02)799-1080, FAX: 02)790-3089, E-mail: syhong@dku.edu

접수일: 2003년 4월 16일, 완료일: 2003년 7월 8일

[†] 준회원, 단국대학교 컴퓨터공학과 박사수료

^{**} 정회원, 단국대학교 컴퓨터공학과 부교수

(E-mail: ymnah@dku.edu)

※ 이 연구는 2003학년도 단국대학교 대학연구비의 지원으로 연구되었음.

1. 서 론

최근 멀티미디어 분야에 발전이 급속도로 이루어지고 있으며, 사용량도 많아지고 있다. 따라서, 멀티미디어의 활용은 특정 분야가 아닌 다양한 산업 분야와 IT분야에서 쉽게 접해 볼 수가 있다. 특히 최근에 IT산업의 발전으로 인해 유선 인터넷이나 무선 인터넷

넷에서의 멀티미디어 활용도는 급격히 증가하고 있다. 유·무선 인터넷의 문화가 활발하게 증가하고 있으므로, 이에 따른 다양한 형태의 정보 서비스와 변환 방법 등이 활발히 연구되어지고 있다. 또한, 많은 산업 분야에서 다양하고 복잡한 형태의 멀티미디어 데이터를 생성, 가공, 교환하고 있기 때문에, 데이터에 대한 표준안은 중요한 기술 가운데 하나라고 볼 수 있다. 이러한 멀티미디어의 다양한 정보를 조직적으로 기술하기 위한 정의 언어와 스키마가 필요함에 따라 MPEG(Motion Picture Expert Group)은 멀티미디어 정보를 기술하고 풍부한 도구들을 제공하는 MPEG-7을 발표하였다. 그래서, 이러한 MPEG-7을 표준으로 멀티미디어 데이터를 XML로 표현하는 연구가 활발하게 진행되고 있다[1].

멀티미디어 중에서도 가장 많은 분야에 적용되어 사용되는 것은 이미지라고 볼 수 있다. 이미지는 국방용으로는 군사적인 목적으로 사용되기도 하고, 특정 분야인 의료 영상, 생물의학용, 지문인식, 홍채인식과 같은 생체인식기술에 접목되어 생체 이미지를 활용하는 이미지 검색 시스템 구축도 많이 연구되어지고 있다. 최근에 가장 많은 이미지 사용분야는 일반적으로 전자 소품몰의 상품 카탈로그 이미지, 전자상거래, 지리 정보와 같은 분야에서 광범위하게 사용 되어지고 있다. 이러한 이미지의 특징을 보면, 다양하면서 시각적인 효과를 중심으로 제작되어지며, 이미지 자체의 특징 데이터(feature data) 보다는 이미지 내의 상품에 대한 의미를 내포하고 있는 상품 중심의 메타데이터(metadata)나 의미 데이터(semantic data)가 더 중요함을 알 수 있다. 또한 한 이미지내에 여러 개의 객체를 포함하는 이미지의 경우, 각 객체의 특징은 크게 달라 질 수가 있다. 이러한 이미지의 데이터를 효율적으로 저장하고 관리하여 검색하기 위한 방법은 상당히 중요하다고 볼 수 있다. 특히, 인터넷상에서 사용 되어지고 있는 웹 상품 이미지의 경우에는 전자상거래 분야에 많은 영향력을 가지고 있다. 이러한 웹 상품 이미지에 대한 검색을 하기 위한 방법은 이미지와 관련된 상품명, 제작회사, 가격과 같은 관련 정보 등을 구조화하여 나타낼 필요가 있으므로 본 논문에서는 XML기술을 사용하여 모델링하고 사용한다.

본 논문에서는 이미지에 대한 표현방법은 크게 3가지로 구분하여 모델링 되어 진다. 첫 번째는 기존에도 많이 연구되어 오던 이미지 자체의 특징 벡터인

색상, 모양, 공간 그리고 질감을 기반으로 한 데이터 추출과 표현이며, 두 번째는 객체를 중심으로한 메타 데이터 추출과 표현이고, 마지막으로 세 번째는 이미지에 대한 의미 기반 검색을 가능하게 하기 위한 의미 데이터 추출과 변환의 표현방법으로 모델링 되어 진다. 따라서 본 논문에서는 다 계층 구조에 의해서 이미지에 포함된 객체의 특징과 의미를 각각 분리하여 저장 관리 할 수 있는 방법을 제시하고, 서로 상이한 시스템 간의 데이터 교환과 통합 그리고 표준을 고려하여 XML로 자동 변환하여 서비스할 수 있는 지능적인 이미지 검색 시스템을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 이미지 검색 기법과 시스템에 대하여 살펴보고, 내용 정보를 구조화하기 위한 XML 관련 기술을 분석한다. 3장에서는 전체적인 시스템 구성과 설계 구조를 살펴보고, 4장에서는 상품 이미지에 대한 객체 중심의 모델링 기법과 XML-Schema 구조에 대하여 기술한다. 5장에서는 4장에서 제안된 모델로 구축된 시스템으로부터 동적인 XML 문서 검색을 통해 지능적인 이미지 검색 이 가능함을 보이고 실제 구현 내용을 소개한다. 마지막으로 6장에서는 본 논문의 결과를 정리하고 앞으로의 연구 방향을 제시한다.

2. 관련 연구

기존의 이미지 검색 기술은 크게 세 가지 형태로 분류하여 연구되어져 왔다. 첫 번째는 키워드(keyword)를 기반으로 하는 단순한 검색 기술이다. 이미지에 대한 전체적인 주제나 관련 단어를 추출하여 데이터베이스나 파일 시스템에 저장 관리하여 검색 대상이 되도록 한다. 보통 키워드를 추출하는 방식은 자동으로 이루어지기 어려워 수동으로 표현하는 방식을 사용하여 이미지의 내용 정보를 구축하는 방법을 사용한다. 최근에는 웹 문서로부터 키워드를 자동으로 추출하여 인덱스로 구성하는 시스템이 사용되기도 한다. 이 방법은 분야별 해당 이미지에 대한 전문지식에 의한 기술이 필요하며, 기술방식에 차이로 인해 검색시에 의미를 해석하거나 단어의 정확도를 맞추기가 어렵다는 단점이 있다. 또한, HTML과 같은 웹 문서로부터 자동 추출할 경우 관련성 없는 단어나, 불필요한 내용이 포함될 수 있어, 검색의 효율성을 저하 시킬 수 있다.

두 번째는 이미지의 특징 벡터를 기반으로 검색하는 내용기반 이미지 검색 기술이다. 이미지의 대표적인 특징 벡터는 색상(color), 모양(shape), 공간(spatial), 그리고 질감(texture)과 같은 이미지 자체가 내포하고 있는 데이터가 있다. 이러한 특징 벡터를 데이터베이스 시스템이나 파일 시스템에 저장 관리하며 벡터 기반의 데이터를 비교하여 비슷한 이미지를 검색하는 내용기반 이미지 검색 기법이 많이 연구되어져 왔다. 보통 내용기반 이미지 검색을 지원하기 위해 이미지 특징 벡터를 자동으로 추출하여 저장하게 되는데, 시스템의 성능과 검색 시간을 고려하여 전처리(preprocessing) 과정을 거쳐 데이터를 추출하게 된다. 자동으로 이미지 데이터를 추출하는 것은 좋지만, 추출된 데이터 용량이 상당히 크며, 전처리 과정을 거치기 위한 비용도 상당히 크기 때문에 대용량의 이미지 검색을 위한 방법에 한계가 있다. 또한 자동으로 추출된 특징 벡터를 기반으로 검색을 할 경우, 사용자의 검색 의도나 이미지가 포함하고 있는 객체의 식별성이나 의미성을 고려하지 못해 검색의 정확도를 저하시키게 된다[2-5].

세 번째는 위의 두 가지 방식을 혼합하여 검색의 효율성을 증가시키는 기술이다. 이미지로부터 전처리 과정을 거쳐 이미지의 벡터 데이터를 자동 추출하고, 자동으로 추출할 수 없는 키워드 정보나 의미 정보, 시각 정보 등과 같은 메타 정보를 수동으로 표현하는 방법이다. 그러나, 이러한 검색 방법을 지원하는 시스템도 사용자의 다양한 검색 방식과 의미성을 해석하여 검색에 대한 적절하게 변환된 형태로 결과를 제공할 수 없으며, 다른 시스템과의 통합이나 교환시에 문제가 발생할 수 있다는 문제는 여전히 남아 있다. 그래서 다양한 환경에서 다양한 방식으로 검색이 가능한 데이터 변환 기술이 필요하며, 서로 상이한 시스템간의 데이터 교환을 위한 멀티미디어 표준 기술의 도입이 필요하게 되었다. 이미지의 내용 정보를 구조화하기 위해서 차세대 표준 언어인 XML 기술을 사용하고, XML 문서로부터 데이터를 질의하기 위한 XQL (XML Query Language), 문서 변환을 위한 XSL (eXtensible Stylesheets Language)의 기술을 사용한다.

2.1 기존 이미지 검색 시스템

기존의 이미지 검색 시스템들은 키워드, 색상, 모

양, 질감과 같은 내용 기반 검색 기법을 제공하고 있다. 개발되어진 이미지 검색 시스템을 살펴보면 IBM 연구소의 QBIC 시스템이나 Columbia 대학의 SAFE, VisualSEEK, WebSEEK 시스템이 있으며, Stanford 대학의 WEIS, SIMPLcity 시스템이 있고, MIT의 Photobook, 그리고 U.C.Berkeley의 Chabot, Blobworld와 같은 시스템들이 있다[6-12]. 표 1은 기존의 내용 기반 이미지 검색 시스템의 검색 방식과 특징 정보 추출방식, 그리고 개선점을 비교 설명하고 있다.

표 1. 이미지 내용 검색 시스템 비교

시스템	검색 방식	특징 정보 추출방식	개선점
QBIC	사용자 스케치 검색, 샘플 이미지에 의한 유사성 검색, 색상, 모양 및 텍스처에 기반한 검색	점(point)으로 특징 벡터 표현, R+-트리를 이용해 인덱싱	통합되지 못한 검색 인터페이스, 의미 기반 정보 검색 불가능, 지식기반 검색 불가능
Chabot	텍스트 및 색상 기반 검색	평균 색상 히스토그램과 수동적인 키워드 입력	복잡한 질의 인터페이스, 내용정보간의 관계성 부족
Visual-SEEK	평균 색상 및 공간 검색, 시각적 특징에 의한 검색	영역적 평균 색상 히스토그램추출과 자동화된 영역 추출	색상정보 검색의 큰 비중, 비효율적인 공간 관계 연산 시간
SIMPLcity	영역 매칭 검색	영역 세그멘테이션 분할, 의미적 분류, 영역적 특징 추출	지식기반 검색 불가능, 영역적 연산 시간 소요
Photobook	객체 기반 검색, 색상 및 질감, 모양에 기반한 검색	객체(윤곽선)정보 추출, 색상, 질감, 모양 특징 추출	통합되지 않은 검색 인터페이스, 의미 기반 검색 불가능, 객체간의 관계성 부족

2.2 XML 관련 기술

XML은 웹 상에서 데이터 교환을 위해 제안된 표준 언어이다. 웹 문서들을 효율적으로 저장하고 검색, 이용하기 위해 W3C에서는 차세대 웹 문서를 기술하기 위한 표준 마크업 언어인 XML을 제안하였다. 다양한 정보 형태를 가진 웹 문서의 효과적인 관리를 위해 XML 관련 연구들이 현재 국.내외에서 활발히 진행되고 있다. 특히, XML문서들을 저장, 관리 및 검색할 수 있는 XML 저장 관리 시스템 개발에 대한 연구와 XML 관련 질의어에 관한 연구가 대두되고 있다. XML 데이터를 데이터베이스에 저장하여 질의하기 위한 방법은 많이 연구되어 왔다[13-16]. 이러한 방법들에는 XML을 위한 XML전용 데이터베이스 시스템을 만들거나, RDB 혹은 ORDB와 같은 기존의 데이터베이스 시스템에 저장 관리하는 것들이다. 기존의 데이터베이스 시스템에 저장할 때 데이터 모델의 이질성으로 인하여 이를 저장하기 위한

다양한 방법이 제시되었으며, 이는 기존의 데이터베이스 시스템의 특성을 잘 이용하여 XML 질의 수행을 빠르게 하도록 지원하고 있다. 또한 기존 데이터베이스 시스템에 저장된 데이터를 XML 문서로 변환하는 도구 개발에 관한 연구가 많이 이루어지고 있으며, 웹 환경에서 이질적인 정보에 대해 데이터의 추출 및 통합하는 시스템 개발도 연구되어지고 있다.

현재 XML 관련 질의어는 XPath, XML-QL, XQL, Quilt, XQuery 등이 있으며, 이들은 XML문서의 구조적 특성을 반영한 구조-내용기반 검색을 지원하고 있다[17-21]. 표 2는 XML 질의어에 대한 비교 분석 내용을 보이고 있다. 이러한 XML 질의어를 활용하면 내용 기반 검색, 구조 기반 검색 등과 같은 다양한 형태의 XML 문서 검색 질의를 표현할 수 있다.

XML의 또 다른 특징은 다양한 형태로의 문서 변환 기술이다. XML문서를 다른 형식의 데이터로 변환하기 위해 XSL을 적용하며, XSLT에 의해 변환된 문서로 제공되어 진다[22]. 그림 1은 무선 인터넷 사용자에게 의한 이미지 검색을 할 경우 XML문서를 XSLT에 의해 WML(Wireless Markup Language) 문서로 변환 서비스 해야할 필요성을 설명하고 있다.

표 2. XML 질의어 비교 분석

질의어 요약내용	XPath	XML-QL	XQL	Quilt	XQuery
특징	문서 자체의 위치 경로를 통한 정확한 노드 정보 검색 언어	반구조적 데이터용 질의 언어이며, 사용자 정의가 가능한 태그와 입력 엘리먼트를 다루는 것이 가능한 선언적 질의 언어	반용적 질의 언어로 경로 표현식 사용하며 XSL에서의 패턴 매치 구문과 유사한 특징적 질의 언어이며 사용하기 편리함.	XQL에서 계층화된 문서 구조 회하기 위한 구문을 제공하고 XML-QL에서의 변수의 활용과 새로운 구조의 XML을 만드는 기능을 제공.	Query 기반 질의어인 기존의 XML 질의어의 XML-QL, XQL 뿐만 아니라 SQL/OLAP과 같은 질의어의 특징과 함께 표현 가능.
질의 구조	위치 경로	WHERE절과 CONSTRUCT 절 구조	경로 표현, 논리식, 비교 연산자, 패턴 매치, 필드 연산자 등	FOR, LET, WHERE, RETURN 절	경로 표현, 엘리먼트 생성자, FOR, LET, WHERE, RETURN 절
질의 결과	노드의 집합	XML 데이터	노드, XML 문서, 배열, 기타 구조	XML 데이터	XML 데이터
내용/구조 기반 검색지원 사항	지원	지원	지원	지원	지원
질의 가능 영역	한 문서 또는 XML 저장소 내 모든 문서에 대한 질의 지원	한 문서 또는 XML 저장소 내 모든 문서에 대한 질의 지원	한 문서 또는 XML 저장소 내 모든 문서에 대한 질의 지원	한 문서 또는 XML 저장소 내 모든 문서에 대한 질의 지원	한 문서 또는 XML 저장소 내 모든 문서에 대한 질의 지원

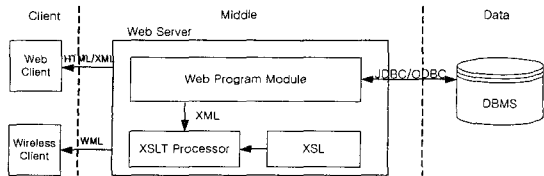


그림 1. XSLT에 의한 XML문서 자동 변환 구조

이러한 시스템구조는 사용자의 사용 환경에 따라 자동으로 XML문서를 변환하여 서비스하게 된다.

3. 시스템 구조

본 논문에서는 동적인 XML문서를 활용하여 지능적인 이미지 검색을 지원할 수 있는 시스템을 제안한다. 제안된 시스템의 구조는 기존의 데이터베이스 시스템을 그대로 사용하면서 동적인 XML문서를 생성할 수 있는 시스템이다. 데이터베이스 시스템을 그대로 사용하는 목적은 데이터 관리의 안전성과 데이터베이스의 성능을 충분히 활용하는데 목적이 있다. 또한, 기존의 검색 방법을 지원하면서 이미지 메타 데이터를 좀 더 효율적으로 저장 관리하기 위해서이다. 그림 2는 본 논문에서 제안한 시스템의 전체 구성도를 나타내고 있다.

제안된 시스템은 크게 세 분류로 나누어 볼 수 있다. 기본적으로 이미지에 대한 특징 벡터 데이터와 메타 데이터, 의미 기반 데이터 등을 저장관리하기 위한 데이터 저장소이며, 두 번째는 이미지에 대한 정보를 추출하기 위한 데이터 추출기가 있다. 데이터 추출기는 이미지 자체의 특징 벡터를 전처리 과정을 거쳐 추출되어지며, 다중 객체를 가지는 이미지의 경우 다중객체를 식별하여 추출하므로써 객체 중심의 검색을 가능하게 한다. 만일, XML 문서로 작성되어 있는 경우, XML 파서에 의해 DBMS 구조에 매핑되어 데이터를 저장하게 된다. 본 논문에서는 이미지 특징 벡터에 대한 전처리 과정은 이루어진 것으로 가정한다. 세 번째는 검색에 대한 질의 처리 구조이다. 이미지에 대한 검색은 다양한 방법으로 검색될 수 있는 IQI(Intelligent Query Interface)를 통해 검색 인터페이스를 제공하며, 검색 데이터는 특징 분석기에 의해 분석되어지고, 검색되어진 데이터는

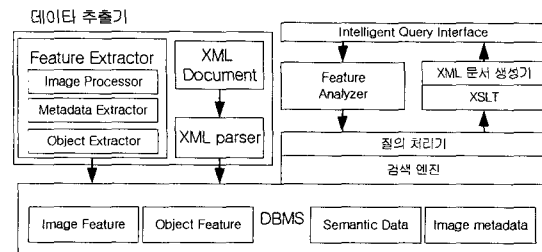


그림 2. 시스템 구조

XSLT의 변환 구조에 의해 HTML문서나 WML문서의 형태로 제공되어 진다.

본 논문에서 제시한 데이터 저장 구조는 XML문서와의 매핑관계를 고려하여 설계 되어졌으며, 이미지 내의 객체를 중심으로 데이터를 저장하는 구조로 설계되어져 있다. 또한, 이미지의 특징 벡터와 메타데이터 이외에도 MPEG-7에서 제안한 표준안을 고려하여 멀티미디어를 표현하기 위한 문서 구조를 적용하여 설계하였다. 그림 3은 데이터베이스 스키마 구조를 E-R 다이어그램으로 나타내고 있다. 이미지에 대한 특징 벡터는 중심 객체를 중심으로 색상(color), 모양(shape), 질감(texture), 그리고 공간(spatial) 테이블에 각각 저장되며, 이미지의 의미정보는 SemanticDS 테이블에 저장된다. 또한 이미지의 메타데이터는 Metadata InformationDS에 저장한다. 즉, 한 이미지에 객체를 중심으로 데이터가 분리되어 저장된다. 이렇게 저장된 데이터는 다른 이미지의 같은 객체에 대하여 별도의 데이터를 가지지 않고도 객체의 OID(Object ID)만으로 이미지에 포함된 모든 객체의 의미성과 특징 벡터 그리고 메타데이터를 공유하여 사용할 수 있다.

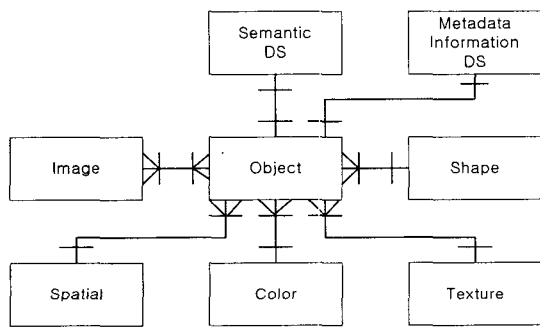


그림 3. 데이터베이스 스키마 구조

4. 이미지 메타데이터 모델링

4.1 다계층 메타데이터

본 논문에서는 지능적인 이미지 검색을 위한 다계층 메타데이터의 표현 구조를 제안하고, XML로 표현할 수 있는 스키마 구조를 나타낸다. 메타데이터란 데이터를 위한 데이터라고 정의할 수 있다. 그러므로, 이미지 메타데이터는 이미지 데이터를 위한 데이터를 의미한다. 보통 이미지를 위한 메타데이터는 다

음과 같이 두 가지로 나누어 볼 수 있다.

- 등록 메타데이터(registration metadata) : 이미지 크기, 비율, 타입, 색상 분포등과 같이 이미지의 속성 데이터를 말한다. 이러한 메타데이터는 이미지의 조작이나 표현방법에 따라서 변화될 수 있다.
- 설명 메타데이터(description metadata) : 이미지 주제, 제목, 키워드, 설명, 파일명등과 같이 언어로 표현되는 텍스트 데이터를 말한다.

이러한 등록 메타데이터나 설명 메타데이터는 내용기반 검색을 지원할 수는 없다. 그러기 때문에 내용기반 검색을 지원하기 위한 이미지 메타데이터를 다음과 같이 다계층적인 구조로 정의한다.

- 전역 특징 메타데이터 레이어(global feature metadata layer) : 한 이미지의 전체적인 평균 색상이나 텍스처 패턴, 영역 모양이 대표적인 전역 특징 메타데이터의 예라고 할 수 있다. 기존에 내용기반 이미지 검색 시스템에서 가장 많이 사용하던 이미지 메타데이터이다.
- 지역 특징 메타데이터 레이어(local feature metadata layer) : 한 이미지로부터 추출되어진 영역이나 객체에 대한 색상, 텍스처, 모양을 지역 특징 메타데이터라고 한다. 실제적으로 객체에 대한 이미지가 존재하지 않으며, 개념적인 이미지로 표현되게 된다. 현재의 내용기반 이미지 검색에서는 많이 고려가 되지 못하고 있는 이미지 메타데이터이다.
- 의미 내용 메타데이터 레이어(semantic contents metadata layer) : 의미 내용 메타데이터는 지식기반 레이어라고 할 수도 있다. 이미지에 포함되어 있는 개념 혹은 의미적 내용이나 지식적인 내용과 같은 것을 의미 내용 메타데이터라고 한다. 또는 이미지 검색을 위한 유용한 정보들을 나타내기도 한다.

그림 4는 이미지로부터 다계층 메타데이터 생성을 하는 모델을 설명하고 있다. 한 이미지에는 3개의 객체가 존재하며 각 객체를 포함하는 전역 특징 메타데이터 레이어에는 전체 평균색상(예:붉은색)을 가질 수 있으며, 이미지 전체의 의미적인 내용이나 지식적인 내용을 표현할 수 있다. 또한 각각의 객체로 표현된 지역 특징 메타데이터 레이어에서는 각 객체의 특징 벡터들을 가질 수 있다. 각 객체의 의미적인 내용을 표현하기 위해 의미 내용 메타데이터 레이어

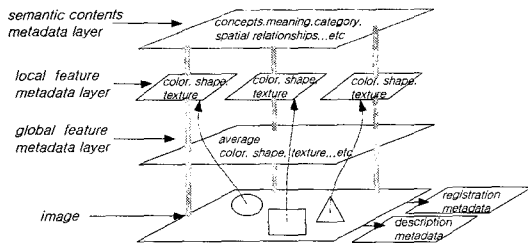


그림 4. 다계층 메타데이터 구조

에서는 각 객체의 의미적 내용이나 지식적인 내용을 표현한다.

상품 카탈로그 이미지의 경우에는 한 이미지에 다중 객체를 내포하고 있는 경우가 많다. 객체 각각에 부여되는 의미나 특징 벡터는 검색에 중요한 요소가 될 뿐만 아니라, 검색의 효율성을 높이게 된다. 기존의 이미지 검색 시스템들은 이미지 전체에 대한 내용을 묘사하거나 특징 벡터를 추출하기 때문에 검색 결과에 대한 사용자의 만족도를 높이기 어렵다. 그림 5는 실제 다중 객체를 가지는 웹 상품 카탈로그 이미지의 예를 보이고 있다.

그림 5에 첫 번째 이미지의 경우 한 이미지 속에 “의자”라는 중심 객체와 “테이블”이라는 중심 객체

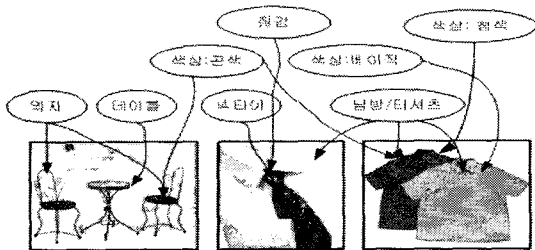


그림 5. 다중 객체를 가지는 상품 카탈로그 이미지

가 다중으로 존재하고 있다. 또한 두 번째 이미지의 경우에는 “넥타이”와 “티셔츠”라는 중심 객체를 내포하고 있다. 세 번째 이미지 또한 같은 “티셔츠”라는 중심 객체를 가지고 있지만, 각각 다른 색상을 가지고 있음을 알 수 있다. 이렇게 중심 객체는 같은 의미를 가지고 있지만, 다른 특징 벡터 데이터를 가질 수도 있고, 다른 의미를 가지고 있지만, 같은 특징 벡터를 가질 수도 있다.

4.2 이미지 메타데이터의 XML 표현

본 논문에서는 이러한 다중 객체를 중심으로 데이터를 구조화 하기 위해서 MPEG-7을 고려한 그림 6과 같은 트리구조를 설계하였다.

그림 7은 그림 6의 데이터 정의에 따라 다중 객체를 가진 이미지 내용 정보를 구조화 하여 나타낸 XML 문서의 예를 보이고 있다.

XML문서의 형태를 정의하기 위해 본 논문에서는 XML-Schema 문법을 사용한다. 기존에 XML문서 저장 시스템에서는 DTD(Document Type Definition)로 정의하여 사용하였지만, DTD의 표현 한계성과 다양하지 않은 데이터 타입으로 인해 XML문서의 형태를 정의하는데 문제가 있다. 따라서 본 논문에서는 그림 8과 같이 다중 객체를 가지는 이미지를 요소(element) 중심으로 각 객체를 표현할 수 있는 XML-Schema를 설계 하였다. 그림 8의 ‘object-1’은 객체의 표현을 위해 모든 요소를 참조(reference)하여 사용하고 있으며, ‘object-2’는 객체의 표현을 위해 일부 요소를 참조해 사용하고 있다. 그림 8은 요소(element)를 중심으로 XML-Schema를 모델링 한 것이다.

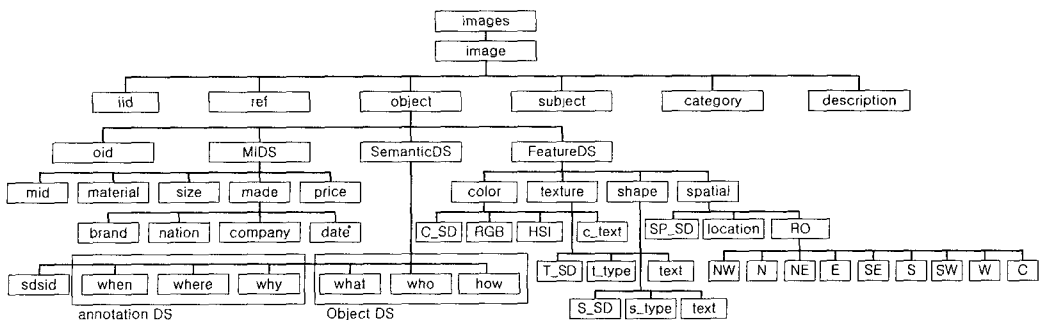


그림 6. 다중 객체를 위한 메타데이터 구조

```

<?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 </naticn>
          <brand>SMC</brand>
          <date> 2002-10-15</date> </made>
        <price>33,000</price>
      </MIDS>
      <semanticDS> <sdsid>sd0001</sds.d>
        <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>
  
```

그림 7. 다중 객체 이미지를 위한 XML 문서

본 논문에서 설계하고 제시한 다중 객체 이미지에 대한 XML-Schema 모델은 요소의 정의를 재사용하여 정의할 수 있을 뿐만 아니라, 요소내의 데이터에 대한 패턴(pattern)이나 모델을 정의할 수 있다. 그림 9는 그림 8에서 모델링한 XML-Schema에 대한 문서를 보이고 있다.

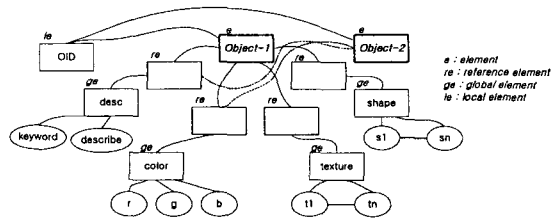


그림 8. 다중 객체 이미지 XML-Schema 모델링

```

<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:complexType name="semanticDSType">
    <xsd:all>
      <xsd:element name="sdsid" type="xsd:string"/>
      <xsd:element name="when" type="xsd:string"/>
      <xsd:element name="where" type="xsd:string"/>
      <xsd:element name="why" type="xsd:string"/>
      <xsd:element name="what" type="xsd:string"/>
      <xsd:element name="who" type="xsd:string"/>
      <xsd:element name="how" type="xsd:string"/>
    </xsd:all>
  </xsd:complexType>
  
```

```

<xsd:complexType name="featureDSType">
  <xsd:sequence>
    <xsd:element name="color" type="colorType"/>
    <xsd:element name="texture" type="textureType"/>
    <xsd:element name="shape" type="shapeType"/>
    <xsd:element name="spatial" type="spatialType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="colorType">
  <xsd:all>
    <xsd:element name="cid" type="xsd:string"/>
    <xsd:element name="r" type="rgbType"/>
    <xsd:element name="g" type="rgbType"/>
    <xsd:element name="b" type="rgbType"/>
    <xsd:element name="h" type="xsd:string"/>
    <xsd:element name="s" type="xsd:string"/>
    <xsd:element name="i" type="xsd:string"/>
    <xsd:element name="color_SD" type="xsd:string"/>
    <xsd:element name="color_text" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>

<xsd:simpleType name="rgbType">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="255"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="textureType">
  <xsd:all>
    <xsd:element name="tid" type="xsd:string"/>
    <xsd:element name="texture_type" type="bitType"/>
    <xsd:element name="texture_SD" type="xsd:string"/>
    <xsd:element name="texture_text" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>

<xsd:complexType name="shapeType">
  <xsd:all>
    <xsd:element name="sid" type="xsd:string"/>
    <xsd:element name="shape_type" type="bitType"/>
    <xsd:element name="shape_SD" type="xsd:string"/>
    <xsd:element name="shape_text" 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>

```

그림 9. XML-Schema 문서

이렇게 정의된 XML-Schema 문서는 다중 객체 이미지를 위한 XML 문서에 대하여 유효성(validation)을 검증하고, 표준으로 정의되어 사용되므로 상이한 시스템간의 XML 문서 교환이 용이하게 된다. 그림 10은 다중 객체 이미지를 위한 XML 문서에 대하여 DTD를 매핑하는 관계도를 보이고 있다.

그림 11은 XML-Schema 문서의 이미지 메타데이터 정보(MIDS)를 정의한 부분과 관계형 데이터베이스의 테이블 스키마간의 매핑 구조를 일부분 설명하고 있다. XML-Schema 문서에서는 요소의 자식을 정의하기 위해 "madeType"이라는 사용자 정의 타입(type)을 정의하여 사용하고 있으며, 이러한 부모 요소와 자식 요소의 정의는 데이터베이스 테이블의 스키마에 매핑되어 XML문서의 데이터를 저장 관리하게 된다.

5. 시스템 구현

본 장에서는 논문에서 제안한 시스템 설계를 기반으로 동적인 XML문서를 활용한 지능형 이미지 검색 시스템을 구현하고, 내부의 구현 기술을 설명한다. 제안된 시스템은 웹 환경에서 XML 데이터를 수용할 수 있으며, 동적인 XML문서 생성을 지원하는 MS-SQL SERVER 2000 과 IIS 웹 서버를 사용하였다[23].

5.1 XML 문서 검색

본 논문에서 제안한 프로토타입 시스템의 구현 환경은 크게 5가지로 분류하여 나누어 볼 수 있다. 첫 번째 서버 운영 체제는 Windows 2000 Server 환경이며, 두 번째 데이터 저장소는 Microsoft 사의 MS-SQL SERVER 2000 데이터베이스 시스템을 사용하였다. 세 번째로 웹 클라이언트는 Microsoft Internet

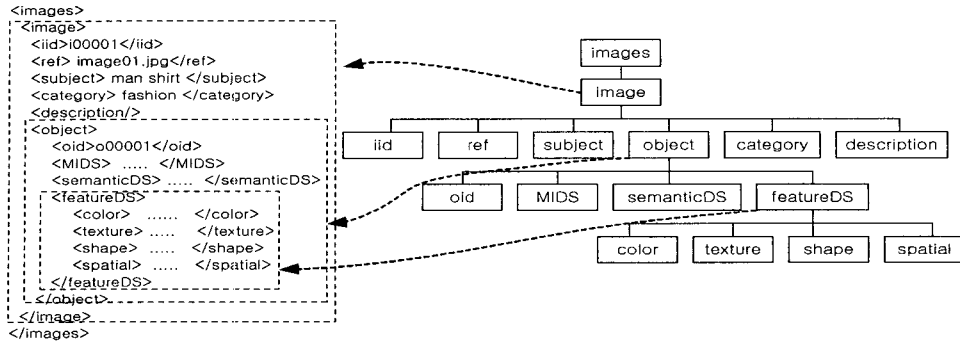


그림 10. XML 문서와 DTD의 매핑 관계도

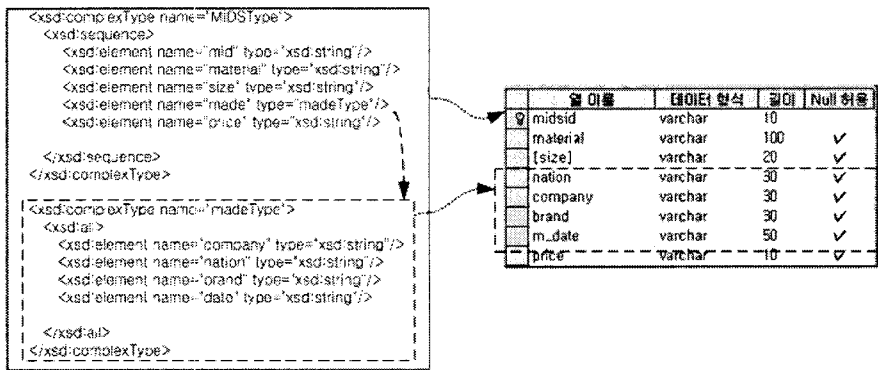


그림 11. XML-Schema 문서와 테이블 스키마의 매핑구조

Explorer 6.0과 Openwave사의 Openwave SDK 5.1을 사용하였으며, 개발 도구 언어는 XML문서의 파싱과 관련API를 지원하는 DOM(Document Object Model), SAX(Simple API for XML)를 기반으로 JAVA와 JSP(Java Server Page)를 사용하였다. 마지막으로 웹 서버는 SQL server에서 지원하고 있는 IIS 가상 디렉토리 서버를 사용하였다. SQL server 2000에서 새롭게 제공되는 기능인 HTTP를 이용한 처리를 이용해 인터넷이나 인터넷 기반 애플리케이션을 구축할 수 있고, XML 데이터를 처리할 수 있으며 또한 매우 유용하다. HTTP로 데이터 처리하는 이 기능은 ASP 애플리케이션보다 좀 더 적은 코드를 이용해 쉽고 빠르게 데이터 중심적(data-centric) 웹 사이트를 구축할 수 있도록 한다. XSL 스타일 시트를 XML 데이터에 적용해 브라우저 기반 클라이언트에게 전송하거나 WML과 같은 문서로 보내거나 WAP(Wireless Application Protocol)같은 형식으로 휴대폰에서 사용 가능한 문서의 형태로 변환 서비스 할 수 있다. 그림 12는 SQL SERVER 2000

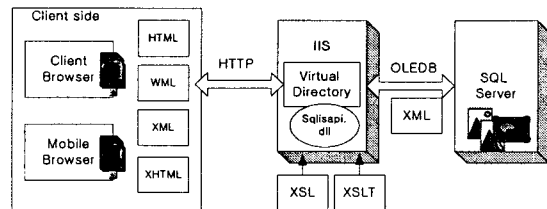


그림 12. SQL Server 2000의 HTTP 데이터 접근 아키텍처

의 HTTP 데이터 접근 아키텍처를 보이고 있다.

SQL Server의 HTTP 처리는 SQLISAPI라고 명명된 SQL Server로부터 제공된 ISAPI 애플리케이션 이용이 가능해진다. IIS에 가상 디렉토리 루트를 생성해 HTTP URL을 이용해 SQL Server의 데이터베이스에 접근을 가능하게 하며 이는 "http://dbmm.dankook.ac.kr/iir" 식으로 수행할 수 있다. SQLISAPI 애플리케이션은 SQL Server 2000의 OLE-DB 프로바이더를 이용해 XML이나 HTML 데이터를 클라이언트로 제공한다.

클라이언트 애플리케이션은 다음과 같은 4가지

방식으로 데이터를 요청 할 수 있다.

- FOR XML 쿼리를 쿼리 스트링(Query String)으로 URL에 보내는 방법.
- XML Query Template을 ISAPI 애플리케이션에 포스팅 하는 방법.
- XML Query Template을 웹 서버의 가상 디렉토리에 지정하는 방법.
- XML schema를 웹 서버의 가상 디렉토리에 정의하는 방법.

이러한 다양한 질의 방식 제공은 동적인 XML문서를 생성하는데 효율적이다. 그림 13은 URL 쿼리를 이용해 데이터베이스 시스템의 이미지 메타데이터를 XML문서로 처리하는 예를 보이고 있다. 질의에 대한 기본적인 출력 형식은 속성 혹은 요소를 중심으로 표현될 수 있으며, 요소와 속성을 혼합한 표현도 가능하다.

또한 XML 문서 처리를 위해 XML 템플릿을 사용할 수 있다. 그림 14는 데이터베이스 시스템으로부터 이미지 메타데이터를 XML 문서 처리하기 위한 XML 템플릿 문서와 템플릿에 의한 질의 결과를 보이고 있다.

5.2 XML 문서 저장

본 논문에서는 서로 상이한 시스템간의 이질성을 해결하기 위해서 XML 문서를 사용한다. 제안 시스템에서는 이러한 이질성 문제를 해결하고 XML문서로 전달받은 데이터를 데이터베이스에 저장할 수 있다. XML 문서로부터 데이터를 추출하여 데이터베이스 시스템으로 저장하는 과정은 그림 15에서 설명한다.

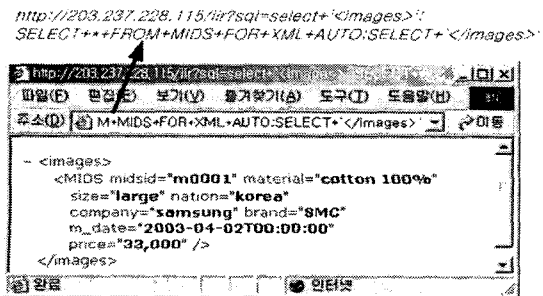


그림 13. URL 쿼리를 이용한 XML 문서 처리

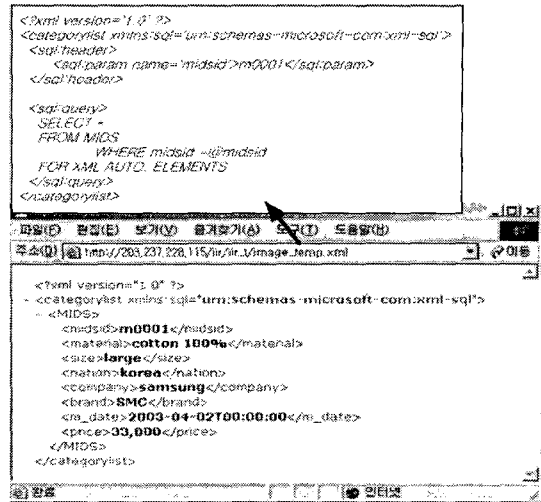


그림 14. XML 템플릿을 이용한 XML 문서 처리

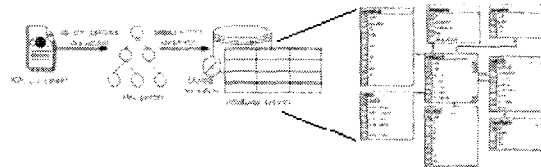


그림 15. XML 데이터 저장 방법과 구조

데이터를 저장하기 위해 OpenXML 함수를 사용하기 전에 XML 문서는 파싱되고 문서의 노드 구조를 메모리 상에 트리 구조로 매핑하게 된다. sp_xml_preparedocument 저장 프로시저로 문서를 읽고 유효한 XML 문서인지 검증한다. 검증된 후에는 저장 프로시저에 의해 요소와 속성으로부터 데이터를 추출해 사용할 수 있는 노드 트리 핸들(handle)을 리턴한다. 노드 트리가 생성된 후에 XML문서의 로우 셋 데이터를 리턴하기 위해 OpenXML을 사용하여 테이블에 데이터를 저장하게 된다. 그림 16은 이미지 메타데이터를 기술한 XML 문서의 일부를 데이터베이스 테이블에 저장하기 위한 프로시저를 나타내고 있다.

5.3 동적 XML 데이터 문서 변환

저장되어진 이미지 정보는 사용자의 검색 인터페이스에 따라 XSLT 변환 처리하여 제공된다. 웹 서버에 접근하는 웹 브라우저는 필요한 페이지를 요청하기 전에 현재 사용자가 사용하는 접속 기기와 웹 브라우저에 대한 다양한 정보를 HTTP헤더에 저장하

```

CREATE PROCEDURE INSERTXMLDOC
AS
DECLARE @xmlDATA VARCHAR(2000)
SET @xmlDATA = '
<?xml version="1.0"?>
<images>
  <image>
    <iid> i00002</iid>
    <subject> samsung cellular phone </subject>
    <category> Phone </category>
    <ref> image02.jpg </ref>
    <description> good cellular phone </description>
  </image>
</images>
'

DECLARE @iTree INTEGER
EXEC sp_xml_preparedocument @iTree OUTPUT, @xmlDATA
INSERT images (iid, subject, category, type, descript)
SELECT iid, subject, category, type descript
FROM
OPENXML(@iTree, 'images image', 1)
WITH (
  iid varchar(10) 'iid',
  subject varchar(100) 'subject',
  category varchar(30) 'category',
  ref varchar(10) 'ref',
  description varchar(500) 'description' )
    
```

그림 16. XML 문서 저장 프로시저

여 웹 서버에 전달한다. HTTP 요청 패킷과 함께 전송된 HTTP 헤더 정보에는 사용자의 IP, 운영체제, 브라우저의 종류, 브라우저가 처리할 수 있는 문서의 종류, 로그인 정보 등에 사용되는 쿠키 정보 등이 저장되어 있다. 다양한 접속기기를 지원하는 유.무선 통합 개발이나, 콘텐츠의 다양한 포맷을 지원하는 경우에는 XML 형태로 정보를 저장하고, 접속 기기가 판별되면 각각의 접속 기기에 알맞은 콘텐츠 스타일로 XSLT 변환을 한 후 출력 결과를 전송한다. 이런 과정을 위해서 접속 기기 판별에 HTTP헤더 정보를 사용한다. 헤더 정보를 검색하기 위해서는 request 객체의 getHeader 메소드를 사용하고, 접속 기기 판별에는 "ACCEPT" 속성을 사용하며, 접속 브라우저 판별에는 "USER-AGENT" 속성을 사용한다. 휴대폰으로 무선 인터넷을 통해 웹 서버에 접속하는 경우 "ACCEPT" 속성 값에는 "wap" 문자열이 포함되어 있게 되고, 이 문자열을 "indexOf()" 메소드로 검색하여 유.무선 인터넷 기기를 구분하게 할 수 있다. 그림 17은 동적인 XSLT 변환 처리를 하기 위한 알고리즘을 나타내고 있다.

```

// 접속 브라우저 판별 (WAP 브라우저인 경우 true)
boolean isWAP = (request.getHeader("accept").indexOf("wap") != -1)? true:false;
// XML 데이터 소스를 StreamSource 객체로 저장.
StreamSource xmlSource = new StreamSource("Link.URL");

// WAP 브라우저이면
if(isWAP)
//MIME 타입은 text/vnd.wap.wml
response.setHeader("content-type", "text/vnd.wap.wml;encoding=euc-kr");

// XSLT 스타일시트는 WAP용으로 지정
xsJURL "for wap xml document";
else
//MIME 타입은 text/html
response.setHeader("content-type", "text/html;encoding=euc-kr");
//XSLT 스타일시트는 HTML용으로 지정
xsJURL "for html xml document";
    
```

그림 17. XSLT 변환 처리 알고리즘

그림 18은 동적인 XSLT 변환 처리 알고리즘을 적용하여 이미지 분류 검색을 한 예를 보이고 있다.



그림 18. 이미지 분류 검색

6. 결론 및 향후과제

인터넷의 발전으로 인해 최근 XML 관련 연구가 활발히 진행되고 있다. 또한, 응용 분야에서 XML의 장점을 이용한 다양한 응용 시스템들이 개발되고 있다. 그러나 아직까지 멀티미디어 정보 검색을 동적으로 지원하기 위한 XML 기술 응용은 거의 없는 실정이다. 멀티미디어 가운데 이미지 정보의 양은 급격히 증가하고 있으며, 사용자들은 다양한 환경에서 이미지 정보를 정확하게 빨리 찾기를 원하고 있고, 질의

방식 또한 다양한 형태로 이루어지고 있다.

본 논문에서는 사용자의 검색 질의에 대해 지능적으로 대처할 수 있는 시스템을 제안하였다. 기존의 데이터베이스 시스템에 이미지의 정보를 저장할 수 있는 모델과 저장되어진 데이터를 사용자 환경에 맞추어 XML로 자동 변환하여 검색할 수 있는 방법을 제시하였다. 또한 의미기반 이미지 검색과 내용기반 이미지 검색을 동시에 지원할 수 있는 XML-Schema를 설계하고 동적인 XML문서를 활용하여 다양한 검색을 지원할 수 있는 시스템을 구현 하였다. 또한 상품 카탈로그와 같은 복잡한 다중 객체를 보유하고 있는 이미지에 대하여 효율적으로 저장 관리하며 검색할 수 있는 다계층 모델을 제시하고 구조화된 XML 스키마로 설계하였다. 본 시스템의 활용은 기존의 웹 브라우저 환경 이외에 WAP이나 PDA와 같은 단말기로의 콘텐츠 제공을 지능적으로 제공하므로써 사용자의 검색 만족도를 높일 수 있다.

향후 연구과제로는 본 논문에서 제안한 시스템을 기반으로 이미지나 동영상과 같은 멀티미디어 자체의 데이터를 사용자의 검색 환경에 맞게 자동 변환해 줄 수 있는 모듈을 추가 하는 것이며, 또한 XML문서의 활용도를 증대시키기 위해 좀 더 다양한 콘텐츠 변환 XSLT를 설계하여 사용자의 만족도를 높일 수 있는 연구가 필요하다.

참 고 문 헌

- [1] <http://www.w3.org/2001/05/mpeg7/w4032.doc>.
- [2] H. V. Jagadish, "A retrieval technique for similar shapes," Proceedings of the 1991 ACM SIGMOD international conference on Management of data, pp. 208-217, 1991.
- [3] D. Androutsos, K. N. Plataniotis and A. N. Venetsanopoulos, "Image Retrieval Using the Directional Detail Histogram," Storage and Retrieval for Image and Video Database SPIE (1998) 129-139.
- [4] Hong, S., Lee, C., and Nah, Y., "An Intelligent Web Image Retrieval System," Proceeding of SPIE: Internet Multimedia Management System II, Vol.4519, August 2001, pp. 106-115.
- [5] Chu, W.W., leong, I.T., and Taira, R.K., "A Semantic Modeling Approach for Image Retrieval by Content," VLDB Journal, 3, 1994, pp. 445-477.
- [6] Flickner, M. et al., "Query by Image and Video Content: The QBIC System," IEEE Computer, Sept. 1995, pp.23-32. (<http://www.qbic.ibm.almaden.com>)
- [7] Ogle, V.E. and Stonebraker, M., "Chabot: Retrieval from a Relational Database of Images," IEEE Computer, Sept. 1995, pp. 40-48.
- [8] John R. Smith and Shin-Fu Chang, "VisualSEEK: a fully automated content-based image query system," ACM Multimedia 96, pp. 87-98, 1996.
- [9] 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.
- [10] Wang, J.Z., Li, J., and Wiederhold, G., "SIMPLcity: Semantics-Sensitive Integrated Matching for Picture Libraries," IEEE TKDE, 23(9), 2001, (<http://www-db.stanford.edu/IMAGE/>).
- [11] Jia.Li, James Z. Wang, Gio Wiederhold, "IRM: Integrated region matching for image retrieval," Proc. ACM Multimedia, pp. 147-156, Los Angeles, ACM, October 2000.
- [12] Jia Li, James Z. Wang, Gio Wiederhold, "Classification of textured and non-textured images using region segmentation," Proc. IEEE International Conference on Image Processing (ICIP), Vancouver, BC, Canada, pp. 754-757, IEEE, September 2000.
- [13] Alin Deutsch, Mary Fernandez, and Dan Suciu., "Storing Semistructured Data with STORED," SIGMOD, 1999.
- [14] Daniela Florescu and Donald Kossmann., "Storing and Querying XML Data using an RDBMS," Data Engineering Bulletin, 22(3), September 1999.

- [15] Jayavel Shanmugasundaram, Kristin Tufte, Gang He, Chun Zhang, David DeWitt, and Jeffrey Naughton., "Relational Databases for Querying XML Documents: Limitations and Opportunities," VLDB, 1999.
- [16] Takeyuki Shimura, Masatoshi Yoshikawa, and Shunsuke Uemura., "Storage and Retrieval of XML Documents using Object-Relational Databases," DEXA, 1999.
- [17] W3 Consortium, <http://www.w3.org/XML>, 2001.
- [18] <http://www.almaden.ibm.com/cs/people/chaemberlin/quilt.html>
- [19] <http://www.ibiblio.org/xql/>
- [20] <http://www.w3.org/TR/NOTE-xml-ql/>
- [21] <http://www.w3.org/TR/xquery/>
- [22] <http://www.w3.org/TR/xslt20/>
- [23] <http://msdn.microsoft.com/xml/>



홍성웅

1998년 우송대학교 컴퓨터학과 (공학사)
 2000년 단국대학교 대학원 컴퓨터공학과 (공학석사)
 2000년~현재 단국대학교 컴퓨터공학과 (박사수료)

관심분야 : 데이터베이스 모델링, 멀티미디어 데이터베이스, 멀티미디어 정보검색, 데이터마이닝, 지능정보 시스템, 시맨틱 웹

E-mail : syhong@dku.edu



나연묵

1986년 서울대학교 컴퓨터공학과 (학사)
 1988년 서울대학교 대학원 컴퓨터공학과 (석사)
 1993년 서울대학교 대학원 컴퓨터공학과 (박사)
 1991년 IBM T. J. Watson Research Center 방문연구원

1993년~현재 단국대학교 전기전자컴퓨터공학부 부교수
 2001년~2002년 Univ. of California, Irvine 방문연구원

관심분야 : 데이터베이스, 객체지향 데이터베이스, 멀티미디어 데이터베이스, 멀티미디어 정보검색, 이동객체 데이터베이스

E-mail : ymnah@dku.edu