

MAF(Multimedia Application File Format) 기반 멀티미디어 검색 시스템의 설계 및 구현

(A Design and Implementation of Multimedia Retrieval System based on MAF(Multimedia Application File Format))

강 영 모 [†] 박 주 현 [†] 방 형 진 [†] 낭 종 호 ^{**} 김 형 철 ^{***}
(Youngmo Gang) (Joohyoun Park) (Hyunggin Bang) (Jongho Nang) (Hyungchul Kim)

요약 최근 MPEG-A(ISO/IEC 23000)에서는 MAF(Multimedia Application File Format)[1]라는 새로운 멀티미디어 파일 포맷을 제안하고 있다. MAF 파일포맷은 기존의 MPEG 압축 규격을 갖는 미디어 파일에 더해, 미디어 내용 정보를 기술하는 메타데이터를 MPEG-7을 사용하여 하나의 시스템 포맷 내에 포함할 수 있도록 해준다. 하지만 현재로서는 이러한 표준을 구현한 실제 시스템이 없기 때문에, 응용 측면에서 MAF가 얼마나 유용한지 검증하기 어렵다는 문제점을 가지고 있다. 본 논문에서는 PC와 모바일 단말기상에서 MPEG-A 표준을 기반으로 한 멀티미디어 검색 시스템을 설계 및 구현한다. 또한 MPEG-A에서는 아직 동영상에 관련된 메타데이터에 대해서는 제안된 바가 없기 때문에 MPEG-7 MDS(Multimedia Description Scheme)[2]와 TV-Anytime[3]을 기반으로 모바일 환경에서도 저작 및 사용하기 편리한 메타데이터를 설계한다. 전체 시스템을 설계하기 위해서 MAF 검색 시스템이 가져야 할 5가지 요구사항인 이식성, 확장성, 호환성, 적응성, 효율성을 정의하고 이러한 요구사항을 바탕으로 하여 MAF 검색 시스템의 전체 구조를 응용 계층, 미들웨어 계층, 플랫폼 계층으로 설계한다. 제안하는 검색 시스템은 MAF 형식의 멀티미디어 데이터를 저작, 재생, 검색할 수 있는 클라이언트 파트와, 대용량의 MAF 형식의 멀티미디어 파일과 MAF 파일로부터 추출한 MPEG-7 형태의 메타데이터를 저장, 관리할 수 있는 서버 파트로 구성된다. 그리고 설계한 MAF 검색 시스템의 유용성을 검증하기 위하여, PC에서는 MS 윈도우 플랫폼 그리고 모바일 단말기에서는 WIPI 플랫폼 상에서 클라이언트 시스템을 구현하고 시스템 요구사항들을 모두 만족하는지 여부를 확인한다. 제안하는 검색 시스템은 MPEG-A 표준을 검증하고 그 유용성을 확인하기 위해 사용될 수 있다.

키워드 : 멀티미디어 검색, MAF, MPEG-A, 동영상 메타데이터

Abstract Recently, ISO/IEC 23000 (also known as "MPEG-A") has proposed a new file format called "MAF(Multimedia Application File Format)[1]" which provides a capability of integrating/storing the widely-used compression standards for audio and video and the metadata in MPEG-7 form into a single file format. However, it is still very hard to verify the usefulness of MPEG-A in the real applications because there is still no real system that fully implements this standard. In this thesis, a design and implementation of a multimedia retrieval system based on MPEG-A standard on PC and mobile device is presented. Furthermore, an extension of MPEG-A for describing the metadata for video is also proposed. It is selected and defined as a subset of MPEG-7 MDS[4] and TV-anytime[5] for video that is useful and manageable in the mobile environments. In order to design the multimedia retrieval system based on MPEG-A, we define the system requirements in terms of portability,

· 이 논문은 정보통신부에서 시행한 산업경쟁력강화사업의 연구비 지원으로 수행하였습니다.

[†] 학생회원 : 서강대학교 컴퓨터학과
jec1979@mlneptune.sogang.ac.kr
jeen798@mlneptune.sogang.ac.kr
parkjh@sogang.ac.kr

^{**} 종신회원 : 서강대학교 컴퓨터학과 교수

jhnang@sogang.ac.kr

^{***} 종신회원 : (주)씬멀티미디어(TMI) 이사

hckim@thinmultimedia.co.kr

논문접수 : 2006년 5월 17일

심사완료 : 2006년 8월 16일

extensibility, compatibility, adaptability, efficiency. Based on these requirements, we design the system which composed of 3 layers: Application Layer, Middleware Layer, Platform Layer. The proposed system consists of two sub-parts, client-part and server-part. The client-part consists of MAF authoring tool, MAF player tool and MAF searching tool which allow users to create, play and search the MAF files, respectively. The server-part is composed of modules to store and manage the MAF files and metadata extracted from MAF files. We show the usefulness of the proposed system by implementing the client system both on MS-Windows platform on desk-top computer and WIPI platform on mobile phone, and validate whether it to satisfy all the system requirements. The proposed system can be used to verify the specification in the MPEG-A, and to proves the usefulness of MPEG-A in the real application.

Key words : Multimedia Retrieval, MAF, MPEG-A, Video Metadata

1. 서론

최근 MPEG 그룹에서는 기존의 여러 압축포맷들을 갖는 여러 멀티미디어 파일들을 하나로 통합할 수 있는 MPEG-A[4]라는 새로운 국제 표준을 제정하고 있다. MPEG-A에서 제안하고 있는 MAF(Multimedia Application File Format)라는 새로운 파일포맷은 기존의 MPEG 압축 규격을 갖는 미디어 파일 뿐만 아니라, 해당 미디어의 내용 정보를 기술하는 메타데이터를 MPEG-7 [2]형태로 포함한다는 특징이 있다. 하지만 현재로서는 이러한 MAF 파일 포맷을 사용하여 직접 멀티미디어 파일을 생성하거나 검증해볼 수 있는 체계적인 시스템이 갖추어있지 않다. 따라서 MAF 형식으로 저장된 대량의 멀티미디어 데이터들을 생성(Creation), 교환(Exchange), 변경(Modification), 검색(Search), 재생(Play)에 이르기 까지, 사용자가 MAF 형식의 파일들을 효율적으로 관리할 수 있는 검색 시스템의 개발이 필요하다.

본 논문에서는 MAF 기반의 멀티미디어 검색 시스템을 설계 및 구현한다. 전체 검색 시스템에는 MAF 형식의 멀티미디어 데이터를 저작, 재생, 검색할 수 있는 클라이언트 시스템과, 대용량의 MAF 형식의 멀티미디어 파일들을 저장, 관리할 수 있는 서버 시스템을 포함하고 있다. 특히, 동영상에 관련된 메타데이터에 대해서는 아직 MPEG-A 그룹에서 제안한 바 없기 때문에 MPEG-7 MDS(Multimedia Description Scheme)[2]와 TV-Anytime[3]과 같은 기존의 동영상 관련 메타데이터들을 분석하여 검색에 필요한 동영상 관련 메타데이터 프로파일(Profile)을 설계한다. 그리고 설계한 MAF 검색 시스템의 유용성을 검증하기 위하여 윈도우와 핸드폰 환경에서 클라이언트 시스템을 구현하고, 특히 단말기 내의 검색을 위해서는 간단한 인덱스 파일을 구현하여 메타데이터의 빠른 검색이 가능하도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 사용할 MAF 파일 포맷과 기존의 동영상 메타데이터 사양들을 살펴본다. 3장에서는 본 시스템의 전체적인

구조를 설계하기 앞서, 동영상 검색을 하기 위해 필요한 메타데이터를 설계하도록 한다. 4장에서는 MAF 기반 멀티미디어 검색 시스템이 가져야 할 요구사항을 정의하고, 그 요구사항을 만족시키기 위해 전체 시스템의 구조를 설계한다. 5장에서는 4장에서 제안한 설계를 바탕으로, 실제 윈도우와 모바일 환경에서 구현한 멀티미디어 검색 시스템에 대해 설명하고 구현 결과를 분석한다. 마지막으로 6장에서는 본 연구에 대한 결과와 앞으로의 연구방향에 대해 서술하고 결론을 맺도록 한다.

2. 연구배경

이번 장에서는 MPEG-A에서 제안하는 MAF(Multimedia Application File Format)의 내부구조와 MPEG-7 MDS와 TV-Anytime과 같은 기존의 동영상 메타데이터 사양들을 살펴보도록 한다.

2.1 MAF(Multimedia Application File Format)

MPEG-A에서 제안하는 파일 포맷인 MAF는 ISO Base Media File Format[5]을 기반으로 한 파일 시스템이다. 현재 MPEG 그룹에서는 MPEG-4 시스템 포맷 [6]과 MPEG-21 시스템 포맷[7]을 모두 MAF로 사용할 수 있도록 권장하고 있다. 하지만, 표준화가 진행된 정도와 최근에 각광받고 있는 모바일 환경에서의 멀티미디어 검색을 고려했을 때, MPEG-4 시스템 포맷이 다양한 응용에 적용하기에 좀 더 보편적인 추세이다. 그림 1에서는 MPEG-4 시스템 포맷을 기반으로 한 MAF 파일 형식을 보여주고 있다. ISO Base Media File Format을 기반으로 한 시스템 포맷들의 특징은 각각 필요한 정보들을 담은 계층적인 박스들의 형태로 이루어져 있다는 점이다. 가장 상위 레벨의 세 개의 박스는 mdat 박스, moov 박스, meta 박스로 구분된다. 이 중, mdat 박스 내에는 실제 비디오나 오디오와 같은 미디어 스트림을 저장하는 박스이고, moov 박스는 mdat 내에 저장된 미디어 스트림에 관한 시간적, 공간적인 위치 정보와 코덱 정보들을 담은 박스이다. 그리고 meta 박

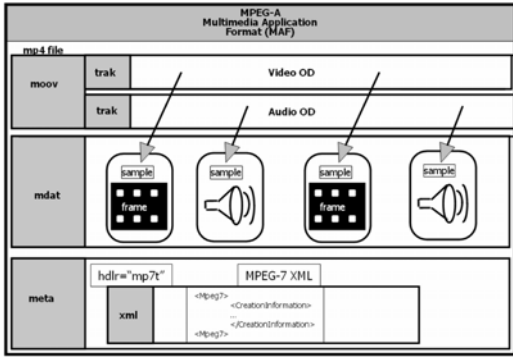


그림 1 MPEG-4 시스템 포맷을 사용한 MAF의 구조

스는 MPEG-7 XML 형태(바이너리 혹은 텍스트)로 메타데이터를 탑재할 수 있는 박스이다.

MPEG-A의 제2부 규격인 MPEG Music Player Application Format[8]과 Photo MAF Player[9]의 표준화 작업에서는 JPEG의 메타데이터의 형태인 EXIF (Exchangeable Image Format)[10]과 MP3의 메타데이터의 형태인 ID3[11]를 MPEG-7 XML 형식으로 사상(Mapping)하는 방법과, MAF에서 필요로 하는 의미적(Semantic)인 정보들을 MPEG-7 XML 형식으로 표현하도록 명시하고 있다. 하지만, 동영상에 관련된 메타데이터에 대해서는 아직 MPEG-A 그룹에서 제안한 바 없기 때문에, 본 논문에서는 기존의 동영상 관련 메타데이터들을 분석하여 검색에 필요한 동영상 관련 메타데이터 프로파일(Profile)을 설계한다.

2.2 동영상을 위한 기존의 메타데이터 사양들

멀티미디어 데이터의 내용을 기술하기 위한 표준으로서 기존에 가장 널리 사용되고 있는 것은, MPEG 그룹에서 제정한 MPEG-7 MDS(Multimedia Description

Scheme)와 디지털 방송의 핵심 기술에 대한 국제 표준을 주도하고 있는 TV-Anytime이다. TV-Anytime이 다루는 메타데이터의 특징은, 실제 방송 프로그램에 필요한 필수적인 메타데이터들을 대부분 MPEG-7 MDS에서 선택적으로 조합하여 표준화를 진행하고 있다는 점이다. 그림 2에서는 MPEG-7 MDS와 TV-Anytime에서 다루고 있는 메타데이터의 카테고리 및 범위를 설명해주고 있다. 본 논문에서는 TV-Anytime보다는 좀더 포괄적인 MPEG-7 MDS를 바탕으로 하여 MAF에 필요한 동영상 메타데이터를 설계한다.

3. 동영상을 위한 MAF 메타데이터 설계

본 장에서는 MAF 검색 시스템에서 필요한 메타데이터 요구사항을 바탕으로 MPEG-7 MDS를 분석하고, 필요한 동영상 메타데이터를 설계하도록 한다.

3.1 요구사항 정의

- **관리용이성(Manageability)**: 모바일 환경에서 사용자가 저작하기 용이해야 한다.
- **간결성(Compactness)**: 모바일 환경에 적합하게 메타데이터의 표현이 간결해야 한다.
- **유용성(Availability)**: 모바일 환경에서 검색할 때 유용하게 사용할 수 있어야 한다.
- **보편성(Generality)**: 특정 분야에 해당하는 메타 데이터보다는 다양한 분야에서 공통적으로 사용될 수 있어야 한다.
- **자동성(Automaticity)**: 최신 멀티미디어장치(디지털 카메라, 핸드폰 등)로부터 생성되는 메타데이터들을 적극 활용해야 한다.

3.2 동영상 메타데이터의 설계

MPEG-7 MDS의 총 5가지 카테고리를 각각은 멀티미디어 콘텐츠에 대해 서로 다른 메타데이터를 기술하

Simple Types Complex Types	Media Info. Creation & Production Info. Usage Info.	Structure Info. Semantics Info.	Summaries Info. Views Info. Variations Info.	Collections Info. Models Info.	History Info. Preference Info.
TV-Anytime Basic Types	TV-Anytime Program Genre/Format/ Locator, etc.	TV-Anytime Temporal Segment Type	TV-Anytime Summary Type		TV-Anytime User History / Preference Type
MPEG-7 Basic Elements	MPEG-7 Content Metadata	MPEG-7 Content Description	MPEG-7 Navigation&Access	MPEG-7 Content Organization	MPEG-7 User Interaction

그림 2 MPEG-7 MDS와 TV-Anytime에서 메타데이터 포함관계

고 있다. 이 중 Navigation & Access, Content Organization, User Interaction 카테고리들은 정의한 요구사항의 만족도 측면에서 볼 때, 사용자 입장에서 저작이 힘들고 보편적인 검색에는 사용할 수 없는 메타데이터를 포함하고 있다. 따라서 본 논문에서는 동영상 메타데이터를 위한 MPEG-7 MDS의 카테고리로서, Content Meta-data와 Content Description 카테고리를 선정하였다.

동영상 메타데이터 설계 시에 한 가지 더 고려해야 할 사항으로는, 이렇게 선택한 MPEG-7 MDS 카테고리 중에서도 가급적 최소한의 메타데이터 기술자(Descriptor)만을 선택하여, 최종적으로는 하나의 SMP(Simple Metadata Profile)의 형태로 구성해야만 한다는 점이다. MPEG-A 그룹에서는 'Jon4'[9]에 필요한 메타데이터 선정을 위해 MPEG-7 MDS의 Content Descriptor 카테고리에서 Who, Where, When, WhatObject, WhatAction, Why, How와 같은 의미적인 기술자들을 SMP에 포함하도록 하였다. 본 논문에서는 앞서 분석을 통해 선택한 Content Metadata, Content Descriptor 카테고리에 속한 여러 세부 기술자들을 중심으로, 동영상 메타데이터를 위한 SMP를 설계하였다. 다음은 Content Meta-data, Content Descriptor 카테고리 중에서, 다시 한번 요구사항을 바탕으로 하여 동영상 SMP에 포함하거나 배제한 세부기술자들에 대한 설명이다.

• Content Metadata 카테고리

• **Media Descriptors:** 미디어 정보, 미디어 구분, 미디어 포맷, 미디어 품질(Quality) 정보-보편적(General)인 정보이기는 하지만, 사용자가 저작하기 불편하고 ISO Based File Format을 기반으로 한 MPEG-4 시스템 파일포맷, MPEG-21 시스템 파일포맷 내부에는 이러한 정보들을 담기 위한 moov 박스가 이미 존재한다.

• **Creation & Production Descriptors:** 미디어 제목(Title), 미디어 장르(Genre), 미디어 가이드(Guide), 미디어 생성자(Author), 미디어 생성 시간(Creation Time), 미디어 생성 장소(GPS Info.)-동영상의 내용을 표현하기에 보편적인 내용으로써, 이미 TV-Anytime의 경우에도 이러한 기술자들을 포함하고 있는 추세이다. 이 중 일부는 미디어의 저작자에 의도를 보호하기 위한 메타데이터들이다. PG(Parental Guide) 또는 Copyright 기술자들은 동영상 메타데이터에 있어서 보편적이고 유용하게 사용될 수 있는 메타데이터들이다.

• **Usage Descriptors:** 사용 권한(Availability), 비용(Financial), 유포(Dissemination)에 관련된 기술자-멀티미디어 데이터에 대한 저작권 문제, 사용 요금에 대해서 자세한 기술자들을 포함하고 있어서 콘텐츠 제공자(Contents Provider)에게 유용하게 사용될 수 있다.

• Content Description 카테고리

• **Structure Descriptor:** 공간적인 세그먼트(Segment), 샷(Shot) 또는 신(scene) 정보-사용자가 모바일 단말기 상에서 저작하기 불리할 뿐만 아니라, 비디오 클립을 대상으로 할 경우에는 검색할 때 유용하게 사용하기 힘들다.

• **Semantic Descriptors:** 누가(Who), 언제(When), 어디서(Where), 무엇(What)을, 어떻게(How), 왜(Why)와 같은 의미적인 정보-보편적인 개념이며, 시각적 미디어인 동영상의 검색에서도 유용하게 활용될 수 있다.

앞서 분석을 통해 최종적으로 설계한 동영상 메타 데이터는 표 1과 같다.

4. MAF 기반 멀티미디어 검색 시스템의 설계

본 장에서는 MAF 기반의 멀티미디어 검색 시스템이 가져야 할 요구사항들을 정의하고, 요구사항을 근거로 하여 검색 시스템의 구조를 설계하도록 한다. 검색 시스템은 크게 2가지 부분으로, 저작 도구, 검색 도구, 재생 도구를 포함하는 MAF 클라이언트 시스템과, 저작한 MAF 파일들을 저장하고 클라이언트로부터 검색 질의(Search Query)를 받아 검색을 수행하는 MAF 검색 서버로 구성된다.

4.1 MAF 기반 클라이언트 시스템 설계

4.1.1 요구사항 정의

본 논문에서 제안하는 MAF 기반의 멀티미디어 검색 시스템은 PC뿐만 아니라, 핸드폰이나 PDA와 같은 모바일 환경에서의 멀티미디어 검색을 대상으로 하고 있다. 클라이언트 시스템의 설계에 앞서, 시스템 측면에서 필요한 이러한 요구사항들을 표 2와 같이 분석 및 정의하였다.

4.1.2 MAF 기반 멀티미디어 단말기의 시스템 구조

MAF 클라이언트 시스템은 앞서 살펴본 요구사항을 바탕으로 하여, 계층적인 접근방식(Layered Approach)을 통해서 그림 3과 같이 설계하였다. 전체 시스템은 크게 3개의 계층 응용 계층, 미들웨어 계층, 플랫폼 계층으로 구성된다, 특히 미들웨어 하부에 위치한 PAL(Platform Abstraction Layer) 계층에서는 외부적으로 동일한 API를 제공하면서 동시에, 내부적으로는 실제 사용하고자 하는 플랫폼에 해당하는 API로 매핑해주는 역할을 담당한다. 따라서 구현 시에 필요한 이식성(Portability)과 확장성(Extensibility)을 높여주는 장점이 있다.

• MAF 응용 계층

MAF 저작/검색/재생 도구와 같이, PC 혹은 모바일 단말기 상에서 사용자가 사용하는 응용 프로그램을 포함한다. 응용 계층에서는 MAF 미들웨어 계층의 모듈들 중에서 필요한 모듈들을 선택적으로 조합하여 구성된다.

표 1 본 논문에서 제안하는 동영상을 위한 메타데이터들

Metadata Name	Description	MPEG-7 Path
Title	Title of Video Clip	CreationInformation/Creation/Title
Author	Creator—who take Video Clip	CreationInformation/Creation/Creator
Creator	Description (not an end-user annotation)	CreationInformation/Creation/Abstract/FreeTextAnnotation
Description		
Genre	Genre of Video clip	CreationInformation/Classification/Genre
Date & Time	Date/Time	CreationInformation/Creation/CreationCoordinates/Date
GPSLatitudeRef	Where was Video Clip taken i.e., location generated from devices	CreationInformation/Creation/CreationCoordinates/Location/GeographicPosition/Point/[longitude="", latitude="", altitude=""]
GPSLatitude		
GPSLongitudeRef		
GPSLongitude		
GPSAltitudeRef		
GPSAltitude		
UserComment	Free Annotation by user	CreationInformation/Creation/Abstract/FreeTextAnnotation
Copyright	Copyright	CreationInformation/Creation/CopyrightString
Rating	Parental Guide	CreationInformation/Classification/ParentalGuidance/ParentalRating
Who	Who in Video Clip	CreationInformation/Creation/Abstract/StructuredAnnotation/Who
When	Who in Video Clip	CreationInformation/Creation/Abstract/StructuredAnnotation/When
Where	Where took a Video Clip	CreationInformation/Creation/Abstract/StructuredAnnotation/Where
What	WhatActions in Video Clip	CreationInformation/Creation/Abstract/StructuredAnnotation/WhatObject
Why	Why took a Video Clip	CreationInformation/Creation/Abstract/StructuredAnnotation/Why
How	How took a Video Clip	CreationInformation/Creation/Abstract/StructuredAnnotation/How
Right	Rights for Video Clip	UsageInformation/Availability/Rights
Financial	Financial Info for Video Clip	UsageInformation /Availability/FinancialResults/AccountItem
Disseminator	Disseminator for Video Clip	UsageInformation /Availability/Dissemination/Disseminator

표 2 MAF 기반 멀티미디어 단말기 설계를 위한 요구사항

요구사항	요구사항 설명
이식성(Portability)	데스크탑과 모바일 단말기 상에 동일하게 적용이 가능해야만 한다. MAF 저작 도구나 검색 도구 및 재생 도구는 모바일 단말기에 탑재가 가능해야 한다.
적용성(Adaptability)	시스템의 요구사항이 변경되었을 때, 그 요구사항들을 쉽게 시스템에 반영할 수 있는 구조를 가져야 한다. 예를 들어, 메타데이터의 변경이 있을 경우 전체 시스템의 큰 변경없이도 변경된 메타데이터를 적용할 수 있어야 한다.
확장성(Extensibility)	MAF 클라이언트 시스템에 필요한 기능들을 모듈화하고 모듈의 재사용을 통해, 새로운 어플리케이션의 개발이 쉽도록 한다. 다시 말해, 본 시스템에서 제공하는 핵심 모듈을 조합하여 MAF에 필요한 새로운 응용 도구를 구성할 수 있는 특성을 말한다.
호환성(Compatibility)	본 시스템에서 생성한 MAF 파일이 기존의 다른 시스템들과 호환 가능해야 한다. 저작도구에서 생성하게 되는 MAF 파일들은 기존의 다른 MP4 재생기에서 MPEG-7 메타데이터를 제외하더라도, 미디어 자체는 재생이 가능해야만 한다.
효율성(Efficiency)	MAF 저작, 재생, 검색과 관련된 모든 핵심 모듈들은 시간과 리소스 측면에서 빠르고 효율적으로 동작해야만 한다.

예를 들어, 응용 계층에서의 MAF 저작도구는 하위 계층에 있는 ID3/EXIF 추출기, MAF 시스템 인코더/디코더, MPEG-7 XML 인코더/디코더, 3on4 포맷터/디포맷터, 재생 모듈, 네트워크 모듈들을 조합한 다음, 각각의 모듈이 제공하는 API를 사용하면 된다.

• **MAF 미들웨어 계층**

MAF 미들웨어 계층은 MAF 검색 시스템을 구현하는 데에 필요한 공통 핵심 모듈들이 담겨있는 계층이다.

• **ID3/EXIF Extractor**

MP3 파일 내에 포함된 ID3 태그 혹은 JPEG 파일 내에 포함된 EXIF 태그를 추출하는 모듈이다.

• **MPEG-7 XML Coder/Parser**

메타데이터를 표준[8,9]에 따라 MPEG-7 XML 형태

로 인코딩하거나, MPEG-7 XML을 파싱하기 위한 모듈이다.

• **Extended MPEG-4 System Encoder/Decoder**

이 모듈은 미디어 스트림(JPEG, MP3, MP4 Video/Audio)과 MPEG-7 XML 스트림을 MAF 형식으로 인코딩 혹은 디코딩하기 위한 모듈이다.

• **MP3onMP4 Formatter/Deformatter**

MP3 포맷을 MAF 형식으로 인코딩하기 위해서 MP3의 프레임들을 MP4에서의 개별적인 AU(Access Unit) 단위로 변환해주는 Formatter와, 반대로 디코더 측에서는 AU들을 원래의 MP3 스트림으로 역 변환해주는 Deformatter를 포함한다[12].

• **Player Module**

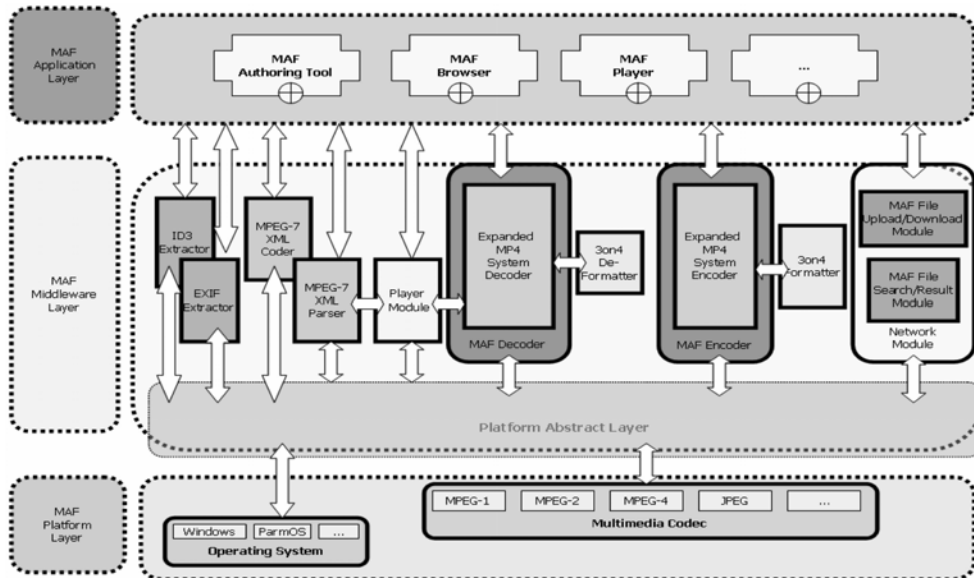


그림 3 MAF 기반 멀티미디어 단말기를 위한 시스템 구조

All-In-One 재생기를 위한 모듈로써, 기존의 JPEG, MP3, MP4파일과 모든 MAF 형식의 파일이 재생 가능하다. 재생모듈에서는 MPEG-7 XML 파서와 MAF 디코더 모듈을 사용한다.

• Network Module

MAF 파일의 등록 및 검색을 위한 네트워크 관련 API를 제공해준다. MAF 파일의 업로드(등록)와 질의 메시지의 전송을 통한 MAF 파일의 검색, 그리고 서버로부터 검색된 결과를 돌려 받는 기능, 사용자가 임의로 선택한 MAF 파일을 서버로부터 다운로드 받는 기능들이 모두 이 모듈에 포함되어 있다.

• MAF 플랫폼 계층

MAF 플랫폼 계층은 실제 MAF 검색 시스템이 탑재된 환경을 말한다. 이 계층에서는 윈도우, 리눅스와 같은 OS와 해당 OS에서 제공하는 서비스들, 그리고 WIPI나 BREW와 같이 무선 단말기에서 제공하는 여러 플랫폼들을 포함한다. 그리고 각 플랫폼마다 사용하는 다양한 멀티미디어 코덱들(소프트웨어, 하드웨어)이 이 계층에 포함된다.

4.2 검색 서버의 설계

MAF 검색 서버는 사용자가 클라이언트 저작도구를 통해 생성된 MAF 파일들과 메타데이터를 저장하고, 클라이언트 검색도구를 통해 검색 질의(Query)를 요청하면 관련된 MAF 파일의 리스트를 사용자에게 돌려주는 기능을 담당한다. 서버는 크게 원격 서버와 로컬 서버로 구분할 수 있다.

4.2.1 원격 서버와 로컬 서버

• 원격 서버의 설계

원격 서버의 경우에는 상용 DBMS를 사용하여, 저장된 MAF 파일의 메타데이터를 별도의 테이블 형태로 저장한 구조를 갖는다. MAF 파일의 메타데이터들은 대부분 키워드 형식이기 때문에, 문자열 타입의 변수들로 표현할 수 있다. 원격 서버에서 필요한 DB 스키마는 Jon4[8], 3on4[9]의 메타데이터 표준과 본 논문에서 설계한 동영상 메타데이터를 사용하도록 설계하였다.

• 로컬 서버의 설계

로컬 서버란 대용량의 스토리지를 갖는 모바일 장치 내에 저장된 MAF 파일을 검색하는 경우를 말한다. 모바일 환경에서는 리소스의 제약을 감안하여, 가급적 단순한 방식의 저장 방법을 사용하면서 동시에 검색 속도를 향상시킬 수 있는 별도의 인덱싱 방법을 사용해야만 한다. 가장 단순하면서도 널리 사용되는 인덱싱 방법으로는 이진 트리 기반(Binary Tree-Based)의 방법과 해싱 기반(Hashing-Based)의 방법이 있다. 두 가지 인덱싱 방법을 단말기 환경에서 적용하기 위해서는 먼저, 텍스트 검색을 했을 때 각각의 방법에 대한 검색 시간과 최대 메모리 요구량을 비교해볼 필요가 있다. 표 4에서는 가장 단순한 해싱 방식(Radix Hashing, Bit-Wise Hashing)과 가장 단순한 이진 트리(Binary Tree, Splay Tree) 방식을 사용하여 텍스트 데이터에 대한 인덱스 파일을 구성한 다음, 데이터베이스 크기에 따라서 검색 시간(초) 및 최대 메모리 요구량(Mb)을 측정한 결과[13]이다. 실험에 사용된 텍스트 데이터베이스의 크기는 표 3과 같다. 표 4를 통해 알 수 있는 특징 중 하나

표 3 텍스트 데이터베이스의 크기 [13]

	Small AP	TREC1	Stopwords	Small Web	Large Web
Size(Mb)	23	1,206	292	2,146	31,745
Distinct words	74,439	618,443	589	1,335,011	9,211,024
Word Occurrences	3,693,936	183,710,119	74,940,753	238,894,460	2,009,762,446

표 4 검색 속도(초)와 최대 메모리 요구량(괄호, MB) [13]

	Small AP	TREC1	Stopwords	Small Web	Large Web
Binary Tree	9.3(2)	521.9(13)	89.2(1)	1126.4(30)	6232.0(208)
Splay, all searches	12.5(2)	605.4(15)	120.8(1)	1877.1(35)	7170.0(243)
Splay, intermittent	10.1(2)	494.5(15)	88.2(1)	1405.9(35)	6107.0(243)
Radix Hashing	5.6(2)	288.8(15)	71.0(5)	1408.7(28)	3127.0(177)
Bit Wise Hashing	2.5(2)	123.9(15)	30.2(5)	404.1(28)	1361.1(177)

는, 단어의 수가 적은 환경(Stopwords)일수록 해싱 기반의 방법이 이진 트리 기반의 방법보다 검색 속도가 빠르다는 점이다. 그리고 요구하는 메모리의 양은 두 가지 방법 모두 비슷하지만, 해싱의 경우는 해싱 테이블의 크기가 고정된다는 문제점이 있기 때문에, 단어의 수가 적음에도 불구하고 일정한 양의 메모리를 소모한다는 사실을 알 수 있다.

특히 모바일 환경 즉, 로컬 서버의 경우에는 단말기 내부에 저장된 MAF 파일만을 검색 대상으로 하기 때문에, 저장되는 메타데이터의 크기가 비교적 크지 않다는 특징이 있다. 따라서 이러한 특성을 고려했을 때, 보다 빠른 로컬 검색을 위해서는 이진 트리 기반의 인덱싱 방법보다는 해싱 기반의 인덱싱 방법을 선택하는 것이 자원 활용 측면에서 효율적이다. 최대 메모리 요구량을 감안했을 때에도, 작은 크기의 데이터베이스 환경에서 두 방법 모두가 5MB 이하의 메모리를 필요로 하기 때문에, 해싱 기반의 인덱싱 방법을 사용하더라도 대부분의 단말기 환경에서 큰 문제없이 동작이 가능하다. 본 논문에서는 이러한 근거를 바탕으로 하여, 로컬 서버의 검색을 위한 해싱 기반의 인덱싱 파일을 WIPI 환경에서 구현하였고 성능을 분석하였다.

5. 구현 및 분석

본 장에서는 앞서 4장에서 설계한 MAF 검색 시스템의 유용성을 검증하기 위하여 윈도우와 핸드폰 환경에서 클라이언트 시스템을 구현하고 구현 결과를 분석한다.

5.1 윈도우 상에서의 구현

MAF 저작 도구의 구현 환경은 펜티엄-IV 프로세서를 가진 PC의 MS-Visual C++ 6.0을 사용하였으며, 미들웨어의 모든 핵심 공통 모듈들은 .c 파일로 구현하였고 GUI에 해당하는 부분은 .cpp 파일(MFC 사용)로 구현하였다.

윈도우 버전의 저작 및 검색도구의 GUI는 그림 4와

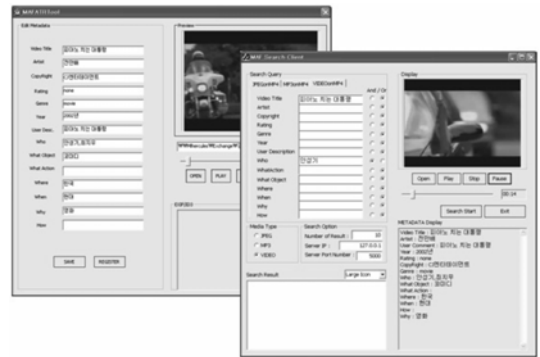


그림 4 MAF 저작 및 검색 도구의 GUI(윈도우 버전)

같다. 저작 도구의 입력으로는 MP3 File / JPEG File / MAF File의 셋 중 하나의 형태가 주어지며, MP3나 JPEG의 경우 ID3 추출 모듈과 EXIF 추출 모듈을 통해 추출된 메타데이터를 GUI상에 표시하고, 동시에 재생 모듈을 통해 MP3 스트림이나 JPEG 이미지가 재생되게 된다. MAF 검색 도구에서는 사용자의 편의를 위해 Jon4, 3on4, MP4 동영상에 따른 메타데이터의 입력 창을 각각 별도의 탭으로 구성하였다. 예를 들어 찾고자 하는 MAF에 관련된 메타데이터를 키워드로 입력하면 서버에서는 각각의 키워드에 따라 검색된 MAF파일들의 메타데이터만을 먼저 사용자에게 보여주게 된다. Jon4와 MP4비디오 형식의 MAF 파일의 경우에는 서버에서 썸네일 이미지(Thumbnail Image)를 전송함으로써, 사용자가 원하는 MAF 파일의 여부를 직접 확인하도록 하였다. 사용자는 클라이언트 검색 도구를 통해 모든 메타데이터의 필드의 값을 입력하거나, 부분적으로 필요한 필드만을 입력할 수 있다. 검색 시에는 비교 연산(AND 또는 OR 연산)을 조합하여 질의(Query)를 전송할 수 있도록 구현하였기 때문에, 다양한 검색 시나리오 역시 가능하다. 또한 검색 도구에 서버 IP 주소를 입력할 수 있는 GUI를 별도로 추가함으로써, 클라이언트

는 상황에 따라 원격 서버 또는 로컬 서버를 선택하여 검색을 진행할 수 있다.

5.2 WIPI 플랫폼 상에서의 구현

WIPI 플랫폼 상의 구현에서는 윈도우 버전에서 이미 구현한 핵심 모듈을 코드 수정 없이, ARM Developer Suite(ADS) Version 1.2와 WIPI Binary Maker Version 1.2[14]를 통해 폰 상에서 다운로드 가능한 어플리케이션 파일(.daf)로 생성하였다. 사용한 단말기의 버전은 SKY IM-8500으로, 본 시스템에서 필요로 하는 미디어 파일들인 JPEG과 MP3, MP4 비디오의 재생을 지원해준다. 그림 5는 핸드폰 상에서 구현한 MAF 저작도구를 보여주고 있다.

5.3 검색 서버의 구현

본 논문에서는 원격 서버의 구현을 위해 MS-SQL 2000 Personal Edition을 사용하였고 Jon4[8], 3on4[9]의 메타데이터 표준과 본 논문에서 설계한 동영상 메타데이터를 사용하여 DB 스키마를 구성하였다.

본 구현에서는 모바일 환경에서의 검색 속도 문제를 해결하기 위하여, 해싱 기반의 인덱싱 방법을 사용하여 로컬 서버를 구현하였다. 참고한 소스코드는 QDBM(Quick Database Management)[16]으로써, 소규모의 데이터 베이스에 적합한 간단한 API를 제공하는 DBMS 도구이다. 하지만 본 논문에서는 모바일 상에서의 검색과 이식성 측면을 고려하여, QDBM에서 해싱 기능을 제공하는 API만을 별도로 추출하여 모듈화하였다. 따라서 본래의 QDBM 소스코드 크기는 806KB이나, 간소화한 해싱 관련 소스 코드의 크기는 76KB로 줄일 수 있었다.

5.4 분석

이번 절에서는 시스템 설계 시에 고려한 요구사항을 바탕으로 전체 시스템을 분석한다. 시스템 설계 시에 고려한 요구사항은 5가지로써, 이식성, 적응성, 확장성, 호환성, 효율성이다.

먼저 이식성(Portability) 측면에서 살펴봤을 때, 본 시스템의 미들웨어에서 제공하는 PAL(Platform Abstraction Layer)[15] API들은 PC와 모바일 단말기의 운영체제에 상관없이 MAF 클라이언트 모듈에 필요한 파일, 메모리, 네트워크 기능들을 원활하게 제공해준다. 실제로 본 논문에서 구현한 모든 모듈들은 PAL에서 제공하는 API를 사용하고 있으며, 따라서 윈도우 버전과 모바일 버전의 클라이언트 시스템을 구현할 때에도 코드의 수정 없이 동일한 인터페이스를 통해 구현할 수 있었다. 또한 PAL 계층에서는 윈도우와 WIPI 플랫폼뿐만 아니라, 최근에 모바일 플랫폼으로써 각광받고 있는 BREW와 SYMBIAN 등에 대해서도 필요한 기능들을 제공할 수 있다.

적응성(Adaptability)은 MAF 검색 시스템에서 사용하는 메타데이터의 변경이 있더라도, 코드의 수정 없이 변경사항을 시스템에 반영하기 위한 요구사항이다. 본 시스템에서는 이러한 적응성 측면을 고려하기 위하여, 메타데이터를 정의한 별도의 스키마 파일을 입력으로 받도록 설계 및 구현하였다. 따라서 동영상에 필요한 메타데이터가 앞으로 표준화되거나 응용에 따라 필요한 메타데이터를 정의하더라도, 본 시스템에서는 스키마 입력 파일만 수정함으로써 변경사항을 쉽게 반영할 수 있

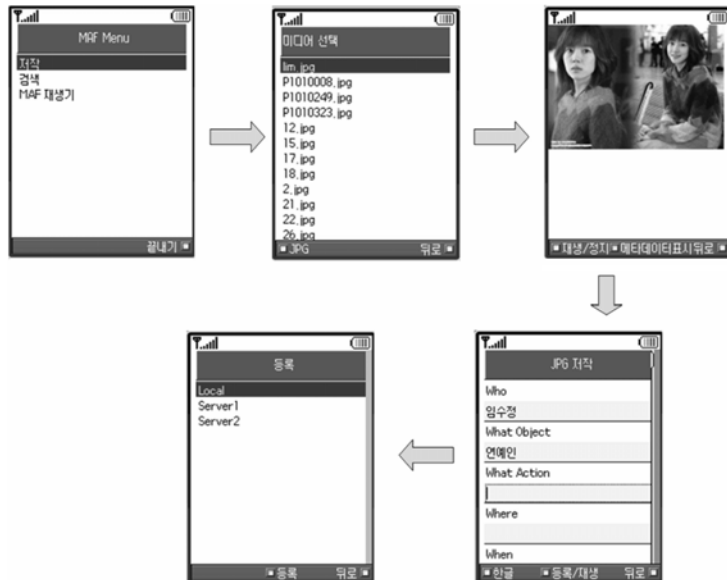


그림 5 WIPI 플랫폼 상에서의 저작도구 GUI

는 특징이 있다.

확장성(Extensibility)은 본 시스템에서 설계 및 구현한 핵심 모듈을 사용하여 MAF와 관련된 응용 프로그램의 개발을 보다 효율적으로 하기 위한 요구사항이다. 본 시스템에서 제공하는 핵심 모듈들을 사용하거나 기능에 따라 필요한 모듈(예를 들어, 카메라 캡춰 모듈)을 미들웨어 계층에 추가한다면, 좀 더 효율적이고 빠른 어플리케이션의 개발이 가능하다.

호환성(Compatibility)은 본 시스템에서 생성한 MAF 파일이 기존의 다른 재생도구에서 동일하게 사용할 수 있는 특성을 말한다. 본 시스템에서는 MP4 시스템 파일 포맷을 사용하여 MAF 파일을 생성하기 때문에, MP4 재생이 가능한 기존의 재생 도구들을 사용하여 호환성을 검증하였다.

효율성(Efficiency) 분석을 위해서는 본 논문에서 구현한 해시 기반의 인덱싱 방법과 로그 방식의 인덱싱 방법(메타 데이터를 순차적으로 파일에 쓴 경우)을 검색 속도 및 인덱스 파일 크기의 관점에서 비교하였다. 그림 6에서와 같이, 단말기 내에 저장된 MAF 파일의 개수가 700개 미만일 때에는 로그 방식이 더 빠른 검색 성능을 보이지만, MAF 파일의 개수가 700개 이상일 경우에는 본 논문에서 구현한 해시 기반의 인덱싱 방법이 탁월한 성능을 보인다. 그림 7에서는 두 방법을 사용했을 때 생성되는 인덱스 파일의 크기를 비교한 것이다. MAF 파일의 개수에 상관없이 모든 경우에 대해서, 해시 기반의 인덱스 파일이 로그 방식의 인덱스 파일보다 크기 면에서 5~8배 크다는 사실을 알 수 있다. 하지만 이러한 인덱스 파일의 크기는 실제 단말기 내에 저장된 MAF 파일들의 크기와 비교했을 때, 상대적으로 아주 작은 크기임이 분명하다. 예를 들어, 3on4의 파일(평균 4MB 가정)이 단말기 내에 3000개 있다고 가정(전체 12GB)했을 때, 인덱스 파일(0.9MB)의 크기는 상대적으로 0.008% 밖에 저장 공간을 차지하지 않는다. 따라서, 본 논문의

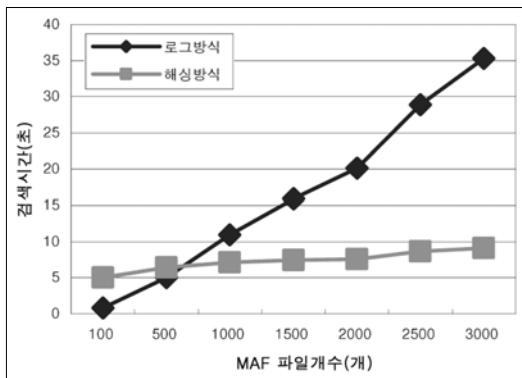


그림 6 로컬 서버의 인덱싱 검색 속도 분석

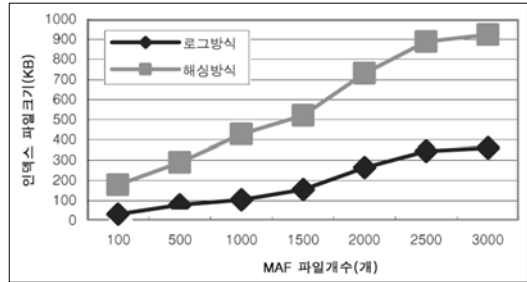


그림 7 로컬 서버의 인덱스 파일 크기 분석

구현한 해시 기반의 인덱싱 방법은 대량의 MAF 파일을 저장할 수 있는 모바일 단말기일수록 효율성이 높은 반면, 작은 크기의 스토리지를 갖는 장치에서는 상대적으로 효율성이 떨어진다는 사실을 알 수 있다.

6. 결론

최근 MPEG 그룹에서는 기존의 여러 압축포맷들을 갖는 멀티미디어 파일들을 하나로 통합할 수 있는 MPEG-A라는 새로운 국제 표준을 활발하게 제정하고 있다. 이러한 MPEG-A에서 제안하고 있는 MAF 파일 형식은 사용자가 멀티미디어 파일을 생성 즉시, 키워드를 통해 직접 메타데이터를 입력하고 그것을 파일 포맷 내에 저장할 수 있도록 해준다. 하지만 현재로서는 MAF 파일 포맷 내에 저장된 이러한 메타데이터를 활용하여 직접 멀티미디어 파일을 검색해볼 수 있는 체계적인 시스템이 갖추어지지 않았다. 본 논문에서는 MAF 형식으로 저장된 대량의 멀티미디어 데이터들을 생성, 교환, 변경, 검색, 재생에 이르기까지, 사용자가 MAF 형식의 파일들을 효율적으로 관리할 수 있는 검색 시스템을 설계 및 구현하였다.

본 논문에서는 MAF 형식의 멀티미디어 데이터를 저장, 재생, 검색할 수 있는 클라이언트 시스템과, 대용량의 MAF 형식의 멀티미디어 파일들을 저장, 관리할 수 있는 서버 시스템을 설계하였다. 본 검색 시스템에서 다루고 있는 MAF 파일들은 현재 MPEG-A 그룹에서 제안한 3on4, Jon4 형식을 포함하고 있으며, 추가적으로 MP4 비디오 형식의 MAF 파일을 포함하도록 하였다. 특히, 동영상에 관련된 메타데이터에 대해서는 아직 MPEG-A 그룹에서 제안한 바 없기 때문에, 본 논문에서는 기존의 동영상 관련 메타데이터들을 분석하여 검색에 필요한 동영상 관련 메타데이터를 설계하였다. 전체 시스템을 설계하기 위해서는, MAF 검색 시스템이 가져야 할 5가지 요구사항인 이식성, 확장성, 호환성, 적응성, 효율성을 정의하였다. 그리고 이러한 요구사항을 바탕으로 하여 MAF 검색 시스템의 전체 구조와 필요한 핵심 모듈들을 설계하였다. 그리고 설계한 MAF 검

색 시스템의 유용성을 검증하기 위하여 윈도우와 단말기 환경에서 클라이언트 시스템을 구현하였고, 특히 단말기 내의 검색을 위해서는 간단한 인덱스 파일을 구현하여 메타데이터에 대한 빠른 검색이 가능하도록 하였다. 그리고 구현을 통해 각 핵심모듈의 동작 여부와 함께 시스템 설계 시에 정의한 요구사항들을 모두 만족하는지 여부를 확인하였다.

본 논문에서 설계 및 구현한 MAF 검색 시스템을 활용도 측면에서 본다면, PC 혹은 단말기 환경에서 사용자가 생성한 MAF 파일들을 그 즉시 저작하여 서버 혹은 단말기 내에 저장한 다음, 추후 의미적인 키워드를 통해 검색하고 재생까지 할 수 있는 멀티미디어의 획기적인 관리 및 검색 시나리오가 가능하다. 또한 본 논문에서 다른 MAF 저작도구/검색도구/재생도구 이외에도 MAF를 이용한 또 다른 응용 프로그램을 개발할 때, 본 시스템의 미들웨어에서 제공하는 핵심 모듈들을 이용하거나 기능에 따라 필요한 모듈(예를 들어, 카메라 캡처 모듈)을 미들웨어 계층에 추가한다면, 좀 더 효율적이고 빠른 어플리케이션의 개발이 가능하다.

앞으로의 연구에서는 모바일 플랫폼 상의 제약을 고려하여 좀 더 효율적인 인덱싱 방법이 제안되어야 할 것이다. 또한 본 논문에서 제안한 검색 시스템에서는 메타데이터에 대한 MPEG-7의 표현을 텍스트 형식(TeM)으로만 지원하고 있기 때문에, 추후 바이너리 형식(BiM)으로도 지원할 수 있도록 보완할 예정이다.

참 고 문 헌

- [1] ISO/IEC, *MPEG-A Multimedia Application Format Overview and Requirement (v.2)*, JTC1/SC29/WG11/ N6832, 2004.
- [2] ISO/IEC, *Information Technology-Multimedia Content Description Interface-Part 5: Multimedia Description Schemes*, 15938-5, 2003.
- [3] TV-Anytime Forum, *TV-Anytime Specification Series: S-3 on Metadata (Normative) SP003v1.1*, <http://www.tv-anytime.org>, 2001.
- [4] ISO/IEC, *MPEG-A Multimedia Application Format Overview and Requirement (v.2)*, JTC1/SC29/WG11/ N6832, 2004.
- [5] ISO/IEC, *Information Technology - Coding of Audio-Visual Objects-Part12: ISO Base Media File Format*, 14496-12, 2005.
- [6] ISO/IEC, *Information Technology - Coding of Audio-Visual objects -Part1: Systems*, 14496-1, 1998.
- [7] ISO/IEC, *Information technology - Multimedia Framework (MPEG-21) - Part 2: Digital Item Declaration (DID)*, 21000-2, 2004.
- [8] ISO/IEC, *MPEG Music Player Application Format*, JTC1/SC29/WG11/N7156, 2005.
- [9] ISO/IEC, *Photo Player Multimedia Application*

Format, JTC1/SC29/WG11/N7093, 2005.

- [10] 일본전자공업진흥협회, *Digital Still Camera Image File Format Standard (Exchangeable image file format for Digital Still Camera:EXIF) ver. 2.1*, JEIDA-49, 1998.
- [11] ID3 표준화협회, *ID3 made easy: ID3 ver. 1.1*, <http://www.id3.org/id3v1.html>, 2005.
- [12] ISO/IEC, *Information Technology - Coding of Audio-Visuals Part3: Amendment 3, MPEG-1/2 in MPEG-4 Audio*, 14496-3, 2001.
- [13] J. Zobel, S. Heinz, and H.E. Williams, "In-memory Hash Tables for Accumulating Text Vocabularies," *Information Processing Letters*, Vol. 80(6), pp. 271-277, 2001.
- [14] 한국무선인터넷 표준화 포럼, *WIPI(Wireless Internet Platform for Interoperability)-C API ver.1.2*, <http://www.wipi.or.kr/>, 2005.
- [15] Thin Multimedia Inc., *Platform Abstraction Layer ver.3.4*, <http://www.thinmultimedia.com/>, 2005.
- [16] Mikio Hirabayashi, *Quick Database Manager: QDBM ver.1.8.34*, <http://qdbm.sourceforge.net/>, 2005.



강 영 모

2004년 서강대학교 컴퓨터학과 졸업(학사). 2006년 서강대학교 대학원 컴퓨터학과(공학석사)



박 주 현

1999년 서강대학교 컴퓨터학과 졸업(학사). 2002년 서강대학교 대학원 컴퓨터학과(공학석사). 2002년~현재 서강대학교 대학원 컴퓨터학과 박사과정 재학 중



방 형 진

2005년 서강대학교 컴퓨터학과 졸업(학사). 2005년~서강대학교 대학원 컴퓨터학과 재학 중



남 중 호

1986년 서강대학교 전자계산학과 졸업 (학사). 1988년 한국과학기술원 전자계산학과(공학석사). 1992년 한국과학기술원 전자계산학과(공학박사). 1992년~1993년 Fujitsu 연구소 연구원. 1993년~현재 서강대학교 컴퓨터학과 교수



김 형 철

1986년 인하대학교 전자계산학과(학사)
1988년 한국과학기술원 전산학과(공학석사). 1995년 한국과학기술원 전산학과(공학박사). 1988년~2000년 ETRI 멀티미디어 연구부 선임연구원. 2000년~2003년 (주)나다텔 기술이사. 2004년~현재 (주)씬멀티미디어(TMI) CTO. 관심분야는 Mobile Multimedia, Mobile TV, Multimedia Processing