

# 관계 데이터 모델 기반 라이프로그 관리 시스템과 그 응용

(A Lifelog Management System Based on  
the Relational Data Model and its Applications)

송 인 철 <sup>†</sup>	이 유 원 <sup>†</sup>	김 현 규 <sup>†</sup>
(Inchul Song)	(Yu Won Lee)	(Hyeon Gyu Kim)
김 항 규 <sup>†</sup>	함 덕 민 <sup>†</sup>	김 명 호 <sup>**</sup>
(Hangkyu Kim)	(Deokmin Haam)	(Myoung Ho Kim)

**요 약** 하드 디스크 저장 매체의 가격이 하락함에 따라 가까운 시일 안에 개인 컴퓨터에 1TB가 넘는 하드 디스크가 기본으로 장착될 것으로 예상된다. 한 사람이 한 달에 1GB의 데이터를 저장한다고 가정하면 대략 1TB면 그 사람의 일생 동안에 걸친 데이터를 저장할 수 있다. 이에 따라 개인이 일상 생활에서 보고 들은 것을 기록한 라이프로그(lifelog)를 효과적으로 관리하는 라이프로그 관리에 대한 연구가 활발히 이루어지고 있다. 지금까지 관계 데이터 모델 기반, 온톨로지 기반, 파일 시스템 기반 라이프로그 관리 시스템(LMS: Lifelog Management System) 등 다양한 LMS들이 제안되었지만, 관계 데이터 모델 기반 LMS는 질의 처리 성능이 뛰어난 반면 복잡한 질의를 잘 처리하지 못하고 온톨로지 기반 LMS는 복잡한 질의를 처리할 수 있는 반면 질의 처리 성능이 떨어지며 파일 기반 LMS는 키워드 질의만 지원하는 등 저마다 장단점을 가진다. 또한 이들 시스템들은 라이프로그 그룹을 효과적으로 관리하기 위한 기능을 제공하지 못하고 있고 효과적인 검색을 위해 라이프로그의 태그(메타데이터)를 수정하거나 새로운 태그를 추가하기 위한 편리한 인터페이스를 제공하지 못하고 있다. 본 논문에서는 이러한 문제점들을 해결하기 위해 관계 데이터 모델 기반 라이프로그 관리 시스템을 제안한다. 제안하는 시스템은 라이프로그를 관계 데이터 모델로 모델링하고 사용자 질의를 SQL로 변환해 처리함으로써 질의 처리 성능이 뛰어나다. 또한 복잡한 질의를 잘 처리하지 못하는 관계 데이터 모델 기반 LMS의 단점을 보완하기 위해 찾으려는 라이프로그와 직접적으로 관련 있는 라이프로그에 대한 정보에 기반해 라이프로그를 검색하는 단순화된 관계 질의를 지원한다. 이와 더불어, 제안하는 시스템은 라이프로그 그룹 생성, 편집, 검색, 플레이 및 공유 기능을 제공함으로써 라이프로그 그룹에 대한 효과적인 관리를 지원한다. 마지막으로 제안하는 LMS에서 제공하는 라이프로그 태그 도구는 태그 추상화를 통해 여러 종류의 태그를 손쉽게 수정하거나 추가할 수 있는 기능을 제공한다. 본 논문에서는 제안하는 시스템의 설계 및 구현을 설명하고 이 시스템을 활용한 다양한 응용을 소개한다.

**키워드** : 라이프로그 관리 시스템, 관계 데이터 모델, 관계 질의, 라이프로그 그룹, 태그

**Abstract** As the cost of disks decreases, PCs are soon expected to be equipped with a disk of 1TB or more. Assuming that a single person generates 1GB of data per month, 1TB is enough to store data for the entire lifetime of a person. This has led to the growth of researches on lifelog manage-

\* 이 논문은 2007년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R0A-2007-000-10046-0)

논문접수 : 2009년 4월 8일

심사완료 : 2009년 7월 30일

<sup>†</sup> 학생회원 : KAIST 전산학과  
icsong@dbserver.kaist.ac.kr  
ywlee@dbserver.kaist.ac.kr  
hgkim@dbserver.kaist.ac.kr  
hkkim@dbserver.kaist.ac.kr  
dmhaam@dbserver.kaist.ac.kr  
<sup>\*\*</sup> 종신회원 : KAIST 전산학과 교수  
mhkim@dbserver.kaist.ac.kr

Copyright©2009 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제9호(2009.9)

ment, which manages what people see and listen to in everyday life. Although many different lifelog management systems have been proposed, including those based on the relational data model, based on ontology, and based on file systems, they have all advantages and disadvantages: Those based on the relational data model provide good query processing performance but they do not support complex queries properly; Those based on ontology handle more complex queries but their performances are not satisfactory; Those based on file systems support only keyword queries. Moreover, these systems are lack of support for lifelog group management and do not provide a convenient user interface for modifying and adding tags (metadata) to lifelogs for effective lifelog search. To address these problems, we propose a lifelog management system based on the relational data model. The proposed system models lifelogs by using the relational data model and transforms queries on lifelogs into SQL statements, which results in good query processing performance. It also supports a simplified relationship query that finds a lifelog based on other lifelogs directly related to it, to overcome the disadvantage of not supporting complex queries properly. In addition, the proposed system supports for the management of lifelog groups by providing ways to create, edit, search, play, and share them. Finally, it is equipped with a tagging tool that helps the user to modify and add tags conveniently through the abstraction of various tags. This paper describes the design and implementation of the proposed system and its various applications.

**Key words** : Lifelog Management System, Relational Data Model, Relationship Query, Lifelog Group, Tagging

## 1. 서론

하드 디스크 저장 매체의 가격이 하락함에 따라 가까운 시일 안에 개인 컴퓨터에 1TB(B는 byte를 나타냄)가 넘는 하드 디스크가 기본으로 장착될 것으로 예상되고 있다. 한 사람이 한 달에 1GB의 데이터를 저장한다고 가정하면 대략 1TB면 그 사람의 일생 동안에 걸친 데이터를 저장할 수 있다[1]. 이렇게 개인의 일생에 대한 정보를 저장할 수 있게 되면서 개인이 일상 생활에서 보고 들은 것을 기록한 라이프로그(lifelog)를 효과적으로 관리하는 라이프로그 관리(lifelog management)에 대한 연구가 활발히 이루어지고 있다[1-6].

라이프로그에는 다양한 종류가 있으며 개인의 생활 패턴이 변하거나 새로운 기술이 출현함에 따라 새로운 종류의 라이프로그가 계속해서 생겨나고 있다. 기본적인 라이프로그에는 사진, 비디오, 문서, 이메일, 일정 등이 있으며, 나아가 대화의 내용, 모임의 내용, 컴퓨터 사용 내역 등을 기록한 라이프로그도 있다. 그리고 센서 기술의 발달로 인해 다양한 센서에서 측정된 값이나 건강 상태의 기록 같은 라이프로그도 생겨나고 있다. 개인 정보기기와 저장 기술이 발전하면서 개인 콘텐츠를 손쉽게 생성할 수 있게 되었고 유비쿼터스 컴퓨팅 기술의 발달로 지속적인 라이프로그 생성이 가능해지는 상황에서 라이프로그를 효과적으로 관리할 수 있게 되면 개인의 생산성 향상, 소셜 릴레이션십 강화, 문화 수준 증진, 삶의 질 향상, 개인화된 비즈니스 창출 등 다양한 효과를 기대할 수 있다.

이렇게 라이프로그 관리의 중요성에 대한 인식이 확산되면서 라이프로그를 효과적으로 관리하기 위한 라이프

로그 관리 시스템들이 제안되었다. 하지만, 기존 라이프로그 관리 시스템들은 저마다 장단점을 가지고 있다. 기존 라이프로그 관리 시스템들은 기반 데이터 모델에 따라 크게 세 가지 부류로 나눌 수 있다. 관계 데이터 모델 기반 라이프로그 관리 시스템[1]은 라이프로그를 관계 데이터 모델로 모델링하고 라이프로그에 관한 질의를 SQL로 변환해 처리한다. 이러한 시스템은 질의 처리 성능이 뛰어난 반면 라이프로그 간 복잡한 관계에 기반한 관계 질의 처리를 제대로 지원하지 못한다. 온톨로지 기반 라이프로그 관리 시스템[2,5,6]은 라이프로그를 자유로운 구조를 가지는 그래프로 모델링함으로써 복잡한 관계 질의를 가능하게 한다. 하지만, 이러한 시스템은 질의 작성이 어렵고 질의 처리 성능이 떨어진다. 구글 데스크탑[7]이나 SIS[3] 같이 PC에 있는 모든 파일의 메타데이터와 콘텐츠에 대해 텍스트 인덱스를 생성하고 이를 기반으로 키워드 질의를 지원하는 파일 기반 라이프로그 관리 시스템도 존재한다. 이러한 시스템들은 라이프로그에 대한 키워드 검색만을 지원하고 관계 질의를 지원하지 못한다.

개별 라이프로그들이 관리되는 상황에서 사람들이 더욱 관심을 가지게 되는 것은 여행, 결혼식, 돌 잔치 등 기억에 남는 사건(본 논문에서는 이를 라이프이벤트라 부른다)들일 것이다. 라이프로그 관리 시스템은 사용자의 이러한 요구사항을 충족시키기 위해 개별 라이프로그 관리에서 한발 더 나아가 라이프로그 그룹인 라이프이벤트를 생성, 편집, 검색, 플레이 및 공유할 수 있는 기능을 제공해야 한다. 기존 라이프로그 관리 시스템들은 라이프로그 그룹을 생성하고 브라우징하기 위한 간단한 기능만을 제공할 뿐 총체적인 라이프이벤트 관리

기능을 제공하지 못하고 있다.

사용자 질의에 대해 풍부한 결과를 제공하기 위해서는 수집된 라이프로그에 충분한 정보가 태깅되어 있어야 한다. 또한 라이프로그에 태깅된 정보가 잘못되었을 경우 이를 수정할 수도 있어야 한다. 그러나 기존 라이프로그 관리 시스템에서는 라이프로그에 추가 정보를 간단히 태깅하는 기능만을 제공할 뿐 기존 태그 정보를 수정하는 방법을 제공하고 있지 않거나 편리한 태깅 인터페이스를 제공하지 못하고 있다.

본 논문에서는 이러한 기존 라이프로그 관리 시스템의 문제점을 해결하기 위해 관계 데이터 모델 기반 라이프로그 관리 시스템(LMS: Lifelog Management System)을 제안한다. 제안하는 LMS는 라이프로그를 관계 데이터 모델로 모델링하고 라이프로그에 관한 질의를 SQL로 변환하여 처리함으로써 질의 처리 성능이 뛰어나다. 또한 기존 관계 데이터 모델 기반 LMS가 라이프로그 간 관계에 기반한 관계 질의를 표현하기 어렵다는 점을 보완하기 위해, 찾으려는 라이프로그와 직접적으로 관련이 있는 라이프로그에 대한 정보를 기반으로 라이프로그를 검색할 수 있게 하는 단순화된 관계 질의를 지원한다. 더불어 제안하는 LMS는 라이프이벤트를 효과적으로 관리할 수 있는 기능을 제공한다. 제안하는 LMS에서 라이프이벤트는 라이프로그의 한 종류로 취급되고 사용자는 라이프이벤트를 수동 및 자동으로 생성, 편집, 검색, 플레이할 수 있고 블로그를 통해 다른 사람과 공유할 수 있다. 마지막으로 제안하는 LMS에서 제공하는 라이프로그 태깅 도구는 많은 종류의 라이프로그를 공통된 특성에 따라 추상화함으로써 사용자가 기존의 태그를 수정하거나 새로운 태그를 추가하는 일을 편리하게 할 수 있게 한다. 본 논문에서는 제안하는 LMS의 설계와 구현에 관해 상세하게 설명하고 그 응용으로서 일정 라이프로그를 관리하고 효율적인 일정을 추천하는 지능형 일정 관리 기능, 얼굴 인식 기술을 기반으로 한 인물 기반 라이프로그 검색 기능을 소개한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 제안하는 LMS의 아키텍처를 제시하고 라이프로그 저장을 위한 저장소 모델을 설명한다. 이어지는 장에서는 LMS 아키텍처의 핵심 모듈인 로깅 모듈, 태깅 모듈, 질의 처리 모듈에 관해 상세하게 설명한다. 3장에서는 로깅 모듈을, 4장에서는 태깅 모듈을, 5장에서는 질의 처리 모듈을 설명한다. 6장에서는 라이프이벤트 관리 기능을 설명한다. 7장에서는 LMS를 기반으로 한 응용 두 가지를 소개한다. 8장에서는 관련 연구를 설명하고 마지막으로 9장에서 결론과 추후 연구를 제시한다.

## 2. LMS 아키텍처

그림 1은 LMS의 아키텍처를 도식화한 것이다. LMS는 크게 로깅 모듈, 태깅 모듈, 질의 처리 모듈로 나누어진다. 로깅 모듈은 PDA, 카메라, 센서, PC 등 다양한 로깅 디바이스에서 작동하는 라이프로그 수집 도구를 통해 라이프로그를 수집하고 수집된 라이프로그를 LMS에 저장하는 역할을 담당한다. 태깅 모듈은 라이프로그 검색을 돕기 위해 라이프로그에 추가정보를 붙이는 역할을 수행한다. 추가정보에는 간단한 텍스트 설명, 라이프로그가 수집된 곳의 위치, 다른 라이프로그와의 관계 등이 있다. 질의 처리 모듈은 사용자로부터 다양한 질의를 입력 받아서 처리하고 결과를 반환한다.

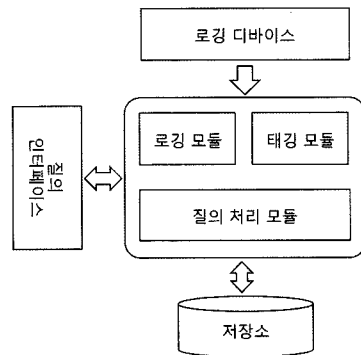


그림 1 라이프로그 관리 시스템 아키텍처

### 2.1 라이프로그 저장소 모델

제안하는 LMS에서는 라이프로그를 관계 데이터 모델로 모델링한다. 그림 2는 LMS에서 사용하는 관계 데이터 모델을 도식화한 것이다. 그림에서 각 사각형은 테이블을, 화살표는 주 키, 외래 키 관계를 나타낸다. 각 테이블에는 몇 가지 핵심적인 칼럼만 나타났다. 그림 2의 가운데 있는 Log 테이블은 모든 라이프로그에 공통적인 정보를 저장한다. Log 테이블의 칼럼으로는 라이프로그의 고유한 아이디인 id, 라이프로그 종류 LogType, 라이프로그에 대한 텍스트 설명을 담은 Description, 라이프로그가 저장된 시간을 나타내는 Date가 있다.

그 밖에 라이프로그 종류별로 테이블이 하나씩 존재한다. 현재 LMS에서는 총 8가지의 라이프로그에 대한 관리를 지원한다. 지원하는 라이프로그에는 웹 페이지 방문 기록을 나타내는 Web, GPS 위치 정보를 나타내는 GPS, 주고 받은 이메일을 나타내는 Email, 음악인 Audio, 동영상인 Video, 사진인 Picture, doc, ppt, pdf 등 다양한 문서를 나타내는 Document, 연락처를 나타내는 Contact가 있다. 라이프로그 중 텍스트 내용을 가지는 라이프로그는 텍스트를 저장하기 위한 칼럼을 가진다. 예를 들어, Web 테이블에는 웹 페이지 텍스트를

답는 Content, Email 테이블에는 이메일의 내용을 담은 Message 등이 있다. 또한 Picture, Audio, Video, Document 같은 파일 형태의 라이프로그는 파일 바이너리 데이터를 담은 RawData 칼럼을 가진다. Document 라이프로그의 텍스트 내용은 파일 바이너리 데이터로부터 추출하기 때문에 Document 테이블에는 텍스트 내용을 담은 칼럼이 존재하지 않는다. 모든 라이프로그 테이블에는 id 칼럼이 존재하고 이 칼럼은 Log 테이블의 id 칼럼을 참조한다.

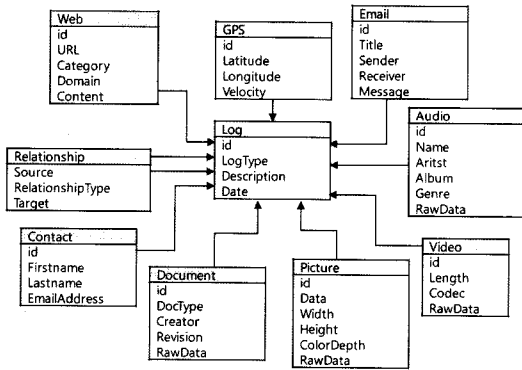


그림 2 라이프로그 관계 모델

그림 2에는 라이프로그 종류별로 생성한 8 개의 테이블 이외에 Relationship 테이블이 존재한다. Relationship 테이블은 라이프로그 간 관계 정보를 저장한다. 이 테이블은 소스 라이프로그를 나타내는 Source 칼럼, 타겟 라이프로그를 나타내는 Target 칼럼, 관계의 의미를 나타내는 RelationshipType 칼럼을 갖고 "Source 라이프로그는 Target 라이프로그에 RelationshipType이라는 관계로 연결되어 있다"는 의미를 표현한다. 이러한 관계 정보를 통해 두 라이프로그 간의 다양한 관계를 표현할 수 있다. 예를 들어, Source로 Picture 라이프로그 id를 갖고 Target으로 Contact 라이프로그 id를 갖고 Type 칼럼에 "PersonInPicture"라는 값을 갖는 Relationship 테이블의 row는 사진 안에 등장하는 인물이 누구인지를 나타내는 정보를 표현한다.

### 2.2 인덱스

키워드 질의로 라이프로그를 검색하기 위해서는 라이프로그에 대해 역인덱스(inverted index)[8]를 생성해야 한다. 테이블의 특정 칼럼에 대해 역인덱스를 생성하면 해당 칼럼에 들어 있는 각 키워드를 그 키워드를 포함하는 row 집합으로 매핑하는 역인덱스가 만들어진다. 제안하는 LMS에서는 Log 테이블의 Description 칼럼과 기타 다른 테이블에서 텍스트 내용을 담은 칼럼에 역인덱스를 생성했다. 또한 Document 테이블의 Raw-

Data 칼럼에도 역인덱스를 생성했다. 그리고 라이프로그의 텍스트 내용 뿐만 아니라 메타데이터에 대한 키워드 질의도 지원하기 위해 Document 테이블의 Creator, Email 테이블의 Sender와 같은 메타데이터 칼럼에도 역인덱스를 생성했다.

한 테이블의 여러 열에 대해 생성되는 역인덱스를 복합칼럼(multi-column) 역인덱스라고 부른다. 제안하는 LMS에서는 라이프로그 저장소로 오라클 데이터베이스를 사용했고 오라클 텍스트[9]를 이용해 복합칼럼 역인덱스를 생성했다. 복합칼럼 역인덱스를 생성할 때 특정 칼럼에 대해 텍스트 내용 추출 과정을 수행할 것인지를 지정할 수 있다. Document 라이프로그의 내용에 대한 키워드 검색을 지원하기 위해 Document 테이블의 RawData 칼럼에는 텍스트 내용 추출 과정을 수행하도록 설정했다. 이렇게 하면 RawData 칼럼의 내용이 doc, ppt, xls, pdf 파일인지에 따라 파일 포맷 분석을 통해 텍스트가 추출되고 추출된 텍스트를 바탕으로 역인덱스가 만들어진다. 각 복합칼럼 역인덱스는 인덱스를 구별하기 위한 식별칼럼을 갖고 이 식별칼럼은 질의 생성 시 사용할 인덱스를 지정할 때 쓰인다. 예를 들어, Email 테이블의 복합칼럼 역인덱스는 Title 칼럼을 식별칼럼으로 갖는다.

### 3. 로깅 모듈

라이프로그를 수집하기 위한 과정이 복잡하고 번거로우면 사용자가 LMS를 사용하지 않을 것이다. 또한 사용자가 라이프로그를 수동으로 입력해야 하면 불편함 때문에 수집된 라이프로그의 양과 질이 저하될 수 있다. 본 시스템에서는 웹 페이지, 위치 정보, 사진, 문서, 음악, 동영상 파일 등 다양한 라이프로그의 자동 로깅을 지원한다. 이어지는 절에서는 그 중에서 웹 페이지, 위치 정보, 파일에 대한 자동 로깅 방법을 설명한다.

#### 3.1 웹 페이지 로깅

제안하는 시스템에서는 사용자가 인터넷 서핑을 하는 도중 방문하는 모든 웹 페이지의 제목, URL, 텍스트 내용 등을 Web 테이블에 자동으로 저장한다. 또한 나중에 검색이 용이하도록 해당 웹 페이지에 대해 다른 사람들이 키워드로 태깅한 정보를 del.icio.us[10]로부터 자동으로 가져와 Web 테이블에 입력한다.

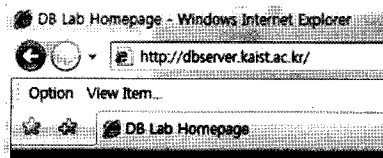


그림 3 웹 로그 수집을 위한 LMS 인터페이스

LMS를 설치하고 웹 브라우저를 실행하면 그림 3과 같이 Option 및 View Item이라는 메뉴를 포함한 톨바가 생성된다. Option 메뉴를 선택하면 나타나는 Option 창에서는 URL만 로깅할 것인지, 아니면 웹 페이지 내용을 함께 로깅할 것인지 설정할 수 있다. 또한 Option 창에서 Automatic Tagging 옵션을 선택해서 del.icio.us로부터 추출된 북마크 정보를 자동으로 태깅되게 하거나 Manual Tagging 옵션을 선택한 다음 직접 태깅을 할 수도 있다. View Item 메뉴를 선택하면 나타나는 View Item 창은 로깅될 최종 내용을 보여준다.

del.icio.us로부터 사용자가 현재 방문 중인 웹 페이지에 대한 태그 정보를 받아오는 데는 다음과 같은 URL 요청을 사용한다. hash 인자 값은 웹 페이지 URL의 MD5 해시 값이다.

<http://badges.del.icio.us/feeds/json/url/data?hash=46efc577b7ddef30d1c6fd13311b371e>

위 URL 요청을 보내면 결과로 JSON 포맷[11]의 데이터가 반환된다. 반환되는 내용에는 웹 사이트에 대한 태그들이 포함되어 있다.

**3.2 위치 정보 로깅**

본 시스템에서는 위치 정보 자동 로깅을 지원한다. GPS 장치가 PC에 연결되면 GPS 로거는 이를 감지해 GPS 장치에 들어 있는 위치 정보를 LMS에 자동으로 입력한다. 현재 Sony사에서 제작한 GPS-CS1[12]을 이용해 시스템을 구현하고 테스트했다. GPS-CS1은 위치 정보를 15초마다 기록하고 각 위치 정보는 NMEA 0183 포맷[13]으로 저장된다. 기록되는 정보에는 수집 시간, 위도, 경도, 속도 등이 있다.

GPS 로거는 그 밖에도 위치 정보를 다른 라이프로그에 자동으로 태깅하는 기능을 지원한다. 사용자가 위치 정보 동기화 명령을 내리면 GPS 로거는 라이프로그들을 시간 순으로 정렬한 후 라이프로그가 기록된 시간과 위치 정보가 기록된 시간을 비교해보고 시간 상 가장 근접한 위치 정보를 라이프로그에 태깅한다.

**3.3 파일 로깅**

파일 로거는 PC 상의 파일을 LMS에 로깅하는 기능을 지원한다. 윈도우 탐색기에서 사용자가 특정 파일이나 디렉토리를 선택하고 마우스 오른쪽 버튼을 눌렀을 때 나타나는 팝업 메뉴에서 LMS 로깅 명령을 선택하면, 선택한 파일이나 디렉토리 아래에 있는 모든 파일이 LMS에 자동으로 입력된다. 파일 로거는 PDA 상에 있는 특정 디렉토리에 들어 있는 파일을 로깅하는 기능도 지원한다.

파일 로거는 파일을 LMS에 로깅할 때 파일 이름, 크기, 수정 일자 등 파일의 메타데이터를 자동으로 추출하여 함께 입력한다. 그림 4는 현재 파일 로거가 파일별로 자동 추출할 수 있는 메타데이터를 도식화한 것이다.

	DOCX	XLSX	PPTX	MP3	AVI	JPG	GIF
File Name	○	○	○	○	○	○	○
File Date	○	○	○	○	○	○	○
File Size	○	○	○	○	○	○	○
Title	○	○	○	○			
Creator	○	○	○	○			
Album				○			
Genre				○			
Width & Height					○	○	○
Color Depth					○	○	○
Codec					○		

그림 4 파일별 자동 추출 가능 메타데이터 종류

**4. 태깅 모듈**

사진, 음성 또는 동영상 등 멀티미디어 형식을 가지는 라이프로그는 다른 라이프로그에 비해 텍스트 데이터가 부족하기 때문에 추가적인 메타데이터를 달아주어야만 검색이 용이하다. 뿐만 아니라, 라이프로그 간 관계를 활용한 관계 질의를 지원하려면 라이프로그 간 관계를 기술해 주어야 한다. 이와 같이 라이프로그에 대해 추가 정보를 메타데이터 형식으로 기술한 것을 태그(tag)라고 하고, 이러한 태그를 추가하는 과정을 태깅(tagging)이라고 한다.

라이프로그 태깅은 시스템 태깅, 사용자 태깅, 지능형 태깅으로 분류할 수 있다. 라이프로그가 로깅될 때 라이프로그에 대한 기본적인 정보를 자동으로 추출해 태깅하는 것을 시스템 태깅이라고 한다. 예를 들어, 발표자료를 로깅하면 발표자료의 파일명, 파일 형식, 만든 날짜 등이 시스템 태깅에 의해 라이프로그에 자동으로 태깅된다. 사용자 태깅은 사용자가 직접 태깅을 수행하는 것을 말한다. 사용자는 사용자 태깅을 통해 여행 사진 라이프로그에 등장 인물이나 촬영 장소에 관한 정보를 직접 입력할 수 있다. 지능형 태깅은 기존 라이프로그를 분석해 자동으로 태깅을 하는 것을 말한다. 예를 들면, 사용자 일정 라이프로그를 참고해 당일 촬영한 사진 중 결혼식 일정과 겹치는 시간에 찍은 사진에 결혼식에 관한 정보를 자동으로 태깅할 수 있다.

제한하는 LMS에서는 시스템 태깅과 사용자 태깅을 지원한다. 시스템 태깅에 관해서는 3장 라이프로그 로깅에 관해 논의할 때 언급했기 때문에 본 장에서는 사용자 태깅을 상세히 설명한다.

**4.1 태그 추상화**

라이프로그는 매우 다양하기 때문에 라이프로그에 추가할 수 있는 태그도 다양하다. 예를 들면, 사진 라이프로그에는 크기, 색상, 파일의 용량, 촬영 장소, 사진 속

의 인물 정보 등이 태깅될 수 있고, 음악 라이프로그에는 음악 데이터의 코덱 정보, 길이, 음질, 음악의 제작자, 음악 정보 등이 태그로 기술될 수 있다. 본 시스템의 라이프로그 관계 모델에는 라이프로그별로 별개의 테이블이 존재하고 각 테이블에는 많게는 20 개 이상의 칼럼이 존재한다. 이러한 다양한 정보를 사용자에게 모두 공개하고 태깅을 요구하게 되면 태깅 작업이 복잡하고 번거로울 것이다. 따라서, 본 시스템에서는 다양한 태그를 네 종류로 추상화해 사용자가 손쉬운 태깅을 할 수 있게 돕는다.

모든 태그는 문자열 태그, 시간 태그, 위치 태그, 그리고 관계 태그 이 네 종류로 분류된다. 문자열 태그는 텍스트 형태로 표현할 수 있는 특별한 형식이 없는 태그를 가리킨다. 예를 들어, 사진 로그의 경우, 파일명, 파일 크기, 사진 크기, 사용자가 추가한 메모 등이 문자열 태그가 된다. 시간 태그는 라이프로그 생성, 변경 시간 등을 포함하는 시간과 관련된 정보를 나타낸다. 위치 태그는 라이프로그가 생성된 장소를 기록하기 위해 사용된다. 관계 태그는 라이프로그 간 의미 관계를 기술하는 데 사용된다.

#### 4.2 태깅 도구 구현

그림 5는 LMS에서는 제공하는 태깅 도구의 모습을 보여준다. 태깅 도구 가장 위쪽에 있는 로그패널에는 태깅할 라이프로그가 미리보기 형식으로 나열된다. 사용자가 LMS 탐색기를 통해 태깅할 라이프로그를 검색한 후 그 중 몇 개를 선택한 다음 태깅 도구를 띄우면 선택된 라이프로그가 로그 패널에 자동으로 나타난다. 또는 LMS 탐색기에서 태깅 도구로 라이프로그를 끌어놓기 (drag and drop) 해서 로그 패널에 추가할 수도 있다. 로그 패널에서 라이프로그를 선택하면 선택된 라이프로그에 대한 태그 정보가 "메타데이터 이름: 값" 형태로



그림 5 태깅 도구

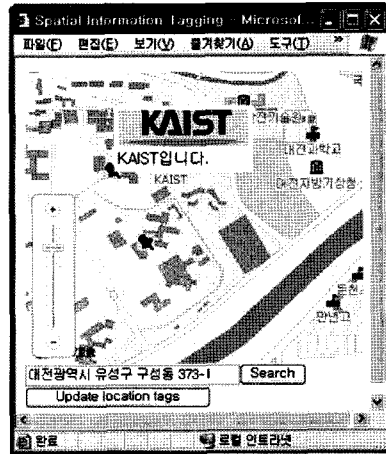


그림 6 위치 정보 태깅을 위한 웹 인터페이스

태깅 도구의 왼쪽 편에 있는 태그 패널에 나타난다. 태그 패널에서 특정 태그를 선택하면 해당 태그에 대한 자세한 정보가 오른쪽 편에 있는 메인 패널에 나타난다. 사용자는 기존 태그를 선택한 다음 수정할 수도 있고 태그 패널 아래에 있는 New Tag 버튼을 클릭해 새 태그를 생성할 수도 있다.

메인 패널의 모습은 태그 종류에 따라 다르다. 문자열 태그는 텍스트 필드로 출력된다. 시간 태그는 달력과 시계 형태로 출력된다. 위치 태그를 선택한 경우에는 웹 브라우저가 실행되고 그림 6에 보이는 위치 정보 태깅을 위한 웹 인터페이스가 나타난다. 사용자는 원하는 위치를 검색한 후 마우스 왼쪽 버튼으로 특정 지점을 클릭해 위치 정보를 태깅할 수 있다. 본 시스템에서는 위치 정보 태깅을 위해 네이버 지도 서비스를 사용했다. 웹 인터페이스에서는 지도 Open API[14]를 사용해 네이버 지도 서비스와 통신한다. 웹 인터페이스에서 사용자가 특정 위치를 클릭하면 지도 Open API를 통해 해당 지점의 위치 정보가 넘어온다. 이 정보는 위치 정보 로깅을 위한 웹 서비스에 전달되고 최종적으로 LMS 데이터베이스에 저장된다.

마지막으로 관계 태그를 선택하면 관계의 종류와 현재 선택된 라이프로그와 연결되어 있는 라이프로그를 보여준다. 사용자는 관계 종류를 다른 것으로 변경하거나, LMS 탐색기에서 태깅 도구 쪽으로 라이프로그 끌어놓기를 해 연결할 라이프로그를 변경할 수도 있다.

#### 5. 질의 처리 모듈

본 장에서는 LMS의 질의 인터페이스와 질의 처리 모듈을 설명한다. LMS의 사용자는 여러 가지 질의를 사용해 라이프로그를 검색할 수 있다. 현재 LMS에서

지원하는 질의 종류는 다음 세 가지다.

- 키워드 질의: 찾고자 하는 라이프로그와 관련 있을 만한 키워드를 입력해 라이프로그를 찾는 질의다. 예를 들어, 설악산 여행을 갔을 때 찍었던 사진을 검색하고 싶으면 “설악산 여행”라는 키워드 질의를 입력하면 된다.
- 메타데이터 질의: 라이프로그 생성 날짜나 라이프로그 종류를 지정해 원하는 라이프로그를 찾는 질의다. 예를 들어, “한달 전에 찍은 사진 라이프로그”라는 조건으로 라이프로그를 검색할 수 있다.
- 관계 질의: 찾고자 하는 라이프로그에 대한 정보가 잘 기억나지 않을 때 관련 있는 다른 라이프로그에 대한 정보를 입력해 라이프로그를 찾는 질의다. 관계 질의를 작성하는 과정은 사람이 연상 작용으로 과거의 기억을 떠올리는 과정과 비슷하다. 관계 질의의 예로 “작년 친구 결혼식에서 들었던 결혼식 축가를 찾아달라”는 질의가 있다.

### 5.1 LMS 탐색기

사용자는 LMS 탐색기를 통해 다양한 질의를 입력하고 결과를 받아본다. 그림 7은 LMS 탐색기의 모습을 보여준다. LMS 탐색기에서 메뉴바 아래에 있는 텍스트 필드에는 키워드 질의를 입력한다. 왼쪽에 있는 Log

Type 패널과 Date Condition 패널에는 메타데이터 질의를 입력한다. Log Type 패널에서는 찾고자 하는 라이프로그 종류를 지정하고 Date Condition 패널에서는 드롭다운 메뉴를 통해 모든 라이프로그, 어제 오늘의 라이프로그, 최근 7일간 라이프로그, 최근 한 달간 라이프로그를 선택하거나, 아래에 있는 텍스트 필드에 직접 날짜를 입력해 원하는 기간에 기록된 라이프로그를 검색할 수 있다. Relationship Condition 패널에는 관계 질의를 입력한다. 이 패널에서 드롭다운 메뉴에는 현재 LMS에 들어 있는 모든 관계의 종류가 나타난다. 원하는 관계 중 하나를 선택하고 오른쪽 텍스트 필드에 찾고자 하는 라이프로그와 선택한 관계를 지나는 라이프로그를 찾는 데 사용할 키워드 질의를 입력한다.

결과 출력 방식에는 Preview와 ListView 두 가지 형태가 있다. Preview 결과 출력 방식은 검색된 라이프로그들을 미리보기 형식으로 출력한다. 미리보기에는 라이프로그의 미리보기 이미지와 요약 설명이 포함된다. ListView 결과 출력 방식은 검색된 라이프로그의 종류, 짧은 설명, 라이프로그 기록 날짜, 긴 설명 등을 테이블 형태로 출력한다.

Search Events 패널은 라이프이벤트를 검색하는 데

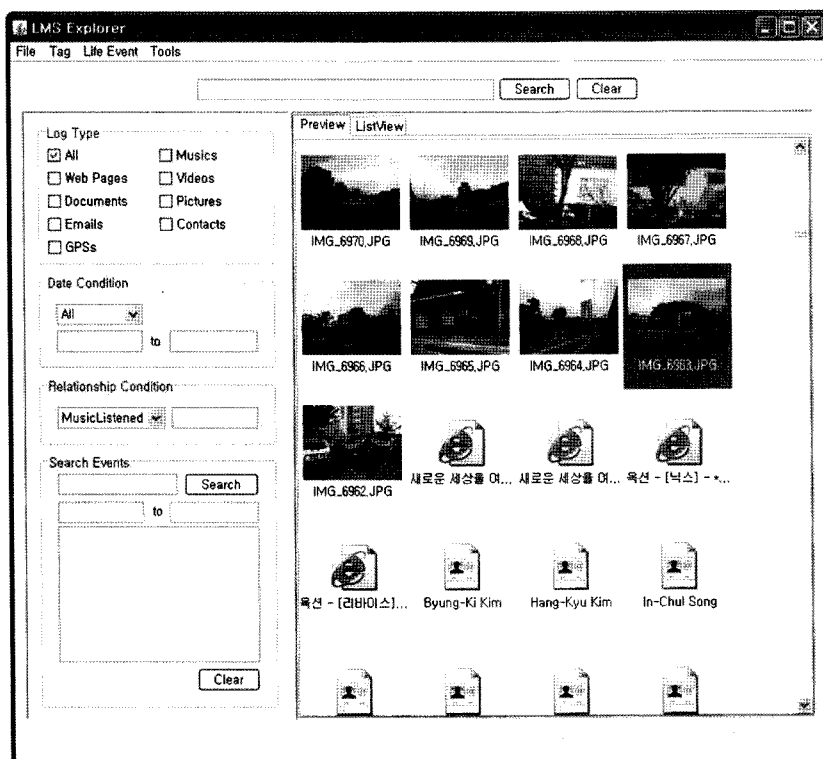


그림 7 LMS 탐색기

사용한다. 라이프이벤트에 관한 내용은 6장에서 설명한다. 그 밖에 LMS 탐색기 메뉴바에는 프로그램을 종료하기 위한 File 메뉴, 태깅 도구를 띄우기 위한 Tag 메뉴, 라이프이벤트 관리를 위한 Life Event 메뉴, 기타 도구를 담은 Tools 메뉴가 있다.

## 5.2 질의 변환

LMS 탐색기에 질의를 입력한 후 Search 버튼을 누르면 질의 처리가 시작된다. 질의 처리 모듈은 사용자가 입력한 질의를 SQL로 변환해 실행한다. 본 절에서는 질의 처리 모듈이 사용자 질의를 SQL로 변환하는 과정을 간략하게 설명한다.

### 5.2.1 키워드 질의와 메타데이터 질의 변환

먼저 사용자가 관계 질의를 제외한 키워드 질의와 메타데이터 질의만 입력했을 때 사용자 질의를 SQL로 변환하는 과정을 설명한다. 질의 처리 모듈은 일단 키워드 질의를 SQL로 변환한다. 사용자가 특정 로그 종류  $t$ 를 검색하기 위해 키워드 질의  $k=(k_1, k_2, \dots, k_n)$ 를 입력했다고 하자. 로그 종류  $t$ 에 대한 정보는 Log 테이블과  $t$  테이블에 나누어서 저장되어 있다. 키워드 질의 변환에 사용되는 기본 SQL문은 다음과 같다.

Q<sub>1</sub>: SELECT \* FROM Log, t WHERE Log.id = tid  
 키워드 질의는 이 기본 SQL문에서 WHERE 절의 조건으로 변환된다. Log 테이블과 t 테이블에 역인덱스가 만들어져 있고 각 역인덱스를 식별하기 위한 칼럼을 Log.i와 t.j라고 하자. 그러면 키워드 질의 k를 반영한 SQL문은 다음과 같아진다.

Q<sub>2</sub>: Q<sub>1</sub> AND (contains(Log.i, k) >  
 0 OR contains(t.j, q) > 0)

여기서 contains 연산자는 오라클 텍스트에서 제공하는 연산자로서 첫 번째 전달자로 식별되는 역인덱스를 사용해 테이블의 각 로우가 키워드 k를 포함하고 있는지 여부를 숫자로 반환한다. 어떤 로우가 키워드 k를 포함하고 있으면 contains는 0보다 큰 숫자를 반환한다.

다음으로 질의 처리 모듈은 메타데이터 질의를 SQL로 변환한다. 메타데이터 질의에는 라이프로그 생성 날짜 조건과 라이프로그 종류 조건이 들어있다. 라이프로그 생성 날짜 조건은 Q<sub>2</sub>의 WHERE 절에서 Log.Date 칼럼에 대한 조건으로 변환된다. 사용자가 라이프로그 종류 조건으로  $T=(t_1, \dots, t_n)$ 을 입력했다고 하면, 각 라이프로그 종류  $t$ 에 대해 앞에서 설명한 것처럼 SQL 변환을 수행해 생성된 SQL문들을 유니온(union) 연산자로 연결해 SQL문을 생성한다.

### 5.2.2 관계 질의 변환

이제 사용자가 관계 질의까지 입력한 경우 사용자 질의를 SQL로 변환하는 방법을 설명한다. 키워드 질의와 메타데이터 질의를 SQL 문으로 변환한 것을 Q<sub>3</sub>라고 하

자. 사용자가 LMS 탐색기의 Relationship Condition 패널에서 관계 종류  $r_1$ 를 선택하고 오른쪽 텍스트 필드에 찾고자 하는 라이프로그와 선택한 관계를 지니는 라이프로그를 찾는 데 사용할 키워드 질의  $r_k$ 를 입력했다고 하자. 먼저 관계 종류  $r_1$ 을 가져오기 위해 다음 질의를 생성한다.

R: SELECT \* FROM Relationship WHERE  
 RelationshipType='r<sub>1</sub>'

다음으로 키워드 질의  $r_k$ 로 라이프로그를 검색하기 위해 키워드 질의  $r_k$ 를 앞에서 설명한 방식대로 모든 라이프로그 종류에 대해 SQL로 변환한다. 이렇게 해서 생성된 SQL을 Q<sub>4</sub>라고 하자. 최종적으로 Q<sub>3</sub>, R, Q<sub>4</sub>를 조인(join)해 라이프로그를 찾는 SQL문 Q<sub>5</sub>를 생성한다.

Q<sub>5</sub>: SELECT \* FROM Q<sub>3</sub>, R, Q<sub>4</sub> WHERE R.Source  
 = Q<sub>3</sub>.id and R.Target = Q<sub>4</sub>.id

## 6. 라이프이벤트 관리

일반적으로 사람들은 개별 라이프로그 자체보다 여행, 결혼식, 아이 돌 잔치 등 기억에 남는 사건, 즉 라이프이벤트를 추억하고 공유하길 원한다. 제안하는 시스템은 라이프이벤트의 생성, 편집, 검색, 플레이 및 공유 기능을 지원한다.

### 6.1 라이프이벤트 저장소 설계

라이프이벤트도 기억에 남는 사건을 기록한 것이므로 라이프로그의 한 종류로 취급한다. 라이프이벤트가 라이프로그와 다른 점은 다른 라이프로그들을 포함한다는 점이다. 일반적인 라이프로그와 마찬가지로 라이프이벤트 정보 중 라이프로그에 공통적인 정보는 Log 테이블에 저장하고 나머지는 테이블 LifeEvent(itemid, eventname, eventfrom, eventto)에 저장한다. LifeEvent 테이블에서 itemid는 Log 테이블의 itemid를 참조하는 주 키이고 eventname은 라이프이벤트 이름, eventfrom은 라이프이벤트 시작 시간, eventto는 라이프이벤트 종료 시간을 의미한다.

테이블 LifeEventDetail(lifeeventid, paraid, type, logid, text)는 특정 라이프이벤트에 속하는 라이프로그들을 저장한다. LifeEventDetail 테이블에서 lifeeventid는 LifeEvent 테이블의 itemid를 참조하고 paraid는 라이프이벤트 안에서 라이프로그의 순서를 기록한다. 라이프로그의 순서를 따로 저장하는 이유는 사용자가 라이프이벤트 안에서 라이프로그들을 원하는 순서대로 배열할 수 있게 하기 위해서다. lifeeventid와 paraid는 LifeEventDetail 테이블의 주 키를 이룬다. type은 'L'과 'T' 문자 중에 하나의 값을 갖는 칼럼으로서 'L'은 라이프로그를 의미하고 'T'는 단순 텍스트를 의미한다. type이 'L'일 때는 logid 칼럼이 Log 테이블의 itemid 칼럼



을 참조하고 'T'일 때는 text 칼럼이 텍스트를 담는다. 이렇게 라이프로그뿐만 아니라 텍스트도 저장하는 이유는 사용자가 라이프이벤트에 자유롭게 설명을 달 수 있게 하기 위해서다.

### 6.2 라이프이벤트 생성

LMS 탐색기의 Life Event 메뉴에서 New Event를 클릭하면 라이프이벤트를 생성하기 위한 New LifeEvent 창이 뜬다. 이 창에서 라이프이벤트 이름, 간단한 설명과 라이프이벤트 발생 기간을 입력하고 Create 버튼을 눌러 빈 라이프이벤트를 생성한다. 사용자는 라이프이벤트 편집 기능으로 새롭게 생성한 빈 라이프이벤트에 라이프로그나 설명을 추가할 수 있다.

### 6.3 라이프이벤트 검색

라이프이벤트 편집을 위해서는 먼저 편집할 라이프이벤트를 검색하고 선택할 수 있어야 한다. 라이프이벤트 검색은 LMS 탐색기의 왼쪽 아래에 있는 Search Events 패널에서 이루어진다. 사용자는 라이프이벤트와 관련 있는 키워드를 입력하거나 라이프이벤트가 발생한 기간을 입력해서 라이프이벤트를 검색할 수 있다.

검색된 라이프이벤트 중 하나를 마우스 오른쪽 버튼으로 클릭하면 라이프이벤트 팝업 메뉴가 뜬다. 팝업 메뉴는 라이프이벤트에 속한 라이프로그 보기(Show Logs), 라이프이벤트 편집(Edit), 라이프이벤트 플레이(Play), 라이프이벤트 블로그 업로드(Upload), 라이프이벤트 삭제(Delete) 명령을 제공한다.

### 6.4 라이프이벤트 편집

라이프이벤트 팝업 메뉴에서 Edit 명령을 클릭하면 Life Event Editor 창이 뜬다. 그림 8은 Life Event Editor 창의 모습을 보여준다. 이 창에서는 라이프이벤트 제목(Title), 기간(Period), 간단한 설명(Description),

그리고 라이프이벤트에 속하는 라이프로그와 텍스트를 편집할 수 있다. 라이프이벤트에 라이프로그를 추가하는 방법은 간단하다. 먼저 추가하고 싶은 라이프로그를 LMS 탐색기에서 검색한다. 검색된 라이프로그 중 라이프이벤트에 추가하고 싶은 라이프로그를 마우스로 라이프이벤트 편집 창으로 끌어놓기를 하면 라이프이벤트에 라이프로그가 추가된다. 또한 라이프로그 편집 창에서 라이프로그 사이에 간단한 설명을 자유롭게 추가할 수 있다. 마지막으로 Save 버튼을 눌러 라이프이벤트를 저장한다.

### 6.5 라이프이벤트 플레이

라이프이벤트 팝업 메뉴에서 Play를 클릭하면 라이프이벤트 플레이어와 웹 브라우저가 나타난다. 라이프이벤트 플레이어는 라이프이벤트에 속한 라이프로그를 슬라이드 쇼 형태로 차례대로 보여주고 웹 브라우저는 현재 화면에 보이는 라이프로그가 기록된 위치를 지도 상에 보여준다.

### 6.6 라이프이벤트 공유

라이프이벤트 공유 기능은 라이프이벤트를 다른 사람과 공유할 수 있게 한다. 라이프이벤트를 공유하는 방법에는 블로그 업로드, CD 굽기 등 다양한 방법이 있지만 현재 LMS에서는 블로그 업로드 기능만을 제공한다.

라이프이벤트 팝업 메뉴에서 Upload를 클릭하면 라이프이벤트 블로그 업로드를 위한 Life Event Upload 창이 뜬다. 여기서 블로그 사이트를 선택하고 사용자 아이디와 암호를 입력하고 Upload 버튼을 누르면 라이프이벤트 내용이 블로그에 업로드된다. 블로그 업로드 기능은 MetaWeblog API[15]를 이용해 구현했고 현재 Tistory 블로그에 대한 업로드만 지원한다.

### 6.7 라이프이벤트 자동 생성

앞에서는 라이프이벤트 수동 생성 방법을 설명했다. 라이프이벤트를 생성과정을 LMS에서 도와줄 수 있으면 사용자가 라이프이벤트 관리하는 수고를 덜 수 있을 것이다. 현재 LMS에서는 일기 라이프이벤트 생성 기능을 제공한다. LMS 탐색기 Life Event 메뉴에서 New Diary를 클릭하면 New Diary 창이 뜬다. 사용자가 이 창에서 일기 제목, 간단한 설명, 날짜를 지정하고 Create 버튼을 클릭하면 지정된 날짜에 기록된 라이프로그를 포함하는 일기 라이프이벤트가 생성된다. 그 밖에도 시스템에 기록된 라이프로그에 데이터 마이닝 기술을 적용해 의미 있는 라이프로그 그룹을 찾고 이 그룹을 라이프이벤트화 하는 것도 유용할 것이다. 이러한 자동 라이프이벤트 생성 기술은 추후 연구로 남긴다.

## 7. LMS 응용

일단 개인의 라이프로그를 관리하기 시작하면 다양한

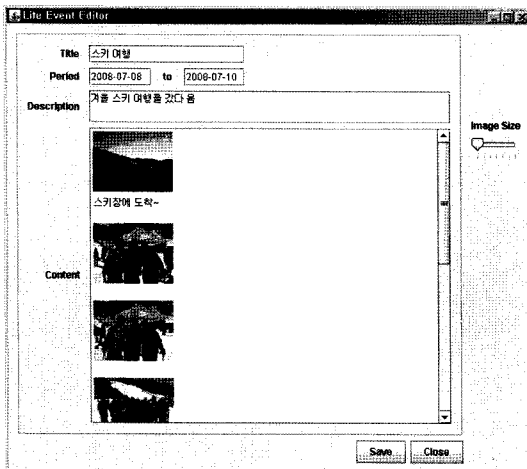


그림 8 라이프이벤트 편집기

응용이 가능하다. 본 장에서는 제안하는 LMS를 기반으로 하는 다양한 응용을 소개한다.

### 7.1 지능형 일정 관리기

LMS가 관리하는 라이프로그 중 중요한 것이 개인 일정에 관한 정보다. 지능형 일정 관리기는 LMS에 저장되어 있는 사용자 일정 관리하고 필요할 경우 기존 일정보다 효율적인 일정을 사용자에게 추천하는 기능을 제공한다.

일정 라이프로그는 테이블 Schedule(schedule\_id, start\_time, end\_time, location, todo, duration, category)에 저장된다. Schedule 테이블의 start\_time과 end\_time은 일정이 완료되어야 하는 시간대를 저장한다. location은 일정이 수행될 장소를, todo는 수행할 일을 저장한다. duration은 일정에 소비할 총 시간을, category는 해당 일정이 한번만 수행되는 것인지 혹은 주기적으로 수행되어야 하는 것인지를 표현한다. 예를 들어, (1, 3:00pm, 4:00pm, 한국대학교, 세미나, 30분, 한번)이라는 로우는 한국대학교에서 세미나를 해야 하며, 대략 3:00pm~4:00pm 사이에 수행하면 되고, 총 소비할 시간은 30분이라는 것을 의미한다.

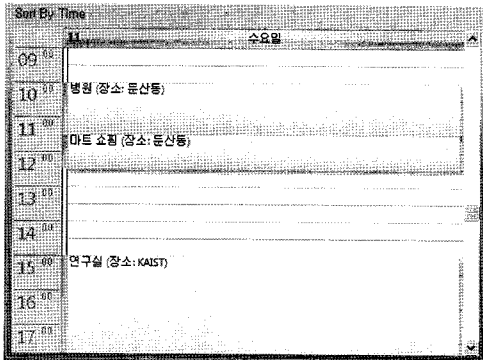


그림 9 지능형 일정 관리기의 일정 추천

지능형 일정 관리기는 LMS에 저장된 일정을 참고해 효율적인 일정을 사용자에게 추천하는 기능을 제공한다. 현재의 구현에서는 수행 시간대가 비슷하고 가까운 장소에서 일어나는 일정들을 묶어서 가능한 짧은 거리를 이동하면서 일정들을 처리할 수 있는 새로운 일정을 추천하도록 구현했다. 예를 들어, 이번 주 내로 둔산동에 있는 마트에 가서 장을 볼 계획이고 이번 주 수요일에 둔산동에 있는 병원에서 아침 10시에 검진을 받기로 하였다 하자. 그림 9는 지능형 일정 관리기에 의해 추천된 일정을 보여준다. 그림을 통해 둔산동에서 하기로 계획된 병원 방문과 마트 쇼핑이 연속된 일정으로 묶인 것을 볼 수 있다.

### 7.2 인물 기반 라이프로그 검색

인간은 다른 사람들과 함께 사는 사회적 동물이다. 그

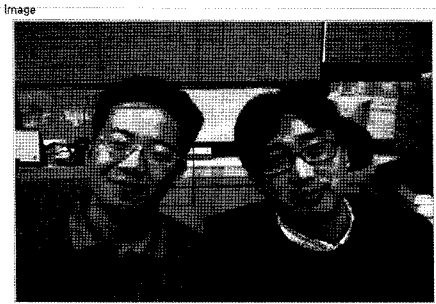
러므로 개인의 일상 생활을 기록한 라이프로그도 다른 사람과 관련된 것이 경우가 많다. 누군가와 함께 찍은 사진, 누군가와 주고 받은 이메일, 누군가와 만날 일정 등 많은 라이프로그가 다른 사람과 관련되어 있고 LMS의 사용자는 그러한 라이프로그를 검색하기 위해 흔히 사람 이름 같은 인물 정보를 질의 조건으로 사용하게 된다. 그러나 사진이나 동영상 같은 멀티미디어 라이프로그에는 텍스트 정보가 부족하고 메타데이터에도 인물 정보는 들어 있지 않다. 이러한 이유로 사진이나 동영상 같은 라이프로그를 사람 이름으로 검색하려면 사용자가 직접 라이프로그에 사람 이름을 태깅해 주어야 한다. LMS에는 많은 라이프로그가 저장될 것이므로 이렇게 사용자가 직접 모든 라이프로그에 인물 정보를 태깅하는 것은 불가능한 일일 것이다. 제안하는 LMS에서는 인물 기반 라이프로그 검색을 가능하게 하기 위해 사진 속에 등장하는 인물을 인식해 사진 라이프로그에 자동으로 태깅하고 이를 바탕으로 인물 정보를 질의 조건으로 사용해 사진 라이프로그를 검색할 수 있게 하는 인물 기반 라이프로그 검색 기능을 제공한다.

인물 정보 자동 태깅 기능은 얼굴 감지 기능과 얼굴 인식 기술을 활용해 구현했다. 두 기술을 활용하기 위해서는 인물별 얼굴 정보가 있어야 한다. 이를 위해 Contact 테이블의 Faces 칼럼에 인물별 얼굴 이미지 목록을 저장한다. 얼굴 이미지 목록은 다양한 각도에서 찍은 얼굴 사진으로 구성된다. 그림 10은 특정 인물의 얼굴 이미지 목록의 예를 보여준다.



그림 10 얼굴 이미지 목록 예

이렇게 인물 데이터베이스가 구축되고 나면, 새로운 사진 라이프로그가 기록될 때 사진 속에 등장하는 인물 정보를 태깅할 수 있다. 먼저, 새로 들어온 사진 라이프로그에 얼굴 감지 기술을 적용해 얼굴들을 추출해낸다. 추출해낸 각 얼굴에 대해 얼굴 인식 기능을 사용해 인물 데이터베이스에 있는 얼굴 이미지 목록의 얼굴들과 비교하고, 비슷한 얼굴이 있을 경우 해당 얼굴을 가지는 Contact 테이블의 로우, 즉 Contact 라이프로그를 주어진 사진 라이프로그에 연결한다. 그림 11은 사진 라이프로그에 인물 정보가 자동으로 태깅된 모습을 보여준다. 본 시스템에서는 The Open Computer Vision Library [16]를 사용해 얼굴 감지 기능을 구현했고 Eigenfaces [17]를 참고해 얼굴 인식 기능을 구현했다.



Index	Last Name	First Name	Email Ad...	Company	Mobile P...
1	김	우람	wkimg@d...	KAIST	011-9393-6...
2	김	태범	kd@dserv...	KAIST	010-8469-1...

그림 11 사진 라이프로그에 인물 정보 자동 태깅

제안하는 시스템에서는 인물 정보 자동 태깅과 인물 데이터베이스를 활용해 Query-By-Example 형태의 라이프로그 검색 기능을 제공한다. 사용자가 찾고자 하는 인물과 관련된 사진을 LMS에 제시하면, LMS는 사진으로부터 인물 정보를 자동으로 추출한 다음 추출한 인물 정보를 질의 조건으로 사용해 해당 인물과 관련된 라이프로그를 검색한다.

### 8. 관련 연구

PC에 개인의 일상에 관한 데이터를 저장할 수 있게 되면서 라이프로그 관리에 대한 연구가 활발히 이루어져 왔다. 몇몇 연구에서는 라이프로그에 대한 개념 모델을 제시하였다. Lifestreams[4]에서는 파일과 디렉토리로 이루어진 기존 PC 환경은 라이프로그를 관리하는데 한계가 있음을 지적하고 라이프로그 관리를 위한 개념적인 모델을 제시했다. 이 연구에서는 라이프로그를 과거로부터 미래로 이어지는 연속적인 스트림으로 모델링하고 라이프로그의 추가, 복제, 전달, 검색, 요약 기능을 제안했다. Dataspaces[18]는 기업, 정부, 도서관, 스마트 홈 등 다양한 데이터 소스로부터 만들어지는 데이터를 통합적으로 다루기 위한 데이터 저장소인 dataspace라는 개념을 제안하였다.

기존 라이프로그 관리 시스템은 기반 데이터 모델에 따라 크게 세 가지 갈래로 나눌 수 있다. 관계 데이터 모델 기반 라이프로그 관리 시스템[1]은 라이프로그를 관계 데이터 모델로 모델링하고 라이프로그에 관한 질의를 SQL로 변환해 처리한다. 이러한 시스템은 질의가 비교적 단순하고 질의 처리 성능이 뛰어난 반면 라이프로그 간 복잡한 관계에 기반한 관계 질의 처리를 잘 지원하지 못한다. 온톨로지 기반 라이프로그 관리 시스템

[2,5,6]은 라이프로그를 자유로운 구조를 가지는 그래프로 모델링하고 라이프로그와 라이프로그 간 관계에 의미를 부여함으로써 복잡한 관계 정보에 기반한 질의를 가능하게 한다. 하지만, 이러한 시스템은 사용자가 데이터 구조를 잘 알지 못할 경우 복잡한 질의를 작성하기가 힘들고 질의 처리 성능이 떨어지며 대부분 메모리 기반 시스템이기 때문에 확장성이 떨어진다. 마지막으로 구글 데스크탑[7]이나 SIS[3] 같이 PC에 있는 모든 파일의 메타데이터와 콘텐츠에 대해 텍스트 인덱스를 생성하고 이를 기반으로 키워드 질의를 지원하는 파일 기반 라이프로그 관리 시스템들이 있다. 이러한 시스템들은 키워드 기반 라이프로그 검색만을 지원하고 있고 관계 질의는 지원하지 못한다. 또한 로깅, 태깅, 질의 처리, 결과 출력 등 총체적인 라이프로그 관리 기능을 제공하고 있지 않다.

서론에서 지적한 바와 같이, 기존 라이프로그 관리 시스템들은 라이프로그 그룹을 효과적으로 관리하기 위한 기능을 제공하지 못하고 있다. 간단히 라이프 이벤트를 생성하고 라이프 이벤트에 라이프로그를 추가하는 기능을 제공하고 있기는 하지만[1,6], 그 밖에 라이프 이벤트 자동 생성, 편집, 검색, 플레이, 공유와 같은 총체적인 라이프 이벤트 관리 기능은 제공하고 있지 않다. 또한 기존 라이프로그 관리 시스템들은 수집된 라이프로그에 추가적인 정보를 태깅하는 기능을 제공하고 있기는 하지만 다양한 종류의 태그를 편리하게 수정하고 추가할 수 있는 기능은 제공하고 못하고 있다.

### 9. 결론 및 추후 연구

본 논문에서는 효과적인 라이프로그 관리를 지원하는 관계 데이터 모델 기반 라이프로그 관리 시스템을 제안하고 그 설계와 구현을 상세히 설명했다. 제안하는 시스템은 라이프로그의 로깅, 태깅, 질의 처리, 결과 출력을 지원한다. 또한 라이프 이벤트의 효과적인 관리를 지원하고 태그 추상화를 통해 편리한 태깅을 지원한다. 본 논문에서는 지능형 스케줄러, 인물 기반 라이프로그 검색 등 제안된 시스템을 기반으로 한 다양한 응용도 소개하였다. 제안하는 시스템은 라이프로그를 효과적으로 관리함으로써 개인의 생산성 향상, 소셜 릴레이션십 강화, 문화 수준 증진, 삶의 질 향상, 개인화된 비즈니스 창출 등 다양한 효과를 가져올 수 있을 것으로 예상된다.

본 논문에서 제안한 시스템은 더욱 효과적인 라이프로그 관리 지원을 위해 다양한 방향으로 확장될 수 있다. 먼저, 개인 콘텐츠 생성이 용이해지고 유비쿼터스 기기의 발전으로 다양한 라이프로그가 생겨나고 있기 때문에 더 많은 종류의 라이프로그의 수집을 지원할 수 있도록 로깅 모듈을 확장해야 한다. 태깅 모듈도 시스템

태깅, 사용자 태깅 뿐만 아니라 지능형 태깅을 지원하는 쪽으로 확장되어야 한다. 또한, 질의 처리 성능과 확장성이 보장되는 내에서 효과적인 질의 처리를 위해 온톨로지를 활용하는 방안을 고려해 볼 수 있다. 이렇게 제안하는 시스템을 다양한 방향으로 확장하는 일은 추후 연구로 남긴다.

### 참고 문헌

- [1] Gemmel, J., Bell, G., and Lueder, R., "MyLifeBits: A Personal Database for Everything," *Comm. ACM*, vol.49, no.1, pp.88-95, 2006.
- [2] Dong, X. and Halevy, A., "A Platform for Personal Information Management and Integration," In *Proc. of CIDR*, 2005.
- [3] Dumais, S., Cutrell, E., Cadiz, JJ, Jancke, G., Sarin, R., Robbins, D. C., "Stuff I've Seen: A System for Personal Information Retrieval and Re-Use," In *Proc. of SIGIR*, 2003.
- [4] Freeman, E. and Gelernter, D., "Lifestreams: A Storage Model for Personal Data," *SIGMOD Record*, vol.25, no.1, pp.80-86, 1996.
- [5] Dittrich, J. and Salles, M. A., "iDM: A Unified and Versatile Data Model for Personal Dataspace Management," In *Proc. of VLDB*, 2006.
- [6] Karger, D. R., Bakshi, K., Huynh, D., Quan, D., and Sinha, V., "Haystack: A Customizable General-Purpose Information Management Tool for End Users of Semistructured Data," In *Proc. of CIDR*, 2005.
- [7] Google Desktop. <http://desktop.google.com>.
- [8] Yates, R. B. and Neto, B. R., "Modern Information Retrieval," Addison Wesley, 1999.
- [9] Oracle Text. <http://www.oracle.com/technology/products/text/index.html>.
- [10] Social Bookmarking. <http://del.icio.us>.
- [11] JSON Script. <http://json.org>.
- [12] Sony GPS-CS1. <http://www.sonymstyle.com/webapp/wcs/stores/servlet/ProductDisplay?catalogId=10551&storeId=10151&langId=-1&partNumber=GPSCS1>
- [13] NMEA 0183 Standard. [http://www.nmea.org/content/nmea\\_standards/nmea\\_083\\_v\\_400.asp](http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp).
- [14] Naver Map API. <http://dev.naver.com/openapi/apis/map/javascript/example>.
- [15] MetaWeblog API. <http://www.xmlrpc.com/metaWeblogApi>.
- [16] The Open Computer Vision Library. <http://sourceforge.net/projects/opencvlibrary/>.
- [17] Turk, M. and Pentland, A., "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol.3, no.1, pp.71-86, 1991.
- [18] Franklin, M., Halevy, A., and Maier, D., "From Databases to Dataspace: A New Abstraction for Information Management," *SIGMOD Record*, vol.34, no.4, pp.27-33, 2005.



송 인 철

2004년 아주대학교 정보 및 컴퓨터공학부(학사). 2004년~2006년 KAIST 전산학과(석사). 2006년~현재 KAIST 전산학과 박사과정. 관심분야는 센서 네트워크, 시맨틱 웹, 정보 검색 등



이 유 원

2005년 부산대학교 전자전기정보컴퓨터공학부 정보컴퓨터공학전공(학사). 2007년 KAIST 전산학과(석사). 2007년~현재 KAIST 전산학과 박사과정. 관심분야는 데이터베이스시스템, 센서 네트워크, 스트림 데이터 처리 등



김 현 규

1997년 울산대학교 전산학과(학사). 2000년 울산대학교 전산학과(석사). 2000년~2001년 한국국방연구원 연구원. 2001년~2004년 LG전자 단말연구소 선임연구원. 2005년~현재 KAIST 전산학과 박사과정. 관심분야는 데이터베이스 시스템, 스트림 데이터 처리 등

트립 데이터 처리 등



김 항 규

2003년 연세대학교 기계전자공학부 정보산업전공(학사). 2003년~현재 KAIST 전산학과 석사사 통합과정. 관심분야는 데이터베이스 시스템, 시맨틱 웹, 온톨로지, 그래프 데이터 처리 등



함 덕 민

2007년 2월 KAIST 전산학과(학사). 2008년 8월 KAIST 전산학과(석사). 2008년 9월~현재 KAIST 전산학과 박사과정. 관심분야는 Keyword Search, Data Mining



김 명 호

1982년 서울대학교 컴퓨터 공학과(학사) 1984년 서울대학교 컴퓨터 공학과(석사) 1989년 Michigan State Univ. 전산학과(박사). 1989년~현재 KAIST 전산학과 교수. 1995년 Univ. of Virginia 방문 교수. 관심분야는 Database, Distributed

Systems, Workflow, Multimedia, OLAP, Data Warehouse