Database Design and Implementation for Personal Life Log Media Framework

Baud Haryo Prananto, Ig Jae Kim, Hyoung Gon Kim
Imaging Media Research Center, Korea Institute of Science and Technology (KIST)
Korea University of Science and Technology (UST)
{baud, kij, hgk} @imrc.kist.re.kr

Abstract

This paper describes the design and implementation of the database which is used in Personal Life Log Media system. The database contains information about media that capture personal experiences and enables the user to retrieve the media in a user friendly ways. The implementation of the database design is done by managing video data, which captures user's personal experiences, with its spatial and temporal information. The database enables the user to retrieve the video by mentioning where and/or when the video has been taken.

1 Introduction

The basic concept of Personal Life Log Media is:

- Save what you see, and
- See what you save.

"Save what you see" means the process of capturing one's personal experience in form of digital multimedia data (e.g. picture, video, audio) that will be saved in data storage.

"See what you save" means the process of reviewing the multimedia data that has been stored in the storage.

Due to the great amount of personal experiences that will be captured in form of digital multimedia data, there is a need to manage those data systematically therefore the user can retrieve them for review. To answer this need, we require a systematic database system to manage the information about the multimedia data (metadata) that enables user friendly search of the multimedia data using human-readable information.

We designed the database system to manage the multimedia data by using spatial, temporal, activity, and environmental information. Then, for the implementation, we built a personal experiences capture system in form of video along with the spatial and temporal information using GPS. The captured video and its metadata are stored in our database. We also built a user interface system to query the video using spatial and temporal information and perform user study to measure the usability of the user interface.

2 Related Work

The project to capture personal experiences, manage and retrieve the captured data has been frequently implemented for example in the LifeLog project initiated by DARPA[1]. There are several papers and commercial products supporting this project.

In 2005, Nitin K. Tripathi from Asian Institute of Technology, Bangkok, built a system for real time data logging using Wireless GIS (w-GIS)[2]. This system enables user to capture pictures, and retrieve the result in form of map.

There is also similar commercial system called GPS-Photo Link[3] that links a picture with GPS information and embed them to the pictures metadata. This system also able to create a stand alone HTML page that shows the location of the picture by connecting to TerraServer (only for USA and Canada).

Chon-In Wu from National Taiwan University built a system called MProducer[4] that enable a PDA user to capture personal experiences in form of video. The system also captures the user's position and activity because the PDA is also equipped with GPS receiver and tilt sensor.

Gamhewage C. de Silva from Aizawa Laboratory, University of Tokyo built an electronic chronicle[5] to capture user's activity in a smart home. Using this electronic chronicle, user can answer questions about past activities.

Most of these papers are focusing their research on the general system design and capture and storage methods. This paper will particularly focus on the database and user interface design. We will propose a database and user interface design that enable LifeLog users to store and query their personal experiences effectively.

3 Database Design

Basically, the database will store the metadata therefore the user can find the media using four axis: time, space, activity, and environment (e.g. person).

This database uses RDBMS because it enables to classify the information and relationships between them while keeping the data logical integrity. Compared to other type of database (e.g. Native XML Database), RDBMS gives better query performances when there's many query conditions[6] like this system (time, space, activity, person, etc.).

3.1 Entity Relationship Diagram

Figure 1 shows the Entity Relationship Diagram of the database.

In the center there is the Media entity that stores the information about the media itself: file name, media type, file time, duration (for audio and video), and other descriptions. For video we can refer on the MPEG-7 metadata.

On the left of the Media entity, there is Places entity that stores high-level spatial information. It can be user-defined (e.g. My Home, My Office, Grandpa's House, etc) or stored permanently in the database if we want to describe public places. It contains the name of the places, address, and geographical coordinates of the place. This Places entity is connected to Media entity with many-to-many relationship.

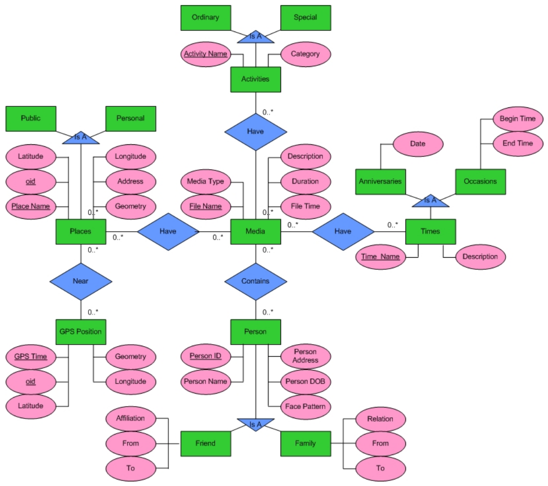


Fig. 1 Entity Relationship Diagram of the Database

Below the Places entity is the GPS entity that stores the low-level spatial information. It contains all GPS readings that inform where the user had ever been. This entity is connected to Places entity with many-to-many relationship.

On the right of Media entity, there is Times entity that stores high-level temporal information. The low-level temporal information (timestamps) is already stored in Media entity. This Times entity contains user defined times (e.g. My Birthday, Monday Weekly Meeting, Lab Seminar, etc) that specialized into two categories: Anniversaries and Occasions. Anniversaries entity stores yearly recurring events therefore it only stores the date. Occasions entity stores events that only occur once therefore it stores begin and end time of the events. Times entity also connected to Media entity with many-to-many relationship.

Above the Media entity there is Activities entity that stores the activities of the user during the capture of the media. The user defines the activities and it will be categorized into Ordinary and Special Activities. Activities entity also connected to Media entity with many-to-many relationship.

Below the Media entity there is Person entity that stores the people appear in the media. Person entity can be specialized as Family and Friend entities. Family entity stores the relationship information with the user (e.g. father, mother, brother, cousin, etc) and the lifespan of that relationship. Friend entity stores the information about the person's affiliation with the user, for instances where the person met the user (e.g. Korea University, KIST, etc). It also stores the lifespan of the affiliation as well. Person entity also connected to Media entity with many-to-many relationship.

3.2 Tabular Representation

Figure 2 shows the tabular representation of the database design. Basically each entity will be represented as one table[7]. For the specialization of Person and Times entities, there will be two additional tables for each entity describing its sub-entities. For the Places and Activities entities, the specialization doesn't require additional table because the sub-entities have no further attributes.

Each many-to-many relationship will be represented as one table as well. This table takes the primary keys of the table it connects and makes them its foreign keys. These tables are very important because the user will look at them to find the media.

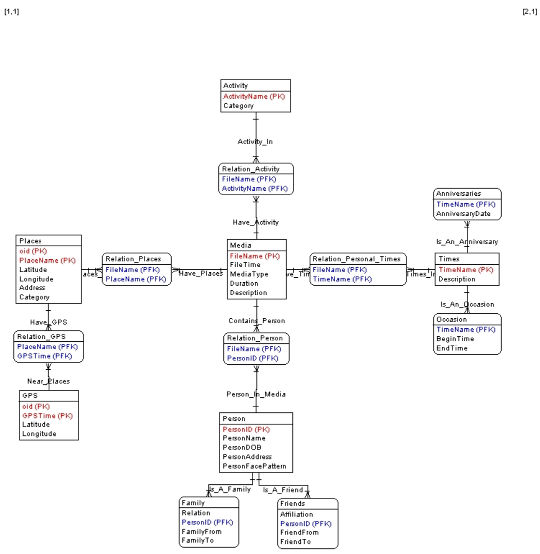


Fig. 2 Tabular Representation of the Database

4 Implementation

The implementation of the database design in our Personal Life Log Media system constrains on the spatial and temporal information only. That means we didn't use the Person and Activity entities.

4.1 System Architecture and Scenario

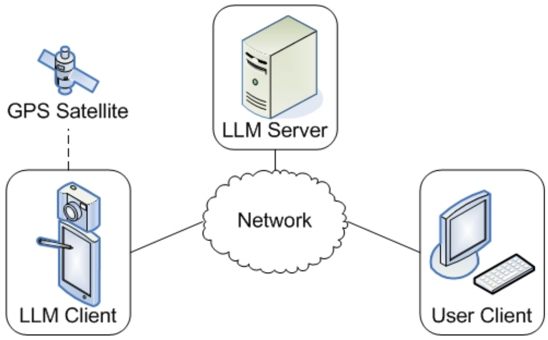


Fig. 3 General System Architecture

The general system architecture can be seen in Figure 3 and the detail breakdown is shown is Figure 4. The user captures personal experiences in form of multimedia data (e.g. picture or video) using Life Log Media (LLM) Client and then upload to the LLM Server (directly or manually after capture).

After media stored in the LLM Server, the user can browse the media in form of a web page through User Client. This User Client can be any computer that has internet browsing capability, and it can also be the same machine with LLM Client.

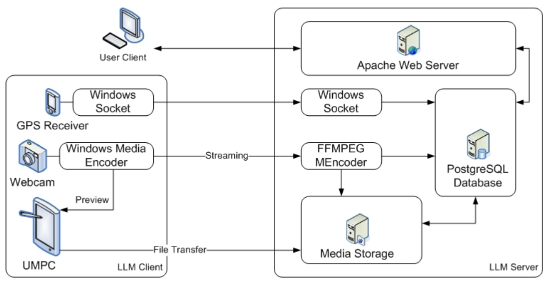


Fig. 4 Details of Client and Server System

4.2 LLM Client

LLM Client performs the capture and transmission function. It captures the media and directly transfers to the LLM Server or stores it first to transfer later. In our system, we use Samsung Q1 UMPC to perform these functions. It is capable to capture high quality video but small enough to use it as mobile device.

LLM Client also captures additional information to support the metadata, for instance the GPS data to provide spatial information.

For video capturing, we use the Windows Media Encoder[8] to encode the captured video from webcam and broadcast to the network through streaming protocol. This broadcast enables the LLM Server to access the video stream and save the video real-time.

For capturing GPS data, we attach a USB GPS receiver and send the data to the LLM Server using Windows Socket.

4.3 LLM Server

LLM Server is the center of the LLM service. It stores the media and the database contains the metadata. It also provides the web server application for user interface.

The designed database is implemented in the server using PostgreSQL and we also install its extension, PostGIS[9], to handle spatial data.

For the user interface, we implemented using web pages provided by Apache web server. The web server will communicate with database server using SQL. We also install the UMN MapServer[10] web application that will interact with PostGIS to present the spatial data stored in database in form of map. Using the web page, the user will able to access the database to perform query and modify the data.

To capture sensor data (e.g. GPS) sent by LLM Client, we use the same Windows Socket program[11] as the one used in LLM Client. This program will directly store the sensor data to the database.

To capture the video, that broadcasted by LLM Client, we use the FFMPEG-based program named MEncoder[12]. Using this program we store the video stream periodically and at the same time write the metadata to the database. In case of the streaming service is unavailable (e.g. no internet connection), the video will stored locally in the LLM Client to be transferred later using file transfer protocol.

4.4 User Client

From User Client, the user can manage and query the media that had already been stored.

With the user interface provided by web server, the user can query using high-level spatial and temporal information. For instance: "I want to see the videos that I've captured in my office during John Doe's birthday". By entering the spatial information of "my office" and temporal information "John Doe's birthday", the user will send the query to the LLM Server. The LLM Server will search on the table that represents Media and Places relationship to find which video files correspond with "my office". It also searches on the table that represents Media and Times relationship to look for which video files captured during "John Doe's birthday". After that the query result will be sent back to the User Client to be evaluated by the user.

5 Result

We have tested the framework and collect some video data using the LLM Client and stored in LLM Server.

The LLM Client captures and broadcast the video using a program employing Windows Media Encoder. This program (Figure 5) also supports FTP upload while there is no internet connection possible for streaming.

The LLM Server will stream the broadcasted video, using a program employing FFMPEG-based MEncoder, and write the metadata to the database.

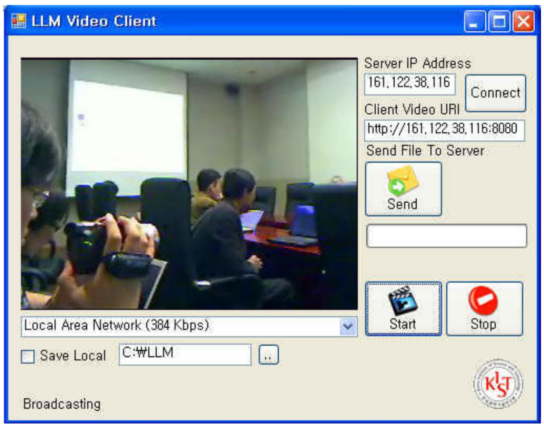


Fig. 5 Program for LLM Client

The basic view of the user interface (Figure 6) is the map of Seoul city, which the user can navigate with panning and zooming control, to provide the spatial search. On the map, there are some icon indicators that indicate the geographical locations that have been captured by GPS receiver. If the user clicks on that icons, a pop-up window will appear to show the corresponding media files that has been captured in that place.

Fig. 6 User Interface Result

The user can filter the icon appearance using high-level spatial and temporal information like "my office" and "my birthday". Using that filter, the map will only shows the icons that corresponds to the information and therefore the user can get the desired media.

6 Evaluation

The evaluation is performed using user study. The users are asked to perform some tasks and give some subjective assessments upon the user interface.

User Profile:

- Respondent number: 8
- Gender: 4 male, 4 female
- All are familiar with internet and digital multimedia
- Only 2 of them familiar with web programming

For subjective assessments users are asked to give the scale of agreement (1 to 4, 1 is for strongly disagree and 4 is for strongly agree) on 10 aspects based on Nielsen's 10 heuristics[13] for usability evaluation:

1. Visibility of system status
2. Match between system and real world
3. User control and freedom
4. Consistency and standards
5. Error Prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

The results are given in Figure 7.

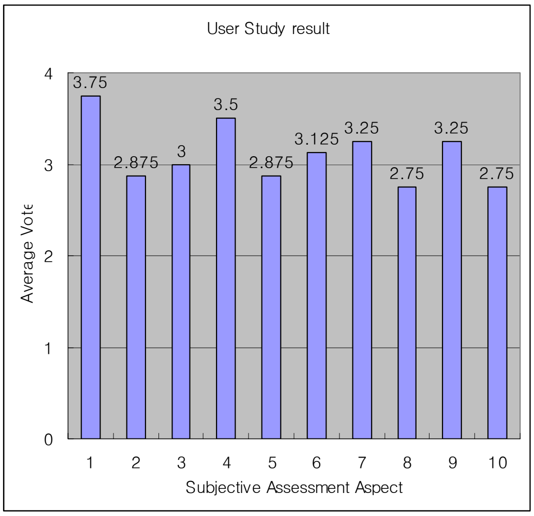


Fig. 7 User Study result

7 Conclusion and Future Work

By using the designed database and the build framework, the user can capture the video and retrieve the media collection using spatial and temporal information. This database design provides the handling of activity and person information and it is planned to be implemented.

The framework built in this work provides quite good usability as can be seen in user study result. The average agreement point are all above 2.5 that means most users are agree on the usability of the user interface.

The activity information can be provided by acceleration sensor placed on the user's body to detect the user's activity. The acceleration sensor will send the data to the LLM Server and it will be processed and determined whether the user is walking, running, sleeping, and so on.

Person contained in the media can be recognized by face recognition system and will be compared by the face pattern stored in the person database.

References:

[1] IPTO DARPA, *LifeLog Project Solicitation, Program Objectives and Description*, 2003.

[2] Nitin K. Tripathi, *Wireless GIS (w-GIS) for real time data logging,* Asian Institute of Technology, Bangkok, 2005.

[3] GPS-Photo Link, http://www.prairie.mb.ca/gps_photolink.htm.

[4] Chon-in Wu, Chao-ming (James) Teng, Yi-chao Chen, Tung-yun Lin, Hao-hua Chu, Jane Yun-jen Hsu, *Point-of-Capture Archiving and Editing of Personal Experiences from a Mobile Device*, ACM/Springer Journal of Personal and Ubiquitous Computing (PUC), Special Issue on Memory and Sharing of Experiences, 2006.

[5] Gamhewage C. de Silva, Toshihiko Yamasaki, Kiyoharu Aizawa, *Creation of an Electronic Chronicle for a Ubiquitous Home: Sensing, Analysis, and Evaluation*, Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06) IEEE, 2006.

[6] Anne Williams, *Performance of Relational Databases versus Native XML Databases*, Research Report Information Science, School of Business, University of Otago New Zealand

[7] Silberschatz, Korth, Sudahshan, *Database System Concepts 4th edition*, McGraw Hill, 2002.

[8] Microsoft Windows Media Encoder, http://www.microsoft.com/windows/windowsmedia/forpros/encoder/default.mspx

[9] PostGIS, http://postgis.refractions.net/.

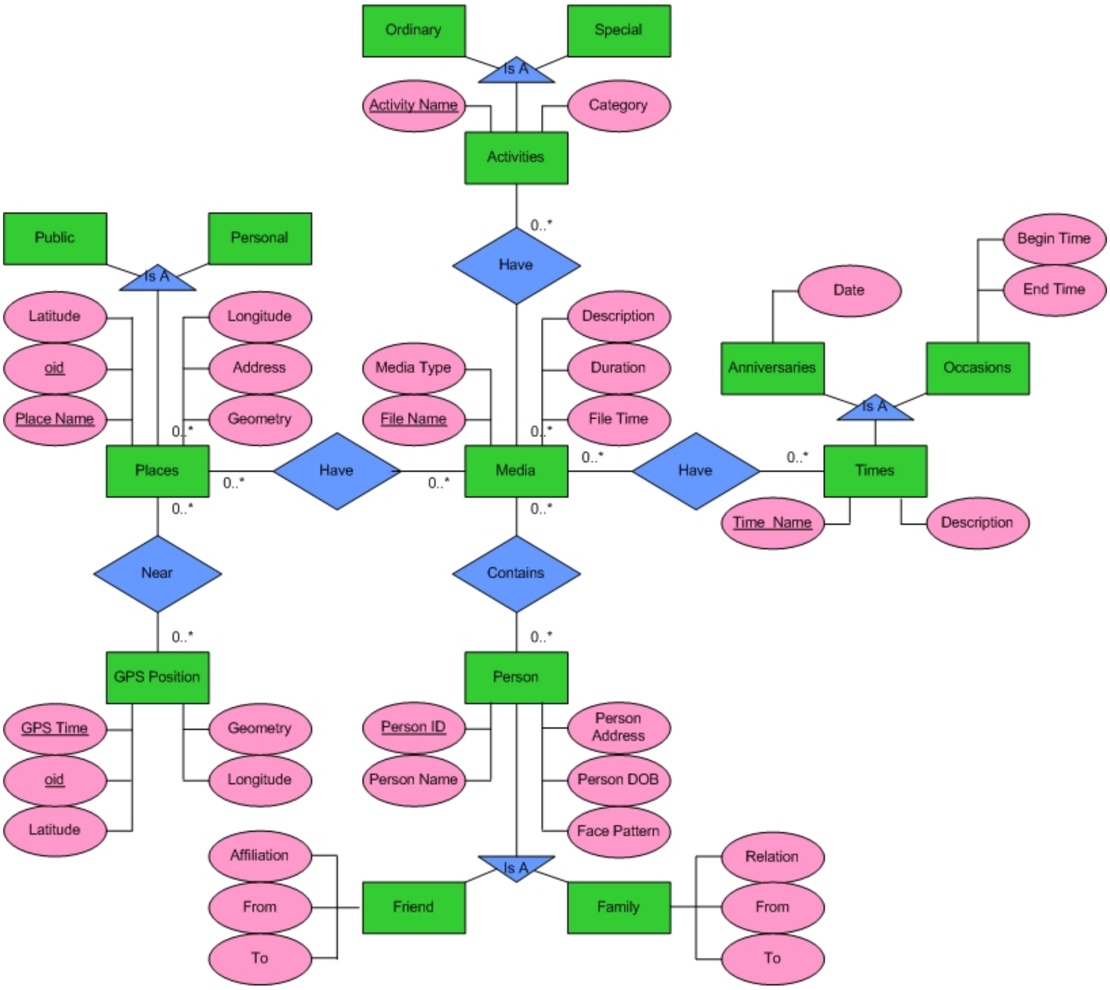
[10] University of Minnesota's Mapserver, http://mapserver.gis.umn.edu/index.html.

[11] Code Project, *C++ Winsock Client to Server File Transfer - Made Easy*, http://www.codeproject.com/useritems/WSFileTransfer.asp.

[12] FFMPEG Multimedia System, http://ffmpeg.mplayerhq.hu.

[13] Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction (Third Edition)*, Prentice Hall, 2003.

Appendix A.
Entity Relationship Diagram of the Database:



Appendix B.
Tabular Representation of the Database:

[1.1]

[2.1]