Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/081L07

Design and Implementation of a Smart Campus Guide Android App

Jaegeol Yim, Jaehun Joo*, Gyeyoung Lee and Kyubark Shim*

Dongguk University at Gyeongju, Gyeongbuk 780-714, Korea

Received: 13 Apr. 2013, Revised: 8 Aug. 2013, Accepted: 9 Aug. 2013
Published online: 1 Apr. 2014

Abstract: This paper introduces our Android application that recognizes the structure (a building and a statue, for example) in which a user is interested and displays useful information about the structure. This application records user's personal information when the user downloads this application, keeps track of the user while it is run, recognizes the structure when the user takes a picture of it, displays the picture along with a text showing some useful information about the structure, and plays a video that is closely related to the structure. This paper introduces our design and implementation of the application in detail. The techniques introduced in this paper can be used in mobile augmented reality, ubiquitous learning, and location based services.

Keywords: Mobile application, VoD, Location-Based, Database .

1 Introduction

As the electronic techniques advance, computing machines have been miniaturized and smart phones are equipped with powerful processors and large memories. In the consequence, various services become available on smart phones. Since a smart phone is a personal belonging, it is an excellent candidate device on which a context-aware service is provided. As an example context-aware service on smart phones, we have picked the campus guide and we are introducing our implementation of it in this paper. The campus guide consists of the server and the client. The main features of the client include determining the current location of the user and the building the user is watching and playing the video which is closely related to the building. In order to realize the client's features, the server consists of many components including the streaming server and a database server. The streaming server takes charge of delivering video content to the client whereas the database server takes charge of storing and retrieving information of the videos. When a new video is obtained and stored at the archive the path of the video file will be stored in the database.

The campus guide consists of the server and the client. The main features of the client include determining the current location of the user and the building the user is watching and playing the video which is closely related to the building. In order to realize the client's features, the server consists of many components including the streaming server and a database server. The streaming server takes charge of delivering video content to the client whereas the database server takes charge of storing and retrieving information of the videos. When a new video is obtained and stored at the archive the path of the video file will be stored in the database.

The service scenario of our campus guide is summarized as follows:

1) When the application is first downloaded by a user, the application takes the user's personal information and saves it in the database.
2) When the user starts running the application, it renders the campus map.
3) When the user takes a picture of a building, it recognizes which building is taken.
4) It displays the picture along with a text showing some information about the picture.
5) It plays the video closely related to the building.
6) After playing a video, it goes back to step 2.

There are many published research results that are similar to ours. For example, [1] introduced a mobile application[2] that displays a multimedia content relative to the object in the photo when the smart phone user takes

* Corresponding author e-mail: yim@dongguk.ac.kr, givej@dongguk.ac.kr

a picture of an article on exhibition in a museum. In [3], a mobile education application[4], which displays educational contents relevant to the object in the photo when a child in a museum or a park takes a picture with a smart phone, is introduced. In [5], a mobile application that figures out the appropriate key words representing objects in the photo when the user takes a picture of a landmark and retrieves images from open image databases with the key words to display them on the smart phone is introduced.

However, none of them plays videos. One of the unique features of our application is that it plays a video which is closely related to the architecture when a user takes a picture of it. In order to make it play a video, we have to install a streaming server.

Another unique feature is the way of identifying the architecture in the picture taken by the user. All the existing applications use an image recognition technique to identify objects in a photo. An image analysis process is time consuming. For example, suppose that a camera image has 10 feature points and the map image has 32 feature points. Then, there are 4 billion different possible 4-point correspondences [6]. In contrast to other existing applications, our application does not use the photo image at all. Our method is originated from the fact that modern smart phones are equipped with pretty accurate sensors. A GPS device on a smart phone measures the location of the phone with as little as 10 meter error. A compass on a smart phone measures the orientation of the phone with less than 2 degree error. The proposed method calculates the formula of the line of sight of the camera with its location and orientation. Then it finds any element of the map that intersects with the line of sight. Among the intersecting elements, the closest one represents the object on the photo if the distance between the camera and element is close to the focus distance of the camera. The time for obtaining the sensor data and calculating the formula of the line of sight is so short that we can ignore it in estimating the process time of our method. The process time is dominated by the time to find the elements of the map that intersect with the line of sight of the camera. The execution time of this process is proportional to the number of elements of the map around the location of the smart phone. There are no more than a few hundreds of such elements. Therefore, our method is radically faster than any existing photo recognition methods. This recognition method was first introduced in [7] and this paper is the first application that makes use of the method.

2 RELATED WORKS

Our campus guide is a kind of VOD (video on demand) system because it plays a video when the user selects the menu. Since it is running on a smart phone it is a mobile VOD service. There are so many published research results related with mobile VOD. [8] proposed cache schemes to reduce the waiting time of the clients of VOD. Ensuring service continuity between fixed and wireless networks has been one of hot research topics. [9] proposed an approach appropriately selecting a broadcasting scheme which minimizes the delay. [10] proposed an apparatus for easily setting up an IPTV interactive digital channel.

Context-aware service[11] is desired but there are not many commercially successful context-aware services. [12] focuses on the methodology of context-aware service design. It starts with building a Petri net model of the system and refines the model into a system design. Context-aware services heavily rely on specific context information and cannot be realized in highly distributed and decentralized environments. [13] addresses this problem by combining the ideas of context-aware frameworks and autonomous data dissemination.

[14] investigates context-aware pervasive service composition (CAPSC) requirements and designs CAPSC architecture. In [15], they determined some essential principles for developing and implementing context-aware applications. Then, they proposed a context-aware framework for an intelligent university mobile organizer by using sprin
g.net technology. Location is one of the essential factors of context[16]. [17] proposed a hierarchical localization scheme. [18] presents design and application of a system for tracking mobile search and rescue robots. [19]

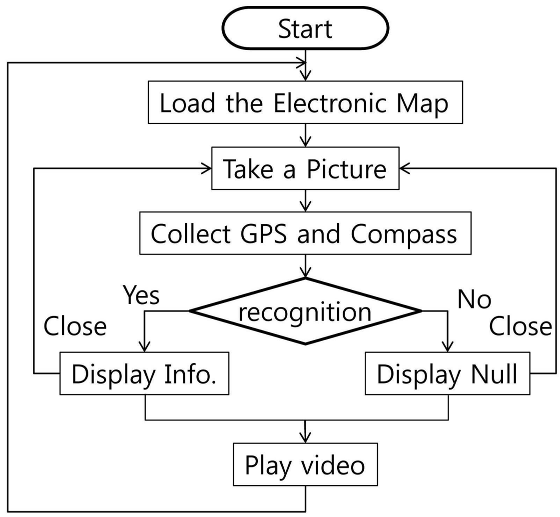

Fig. 1 The process of the application.

3 DESIGN OF THE SYSTEM

Whenever the user takes a picture, the application recognizes the building in the picture taken and plays the video related to the building. The application repeats this process until the user terminates the application. This process is summarized in Figure 1.

Our application plays a video which is related to the building in the picture. Video files are usually huge and stored in an archive system. Information about videos in the archive should be stored in a database so that it can be easily retrieved. Our database has many tables including the following three: 1) genreTable consisting of genreID, genreName, genreIcon, description attributes, 2) videoTable consisting of videoId, videoTitle, videoPath, metadata, ..., attributes and 3) genreVideoTable consisting of genreID and videoId attributes. The type of metadata attribute in videoTable is defined by XML and the metadata file for the video will be saved here.

Since the video files are stored in an archive, our system should be a kind of client-server system as shown in Figure 2. In the server, in addition to a database server, we need a streaming server which takes charge of controlling and delivering media. There are quite a few free streaming servers and implementing a streaming server is quite easy. We are installing Darwin [20] open source streaming server as our streaming server. It will run on Microsoft Windows Server 2008.

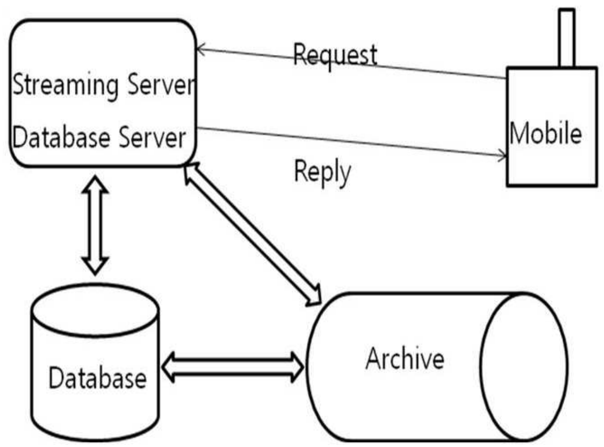

Fig. 2 Main components of our system.

The method of recognizing objects on photos taken by a smart phone camera we are using in this paper makes use of electronic maps. So far, the accessories attached to a mobile phone have not been accurate enough to locate the phone on a map, to determine the camera orientation, and to measure the focus distance of the camera. Therefore, all the existing picture recognition systems rely on image processing techniques. Many of them do make use of GPS and compass data to narrow down the scope of the images to be compared with the photo image, but they all rely on image processing techniques at the final stage of photo recognition. Now, it is changed. They have been improved, and the measurement error of a GPS (compass) on a recently released smart phone is about 10 meters (less than 2 degrees). The process of recognizing the building taken by the camera on the smart phone is described as follows.

1) We obtain an electronic map describing the physical area of the application system. Considering an application of museum guide, we need a drawing (AutoCAD drawing, for example) of the museum building. There exist drawings for all big buildings. The level of detail of the drawing is closely related to the purpose of the application. Considering a campus guide on the level of building, the object recognition process should be able to identify the name of the building on the photo. An example of an electronic map for this application is shown in Figure 3. The electronic map consists of edges representing the outline of Natural Science Building, Gymnasium Building, Student Hall, and so on, where an edge is represented by a pair of points, its start and end. A point is represented by a pair of real numbers, (longitude, latitude).

2) With the sensor data, we determine the location of the smart phone. In Android, ”LocationManager” class provides a method that returns the location of the smart phone. Most of the smart phones are equipped with a GPS receiver, WIFI device, 3G or 4G communication device. ”LocationManager” uses data from these devices to determine the location of the smart phone.

3) We collect the orientation data, namely azimuth, pitch and roll values. In Android, ”SensorManager” class provides a method that returns those values.

4) Using the ”Camera” class, we get the focus distance of the camera. In Android, ”Parameters” class nested in ”Camera” class provides a method, getFocusDistance, which returns the focus distance.

5) We execute our ”Object Recognition Algorithm” to identify the object on the camera. This algorithm calculates the formula representing the line of sight of the camera with the location and orientation data, finds the element (edge) of the electronic map intersecting with the line of sight and its distance is close to the focus distance. If there is no such element, then the algorithm concludes that the object on the photo is not something represented on the electronic map, maybe a person. If there is such an edge (element), then the building (the point of interest) whose outline contains the edge is determined as the object in the photo. Our ”Object Recognition Algorithm” is described in Figure 4 [7].

```
NaturalScience
129.198050, 35.862546
...
129.198050, 35.862546
Gymnasium
129.196986, 35.862514
...
129.196986, 35.862514
StudentsHall
129.195878, 35.862025
...
129.195878, 35.862025
...
```

Fig. 3 A part of an example electronic map of a university campus.

objectRecognitionAlgorithm (fd, location, azimuth, pitch, eMap)
// fd: focus distance ;
// location: (longitude, latitude, altitude)
Step 1: With location, (x, y, z), and azimuth, calculate the following formula whose slope is obtained from the azimuth value.

$ax + b = y$ ---- (eq. 1)

Step 2: Find S = { s | s is an edge in eMap and s intersects eq. 1}.
Step 3: If S is an empty set then return NIL
Step 4: Find the edge e in S which is closest to location. Delete e from S. Let the building whose outline contains e be a "candidate building" and the intersect point of e and eq. 1 be (x', y').
Step 5: Let the distance between (x, y) and (x', y') be d.
Step 6: If "z+d*tan(pitch)" is between the bottom and the top of the "candidate building" then jump to Step 8
Step 7: Go to Step 3
Step 8: Let the distance between (x, y, z) and (x', y', x+d*tan(pitch)) be dist. If (|dist − fd| < threshold) then return "candidate building" else return NIL, where threshold is a small number representing the error of getFocusDistance.

Fig. 4 The algorithm of recognizing the building taken by the camera.

4 IMPLEMENTATION

We implemented our client on Android [21]. For the purpose of rendering a video stream, we can use either MediaPlayer or VideoView. We chose to use VideoView in our MyMediaPlayer because it is easier to handle. What we need to do is to override just one method, onCreate, of Activity as shown in Figure 5. VideoView plays a video streamed by setVideoURI(). If we want to play a local video, we can use setPath() instead of setVideoURI(). path in the code contains video's URL.

```
setContentView(R.layout.videoviewplayer);
videoView = (VideoView) findViewById(R.id.videoView);
videoView.setVideoURI(Uri.parse(path));
MediaController mediaController = new MediaController(this);
videoView.setMediaController(mediaController);
videoView.requestFocus();
videoView.start();
```

Fig. 5 A part of the player module.

In the previous section, we said that the main purpose of the database is to store and retrieve video information. In addition to the main purpose, we are planning to use the database to provide context-aware services. To this end, we have tables of recording subscribers' information, who watched what video, to what genre which video belongs to, and so on as shown in Figure 6.

Our mobile application consists of the classes shown in Figure 7. IdentifySubjectMain contains methods of obtaining GPS information to determine the user's location, of scanning sensors to collect sensor values, of taking a picture with the camera, and so on. Using LocationManager class we can easily obtain GPS information. In order to instantiate this class, we use the following sentence:
Context.getSystemService(Context.LOCATION_SERVICE).
Then, we register the listener (mListener) with the Location Manager (mLocMan) to receive location updates:
mLocMan.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 0, mListener).
The Location Manager (mLocMan) calls onLocationChanged
(Location location) defined in mListener when the user location changes. That means, we have to define onLocationChanged(Location location) in mListener which is a LocationListener().

In order to access the device's sensors, we have to use SensorManager. The following sentence creates a new instance of SensorManager:
getSystemService(SENSOR_SERVICE);
The SensorManager calls onSensorChanged(int sensor, float[] values) when there is change on the registered sensor. So, we register SENSOR_ORIENTATION as follows:
sm.registerListener(this, SensorManager.SENSOR

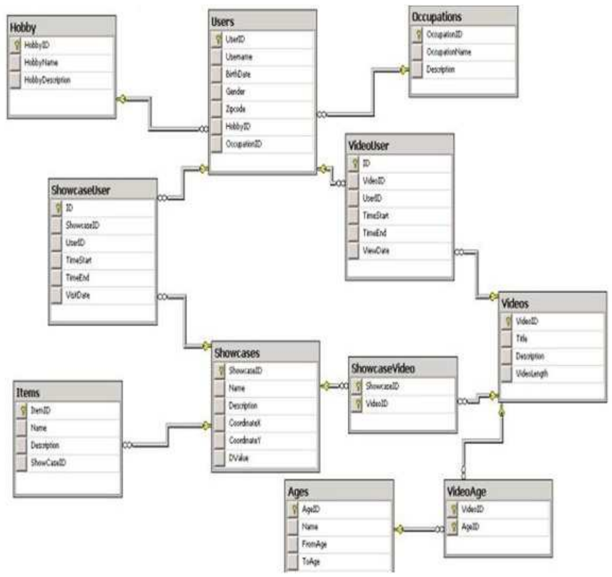

Fig. 6 A description of the database.

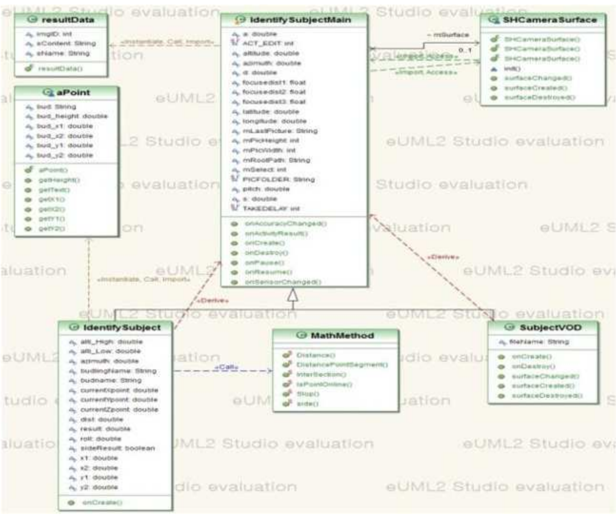

Fig. 7 A class diagram of the application.

_ORIENTATION)

Among the sensor values, the application uses azimuth and pitch as follows:

```
if (sensor ==
    SensorManager.SENSOR_ORIENTATION) {

        azimuth = values[0]+7.5;
        pitch = values[1];

}
```

Note how we get azimuth value. 7.5 is the difference between the magnetic north and the map north. SHCameraSurface contains methods of showing the picture taken by the camera. The class aPoint is to record the coordinates of a point. IdentyfySubject is the class of identifying the building in the picture. The process described in Figure 4 is implemented in this class. In MathMethod class, many mathematical functions are implemented. These functions are mostly used by IdentyfySubject. Given an angle, a point and a line segment, DistancePointSegment() returns the distance from the point to the line if the line defined by the angle and the point intersects the given line segment. Given an angle obtained from the sensor, Slope(angle) returns the slope determined by the angle. Given three points p1, p2, and p3, IsPointOnline() determines if p1 is on the line defined by p2 and p3. SubjectVOD is a class to play a video. The user interface of playing a video consists of three buttons (play, stop, exit) and a screen called a surfaceview on which the video is rendered. On each of these three buttons an OnClickListener is defined. surfaceCreated() creates a media player object and defines the path of the video to be played. The names of the other methods are self-explaining.

5 EXPERIMENTS

We have performed experiments of testing the accuracy of LocationManager at many different locations. An example of the locations is the following: A: 129.196990, 35.861779 We obtained the coordinates from Google Map. The measurement obtained at A by LocationManager on a Galaxy S2 was (129.1968912, 35.8619749). The distance from longitude 129 degree to 130 degree at latitude 36 degree is about 91.29 m and the distance from latitude 35 degree to 36 degree is about 110.941 m. Therefore, we can conclude that the average error of the measurements obtained at location A is about 13.73948193 m. Considering all the measurements obtained from the other locations, we concluded that the average error of LocationManager on Galaxy S2 phone is 11.52493934 m. After testing LocationManager, we performed experiments of testing the accuracy of the compass on a Galaxy S smart phone. In these experiments, we measured azimuth of our Galaxy S while it is aiming at the point C (129.196499, 35.862063) from the point A (129.196271, 35.861620). Referring to these coordinates, we can find that the slope of the line defined by those two points is 2.361227042. The azimuth of the line of sight defined by points A and C can be calculated by the following equation:

$$\text{Azimuth} = 90\text{-Atan(Slope)}$$

The azimuth we obtained was 22.95307631. This is the azimuth on the map. The magnetic north is not the same as the map north. Using a military compass and map, we found out the difference is about 7.5 degree. From these calculations and measurements, we concluded that the azimuth of our Galaxy S should be 30.45307631. After many measurements of the azimuths, we concluded that the average error of our measured azimuths is about 15.4 degree.

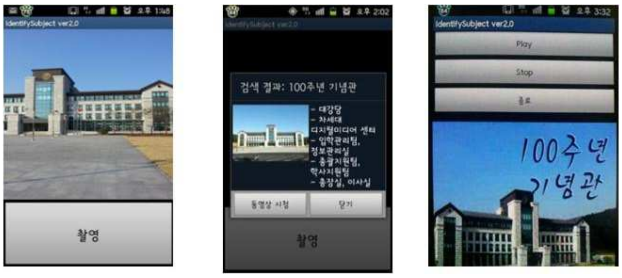

Fig. 8 Screenshots. (a) The initial screen. Click the button at the bottom to take a picture. (b) It shows the picture together with information about the building. (c) It plays the video related to the building.

Finally, we performed experiments of running the application on Galaxy S2 taking pictures of buildings. Three screenshots are shown in Figure 10. The experiments showed that the application 100% correctly plays the right video when it was run within 50 meters from a building.

6 CONCLUSIONS

This paper described the development of our campus guide mobile application. When the user takes a picture of a building, it identifies the building, displays information about the building and plays a video related to the building. The process of identifying the building in a picture is the first implementation of the idea proposed in [7]. Our experimental results showed that the application recognizes the building taken by the camera practically well.

ACKNOWLEDGEMENTS

Yim's research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(NRF-2011-0006942). Shim's Research was supported by Dongguk University Fund(2011).

References

[1] E. Bruns, B. Brombach, O. Bimber, "Mobile Phone-Enabled Museum Guidance with Adaptive Classification," IEEE Computer Graphics and Applications, **28**, 98-102 (2008)

[2] Mert Bal, Rough Sets Theory as Symbolic Data Mining Method: An Application on Complete Decision Table, Information Science Letters, **2**, 35-47 (2013)

[3] K. Mitchell, N. Race, "uLearn: Facilitating Ubiquitous Learning Through Camera Equipped Mobile Phones," IEEE International Workshop on Wireless and Mobile Technologies in Education, (2005)

[4] Nabila Kanwal, Aihab Khan, Efficiency Analysis in Outsourced Database using One Time Signature Scheme, Information Science Letters, **2**, 1-8 (2013)

[5] N. Abe, W. Oogami, A. Shimada, H. Nagahara, R. Taniguchi, "Clickable Real World: Interaction with Real-World Landmarks Using Mobile Phone Camera," IEEE Region 10 Conference, 914-917 (2010)

[6] H. Hile, G. Borriello, "Positioning and Orientation in Indoor Environments Using Camera Phones," IEEE Computer Graphics and Applications, **28**, 32-3 (2008)

[7] A New Method for Recognizing Objects in Photos Taken by Smart Phones, submitted to a journal for possible publication

[8] P. Sumari, A. Rahiman, "Caching Scheme for Handheld Device in Mobile Video-on-demand System," CGIV, 49-54 (2010)

[9] S. Park, M. Hwang, S. Moon, H. Youn, "An Efficient VoD Scheme Providing Service Continuity for Mobile IPTV in Heterogeneous Networks", CIT, 2589-2595 (2010)

[10] M. Lee, "The Service Generation Apparatus for IPTV Interactive Digital Channel," International Journal of Advanced Science and Technology, **10**, 37-50 (2009)

[11] Ilyas Saleem, Hamza Nawaz, Istaqlal Ahmed, S. Muzahir Abbas, Analytical Evaluation of Tri-band Printed Antenna, Information Sciences Letters, **1**, 85-89 (2012)

[12] T. Lu, Q. Hao, "Scenario-Based Context-Aware Service Design," ICSSSM, 1-6 (2010)

[13] C. Jacob, D. Linner, I. Radusch, S. Steglich, "Context-aware Data Dissemination and Service Adaptation," ISTMWC, 1-5 (2007)

[14] J. Zhou, J. Riekki, "Context-Aware Pervasive Service Composition," 2010 International Conference on Intelligent Systems, Modelling and Simulation (ISMS), 437-442

[15] S. H. Mirisaee, A. M. Zin, "A Framework for Context-Aware University Mobile Organizer," International Conference on Future Computer and Communication, 251-255 (2009)

[16] S. M. Abbas, Ilyas Saleem, Bilal Ahmed, Hunaina Khurshid, UWB Antenna with Parasitic Patch and Asymmetric Feed, Information Science Letters, 2, 27-33 (2013)

[17] G. Chen, Y. Zhou, J. He, "A Localization Scheme for Underwater Wireless Sensor Networks," International Journal of Advanced Science and Technology, **4**, 9-16. (2009)

[18] A. Ko, H. Lau, "Robot Assisted Emergency Search and Rescue System with a Wireless Sensor Network" International Journal of Advanced Science and Technology, **3**, 69-78 (2009)

[19] K. Lee, K. Oh, "Ubiquitous Robot and Its Realization," Applied Mathematics and Information Science (AMIS), 311S-321S
[20] D. P. Darwin Streaming Server. Darwin Streaming Server. Available Online, http://dss.macosforge.org (2012)
[21] http://developer.android.com

Jaegeol Yim received the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, Location Based Service, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in Korean Language), and several undergraduate textbooks.

Gyeyoung Lee received the M.S. degree in Computer Science from the Dongguk University in 1982 and the Ph.D. degree in Computer Engineering from the Dankook University in Korea in 1992. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju, Korea. His current research interests include Petri net theory, AI systems and Speech processing systems.

Jaehun Joo is a professor of Department of Information Management at Dongguk University (Gyeongju) in Korea and Editor-in-Chief, Journal of Korea E-commerce Research Aca demy. Also he was a Visiting Professor of Department of Management at University of Nebraska-Lincoln. He received his Ph. D. from Busan National University. His areas of research interest are electronic commerce, business ecosystems, collective intelligence, semantic web, and knowledge management. He published many papers in Int. J. Human-Computer Studies, Journal of Sustainable Tourism, Information Systems Management, International Journal of Industrial Engineering, Expert Systems with Applications, Journal of Computer Information Systems, Decision Support Systems, etc.

Kyubark Shim received the M.S. and the Ph.D. degrees in Statistics from the Dongguk University in 1986 and 1993, respectively. He is a Professor in the Department of Information and Statistics at Dongguk University at Gyeongju, Korea. His current research interests include Computational Statistics, Reliability Test, Analysis of Statistical Data.