A Three-Layered Framework for Supporting the Interactions in a Social Activity

Hsien-Chou Liao and Ke-Chun Liao

Department of Computer Science and Information Engineering, Chaoyang University of Technology, 168 Jifong E. Rd., Wufong Township Taichung County, 41349, Taiwan (R.O.C.)

hcliao@cyut.edu.tw

Abstract

The capabilities of mobile devices are getting powerful. The support of social interactions via mobile devices becomes an important issue in recent years. The interaction of people in a social activity is especially important but has not been addressed in previous studies. Therefore, a three-layered framework, call INTRO, is proposed in this paper. INTRO consists of social, preserving, and promoting layers. A mobile device supports the disseminating and collecting social profiles automatically based on a pre-defined familiar graph in the social layer. The collected profiles are then updated to a computer in the preserving layer. In advance, the social relationships can be updated to a social Web site for promoting the further social interactions. Two experiments of the profile dissemination were designed to measure the transmission time via Bluetooth and Wi-Fi, respectively. Two prototypes of INTRO in the social and preserving layer were also implemented to demonstrate its feasibility.
Keywords: social network, familiar graph, social-mobile application

I. Introduction

The computing power evolution and wireless communication capability of mobile devices extend the application domains of mobile applications. Mobile devices are becoming an important instrument for delivering information technology into people daily life. They also change the way of people interaction. Many researches focus on how to support the social interactions via the mobile devices. For example, N. Eagle et al proposed social-mobile software for smart phone, called social serendipity [1]. It is a mobile-phone-based system using Bluetooth communication for face-to-face interactions between nearby users. Its purpose is mainly to facilitate interactions between physically proximate people through a centralized server. In addition, R. Beale proposed a set of social applications on the smart phone [2]. For examples, the localized dating, file sharing, community building, mobile blogging, *etc*. It demonstrates the ability of smart phone on supporting the social interactions.

However, the interaction of people in a social activity is especially important but has not been addressed in previous studies. A social activity discussed in this paper is defined as an activity considered appropriate on social occasions, e.g., wedding, welcome reception, birthday party, *etc*. If a person knows nobody in a social activity, he has to explore the connections by himself. Oppositely, if the person has several friends in the activity, he can explore the connections faster via his friends. In addition, when a person is presented to his friend’s friend, they may interchange the name cards. The detail profiles must be presented to each other repeatedly since the information on the name card is too simple. It is inefficient and inconvenient way for social interactions. Besides, the profiles kept in the human memory may be lost after a period of time. When a person blunders an unfamiliar friend someday, he has to remind each other again.

Therefore, a three-layered framework, called INTRO, is designed to support the interactions in the social activity in order to overcome the above problems. INTRO consists of the social, preserving, and promoting layers. In the social layer, the relationships among people form a graph, called familiar graph (FG). A mobile device can support the dissemination of social profiles automatically based on the FG. A person can receive the profiles of the others as soon as possible when he joins a social activity. In the preserving layer, the received social profiles can be preserved in a PC or notebook for viewing. In the promoting layer, the preserved social relationships can be updated to a social Web site to increase the further social interactions. Two experiments of the profile dissemination were designed to measure the transmission time via Bluetooth and Wi-Fi, respectively. Two prototypes of INTRO in the social and preserving layer were implemented to demonstrate its feasibility.

The rest of the paper is organized as follows. Section II reviews and summarized related works. Section III presents the familiar graph and three-layered framework of INTRO. Section IV presents the experimental study. Section V presents the prototype of the social layer and preserving layer. Section VI gives the conclusion of our research.

II. Related Works

Some social-mobile applications were proposed in recent years. They attempt to support the social interactions from various aspects. A set of characteristics are defined to analyze these applications. Firstly, N. Eagle and A. Pentland developed architecture to detect an individual to other people nearby [1]. An application of the architecture, called Serendipity, combines an online introduction system to facilitate interactions between physically proximate people through a centralized server. Serendipity consists of three parts: BlueAware (BlueDar) and Serendipity Server. BlueAware can be executed in the background on many Bluetooth phones. BlueAware scans the unique Bluetooth identifier (BTID) number of mobile phones. It then records and timestamps the BTIDs encounters. BlueDar is a variation of BlueAware. It can be deployed in public areas. BlueDar transmits detected BTIDs to the Serendipity server via wireless network. The serendipity server can match the mobile profiles according to the thresholds set from their phone, and thus increase the interactions among physically proximate people.

R. Beale designs seven systems to demonstrate the ability of smart phones on supporting a wide range of social interactions [2]. Systems include Bluedating, BT Communities, BT Share, SmartBlog, *etc*. Bluedating application is a localized dating service. It allows users to enter their interests and desires. The application continually searches for other profiles over Bluetooth. If discovered profile and desired profile is match, the application informs both users by vibrating. BT Communities provides a software framework that can run and manage Bluetooth services. Many applications can be built based on the BT Communities, such as joke-sharing, or chat facility. BT Share is a peer-to-peer file-sharing system. Files can be identified as being public and open for sharing. The sharing files may encourage the fact-to-fact social interaction.

J. Kjeldskov and J. Paay designed a context-aware mobile information system, called Just-for-Us, facilitating sociality in public places [3]. Its design is mainly based on four ideas. Firstly, Just-for-Us makes the invisible visible by augmenting the user's physical surroundings. A "home screen' is pushed to the device displaying information corresponding to the district where the user is located. The home screen consists of the district's name and textual descriptions, an annotated panoramic view, and activity meter. Secondly, Just-for-Us supports Ad-Hoc communication about places, activities, and time. It notifies users when their friends are nearby and displays on the mobile devices automatically. The nearby stores, such as restaurants, coffee shops, *etc*., are also displayed on the screen for arranging the meeting places. Thirdly, Just-for-Us supports indexing recommendations of activities, such as coffee, food, or cultural events. It presents a list of recommendations sorted on the basis of user's history. Lastly, Just-for-Us represents activities within proximity and indexing to familiar places. People often use this information as important cues for where to go and what to do themselves. The prototype of Just-for-Us exemplified how its information and functionality can be tailored to the user's physical and social setting.

MoSoSo (Mobile Social Networking Software) is initially an project by Z. Klein and Y. L. Wong. Now, it represents a class of mobile application to support social interaction among interconnected mobile users [4]. Geographical location and time of a user is associated with its own social network to provide some novel services. For example, it enables a user to find the nearby friends in his social network, or share a photo of a specific location and time through the social network.

S. Konomi *et al.* developed a RFID-based conference support system, call DeaiExplorer [5]. The interconnected social network of a conference participant can be derived dynamically from a publication database. The interconnections of participants are established when they have ever attended the same conference or the co-authors of the same paper. DeaiExplorer is deployed in the conference site. The social network can help participants turn conference experiences into valuable long-term relationship. However, the participants may have privacy concerns by showing their social networks on the public display. And, DeaiExplorer cannot support the social interaction when the attended conferences or published papers of a participant are limited.

J. Leikas *et al.* utilizes a common public display for multi-users mobile applications to support the social interaction [6]. Based on the case study of a multi-player mobile game, three enabling factors for social interaction in a multi-player game are introduced. They are the context of the physical environment, the communication of human-human or human-machine, and the identification of a player under the anonymity and privacy considerations.

G. Kortuem and Z. Segall proposed a new concept, called wearable communities, by augmenting social networks with wearable computers [7]. It is based on the notion of augmented social space, where wireless personal area networks (WPANs) generate a sphere-like digital field. There are four interpersonal communication levels in the social space—intimate, personal, social, and public space. The level defines a region around the body. For example, touch happens in intimate space and group of people set up social distances about 10 feet or less. A wearable device can interchange user profiles with nearby devices when their digital social spaces are overlapped. A social network is then created and maintained using wearable computing devices.

According to the above description, social interaction is achieved by using various technologies. In order to analyze and realize these approaches, five characteristics are defined and listed below:

1. Social Activity/Public Area: Is the approach mainly designed for a social activity or a public area?
2. Automatic/Manual: Does the proposed application operate automatically or manually?
3. Friend/Stranger: Is the approach mainly designed for friends, strangers, or both?
4. Social Network: Is social network used in the approach?
5. Peer-to-Peer (P2P)/Client-Server (CS): Is the proposed system P2P or CS architecture?

The analysis results of the above approaches are summarized in Table 1. The results are not used to show which one is the best, but to reveal the differences among these approaches. They can also help us on realizing the upcoming social-mobile applications. According to the results, we found that most approaches focus on the social interactions of public area and operate automatically. There is no obvious difference among the approaches for the last three characteristics.

Among these related approaches, DeaiExplorer is much similar to INTRO except the architecture. DeaiExplorer uses a client-server architecture, whereas INTRO uses a peer-to-peer one. DeaiExplorer is originally proposed for a conference activity and easily adapted to a general social activity. However, the peer-to-peer architecture based on Bluetooth or Wi-Fi communications enables the detection of nearby mobile devices. These devices are important targets for the profile exchange in a social activity. Oppositely, if a centralized server is used for the clients in all the social activities, some additional information is needed for finding the clients in the same social activity. This means that a client cannot be operated automatically. If an individual server is deployed for every social activity for solving the above problem, the server deployment becomes a troublesome task. Therefore, the peer-to-peer architecture of INTRO is more suitable than the client-server one of DeaiExplorer for the social activity discussed in this paper.

Table 1. The analysis results of related approaches

Characteristics / Approaches	**S**ocial Activity/ **P**ublic Area	**A**utomatic /**M**anual	**F**riend/ **S**tranger	Social Network	**P**2P/**C**S
Serendipity [1]	P	A	S	No	C
Supporting social interaction with smart phones [2]	P	A	S	No	P/C
Just-for-Us [3]	P	A	F	No	P
MoSoSo [4]	P	M	F/S	Yes	P
DeaiExplorer [5]	S	A	F/S	Yes	C
Multi-user mobile applications [6]	P	M	S	No	C
WearCom [7]	P	A	F/S	Yes	P
INTRO	S	A	F/S	Yes	P

III. Approach

When participating in a social activity, a person is introduced to strangers by himself or his friends and thus expands his social relationship. However, it may be troublesome and inefficient for presenting personal profile repeatedly. It is also easy to forget without writing down. INTRO is primary designed to support the interactions by disseminating social profiles automatically. The dissemination is based on a social network, called familiar graph (FG), defined later. It is similar to the social network defined by E. F. Churchill [8]. In this section, FG is defined firstly and then the three-layered framework of INTRO is presented in the second subsection.

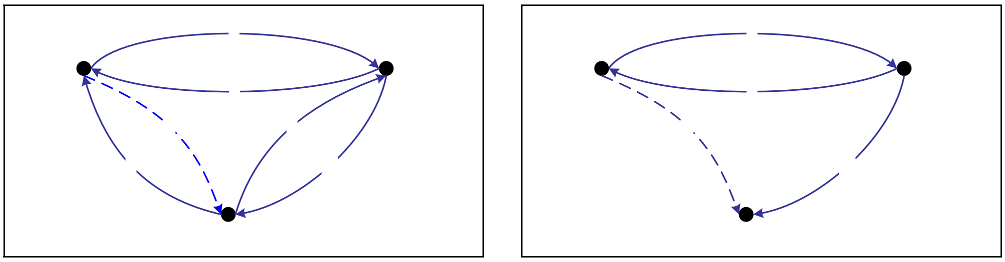
*A. **Familiar Graph***

The relationships between persons can be modeled as a graph. A node represents a person and a link represents the familiarity. Two persons may have different familiar level (FL) to each other. For example, 'Mark' may feel he is closely familiar with 'Ariel', but 'Ariel' may feel she is not so familiar with 'Mark'. The links is directed and weighted to represent the different FL of two persons to each other. Besides, the link of two persons may be established directly or indirectly. An indirect link means the relationship of two persons is established via a path formed by other common friend(s), i.e., node(s).

According to the above description, FG is a weighted directed graph. There are two types of links: direct or indirect, which are represented as solid or dashed line, respectively. The label of the link represents the FL. There are four FLs from zero to three in current definition. FL equals to zero and three means unfamiliar and closely familiar, respectively. The default FL of a link is zero. An example of FG is show in Fig. 1. In Fig. 1(a), there are three persons in the figure. The label of the link from 'Mark' to 'Ariel' is three means that 'Mark' feels that he is closely familiar with 'Ariel'. However, the label of the reversed link is one means that 'Ariel' feels she is not so familiar with 'Mark'. There is an indirect link from 'Ariel' to 'Tommy'. It is derived from the path ('Ariel', 'Mark', 'Tommy'). Since FL from 'Ariel' to 'Mark' is only one, the label of the indirect link from 'Ariel' to 'Tommy' is assigned as one, i.e., the minimal FL in the path. The complete rules of indirect link inference are presented in the later subsection. When the quantity of nodes is large, the FG becomes complicated. Therefore, the link with label zero is ignored in the later materials as shown in Fig. 1(b).

There are different FLs between persons. They are used to determine the profile contents to be transmitted. A person can setup his social profile for different FLs. The basic profile is text-based.

They may include multimedia files, such as picture or video, as shown in Fig. 2. The profile is getting details as the increasing of FLs. Assume the label of a link from a node P_1 to P_2 is n, all the P_1's profile which the FL is smaller or equals to n is combined and transmitted to P_2. A plain text example of the social profile for 'Tommy' with different FLs is also listed in Table 2.



(a) (b)

Fig. 1. Familiar graph examples (a) an example containing links with zero FL (b) a simplified example

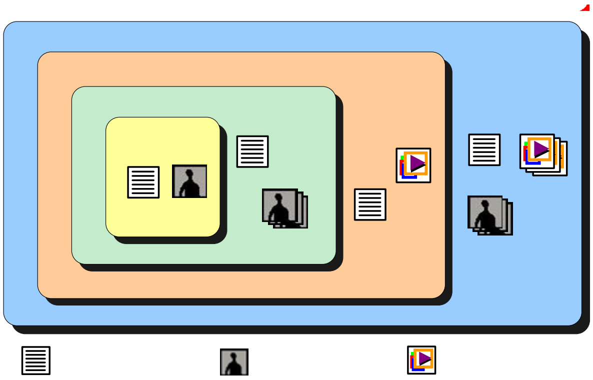


Fig. 2. The setup of profiles for different FLs

Table 2. An example of social profile for different FLs

FL	Social Profile
0	Name: Ke-Chun Liao Nickname: Tommy Sex: Male
1	Title: Graduate Student School: Chaoyang Univ. of Tech. Interest: Basketball, Tennis, PC Game
2	Birthday: 1982/1/1 E-mail: s9427607@cyut.edu.tw
3	Mobile: 0939-555666 Family: One elder brother and two younger sisters Home Addr.: No. 81, Lane 130, Sec. 2, Xinping Rd., Taiping City, Taichung County 411, Taiwan (R.O.C.)

The next issue of FG is the inference of a indirect link and the determination of its FL. For a person, he can determine the FL of out-direct-links. That is, he can actively adjust the details of profiles for every friend individually. Oppositely, the indirect links are established by a link inference mechanism. Assume there is more than one path from one node P_1 to another node P_2, it means that P_1 has several friends familiar with P_2. There should be an indirect link from P_1 to P_2. To determine the FL of the indirect link, the minimal FL of a path is found to ensure that only the profiles under the minimal FL can be transmitted via the path. Then, the maximal value of these minimal FLs is found to allow the profiles can be transmitted via the path with this maximal FL. The algorithm for the indirect link inference of FG and FL determination is listed in Fig. 3.

Algorithm Infer_Indirect_Link *(FG)*
Input: *FG* (a familiar graph)
Output: *FG'* (a modified FG)
begin
  for *every pair of nodes* (p_1, p_2) in *FG*
    explore all the paths from p_1 *to* p_2*, i.e.,* (p_1,…, p_2)
    find the minimal FL (Min_FL_i) *for every paths*
    find the maximal FL (FL_{max}) *among* Min_FL_i
    if FL_{max} > *FL of the current direct link from* p_1 *to* p_2
      set the link type from p_1 *to* p_2 *as 'indirect'*
    set FL of the indirect link from p_1 *to* p_2 *as* FL_{max}
    end if

    explore all the paths from p_2 *to* p_1*, i.e.,* (p_2,…, p_1)
    find the minimal FL (Min_FL_i) *for every paths*
    find the maximal FL (FL_{max}) *among* Min_FL_i
    if FL_{max} > *FL of the current direct link from* p_2 *to* p_1
      set the link type from p_2 *to* p_1 *as 'indirect'*
    set FL of the indirect link from p_2 *to* p_1 *as* FL_{max}
    end if
  next
end

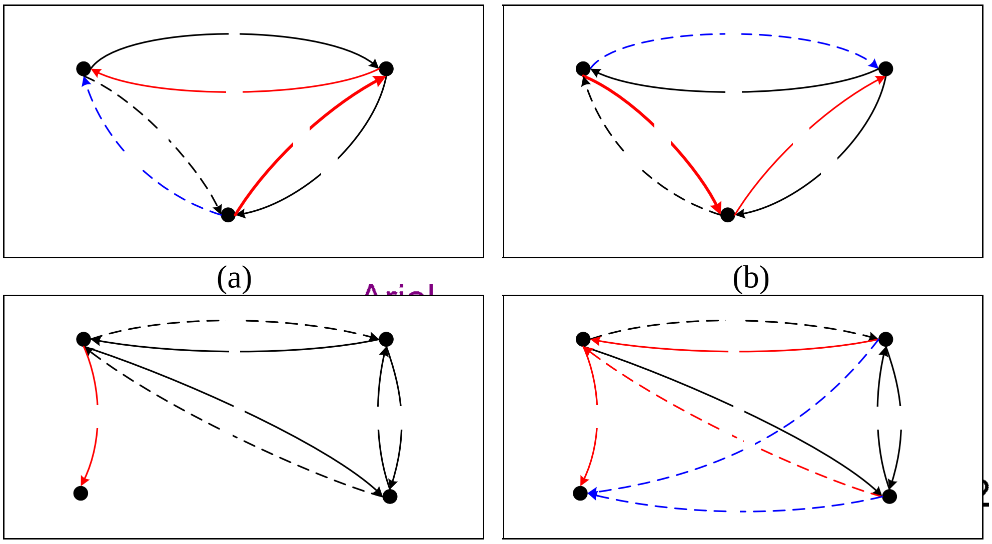
Fig. 3. The algorithm of indirect link inference and FL determination

The inference process is activated when the FL of a link is adjusted by a person or new person are added into the FG. An example is shown in Fig. 4. The initial FG is shown in Fig. 2

The three layers of INTRO are presented as follows:

(1) Social layer: In this layer, the mobile devices exchange social profiles in social activities based on the FG. Many social profiles are collected automatically.
(2) Preserving layer: In this layer, the mobile device can synchronize the collected social profiles to the PC or notebook for preservation. People can view the preserved social profiles on their own computer. It is also user-friendly to view profiles on a PC.
(3) Promoting layer: The PC software can synchronize the social relationship with a social Web site. The update profiles and relationships are helpful to promote and increase further social interactions.

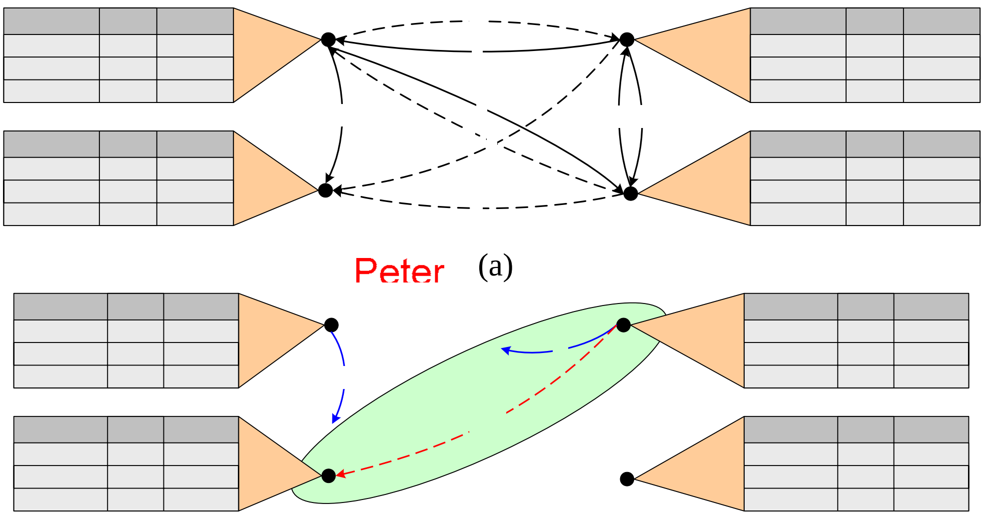
After the conversation of 'Tommy' and 'Mark', 'Tommy' adjusts the FL of 'Mark' from zero to two as the bold red line Fig. 4(a). It causes the FL of the link from 'Tommy' to 'Ariel' is also adjusted from zero to two after the inference process. Similarly, when 'Ariel' adjusts the FL of the link to 'Mark' from one to two, the link type is also changed from indirect to direct. It also causes the FL of the link from 'Ariel' to 'Mark' is adjusted from one to two. The line type is also changed from direct to indirect as show in Fig. 4(b). Then, 'Peter' joins the social activity. He is a friend of 'Ariel' and she is closely familiar with 'Peter' as shown in Fig. 4(c). It causes that two indirect links from 'Mark' and 'Tommy' to 'Peter' can be established after the inference process as show in Fig. 4(d). 'Peter' can receive the profiles of 'Mark' and 'Tommy' as soon as possible.



(a) (b) (c) (d)

Fig. 4. An illustration of link inference

When persons stay in a social activity, the FLs of a FG can keep consistent according to the link inference mechanism. Every mobile devices contains a copy of the FG. However, a person may adjust the FLs of a specific friend after the social activity. When two persons meet at a different social activity someday, the FLs of the same link stored in different mobile devices may be different, i.e., conflict occurred. Thus, a FL conflict resolution mechanism must be defined to solve the problem. An example of the FL conflict is shown in Fig. 5.



(a)

(b)

Fig. 5. An example of FL conflict (a) original FG (b) conflict occurred

The original consistent FG is shown in Fig. 5(a). Every mobile devices stores a copy of the FG represented as a table. The columns 'In' and 'Out' represents the in-link and out-link with the FL and link type. The symbols 'D' and 'I' of the link represent the link is directed or indirect, respectively. In the society activity, the link inference mechanism is activated automatically to keep the consistency of FLs. However, a person may adjust the FL after the social activity. In Fig. 5(b), the FL of the direct link from 'Ariel' to 'Peter' is adjusted by 'Ariel' from three to two. The indirect link from 'Mark' to 'Peter' is established via this direct link. The FL of this indirect link stored in the mobile devices of 'Mark' or 'Peter' cannot be adjusted simultaneously. If 'Mark' and 'Peter' meet in another social activity without 'Ariel', it is debatable to transmit the profiles of 'Mark' with the third level to 'Peter'.

Node	In	Out
Mark	3(D)	2(I)
Tommy	2(I)	2(D)

Ariel

The conditions simply occur on the indirect links since they are built by the link inference mechanism. They should be valid only in the social activity and invalid after a specific period. Therefore, a parameter, called time-to-invalid (TTI), is defined for the FL conflict resolution mechanism. A person can determine the value of TTI, such as one day, one week, never expired, *etc*. Here are the resolution steps:

1. A person setups the value of TTI for indirect links. The default value is one day.
2. When an indirect link is built, the build-time (BT) and the FL of the original direct link (OrgFL) are stored simultaneously.
3. If the social profiles are sent via an indirect link, the current time (CT) is checked whether it exceeds the sum of the link's BT and TTI. If yes, the indirect link is reset back to a direct link with OrgFL. The sending of profiles follows the original direct link. Otherwise; the sending is according to the FL of the indirect link.

The resolution mechanism is activated for the sending of profiles. An indirect link will be reset to the FL of the original direct link when the time exceeds the TTI unless people actively adjust the FL of the link. It is efficient since it is unnecessary to check and reset the FLs of the whole FG periodically.

*B. **Three-Layered Framework***

The design of the FG enables the transmission of social profiles in the social activity. The quantity of received profiles causes that it is unsuitable to keep all the social profiles in a mobile device. On the other hand, many Web sites provide effective ways for people to increase the social interactions, e.g., MySpace, Facebook, Linkist, *etc*. Although the social relationships can be established by yourself or your friends, it is still inefficient to setup in the Web site. Therefore, if social profiles can be collected in a social activity automatically and synchronized with social Web sites, it is helpful to enhance the further social interactions. According to the above concept, a three-layered framework of INTRO is designed as shown in Fig. 6.

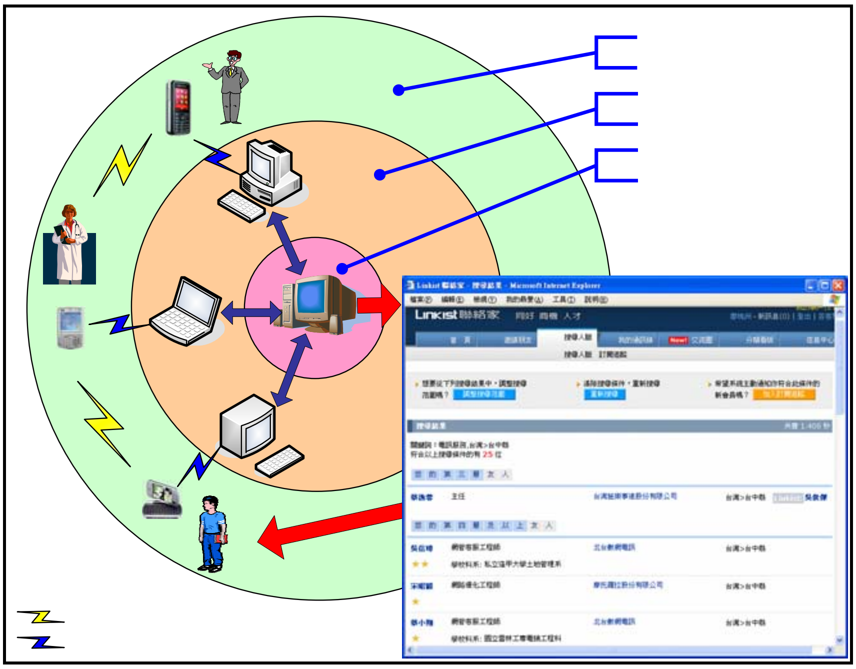


Fig. 6. The three-layered framework of INTRO

Therefore, the three-layered framework can support the social interactions covering from the physical social activities to virtual relationships in the cyberspace.

IV. Experimental Studies

The transmission of social profiles is important for fulfilling social layer in a social activity. Two experiments were designed to evaluate the transmission time with respect to the number of mobile devices and the communication channels, including Bluetooth and Wi-Fi. The experiments are presented in details in the following sections.

A. ***Bluetooth***

In the first experiment, the communication channel is Bluetooth (version 1.1). The mobile device is HP iPAQ Pocket PC h5450. Four different sizes of a social profile, 64KB, 128KB, 256KB, and 512KB are prepared for the transmission. Initially, the same size of social profile is stored in every mobile device. When the experiment is started, every device attempts to transmit its profile to the others. The elapsed time, called transmission time, is measured when all the devices finishing the profile transmission. The transmission time of three, four, and five devices for various profile sizes is listed in Table 3.

Table 3. The transmission time of Bluetooth

Number of Devices	Profile Size (KB)	Total Size (KB)	Transmission Time (min.)	Average Time of a Profile (min.)
3	64	384	2.38	0.40
	128	768	4.67	0.78
	256	1,536	6.43	1.07
	512	3,072	7.84	1.31
4	64	768	5.45	0.45
	128	1,536	6.47	0.54
	256	3,072	8.32	0.69
	512	6,144	11.55	0.96
5	64	1,280	7.13	0.36
	128	2,569	10.58	0.53
	256	5,129	12.45	0.62
	512	10,278	16.67	0.83

Assume the number of mobile devices is denoted as *n* and the profile size is denoted as *S*, the total transmission size equals to $n \times (n-1) \times S$. Therefore, the transmission time is increased as *n* in Table 3. It is not linear since the complexity of total size is $O(n^2)$. In advance, the transmission time of six to twenty devices is predicted and computed according to the equation (1).

$$n \times (n-1) \times \text{average time of a profile} \quad (1)$$

The predicted results are listed in Table 4. The actual and predicted transmission time are also depicted in Fig. 7.

Table 4. The predicted transmission time of Bluetooth

Number of Devices	Profile Size (Average Time)			
	64 KB (0.4 min.)	128 KB (0.62 min.)	256 KB (0.8 min.)	512 KB (1.03 min.)
6	12.07	18.47	23.88	31.03
8	22.54	34.47	44.57	57.92
10	36.22	55.40	71.63	93.09
12	53.12	81.25	105.05	136.53
14	73.24	112.02	144.84	188.25
16	96.59	147.72	191.00	248.24
18	123.15	188.34	243.53	316.50
20	152.93	233.89	302.42	939.04

(unit: minute)

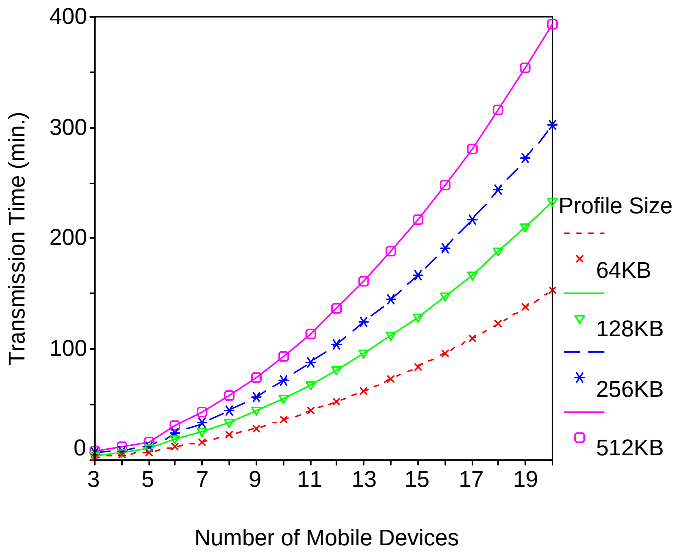


Fig. 7. The actual and predicted transmission time using Bluetooth

According to the predicted results, the transmission time is increased rapidly as the number of mobile devices. One reason is that the complexity of the total transmission size is O(n^2), the other is that Bluetooth is only allowed one connection at a time. Therefore, if one device attempt to connect to another busy device, the device must wait until the finish of profile transmission. Besides, the elapsed time of twenty devices for transmitting 512KB profile is about 939 minutes. Although a new Bluetooth version 2.0 plus EDR (Enhanced Data Rate) technology is three times faster than Bluetooth version 1.1, it seems impractical for large number of devices.

B. ***Wi-Fi***

The second experiment is the measurement of transmission time under Wi-Fi communication. The bandwidth of Wi-Fi is larger than Bluetooth. Therefore, the number of devices and the profile size is increased to twenty and 1MB, respectively. According to the same experimental steps of Bluetooth, the results are listed in Table 5. In advance, the transmission time of thirty to one hundred mobile devices is predicted as listed in Table 6. The actual and predicted transmission time are also depicted in Fig. 8.

Table 5. The transmission time of Wi-Fi

Number of Devices	Profile Size (KB)	Total Size (KB)	Transmission Time (min.)	Average Time of a Profile (min.)
5	64	1,280	0.51	0.025
	512	10,240	1.18	0.059
	1024	20,480	2.12	0.105
10	64	5,760	1.74	0.019
	512	46,080	8.77	0.097
	1024	92,160	17.07	0.189
15	64	13,440	3.42	0.016
	512	107,520	22.01	0.104
	1024	215,040	40.90	0.194
20	64	24,320	5.87	0.015
	512	194,560	41.07	0.108
	1024	389,120	62.44	0.164

Table 6. The predicted transmission time of Wi-Fi

Number of Devices	Data Size (Average Time)		
	64 KB (0.019 min.)	512 KB (0.092 min.)	1MB (0.163 min.)
30	16.63	80.37	142.36
40	29.82	144.11	255.27
50	46.83	226.33	400.91
60	67.67	327.03	579.27
70	92.33	446.20	790.36
80	120.81	583.85	1034.18
90	153.12	739.97	1310.72
100	189.25	914.57	1619.99

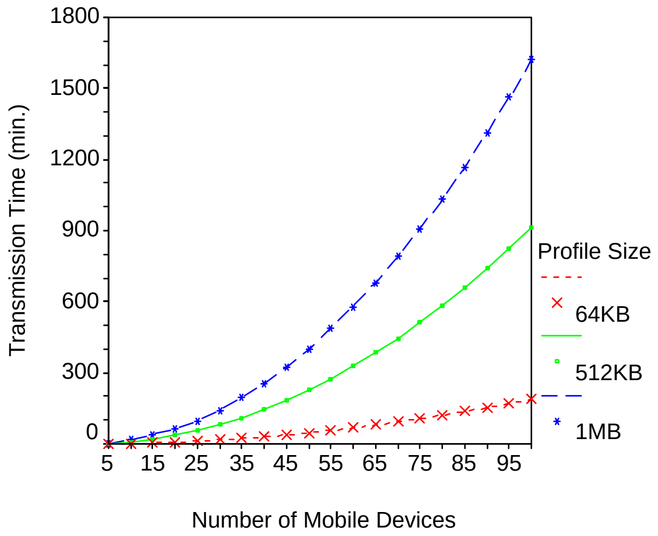


Fig. 8. The actual and predicted transmission time using Wi-Fi

According to the above results, the predicted time of one hundred devices for transmitting 64KB profiles is about 190 minutes. It is still too long for a society activity.

In the above experiments, the transmission time of social profiles is measured to realize the potential problem in the physical environment. According to the above experimental results, some issues are discussed below:

(1) Bluetooth is initially expected as a good communication channel for profile transmission. However, the time may be too long when the data size or the number of mobile devices is not so large. It means that there is scalability problem using Bluetooth. The same problem may be encountered for the previous studies with Bluetooth, such as Serendipity [1] or those systems of smart phones [2].

(2) Although the transmission time with Wi-Fi communication can be shortened, it is still too long when the number of mobile devices is large. Two possible solutions are suggested to solve this problem.

(A) A simple solution is to provide an option for receiving profile, such as “text only” or “file size less than a specific value”. The text-based profile can be used for the initial interaction. Then, they can send interesting multimedia files via peer-to-peer transmission when needed.

(B) A social profile may include multimedia files, such as photo, picture and video. The size of multimedia files causes they are not suitable for exchanging in a social activity. In fact, many Web sites provide free space for storing and sharing multimedia information, such as Blog (Web log) or Vlog (video log). Therefore, multimedia files can be replaced by an URL address. Users can access these multimedia files by themselves when needed. Such a way can reduce the size of a profile dramatically and thus solve the transmission problem of profiles.

Although there is problem on profile transmission, some supplements, such as Blog or Vlog, can be used to reduce the profile size and achieve the purpose of INTRO on supporting social interactions.

V. The Prototypes

In order to demonstrate the feasibility of INTRO, two prototypes of the social layer and preserving layers were implemented using Visual Studio 2005. The screen shots of the prototype for the social layer are shown in the Fig. 9. In Fig. 9(a), a user can click the tags “FL=0”, “1”, *etc*., to define profiles of different familiar levels. A user can also define plaintext, pictures, or video files for every FL. The setup of plaintext, pictures, and videos are shown in Fig. 9(a), (b), and (c), respectively.

The main form of the social layer prototype is shown in Fig. 9(d). Persons in a social activity are listed in the form. There are several functions to handle a person in the list including “send message”, “Adjust FL”, “Trace”, “Delete”, and “Block/Unlock”. Most of the functions are similar to the MSN Messenger expect the “Trace” function which is used to explore the path to the person.

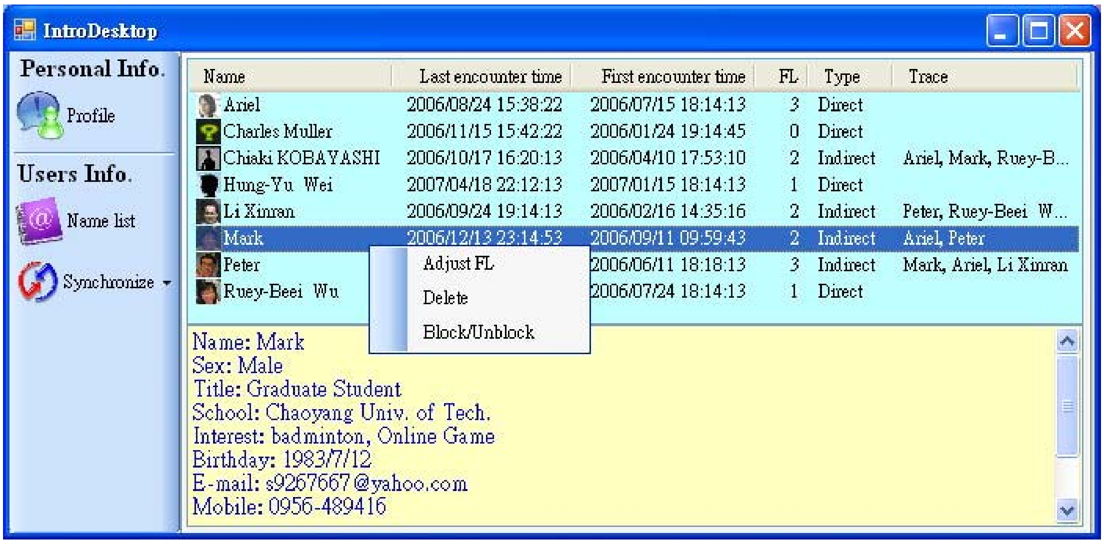
The function of viewing a profile is shown in Fig. 9(e) and (f). In Fig. 9(e), the received profile of a friend with FL three is the merged result of profiles in Fig. 9(a) to avoid the receiver aware the FL set by the friend. In Fig. 9(f), the screen shot is a convenient horizontal mode for viewing photos.

The prototype of preserving layer is shown in Fig. 10. When a user leaves a social activity and comes back to home, he can save the received profiles in the computer. In Fig. 10(a), the profiles in the mobile device are synchronized with that stored in the computer when the user clicks the “Synchronize” button. When the user clicks the “Name list” button, the detailed information, including the first and last encounter time, FL, type (direct or indirect), and trace (the path of the indirect link), is shown on the right hand side. When the user clicks on a specific friend, the corresponding profile is shown on the lower part. In Fig. 10(b), a user can edit his profiles and then synchronize with his mobile device.

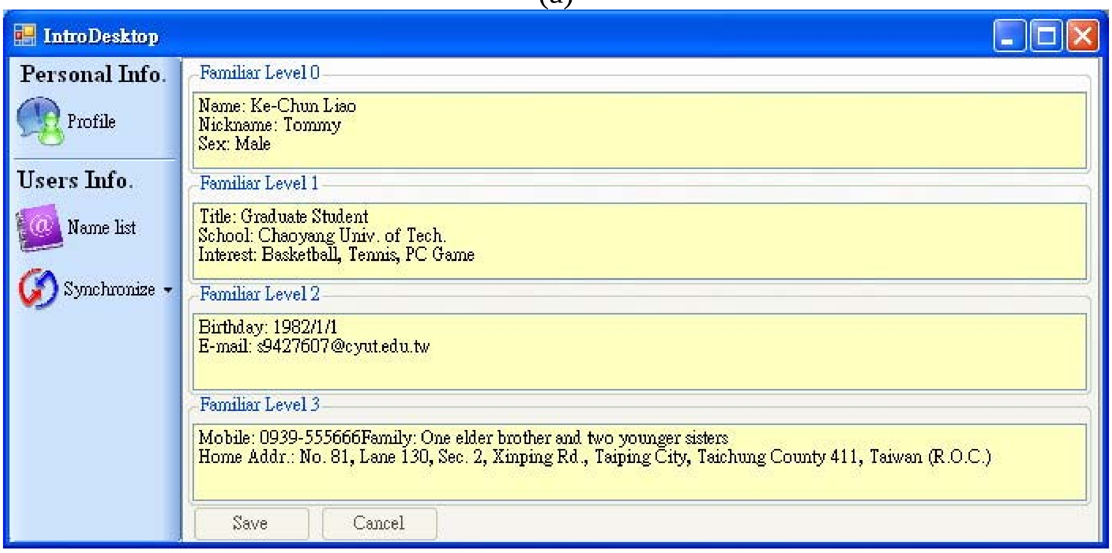
(a) (b) (c)

(d) (e) (f)

Fig. 9. The screen shots of the prototype for social layer (a) the profile setting of text (b) the profile setting of photos (c) the profile setting of videos (d) the list of persons in a social activity (e) the text profile (f) the photo of a specific friend



(a)



(b)

Fig. 10. The screen shots of prototype for preserving layer (a) the friend list (b) the editing of the social profile

According to the above prototypes, some issues about the prototype are discussed below.

(1) For the friend list of the screen shot in Fig. 9(e), the “In” column represents the FL of the friend to the mobile device owner and “Out” column represents the FL of the mobile device owner to the friend. Two FLs shown on the screen may cause a psychological problem when they are not the same. For example, the user “Tommy” may feel bad since he set the highest FL to “Mark”, but “Mark” does not give the same response to him. This problem may disrupt but not support the social interaction. The “In” FL should be hidden to prevent such problem.

(2) The number of near-by persons may be very huge in some social activities, such as a concert. The indirect link inference and profile transmission may exceed the burden of a mobile device. Except to turn off the function of profile transmission, some rules can be designed to overcome such a problem. For example, a profile is exchanged when FL is larger than zero to avoid the transmission with huge number of strangers.

(3) The automatic transmission of profiles may consume power quickly. A manual mode could be used to overcome such problem. For example, when a user is presented to a person, he then transmits his profile to the person manually.

(4) The assigned FL of a friend is not changed permanently. In fact, the FL should be decreased as time. The temporal factor should be used to adjust the assigned FL in order to meet the situation in the life. For example, the FL of a direct link is decreased one every year until to zero.

The above discussions illustrate that the functions of prototypes can be refined to meet user demand on handling social interactions.

VI. Conclusion

Social interactions are important for people. The popular of mobile devices make there exists opportunity for supporting the interactions via these devices. Many research attempts to accomplish such purpose from various ways. However, the support of the interactions in social activities is still an open issue. In this paper, we focus on this issue to propose a three-layered framework – INTRO. In the social layer, a mobile device becomes a useful tool for distributing and collecting social profiles based on the familiar graph (FG). The profile exchange can encourage the face-to-face interactions. The collected profiles can be updated to a computer in the preserving layer. Then, the social relationships can be synchronized with social Web sites to promote the further social interactions. The experimental results also show that Wi-Fi is a better communication channel than Bluetooth for INTRO. INTRO can be a good solution for supporting the social interactions from a physical social activity to the virtual cyberspace.

Acknowledgement

This work was sponsored by the National Science Council with grant number NSC 95-2221-E-324-042.

References

[1] N. Eagle and A. Pentland, “Social Serendipity: Mobilizing Social Software,” IEEE Pervasive Computing, Vol. 4, Issue 2, Jan.-March 2005, pp. 28-34.

[2] R. Beale, “Supporting Social Interaction with Smart Phones”, IEEE Pervasive Computing, Vol. 4, Issue 2, Jan.-March 2005, pp.35-41.

[3] J. Kjeldskov and J. Paay, “Just-for-Us: A Context-Aware Mobile Information System Facilitating Sociality,” Proceeding of the 7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI’05), Salzburg, Austria, Sep. 19-22, 2005, pp. 23-30.

[4] MoSoSo, http://en.wikipedia.org/wiki/MoSoSo

[5] S. Konomi, S. Inoue, T. Kobayashi, M. Tsuchida, and M. Kitsuregawa, “Supporting Colocated Interactions Using RFID and Social Network Displays”, IEEE Pervasive Computing, Vol. 5, Issue 3, July-Sept. 2006, pp. 48-56.
[6] J. Leikas, H. Stromberg, V. Ikonen, R. Suomela, and J. Heinila, “Multi-User Mobile Applications and a Public Display: Novel Ways for Social Interaction”, Proc. of the Fourth Annual IEEE Int’l Conf. on Pervasive Comp. and Comm. (PerCom’06).
[7] G. Kortuem and Z. Segall, “Wearable communities: augmenting social networks with wearable computers”, IEEE Pervasive Computing, Vol. 2, Issue 1, Jan.-Mar. 2003, pp. 71-78.
[8] E.F Churchill and C.A Halverson, “Social Networks and Social Networking”, IEEE Computer Society, Vol. 9, Issue 5, Sept.-Oct. 2005, pp.14-19.

Hsien Chou Liao received the B.S. and Ph.D. degrees in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, Republic of China, in 1991 and 1998. He joined the faculty of Department of Computer Science of Information Engineering of Chaoyang University of Technology, Taiwan, Republic of China, as an assistant professor in 2000. He became an associate professor in 2007. His research interests include computer security, pervasive computing, and software engineering. He is also a senior member of IEEE.

Ke-Chun Liao received the B.S. and M.S. in Computer Science and Information Engineering from the Chaoyang University of Technology, Taiwan, Republic of China, in 2005 and 2007. His current research interests are in the area of social networking and mobile computing.