International Research Journal of Advanced Engineering and Science
ISSN: 2455-9024

Frequent Environment of a Human Being on Community Network Using Robot

Mrs. B. Arulmozhi[1], Ms. K. Indumathi[2], Ms. A. Sivasankari[3]
[1]Assistant Professor Department of Computer Science, DKM College for Women (Autonomous), Vellore, TamilNadu, India
[2]Department of Computer Science, DKM College for Women (Autonomous), Vellore, TamilNadu, India
[3]Head of the department, Department of Computer Science, DKM College for Women (Autonomous), Vellore, TamilNadu, India

Abstract *–In this system that illustrates the social nature of a human being – the need to be always in touch with family and friends – taking into account facilities available on Robot platform. The role of this application is to create a social network in which the users are being alerted when their friends are around. This gives them the possibility to set up a meeting or to avoid one. The users have the possibility to check in some locations and allow their friends to follow their activity. Taking into account the security of the users, we included in the facilities of the application an option which allows close friends or family to check the user's location based on a keyword text message. For this purpose, available Robot location and messages services are used for finding an approximate location of a mobile phone running this program and then sharing it through Social Nature or via SMS. Information is being displayed using default components provided by Robot platform and also more complex elements including heterogeneous lists CWAC, Google Maps and augmented reality using Mixare Library.*

Keywords *– Robot, location and SMS services, augmented reality, social networks*

I. INTRODUCTION

Human need for socialization had been brought to light since ancient times, when Aristotle said: "Man is by nature a social animal" (Aristotle, Politics I), by this he meant to emphasize the fact that human being is destined to live in peers within organized community. As there is no society without communication, so there is no person without social interaction.

Over time, forms of communication and understanding about this process have been expanded based on technology progress. The emergence of the computer and then the Internet has revolutionized the way people communicate, managing to overcome any limitation of time or space dependent, enabling them to exchange information in an efficient way.

However, none of them managed to connect people anywhere and anytime as mobile phone does. The development of smart phones is a consequence of this need. The desire to use the phone not only to call someone or send and receive SMS has been perceived by Google since 2005, reason why it bought the original developer of mobile software Robot Incorporation in August of the same year.

Another significant progress had been accomplished in 2007 when Open Handset Alliance has been founded and joined by 48 companies' hardware, software and telecommunications, willing to help the development of open standards for mobile devices.

According to Wikipedia, Robot architecture consists in a "Linux kernel with middleware, libraries and APIs written in C and application software running on a platform which includes Java-compatible libraries based on Apache", a free type software and open source license, aspect that makes it very attracted among developers.

We will show in this paper how we used this architecture in order to build Social Nature application Social Nature application comes with two new facilities in comparison with existing solutions. First one is related to the fact that users can place their contacts in groups (like *family*, *friends*, *colleagues*, etc.) and they can set different parameters to every group regarding proximity alert so they would be notified when a member is close to them.

The second one combines the first facility with augmented reality and shows on phone camera the direction which must be followed in order to meet somebody and the corresponding distance.

The rest of the paper is structured as follows: Section 2 presents the state of the art in this domain and the most important mobile applications which allow users to stay connected to social networks at any moment; Section 3 details the general architecture of our system and technologies used, while the last Section discusses the conclusions.

II. CONDITION OF THE DEPICTION

In recent years, we can observe that people need to be in touch with their friends and family just by looking at the enthusiasm with which they use applications and social networking websites. This means that with the development of mobile devices also this branch has been expanded, the social programs being ported to the smartphones platform where they have been combined with other innovative technologies such as augmented reality.

In this regard, tracking applications have been developed with which parents can supervise their children in an effective way (Life360) or users can check their friends location via text message (Tehula), in which teenagers interrelate with their friends, meet new people and exchange impressions about the by which people can share their location or the path they go through in a preset period of time (Foursquare, Glympse).

By improving photo camera and introducing GPS system, the augmented reality applications became more accessible among programmers, who starting with using geographic points to create an augmented space and expanded by

Mrs. B. Arulmozhi, Ms. K. Indumathi, and Ms. A. Sivasankari, "Frequent environment of a human being on community network using robot," *International Research Journal of Advanced Engineering and Science*, Volume 1, Issue 4, pp. 41-45, 2016.

developing of 3D object recognition technology (Layer) and barcode scanning products (Junaio).

Life360 – has been designed as a way to keep family in touch, transforming mobile phone in tracking devices. Using GPS technology, the wireless and telephone network, users can share their location or check for the other family members' status. The user has the option to notify his family when he is in a safe place and make a panic call when he is in danger. Being designed to provide safety to people who use it, it also provides location based information about hospitals and police stations located in the area where the user has registered.

Tehula – is an iPhone application through which users can check friends' location via text messages. Their precise location appears through the Google Maps API, allowing the users to pinpoint the position easily.

Google Latitude – is one the main features offered by Google along with Google Maps, 2D/3D maps viewing, consulting street view or real-time traffic in certain areas. Latitude option allows a user to check in a location and share it with his friends.

Gypsii – provides the option to create and save geopoints on Google Maps and label them with photos, videos and messages. It is also possible for users to connect to their Face book and Twitter accounts and share their current position.

LOCiMobile provides two applications for the Robot platform from which customers can choose in order to fulfill their needs. According to their web site, GPS Tracking allows users to find other people, provides access to the Facebook and Twitter social networks, and supports writing messages to one or more friends and uploading photos. On the other hand, Tracking provides benefits to locate and find out real time information about a contact or a group of contacts, using the installed mobile application or the web client.

*Match2Blu***e** – Unlike other applications that are designed to link users with their friends and family, Match2Blue helps customers meet new people with whom they can interrelate. The home view lists the connected users, ordered by distance against the current user. By uploading photos and using interesting posts, they begin to socialize, thus forming new ties of friendship.

Foursquare – is a location-based social networking website for mobile devices. Users "check in" at venues by selecting a place from a list of nearby sights. Location is based on GPS hardware in the mobile device or network location provided by the application. Each check-in awards the user points and sometimes "badges".

Glympse – can be used when a person wants to share their road with other persons for a predetermined period of time. The plus is represented by the fact that users do not need to register in order to share this information and the road can be watched on any web browser by visiting the link received via e-mail.

III. System Architecture

Social Nature application connects its users by sharing a user's position with his friends, colleagues or other family members (see Figure 1). A user has the option to be notified when a contact is near him and helps the others to find out his location even when the application is not open by sending message is received, a background service is checking its content and if it is a match, the GPS is turned on and the location is computed and replayed. In this way, when a user is in danger or his parents are worried or people are waiting for their delayed friend, they can check his location to see if he is safe Information is being displayed taking into consideration the camera's pointing direction, which is computed using the compass, the gyroscope and the motion sensors and a message containing a shared secret string.

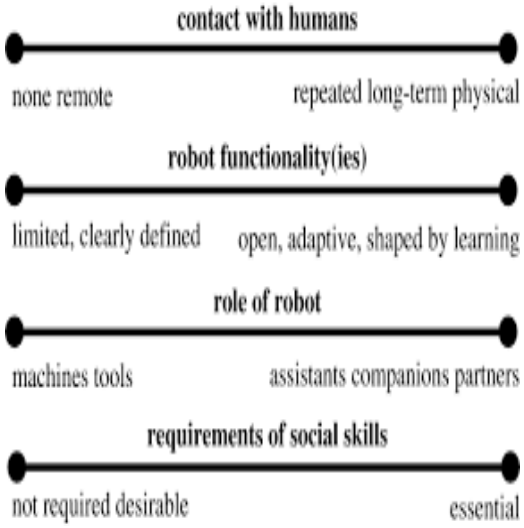

Fig. 1. Social Nature Application.

The location is specified using the latitude and the longitude coordinates and can be viewed on the Google Maps accessing the "check in" to a location, in order for his friends to be aware of places he frequented and the possible activities he conducted. As we stated, a notable feature provided by this program is the possibility to view a friend's locations using augmented reality.

The application presented in this paper consists of three modules corresponding to the three levels of the client-server architecture (3-tier architecture), where elements of interface and data processing are separated. The presentation layer consists of the Robot client by which the information will be displayed in order for users to see and interact with, the data layer is represented by the database storage where the information are hold in and the logic layer, which connects these two levels (performed using Robot services and Php services).

A. Logic layer – Technologies used

Face book API

Facebook social network offers for Robot platform a library through which mobile developers are able to develop an application which let users connect to their Facebook account and posts the desired information. Beside the regular registration made using the email address, the Social Nature users are able to connect also by using their Facebook account. To achieve this, the Social Nature application was

Mrs. B. Arulmozhi, Ms. K. Indumathi, and Ms. A. Sivasankari, "Frequent environment of a human being on community network using robot," *International Research Journal of Advanced Engineering and Science*, Volume 1, Issue 4, pp. 41-45, 2016.

International Research Journal of Advanced Engineering and Science
ISSN: 2455-9024

registered in the Section Mobile Application from the Facebook App Dashboard.

Robot Location Services

Location based information is becoming increasingly important in the world of mobile application development. Robot offers this functionality, using network location providers, which determine the user location using GPS, cell tower and Wi-Fi signals.

To get access to this information, an Robot developer may use the classes from *robot.location* package, which includes members such as: *LocationManager*, *LocationProvider*, *Criteria* and *LocationListener*. The way that location classes communicate can be seen in Figure 2. As far as we can see there are a lot of social tracking applications, which meet some of our features, but none of them notify their users when their friends are nearby or allow them to view their friends (and the distance to them) using augmented reality facilities.

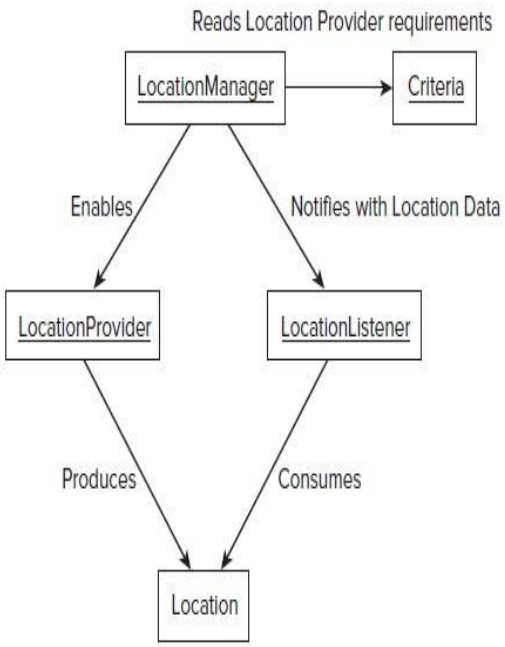

Fig. 2. Robot location components.

Google Maps

To display the items found by Location Manager tool, one can use the Google Maps external library. For this, project has to be built as a Google APIs project, a standard Robot version not being able to reference the required classes. In the Social Nature project, this feature has been used to display user's friends and to show the shortest route to their location.

To obtain the set of coordinates corresponding to the road path between two points on the globe, a request to Google Maps API Service has to be made. The resulted *.kml* file is then parsed and the extracted points are transformed into GeoPoint objects, which are displayed afterwards on the surface of the map (see Figure 3).

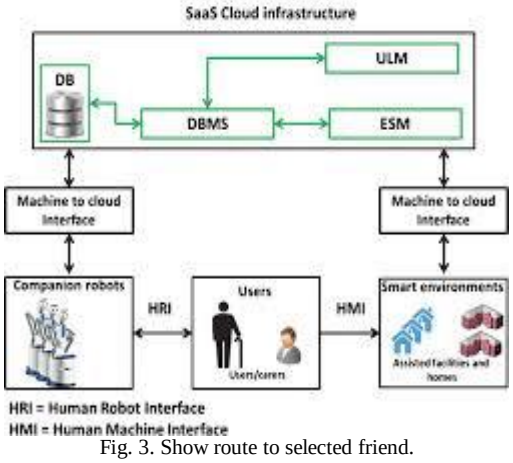

Fig. 3. Show route to selected friend.

Mixare API

Mixare is an open-source library for smart phones, which combines the image captured by the camera lens with the device location. When the phone is being moved, the location is computed and sent to the web services specified in the configuration file. Nearby points of interest are serialized as a JSON object and returned back to application. Information is being displayed taking into consideration the camera's pointing direction, which is computed using the compass, the gyroscope and the motion sensors.

In this application, Mixare library is used to display friends' position on mobile screen. This was achieved by creating an Robot Service which uses a RESTful web service in order to interrogate the database and retrieve user friends' location. Every 30 seconds a request is made using the user identifier and the received result is parsed and mapped to the device screen as displayed in Figure 3.

Short Message Service

One of the useful features available to Robot developers is the possibility of handling the receiving and the sending of text messages. As mentioned before, the application is focused also on user's security. When the application is closed and the user is not answering to his phone, close friends or family can still have access to his location by sending the shared keyword.

When a message is received, a background service is checking its content and if it is a match, the GPS is turned on and the location is computed and replayed. In this way, when a user is in danger or his parents are worried or people are waiting for their delayed friend, they can check his location to see if he is safe.

The Broadcast Message Intent includes information about the received text messages (the message content, the recipient's phone number, the time at which delivery took place), saved as an array of objects in PDU format. To extract the message information, it is necessary to transform the PDU byte array components into Sms Message objects, as illustrated in the following sequence of code.

Tweened animation provides depth, motion and feedback to user's actions at minimal cost resources. Using animation to apply a set of changes in the resources orientation, scaling or

Mrs. B. Arulmozhi, Ms. K. Indumathi, and Ms. A. Sivasankari, "Frequent environment of a human being on community network using robot," *International Research Journal of Advanced Engineering and Science*, Volume 1, Issue 4, pp. 41-45, 2016.

positioning is less expensive than manually redrawing workspace in order to get similar effects.

```
Bundle bundle = intent.getExtras(); SmsMessage[] msgs = null;
if (bundle != null) {
Object[] pdus = (Object[]) bundle.get("pdus");
msgs = new SmsMessage[pdus.length];
for (int i = 0; i < msgs.length; i++){
msgs[i]=SmsMessage.createFromPdu((byte[])pdus[i]);

}
}
```

This layer offers the main facilities of the social nature application. A user can use them via interface layer which is presented below.

B. The Interface Layer

The application interface is all that a user can see and can interact with. All user interface elements are built using View and ViewGroup objects from Robot platform. A View is an element that draws something on the screen, while a ViewGroup is an object that holds other View objects in order to define the layout of the interface, as it can be seen in Figure 5. Robot offers custom views such as layouts objects or input controls for defining the visual structure of a user interface.

Animations

Graphical interfaces for mobile applications have developed on a fast pace, allowing the use of advanced graphics features such as shadows, translucency effects, animations or even OpenGL library.

In the Social Nature application, animations are used for translating views when a button is pressed. More concrete, when the option button is touched, the LinearLayout object, which encapsulates the menu items, will become visible and will translate to the right side of the screen. A second tap will cause the menu translate to the left and become invisible again. To achieve this, we have used the documentation provided on an Robot blog.

Heterogeneous Lists

The ListView items provided by Robot are suited to expose data that have the same format. However, most of the time it is required a more complex structure in order to expose some information. In Social Nature program, these lists are used to group friends, merging in a CWAC list labels with contacts lists.

This layer allows the user to perform the main actions in the system and to use the facilities defined in the logical layer. The results of the interaction between user and this layer are saved in the storage layer.

C. Storage layer

Social Nature application user information is stored in a MySQL database on the server. Almost all interactions with interface layer have as result a communication with storage layer, where information is saved. The Robot Client is displaying the data by interrogating the database using RESTful Php services. The communication between the web services and the Robot Client is performed using JSON Object format messages.

D. Social Nature Operations

A social nature user can perform the following main actions in the system:

a) Register – allows somebody to create a Social Nature account in order to use this application. The registration can be performed in two different ways: first, via the *Get a new Social Nature ID!* option or second, via Facebook login. Using the first one, a user must fill in a form with his personal details while the other one will extract the necessary information from his Facebook account (using the Facebook API). After performing this step, the account will be created and the user will have access to the Social Nature server, where all his information (locations, friends, groups, preferences, etc.) will be managed.

b) Login/logout – when the user is logged in the application, he can use the Social Nature application, otherwise he cannot be informed about his friend's position and vice versa his friends cannot be notified about his current location, except they use the shared keyword.

c) Locations Management (Create/modify /delete) – defines the main operations for managing locations. In order for a user to set his current position, he must first create it (if the location has not been used so far) using the GPS facilities of his phone, when he is located in a point of interest, or by manually inserting the desired location's coordinates.

d) Friends Management (Create/modify/delete) – defines the main operations for managing friends. All users must possess an Robot phone that has the Social Nature application installed. When new friend requests are received, a background service changes the view and notifies the user about those requests. If the requests are accepted, the users will start sharing their location.

e) Groups Management (Create/modify/delete) – defines the main operations for managing groups. At group level, users can add/delete friends and they can customize group parameters (like distances under which a user will be/will not be notified about the presence of a current group member). For example, a user can set:

MaxDistanceForFamily = 500 m, MaxDistanceForFriends = 200 m, MaxDistanceForColleagues = 50 m. Similar, users can specify an area for *home* (where will not be notified when someone from Family is close to them) or can define an area for *office* (where will not be notified when someone from Colleagues is close to them).

f) Settings – manages the shared keyword which will be used by friends to have access to the current user location, when he is not online. The keyword can be changed only by user using his chosen password.

g) Search – the basic search is looking for users in friends' friends while the user is typing the name. The advanced search is looking for a match of the typed name in the entire database.

h) View – is composed from a principal view which changes dynamically, a menu bar and a sliding menu. The menu bar contains 4 buttons: one for menu, one for friends'

Mrs. B. Arulmozhi, Ms. K. Indumathi, and Ms. A. Sivasankari, "Frequent environment of a human being on community network using robot," *International Research Journal of Advanced Engineering and Science*, Volume 1, Issue 4, pp. 41-45, 2016.

requests, one for friends' locations feeds and one for locations management. When the menu button is pressed, the menu windows slides to right and after choosing an option, the menu windows slides to left (becoming invisible) and the principal view changes accordingly to the selection. In the upper side of the menu window, the user can find the search box, which will open by touching it the search window.

i) Check-in – a user may want to add some information to his current location, like the activities he is conducted or the friends he is with. For this to happen, he can check in to a defined location and add a short message which will appear to his profile and also on his friends' feeds window.

E. Technical Details

In present the application was tested using a server with following architecture: Pentium (R) Dual Core Processor, 3 GB Ram, 320 GB Hdd, with Windows 7 SP 1, and MySql 5. For client component we used a phone with following architecture: Samsung I9001 Galaxy S Plus model, 1.4 GHz Scorpion processor, 512 MB RAM, 8GB storage, Robot OS, v2.3 (Gingerbread) with a service that allowed data transfer.

We tested application with 10 users and we didn't remark notable problems until now. In the future we intend to test our application with more than 100 active users in Social Nature application in order to see if the server can handle all the necessary traffic.

IV. CONCLUSION

Social Nature application consists of three modules corresponding to the three levels of the client-server architecture (3-tier architecture), where elements of interface and data processing are separated. The presentation layer consists of the Robot client by which the information are displayed in order for users to see and interact with, the data layer is represented by the database storage where the information are hold in and the logic layer, which connects these two levels (performed using Robot services and Php services).

The role of this system is to create a social network that allows sharing user location-based information and activities they carry out. For this purpose, available Robot location services like GPS technology, wireless and mobile towers are used to find an approximate location of a mobile phone running this program. Information display has been achieved by using both default components provided by Robot platform and more complex elements including heterogeneous lists CWAC, Google Maps and augmented reality using Mixare Library.

The novelty of our approach is related to the fact that a user can be alerted, accordingly to his group preferences, when his friends are in his vicinity. On one hand, it combines features found in other applications like "check in" from Foursquare or Google Latitude, request location via SMS from Tehula, and on the other hand, it adds new options like viewing all friends' location using augmented reality and drawing the shortest road to the selected friend using Google Maps service.

REFERENCES

[1] C. Phua, V. Lee, K. Smith, and R. Gayler, "Comprehensive survey of data mining-based fraud detection research," *Artificial Intelligence Review*, 2005.

[2] C. Elkan, "Magical thinking in data mining: lessons from CoIL challenge 2000," *KDD '01 Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426-431, 2001.

[3] T. Fawcett, "In vivo spam filtering: A challenge problem for KDD," *SIGKDD Explorations*, vol. 5, issue 2, pp. 140-148, 2003.

[4] N. Lavrac, H. Motoda, T. Fawcett, R. Holte, P. Langley, and P. Adriaans, "Introduction: Lessons learned from data mining applications and collaborative problem solving," *Machine Learning*, vol. 57, issue 1-2, pp. 13-34.

[5] "Turkish Health Care Syndicate 2008 Health Care Report" 2008. <http://www.turksagliksen.org.tr/content/view/6271/55/>

[6] "About Basel Committee" 2003, <http://www.bis.org/bcbs/>

[7] J. Lin, M. Hwang, and J. Becker, "A fuzzy neural network for assessing the risk of fraudulent financial reporting," *Managerial Auditing Journal*, vol. 18, issue 8, pp. 657-665, 2003.

[8] T. Bell and J. Carcello, "A decision aid for assessing the likelihood of fraudulent financial reporting," *Auditing: A Journal of Practice and Theory*, vol. 10, issue 1, pp. 271-309, 2000.

[9] K. Fanning, K. Cogger, and R. Srivastava, "Detection of management fraud: A neural network approach," *Journal of Intelligent Systems in Accounting, Finance and Management*, vol. 4, pp. 113-126, 1995.

[10] S. Summers, and J. Sweeney, "Fraudulently misstated financial statements and insider trading: An empirical analysis," *The Accounting Review*, pp. 131-146, 1998.

[11] M. Beneish, "Detecting GAAP violation: implications for assessing earnings management among firms with extreme financial performance," *Journal of Accounting and Public Policy*, vol. 16, pp. 271-309, 1997.

[12] B. Green and J. Choi, "Assessing the risk of management fraud through neural network technology," *Auditing*, vol. 16, issue 1, pp. 14-28, 1997.

[13] P. J. Bentley, "Evolutionary, my dear Watson: Investigating committee-based evolution of fuzzy rules for the detection of suspicious insurance claims," *GECCO'00 Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pp. 702-709, 2000.

[14] C. Von Altrock, *Fuzzy logic and Neurofuzzy Applications in Business and Finance*, Prentice Hall, pp. 286-294, 1997.

[15] B. Little, W. Johnston, A. Lovell, R. Rejesus, and S. Steed, "Collusion in the US crop insurance program applied data mining," *Proceeding of SIGKDD02*, pp. 594-598, 2002.

[16] C. Phua, D. Alahakoon, and V. Lee, "Minority report in fraud detection: classification of skewed Dat"a, SIGKDD Explorations, 6(1) (2004): 50-59.

[17] S. Viaene, R. Derrig, & G. Dedene, "A case study of applying boosting naive bayes to claim fraud diagnosis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, issue 5, pp. 612- 620, 2004.

[18] P. Brockett, R. Derrig, L. Golden, A. Levine, and M. Alpert, "Fraud classification using principal component analysis of RIDITs," *Journal of Risk and Insurance*, vol. 69, issue 3, pp. 341-371, 2002.

[19] B. Stefano and F. Gisella, "Insurance fraud evaluation: A fuzzy expert system," *Proceedings of IEEE International Fuzzy Systems Conference*, pp. 1491-1494, 2001.

[20] E. Belhadji, G. Dionne, and F. Tarkhani, "A model for the detection of insurance fraud," *The Geneva Papers on Risk and Insurance*, vol. 25, issue 4, pp. 517-538, 2000.

[21] M. Artis, M. Ayuso and M. Guillen, "Modelling different types of automobile insurance fraud behavior in the Spanish market," *Insurance: Mathematics and Economics*, vol. 24, issue 1-2, pp. 67-81, 1999.