Computer Science • 13 (4) 2012 http://dx.doi.org/10.7494/csci.2012.13.4.87

Michał Wrzeszcz
Jacek Kitowski

MOBILE SOCIAL NETWORKS FOR LIVE MEETINGS

Abstract

In this article, we present an idea of combining social networking websites and modern mobile devices abilities to transfer social networking activity to a higher level. Nowadays, these devices and websites are used to offer ability of remote communication (phone calls, message exchange etc.), which potentially can be used to notify people about meetings in the real world. Since the current social network models do not provide enough information for such notification (social networking websites are examples of social networks) a new social network model that will be suitable for the above mentioned application is proposed and a new social platform that base on mobile devices is introduced. This platform can notify users when their friends are nearby. The paper presents the model and the simulation that verifies the approach.

Keywords social networks, social networking websites, mobile devices

1. Introduction

Nowadays, social networking websites are very popular (e.g. Facebook had 845 million monthly active users at the end of December 2011 [1]). People use them not only to exchange news and opinions, but also they become tools used to organize various social events. They were even used to organize revolutions [11]. Additionally, the growing popularity of smartphones that displace traditional mobile phones should be noticed [10]. The problem under study is whether the social networking websites take full advantage of modern mobile devices widespread currently.

The mobile devices use sensor data to find the location of the user. This information can be used by applications integrated with social networking websites to report the activity of the user. It allows friends of the user to check his daily activity and location. In this article, we would like to prove that social networking websites combined with mobile devices can offer much more than the traditional approach using, for example, a personal computer.

The rest of the paper is organized as follows. In Section 2, we describe new possibilities of mobile communication. Then, section 3 discusses the kinds of information that social network should provide to allow implementation of the platform that gives users these new possibilities of communication. In section 4, we show extended models of a social network to justify the need of a new model creation. Next, in section 5, the proposed model is mathematically described. Section 6 includes description of implementation of this model. In section 7, we show a simulation of the platform created on the basis of the proposed model. Section 8 contains a description of performance tests of our model. We conclude this paper in section 9 and describe future work directions of our research.

2. New possibilities of communication — mobile phone social platform

Although, we are able to contact with our friends anytime using mobile phones, remote or virtual communication cannot replace communication with other people in the real world. Combining the abilities of social networking websites and mobile devices, we can help people to meet in the real world. Figure 1 shows an evolution of the method of use of mobile devices and social networking websites. The left side of the picture presents the current situation — mobile phone sends the information about activity and location of the user when he pushes the appropriate button in a dedicated application. The right side of the picture presents a new level of social activity proposed by us — an application associated with the platform runs at the mobile device in the background all the time, identifies other users around and notifies the user about their presence. The mobile phone social platform proposed by us implements this new method. It will include an application for mobile devices and a server that will store information about users, detect potential meetings and notify the applications installed at users' mobile phones. This social platform may be considered as a social

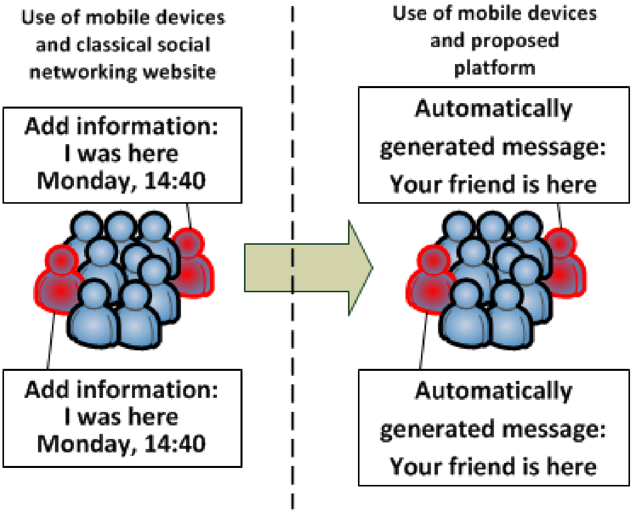


Figure 1. New role of mobile devices in access to social networking websites.

networking website. However, the users of a typical social networking website usually access it via the web browser and rarely use mobile devices to access it while the users of the proposed platform will always use mobile devices to access it. Moreover, their activity will be usually limited to launching the application in the background (additionally, they will rarely configure it, for example, they will sometimes change the lists of their friends).

2.1. Capabilities of the platform

Everyone, who has a device with GPS and access to the Internet, may become a user of the platform. In order to join this social platform the users install the appropriate application at their mobile phones. In this case, contacts from the user's phone book are the basis for creating a list of his friends. The user chooses which telephone numbers should be monitored for him. If a friend of the user who also uses the platform is detected close to the user, notification is sent. Each user may choose the size of the radius of action which will be the basis for notification generation.

Each user can choose if he wants to be 'visible' by other users so only devices which owners use the platform and are 'visible' may be detected by others. Moreover, each user may use white or black lists. If someone is using a white list, only devices of people from this list are able to detect him (even if someone marked his as a friend, he will not be notified when he is not listed). If a user is using a black list, all devices that are not listed are able to find him. Of course, probably only the devices of his friends will be interested in finding him.

The system will detect friends that are near on the basis of the data from GPS sensors. This data will be used to calculate distances between the user and all people that are marked as his friends (and who are 'visible'). Afterwards these distances will be used to check if there is someone about whom the user should be notified and notifications will be sent if needed.

2.2. Types of users

The platform will be mainly used by mobile device users to automatically identify their friends around and to inform them about potential meetings. Additionally, it may be used by companies to promote their services. These companies should register their position in the system (their positions will not be monitored dynamically). If the user wants to find some service (e.g. restaurant or cinema), he can choose the type of service and the platform will show him the nearest companies that fulfil his demands. However, there are many services (e.g. car navigation systems) that offer this functionality so we will not consider it in this article. We will focus on friends finding.

2.3. Comparison with existing solutions

The data from GPS sensors is widely used to track the location of a man, e.g., on the basis of this data Google Earth [2] can provide interesting statistics about user's activity. Tracking of people is also possible without GPS sensors, e.g., INDECT project [3] developers use mobile tracking sensors that obtain an object's position by analysing information which master stations' area it is entering and leaving [4]. It shows that our platform can use well-known solutions to tract the users.

A major challenge is a detection of interactions between all users in real time. Social discovery on location is a hot topic right now. At this early market, many interesting products that use social networks and mobile devices have been proposed. Applications like Banjo, Foursquare or Gowalla give people the ability to search for other people nearby. They use users check-ins, geolocated Tweets or Twitpic photos to identify user location [6]. Contrary to the solutions mentioned above, in our solution, the user does not have to check-in or use any social networking website to be found. It is more convenient. Moreover, when we are using check-ins or Tweets, some of them have been made a few hours earlier so people who have produced them may be far away. Another difference is that most of the existing applications focus on meeting new people while we want to help people to meet with friends they already know. This difference has a serious consequence. The data used by the proposed platform should be up-to-date to find an ability of accidental meeting while the data used by the described platforms do not have to be 'fresh' — a new relationship proposed by the described applications may start from a contact in virtual world when the proposed partner has left a place where the user is.

Aforementioned differences result in differences in the dynamics of the changes of the social network. In our solution the data from GPS sensors will be sent to the server more frequently because many people would like to meet their friends more frequently (so constant monitoring of the possibility of an accidental meeting is desired by them) while most of the people usually try to meet new people only from time to time. Furthermore, the need to use up-to-date data results in an inability to use check-ins or Tweets instead of the data from GPS sensors (users change their positions more frequently than they check-in or use social networking websites) so the data from GPS sensors must be sent more frequently. Therefore, the business model of our solution is different than the business model of the described applications. Our application could be used as a service integrated with some mobile phone network. It would allow us to transfer the data from GPS sensors to the server for free because the operators of the mobile phone network could profit from the increase in the number of clients.

3. Information provided by the social network

Users of social networking websites and relations between them form a social network. Concerning the proposed platform, relations in the social network represent friendship. These relations will be used to choose about whose presence the system should notify the user. However, in our case the social network can provide also the second type of information — it allows checking if two people are close enough to communicate in the real world. Unfortunately, standard social network models do not allow checking for the ability of communication between nodes. This fact encouraged us to study extended social network models.

4. State-of-the-Art Extended Models of social networks

Examples of extended models are Multi-layered Social Networks and Multidimensional Social Networks. A Multi-layered Social Network is defined as a tuple $\langle V, E, L \rangle$ where: V is a non-empty set of nodes, E is a set of edges where each edge belongs to exactly one layer and L is a set of layers [5]. Each layer corresponds to one type of relationship between users [7]. This model offers new abilities — the additional layers can be used to represent some additional information, e.g., frequency of meetings. We can also consider using a selected layer to represent the ability of communication between nodes (i.e. between people) in the real world. However, the users want to communicate only with their friends so this layer would be dependent on the layer that represents friendship. The network should allow for an analysis of each layer separately. It is not possible in this case. Moreover, each layer should represent some type of relation between people — we reckon that the ability to communicate between people should not be equated with any type of relation between them. Although, Multi-layered Social Networks gives us the ability to represent all needed information, in our case this representation will be difficult in use (relations and ability of communication should be separated more clearly).

Multidimensional Social Networks are more complicated than Multi-layered Social Networks and allow for observation of changes of the network. They contain three distinct dimensions: the layer dimension that describes all relationships between the users of a system, the time-window dimension that allows for temporal analysis and the group dimension which describes subsets of users [8]. From our point of view, the time-window dimension is most interesting (see Figure 2).

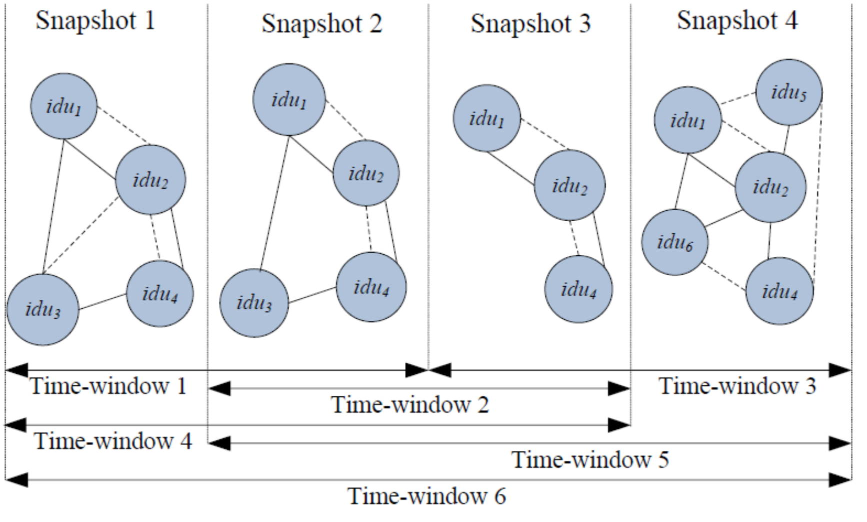


Figure 2. Time windows [9].

A time-window limits the social network to those nodes and relationships that have existed in a period defined by time-window size. We could use time-windows to limit the social network to relations between people that can communicate at the moment. However, it binds relations with the ability of communication. If we like to model a situation when the particular nodes cannot communicate in a particular time-window, we have to delete a relation between these nodes. Afterwards, when the possibility of communication is restored, we have to recreate the relation (in the next time-window). This type of modelling causes a problem. We are not able to change the relation between nodes when these nodes cannot communicate. In real life we can change our opinion about someone in the situation when we are not able to contact him, e.g., we can change our opinion about someone when we receive information about him. This information may encourage us to delete him from friends' list, which is not possible when we use time-windows.

Our studies have shown that despite the fact that extended models of social networks already exist, the extended model that is suitable for our case should be developed separately.

5. A proposed social network model

A new social network model allows representing of the ability of communication between nodes and separates this ability from relations between them. It is based on Multi-layered Social Networks but adds a new abstraction layer — communication channels. This model is defined as a tuple $\langle V, E, L, C, T \rangle$ where:

- V — a non-empty set of nodes;
- E - a set of relations; each relation is a tuple $\langle x, y, l \rangle$, where x,y are different nodes and l is a layer $(x, y \in V, l \in L, x \neq y)$;
- L — a set of layers; a layer is a set of relations of the same type (e.g. friendship layer, family ties layer); maximum two relations between particular nodes (x to y and y to x) belong to each layer (up to $|V| \cdot (|V| - 1)$ relations belong to each layer): $\forall l \in L : (\langle x, y, l \rangle \in E \wedge \langle x', y', l \rangle \in E) \Rightarrow x \neq x' \vee y \neq y'$;
- C — a set of communication channels; each communication channel is a tuple $\langle x, y, t, n \rangle$, where x,y are different nodes, t is a communication channel type and n is a number of the channel $(x, y \in V, t \in T, n \in N, x \neq y)$;
- T — a set of types of communication channels (e.g. face to face communication channel type, telephone communication channel type, e-mail communication channel type); communication channels are directed (e.g. x can know the telephone number of y, while y does not know the telephone number of x); two nodes can be connected by more than one channel of each type, e.g., one person can know two different telephone numbers to a friend: $\forall t \in T : (\langle x, y, t, n \rangle \in C \wedge \langle x', y', t, n' \rangle \in C) \Rightarrow x \neq x' \vee y \neq y' \vee n \neq n'$.

The most important novelty is the set of communication channels. They represent the ability to exchange information between nodes. Furthermore, the model allows us to represent many types of communication channels, as there may exist many ways of communication between nodes.

Another advantage of the proposed approach is that the set of communication channels may be modelled as a multi-graph, so the proposed model of social network can be represented by a multi-graph also. It is a very important feature because it allows us to create an implementation of the model that will be easy in use by programmers for the creation of the considered platform.

6. Implementation of new social network model

The Java language and the JUNG framework were used to create the implementation. The classes that implement the social network model are shown in Figure 3. Package ‘socialNetwork’ contains a representation of nodes and edges of Multi-layered Social Network. Subclasses of class ‘Person’ are nodes of the network while instances of ‘Relation’ class are edges. Each instance of ‘Relation’ belongs to a layer described by the instance of ‘Identifier’ class. Package ‘general’ includes classes that may be used to create attributes for nodes and edges.

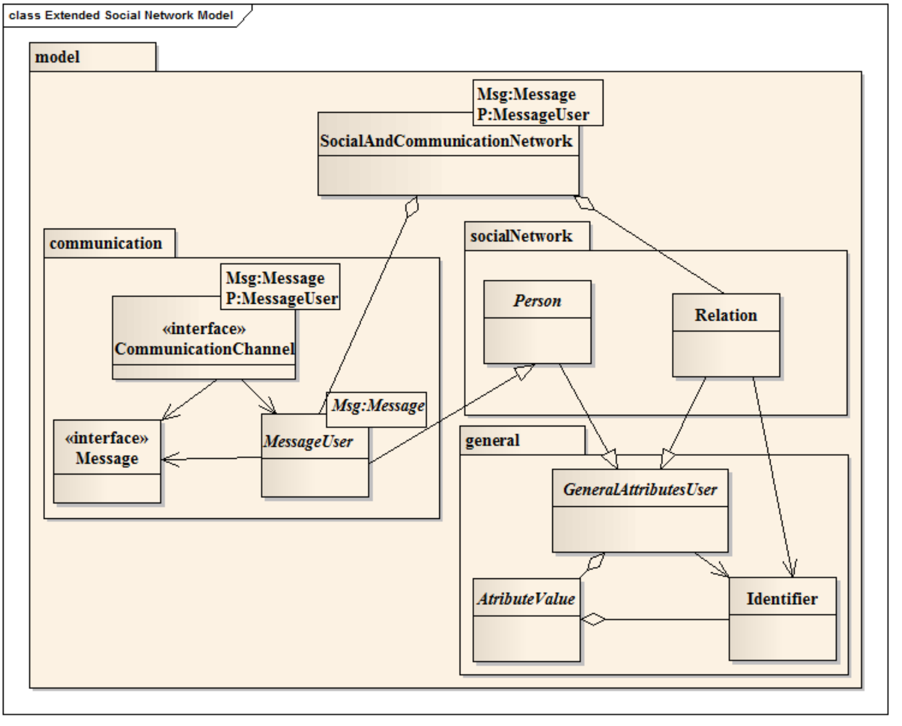


Figure 3. Implementation of an extended social network model.

Package 'communication' implements the proposed new abstraction layer. The user may create any number of 'CommunicationChannel' subclasses. However, each class must give the ability to identify the type of channel and mechanism for propagating messages. The extended social network model is represented by a 'SocialAndCommunicationNetwork' class which uses multi-graph to store information about people, relations between them and communication channels.

7. Simulation of social networking website

To check if the proposed social network model can be used to create described platform we simulated the platform using the agent methodology. The simulation focused on detection of possibilities of meetings. The friendship relations were created randomly. We iterated over all pairs of nodes and used the following expression to decide if the relation was created: $rand() \cdot (dist \div maxDist)^2 < densityRatio$, where:

- rand() — random number generator,
- dist — distance between nodes,

- maxDist — the radius of action (maximal distance which allows nodes to communicate),
- densityRatio — parameter that influences the number of created relations (more relations are expected to be created for bigger densityRatios).

A relation was created when the value of the expression was true. We ran simulations for various density ratios (2–256). During each simulation the agents moved randomly. The automaton that managed communication channels was the most important element of the simulation. It notified agents when new communication channels were created (notified about the possibility of meetings with friends) so it provided the most important functionality of the platform.

7.1. Simulation framework

The framework that enables simulations was created on the basis of MASON framework (see Figure 4). An 'AbstractCrowd' class is a template of the simulation. This class allows for injecting automatons that manage simulation, e.g., that automaton that creates and deletes communication channels dynamically.

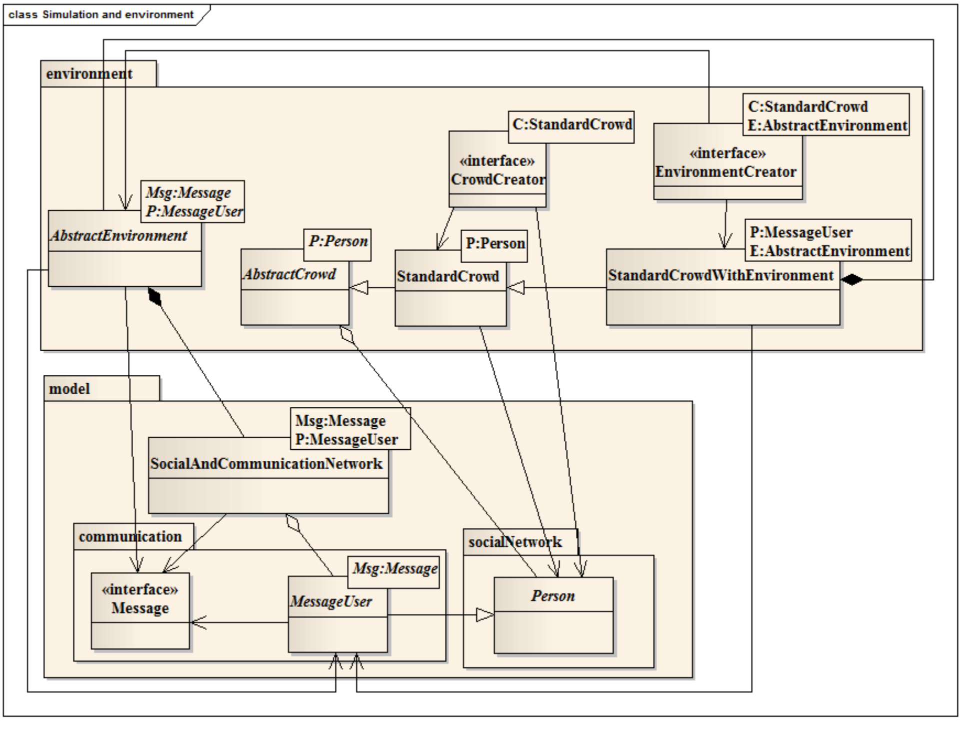


Figure 4. Implementation of simulation and environment templates.

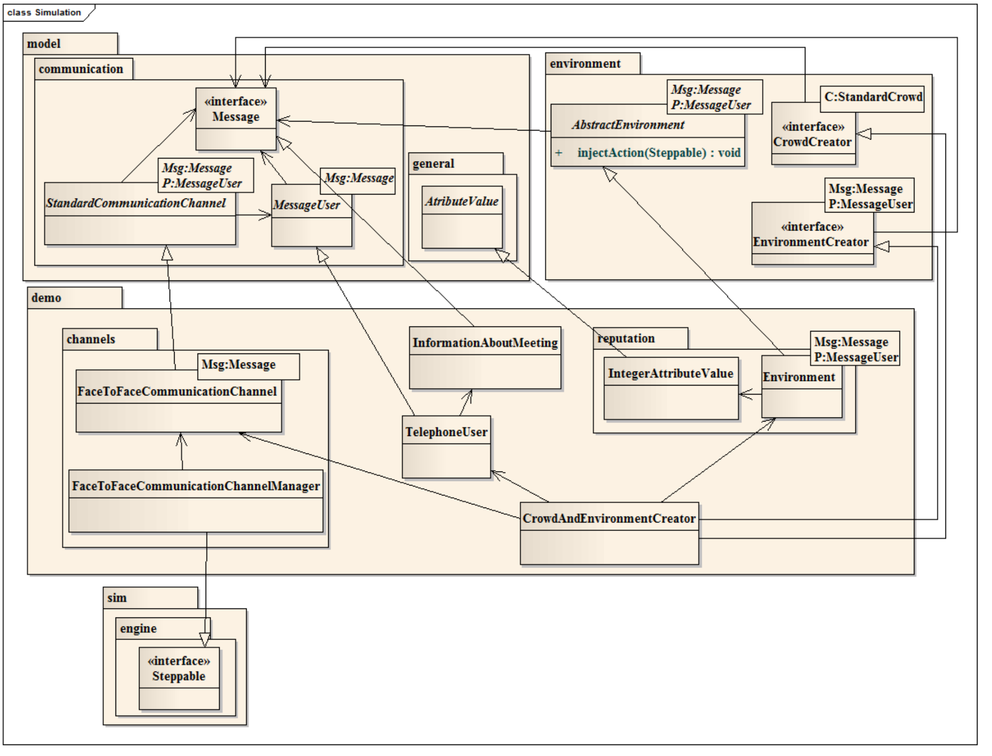


Figure 5. Implementation of simulation.

This mechanism may be used to discover potential meetings. The 'AbstractEnvironment' class contains an extended social network model implementation—it describes relations that connect the nodes of the network (people). The 'StandardCrowd' class adds to the 'AbstractCrowd' an easy way of simulation creation (using crowd creators) while 'StandardCrowdWithEnvironment' binds the simulation with the Environment.

7.2. Simulation implementation

An implementation of the simulation is shown in Figure 5. Simulation is created using a 'CrowdAndEnvironmentCreator'. It initializes agents (instances of 'TelephonUser' class) and relations. Relations are described in the 'Environment' class. The relation between particular nodes exists when the source node marks the destination node as a friend. Additionally, each relation has an integer attribute which counts the number of meetings. To detect potential meetings we created face to face communication channels and the manager that creates and deletes them. When a channel is created, the nodes are notified using 'InformationAboutMeeting' class instance.

8. Performance tests

The tests have shown that the potential meetings were detected correctly regardless of the density ratio and the number of agents. During the performance tests we studied how size of the social network affects the time of the graph operations.

8.1. Tests results

At the beginning we measured nodes adding time. Although, the time of adding 1000 nodes was very low in most cases even for a large graph, we noticed some cases where this time it was high (see Figure 6a). This is probably caused by the fact that at the beginning more space than needed is reserved for nodes (by the graph framework) so adding further nodes is very fast. When the graph is growing and the space becomes insufficient, it is enlarged during the adding operation which takes additional time. To eliminate this phenomenon we measured the time of adding of larger groups of nodes. The results of this test showed that time of nodes adding slowly grows with the number of nodes growth (see Figure 6b).

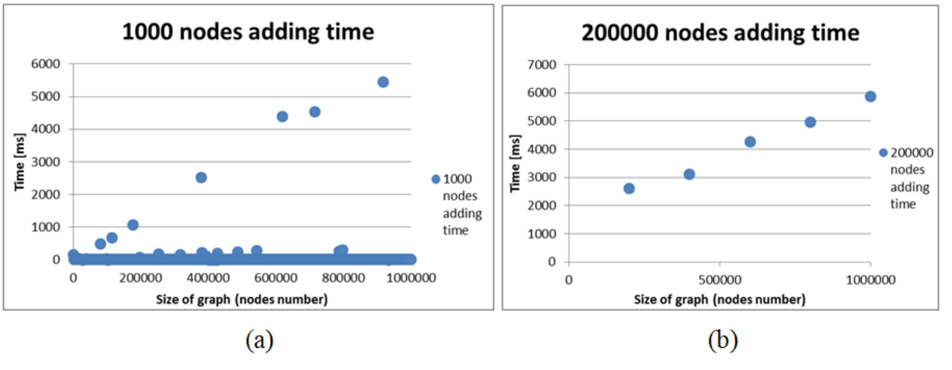
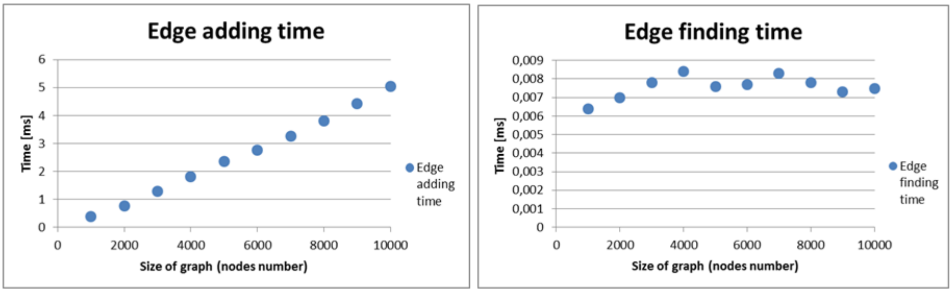


Figure 6. Times of nodes adding.

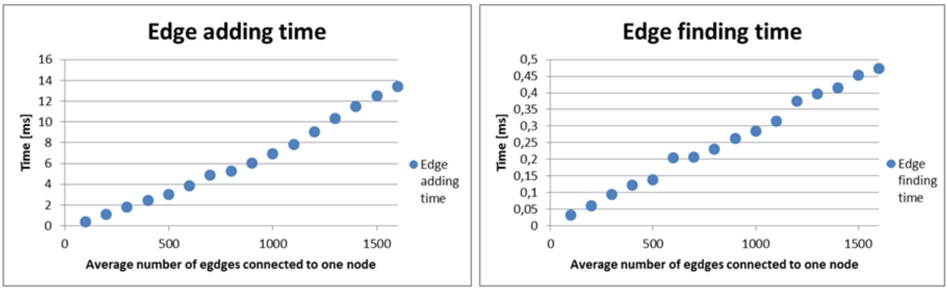
Afterwards we measured the edge adding and finding time. During the first test we added 20 random outgoing edges to each node. Execution time of adding of a single edge is shown in Figure 7a. It grows with the nodes number growth. When all edges were added, we measured average edge finding time (we provided two nodes as the arguments of the function and received object that represents the edge). We did not detect any corelation of this time with the nodes number (see Figure 7b).

We also verified the correlation between these times and average number of outgoing edges. During this test, the number of nodes was constant, but we added 100 outgoing edges to each node before each measurement. We observed that both adding and finding times grow with the number of edge growth (see Figure 8).



(a) (b)

Figure 7. Times of nodes adding and finding according to nodes number.



(a) (b)

Figure 8. Times of nodes adding and finding according to edges number.

8.2. Analysis of results

To interpret the results correctly we should use graph theory and analyse the implementation of the graph framework used by us. We plan to do it in the near future (see chapter 9). However, we can already formulate some conclusions taking into account the application of the model.

After the first period, when the number of users may grow very fast, the nodes will be added rarely so growth of adding time is not a problem. Moreover, we may try to create an algorithm which increases space for nodes when system load is low and the amount of space is small. It should allow us to make node adding time small regardless of the graph's size.

Since people have a limited number of friends usually, the number of edges connected to each node is relatively small in practice and will not grow substantially with the growth of the nodes' number. Unfortunately, the results have shown that the time of edge adding grows also with the growth of the nodes number even when the average number of edges connected to each node is constant. It could be a problem because it results in computational power increase faster than growth of the number of users. If further analysis shows that we cannot avoid this problem (e.g. by changing the graph framework), we will try to split the graph into some smaller structures (or into many smaller graphs).

The first step that may be done very easily is storing relations and communication channels separately. In our case, a graph that stores information about relations is modified rarely (the friendship relation is usually stable) so it may be implemented less optimally (we can use one graph to represent all relations). Unfortunately, communication graph is to be changed very often (communication channels will be dynamically created and deleted) so we have to find the way of increasing of performance of operations connected with this graph.

To resolve the aforementioned problem we have to analyze the characteristics of communication channels. In our case, communication channels represent the ability of face to face communication so they will always connect people that are located close to each other. This observation can be the basis for the creation of a communication graph splitting algorithm that uses user locations in the real world (see chapter 9).

9. Conclusions and future work

The tests have shown that the presented extended social network model can be used to create a platform that helps people meet in the real world. However, further work is needed to increase the performance of its implementation. First, we plan to analyse the results of performance tests once more using graph theory and investigate the implementation of the graph framework. If this analysis confirms our previous observations, we will focus on splitting the communication graph. We will try to use some algorithms that will split users into groups on the basis of their locations in the real world (e.g., using clustering algorithms). Afterwards, we will try to develop an algorithm that will allow for the managing these groups (migrations of users from one group to another, merging and splitting groups etc.).

Acknowledgements

This work is supported by European Union project UDA-POKL.04.01.01-00-367 /08-00.

References

[1] Facebook fact sheet. [online] `http://newsroom.fb.com/content/default.aspx?NewsAreaId=22`.

[2] Google earth home page. [online] `earth.google.com.`

[3] Indect home page. [online] `http://www.indect-project.eu`.

[4] Indect public deliverable: 2.2 description of systems architecture.

[5] Brodka P., Filipowski T., Kazienko P.: An introduction to community detection in multi-layered social network. In *Communications in Computer and Information Science, CCIS 111*, Springer, 2010, pp. 125–130, 2011.

[6] Geron T.: Banjo app connects with the nearby social world. [online] http://www.forbes.com/sites/tomiogeron/2011/06/22/banjo-app-connects-with-the-nearby-social-world/.

[7] Kazienko P., Brodka P., Musial K., Gaworecki J.: Multi-layered social network creation based on bibliographic data. In A. K. Elmagarmid, D. Agrawal, ed., *SocialCom/PASSAT*, pp. 407–412. IEEE Computer Society, 2010.

[8] Kazienko P., Kukla E., Musial K., Kajdanowicz T., Brodka P., Gaworecki J.: A generic model for multidimensional temporal social network. In *Communications in Computer and Information Science, CCIS 171*, Springer, 2011, pp. 1–14, 2011.

[9] Kazienko P., Musial K., Kukla E., Kajdanowicz T., Brodka P.: Multidimensional social network: Model and analysis. In P. Jedrzejowicz, N. T. Nguyen, K. Hoang, eds., *ICCCI (1)*, vol. 6922 of *Lecture Notes in Computer Science*, pp. 378–387. Springer, 2011.

[10] Maisto M.: Smartphone shipments overtake pc sales. [online] http://www.techweekeurope.co.uk/news/smartphone-shipments-overtake-

[11] Sutter J. D.: The faces of egypts revolution 2.0. [online] http://edition.cnn.com/2011/TECH/innovation/02/21/egypt.internet.revolution/index.html.

Affiliations

Michał Wrzeszcz
AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Computer Science, al. A. Mickiewicza 30, 30-059 Krakow, Poland, wrzeszcz@agh.edu.pl

Jacek Kitowski
AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Computer Science, al. A. Mickiewicza 30, 30-059 Krakow, Poland, kito@agh.edu.pl

Received: 14.03.2012
Revised: 2.07.2012
Accepted: 3.09.2012