The MAgNet: Agent-based Middleware Enabling Social Networking for Mobile Users

Marko Basuga, Rebeka Belavic, Ana Slipcevic, Vedran Podobnik, Ana Petric, Ignac Lovrek
University of Zagreb
Faculty of Electrical Engineering and Computing
Department of Telecommunications
Zagreb, Croatia
{marko.basuga, rebeka.belavic, ana.slipcevic, vedran.podobnik, ana.petric, ignac.lovrek}@fer.hr

Abstract—Nowadays social network services play an important role in users’ everyday lives. This is especially emphasized in lives of users that belong to Generation Y. Young people often communicate and share various personal information with their family, friends or colleagues through virtual communities that are based on social relations. By introducing social networking concept into the mobile network domain, proliferation and availability of social networking services could be notably expanded. This paper presents the MAgNet, a middleware based on software agent technology that enables social networking services for users in the mobile network domain. The MAgNet enables mobile users to define and customize their social relationships with other users, as well as it uses created relationships to plan and manage group events.

I. INTRODUCTION

Popularity of Web sites offering Social Networking Services (SNSs, e.g., Facebook1, MySpace2 or LinkedIn3) is constantly increasing [1]. Currently, one out of every fifteen Internet visits in the US goes to the top 20 social networking websites, so it is likely that this trend will continue in the future [2]. These sites are innovative because they introduce new ideas in the way how people connect with each other. By using these sites, users create self-descriptive profiles and establish contacts with other people on the site, creating a network of personal connections4. These connections are based on various criteria depending on users’ preferences, ranging from connections with friends and acquaintances to connections with strangers that share the same hobby, musical taste or even political beliefs. Once they are created, social networks help users collaborate over common activities or interests, or just to stay in touch with their friends or relatives. Social networks can also be used for various business purposes such as finding new career opportunities (e.g., LinkedIn).

Not only that social networks are useful to its users, but they also provide new possibilities for various business entities in the electronic communication market (e.g., telecoms, Internet service providers (ISPs)) or even broader (e.g., advertising companies). Business entities stand to increase their company’s profit by introducing novel business models. Those models can create value based on results obtained from the analysis of social networks: telecoms and ISPs are able to provide personalized services or create and offer innovative group services; advertising companies can utilize social networks as a medium for viral marketing.

However, SNSs commonly require the use of a personal computer with a broadband Internet connection. This fact limits the full potential of SNSs because mobile users are usually handicapped when trying to use such services. The number of mobile devices is constantly increasing and the network support is being continuously enhanced [3]. Additionally, a mobile phone holds a vast amount of personal information about its owner (e.g., a list of friends and acquaintances stored in user’s phonebook, meetings and birthday reminders via calendar entries and many more). By introducing social networking concept into the mobile network domain [4], proliferation and availability of SNSs could be greatly expanded. Mobile users would have access to these services at any given time, regardless of their current location.

Facing strong competition and falling voice revenues, mobile operators are increasingly interested in new business models that would generate profit [5]. Due to the nature of mobile phones, their status as one of the most personal items user can have, they present an ideal platform for implementation and development of SNSs. Mobile operators can increase their level of competitiveness on the market by using data gathered during users’ social network activities for the development of personalized services.

In this paper we present an agent-based middleware for social networking (MAgNet), a proof-of-concept prototype that enables SNSs for users in mobile network domain. The rest of the paper is organized as follows. A general overview of used technologies is given in Section 2. Proposed MAgNet middleware is described in Section 3. This section contains a detailed description of the middleware implementation, its functionalities, as well as two case studies. Section 4 concludes the paper and proposes guidelines for future work.

1 http://www.facebook.com
2 http://www.myspace.com
3 http://www.linkedin.com
4 Such networks which are created based on user-initiated and user-aware interlinking are called explicit social networks. Adversely, social networks can also be autonomously created based on similarity of user profiles: this type of networks is called implicit social networks.
II. AGENT-BASED MIDDLEWARE FOR SOCIAL NETWORKING

Current social networking sites provide a web-based service which is predominately tailored for stationary Internet users. Still, several SNSs can also be accessed by using a mobile device, but they are not fully customized for such use what leads to a significant loss in functionality. The proposed middleware would enable and simplify the use of SNSs on mobile devices. This is possible because mobile devices already have an advanced network support and are even more personalized than a personal computer. With technological advancements mobile phones today have become more feature-packed and they perform many functionalities in tune with the modern lifestyle. Since recently, touch screen phones are becoming more and more widespread. Since they do not use standard keyboard and therefore have larger screens. This enables different multimedia functions, from looking at photographs to watching video clips or TV programs. Those are just some of the reasons why touch screen phones are likely to replace standard mobile phones. Unlike mobile phones with standard keyboard, touch screen phones significantly simplify input of large data. A mobile device holds a large amount of user’s personal information. That information can be provided by the user explicitly or it can be acquired by monitoring users’ activities, from their current location to learning about their behaviour patterns (e.g., tracking the duration of users’ calls). Taking all of this into account, a mobile device represents a significant source of information that is needed to create a user’s profile.

We propose the MAgNet, a middleware based on software agents that enables SNSs for users in mobile network domain. Services provided by the MAgNet middleware can be roughly divided into two groups: creating and managing groups of users and planning group events. Current set of implemented services is used as the MAgNet proof-of-concept prototype but it possible to extend the middleware for provisioning more SNSs.

Recently, few other projects tried to tailor SNSs according to mobile user needs. One of them is the MobiSoc middleware [6] which provides a common platform for rapid development and deployment of mobile social computing applications. The Social Serendipity project [7] uses Bluetooth technology for detecting other nearby users. Although this project is not aimed at creating social services, it has recognized other potential uses for mobile phones. Another project, the SAMOA middleware [8], integrates a set of common management facilities for personalizing location-dependent social networks, and for propagating social networks’ visibility up to the application level. The Active Campus project [9] provides context-aware infrastructure for ubiquitous computing, exploring the challenges of simultaneously supporting extensibility and tight integration. The Whereabouts Diary [10] is an application/service for logging the places visited by the user and labelling them in an automatic way, with descriptive semantic information. Although many of these projects provide SNSs or use mobile phones for developing new applications as does the MAgNet, the main difference between these projects and the MAgNet is in the use of software agent technology for integrating SNSs and mobile phones.

In the following subsections we describe the idea of social networking (which is enabled by the MAgNet middleware), software agents (which are used to implement the MAgNet middleware) and finally user profiles (which enable agent-based personalization).

A. Social networks

A social network is a structure comprising of nodes connected to each other by using various criteria. It is described as a graph $G = (V, E)$. The nodes (i.e., vertices $V$) represent people or organizations connected through various criteria while the edges in the graph (E) represent those connections.

SNSs allow individuals to construct a public or semi-public profile within the system, create a list of other users they want to be connected with, and view and traverse their list of connections as well as those made by others within the system [1]. Nowadays SNSs play an important role in users’ everyday lives, especially those that belong to Generation Y. Young people often communicate and share various personal information through virtual communities that are based on social relations with their family, friends and colleagues. These social connectivity patterns can be used to create a network based on relationships between people. With the current pace of modern day life and relationships requiring a large proportion of one’s time, people are searching for a simpler and more convenient way to stay in touch with as many people as possible without spending too much of their time. As a result, many SNSs are becoming increasingly popular: they represent an extension of real-life social connections. Connections in SNS systems can be simple connections based on a single criteria (e.g., family ties) or complex connections based on a multiple criteria (e.g., music taste, religious views, etc.).

SNSs are not popular only among Internet users, they also attract attention of business entities. In 2007, over 90 social networking sites were active with over 630 million users [11], marking these services as a viable source of profit for various companies (e.g., advertising companies can utilize social networks for viral marketing and market research).

B. Software agents

Software agent technology is a rapidly evolving area of research and probably one of the fastest growing areas in the field of IT. A software agent [12][13][14][15] is a computer program that autonomously acts on behalf of its user. The most important property of an agent is its autonomy, meaning that the agent can act without direct intervention from its user and has control over its own actions and internal state. A multi-agent system (MAS) is a (distributed) system with a number of interacting software agents. Because agents are autonomous, MASs are not bound to a specific system but can be run on any computer or server which supports the execution of an agent platform.

Agents are able to learn users’ preferences and habits over time and adapt according to them if necessary. They are also cooperative which means that they communicate with other agents in the system in order to complete their tasks, but are not bound to the system where they began their execution. Moreover, agents have the ability to react dynamically which makes them suitable for developing robust and fault-tolerant distributed systems. Agents can
also be mobile which means that they can migrate from one host to another in the network. This is a very useful feature because it helps to reduce network load and balance processor load. Instead of exchanging significant amount of data during the communication between two distributed systems, agents can migrate to destination host and interactions can then take place locally. After it has finished its task, agent returns to its initial host with results.

C. User profiles

Information about the user as well as user’s personal information is stored within the user profile [16]. Profiles managed by the MAgNet middleware are described by using Resource Description Framework (RDF) and Friend-of-a-Friend (FOAF) vocabulary [17], what makes them compliant with the idea of Semantic Web. This approach for creating user profiles should also allow the mobile network operator to collect relevant user information and reason upon the collected information in a more meaningful way.

RDF is a language for representing information about resources in the World Wide Web [18]. RDF is used for creating a machine-processable web of data. For describing different types of resources a number of RDF vocabularies were developed. An RDF vocabulary for describing metadata about people, their interests, relationships and activities is FOAF. A simple example of how FOAF can be used for describing metadata about a person is given in Listing 1. This FOAF profile describes a person whose name is Annie, surname Scott, she has a phone number 6431287 and her e-mail address is annie@nomail.com.

Since all information described using FOAF is machine-interpretable, computers can use FOAF profiles to autonomously find people with certain similarities (e.g., all people living in Europe).

The user profile used by the MAgNet middleware currently contains several elements which are not part of the FOAF vocabulary: interest and wish list. These elements were added to support features and services provided by the MAgNet middleware.

III. THE MAGNET MIDDLEWARE

This section firstly describes the MAgNet middleware architecture and afterwards explains how the middleware is implemented. Finally, functionalities of the MAgNet middleware are presented through two proof-of-concept services.

A. The middleware architecture

The architecture of the proposed MAgNet middleware is designed as a MAS system which contains three types of software agents: Graphical User Interface Agent (GA), User Agent (UA) and Social Agent (SA), as shown in Figure 1.
The UA and its accompanying GA are assigned to a single user. The UA is designed to represent the user within the MAgNet middleware. The GA provides an interface between the user and his/her UA: it basically enables the user to access all of middleware functionalities and communicate with other users by presenting all relevant information and personal messages to the user via graphical interface. Finally, the SA provides the support and coordination needed for providing SNSs within the developed MAgNet middleware. Unlike the other two types of agents existing within the middleware (i.e., each agent is assigned to a single user), SA does not have a designated user and acts as a manager for the entire system: the SA is in charge of storing and managing currently existing user groups and coordinating the message exchange between UAs in the system. Agents’ position within the Next-Generation Network (NGN) architecture [19][20] is shown in Figure 2.

All messages exchanged within the middleware – sending friend requests, accepting or rejecting received friend requests, sending event invitations and event notifications (e.g., concerning attendance status change or cancellation of an event) – are displayed to the user via graphical interface of his GA, shown in Figure 3.

B. The middleware implementation

The MAgNet middleware was developed using Java Agent Development framework (JADE)\(^5\), a framework for developing agents and MASs in compliance with Foundation for Intelligent Physical Agents (FIPA)\(^6\) standards.

All agents within the middleware communicate by exchanging Agent Communication Language (ACL) messages. Each ACL message, besides carrying information for the receiving agent, has a type that defines its function (e.g., inform, request, cancel, reject or accept). In addition to message types, each message can be assigned with a conversation identifier (CID). The CID facilitates the detection and filtering of incoming ACL messages. Filtering ACL messages based on their CID and message type simplifies the communication between agents.

Each agent has a number of tasks. Every task is implemented as agent’s behaviour, which is basically a Java class. Adding behaviour to the agent tells an agent to execute the task which the behaviour represents. Since multi-threads are supported in Java, an agent can execute several behaviours concurrently. This allows having a single Java thread per agent which is especially useful in environments with limited resources such as mobile phones.

As stated earlier, there are two types of proof-of-concept services currently available within the MAgNet

\(^5\) http://jade.tilab.com/

\(^6\) http://www.fipa.org/
middleware. The first one is Group Management and the other one is Planning Group Events. Each of these two functionalities is realized by defining a series of tasks that have to be completed, and using one behaviour for each task. Agent’s behaviours within the system can be classified into two groups: general and specialized behaviours. General behaviours are always active, they are usually added to the agent during its setup phase and they are responsible for providing core system functionalities. These include behaviours responsible for communication between UA and GA, etc. Specialized behaviours are added to the agent when it needs to complete a specific task. These behaviours usually end after a certain time has passed or its termination has been requested (e.g., behaviours responsible for planning group events).

C. The middleware functionalities

In order to use the MAgNet, user has to create his user profile using the application Profilko shown in Figure 4. Profilko is suitable for use on mobile devices because it is a Java-based application whose graphical interface size fits the mobile devices’ screen size. The user profile contains the relevant information about the user: personal information such as first and last name, birth date, place of residence, information concerning the user’s interests and finally information describing user’s relations in the context of user groups. After the user agrees to use the MAgNet, the UA obtains information from the created user profile and acts on behalf of its user, minimizing user intervention. Still, user must explicitly agree to any major change. Privacy concepts are obtained from existing SNSs where they have proven to be successful. To ensure user privacy, the MAgNet also enables user to use so-called block lists, which will be explained in more detail later on.

1) Proof-of-concept service #1: Group Management

Each user in the system is assigned with their own UA that reads user’s profile and stores the relevant information: personal user information and information about groups. A sample MAgNet profile is given in Listing 2. This profile is an extension of the FOAF profile shown in Listing 1 and also describes a person whose name is Annie, surname Scott, phone number 6431287 and e-mail address annie@nomail.com. Additionally, we can learn that Annie’s birthday is January 1, 1985, she lives in London, her hobby is paragliding and she wishes a dog.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/TR/rdf-syntax-grammar/
xmlns:rdfs=http://www.w3.org/TR/rdf-schema/
xmlns:foaf="http://xmlns.com/foaf/spec/20071002.html"
xmlns:bio="http://vocab.org/bio/0.1/bio-vocab-20040305"
xmlns:rel="http://vocab.org/relationship/"
<foaf:Person rdf:nodeID="6431287">
<!--Identification Information-->
<foaf:Phone>6431287</foaf:Phone>
<!--Additional Information-->
<foaf:Name>Annie Scott</foaf:Name>
<foaf:givenname>Annie</foaf:givenname>
<foaf:family_name>Scott</foaf:family_name>
<bio:event>
  <bio:Birth>
    <bio:Date>1.1.1985</bio:Date>
  </bio:Birth>
  <bio:placeOfResidence>London</bio:placeOfResidence>
  <bio:habitualResidence>London</bio:habitualResidence>
  <foaf:mbox>annie@nomail.com</foaf:mbox>
  <foaf:interest>Paragliding</foaf:interest>
  <foaf:wishlist>dog</foaf:wishlist>
</bio:event>
</foaf:Person>
</rdf:RDF>
```

Listing 2. Sample User Profile

Figure 4. Application Profilko displaying all three tabs
In the user profile mobile phone number is also used as her/his ID and as the ID of her/his UA (i.e., user Annie Scott with profile defined in Listing 2 has an ID 6431287). Users can create their own personal groups by adding other users, using their IDs, to the desired groups. There are four predefined user groups in the user profile: Family, Friends, Colleagues and VIP. Other than these predefined groups, users can also create new groups. In the sample user profile (Listing 2), the user has only added two members to his/her Family group (i.e., user with ID 6431705 and user with ID 6342356) while other groups are empty. In his/her profile the user can also define a list of users he/she wants to prevent in accessing his/her information – the block list. The block list is not like other four lists used in the system since it represents another way, next to explicit user approval, to ensure user’s privacy. For the middleware, the block list is a tool which constrains certain actions and has to be checked before taking further steps. Although a user can add any other user to his/her groups, whether or not the added user will actually be a member of the group depends on his approval. For example, Alice will be in Bob’s Friends group only after she accepts his request to join his group. After processing the entire user profile, the UA starts the process of registration of user groups with the SA. Steps necessary to complete the registration of all members into a group are shown in Figure 5.

The registration process occurs within the MAgNet during the users’ initial access to the system. At the beginning of the registration process the UA sends a list of members to all members within the same group. After receiving the list with all the members of the user’s group, the SA sends a Friend Request (FR) to each member that has not already agreed to join the group. To keep track of these changes, each user is assigned a status within a group. Initially, all new group members have the newentry status. This status changes to requested after the SA sends the request to join the group (i.e., FR) and finally to approved after the user accepts to join the group. Before sending requests, the SA checks the block list of all users in the group. Furthermore, the SA automatically approves all FRs between every pair of users that wish to add each other in the same group. For example, if Alice wants to add Bob to her Friends group and if Bob also wishes to add Alice to his Friends group, the SA detects this situation and automatically approves both requests without sending, in this case, unnecessary FRs. The same principle applies if Alice wishes to add Bob to her Friends group and she is already a member of Bob’s Friends group; the SA automatically approves her FR sent to Bob. This approach of creating user groups significantly reduces a number of messages exchanged between agents in the created MAS. In all other cases, the user has to explicitly approve the FR in order to join the group. This approach is utilized by various SNSs, including Facebook, making it familiar and comprehensible to users.

![Figure 5. Activity diagram for Group Management service](image)

![Figure 6. GUI – window which contains a list of received FR](image)
The SA stores registered groups with a list of members that have received and approved a FR for that group and those to which a FR has been sent. By storing groups in one place, group related data can be accessed much faster and the amount of messages exchanged within the MAgNet is significantly decreased. The registration process is important because it is a prerequisite for using all other MAgNet middleware services and functionalities such as planning a group event. Received FRs are listed to the user on his GUI as shown in Figure 6.

2) Proof-of-concept service #2: Planning Group Events

The user also has an option to plan an event. This includes creating and specifying a new event, inviting one of his/her groups to attend the event or even cancelling it. Using the graphical interface of his GA, at all time the user can keep track of events he/she had created, events he/she is invited to, attendance status of other invited users at all time and can also change his/her own attendance status for a certain event.

User can create a new group event by selecting a day in the calendar and inviting one of his/her groups to attend the event or even cancelling it. Using the graphical interface which enables him/her to communicate with the MAgNet middleware. Other than selecting the desired group, the date and time of the event, the user can also supply additional event information such as place of the event, event description or a personal message for invited users (Figure 7).

After the user specifies the new group event, his UA sends the Event Request (ER) for planning the event to the SA. The ER contains the information about the event. The actual list of users that will be invited is determined by the SA who uses event information and lists of stored groups. This step is necessary because the group information stored by the SA is constantly being updated and therefore always accurate. The SA then sends Event Invitations (EIs) to UAs whose users are in the invited group list, and have presumably approved a friend request from the user who is planning the event. Therefore, only users who have accepted the friend request (FR) from the user who creates an event will receive the EI for this event. This is an example of the fact that the group defined by the user, and the actual group can be different. While FRs described in previous subsection are sent to all users listed in the registered group, EIs are sent only to those users on the list who have approved FRs and are therefore actual members of the certain group. Each invited user can see all other invited users, their attendance status and can change their own attendance status for the event (attending, maybe attending, not attending). This does not apply only in the case where one of the members of invited group has another member on his block list. The blocked member will not receive and will not be aware of status updates about the user who blocked him. Any change of the attendance status is reported to the SA which then forwards the information about the status change (Event Notification, EN) to other UAs. The UA which received the EN updates the event information and notifies its user by displaying EN on user’s graphical interface. Steps and actions necessary for planning a group event are shown in Figure 8.

Figure 7. Windows for creating a new event and viewing event invitations

Figure 8. Activity diagram for Planning Group Events service
The event can be cancelled in two ways. Firstly, the user who created the event erases it from his calendar or sets his attendance status to \textit{not attending}. The notification about the cancelled event is also sent to the SA which notifies UAs of members invited to this event. Secondly, when the date the event was scheduled for expires, the behaviour assigned to that event is automatically terminated and no further status changes or ENs will be delivered.

IV. CONCLUSION

Current social networking sites provide a web-based service which is predominately tailored for stationary Internet users. We believe that such approach fails to adequately utilize the full potential of these services: by developing the middleware designed for mediation between SNSs and mobile users even a larger number of potential users could be reached. This paper proposed an agent-based middleware, called the MAgnet, which enables mobile users to use Social Networking Services (SNSs).

The MAgnet was developed by using JADE framework which is fully implemented in Java programming language. JADE was used because it simplifies the implementation of multi-agent systems and because Java is a good choice for developing applications which should be deployed on a variety of end devices, such as mobile devices.

Two proof-of-concept services showed that the MAgnet can enable mobile users to define and customize their social relations with other (mobile) users, as well as use the created relations to plan and manage group events. We believe this proves that software agents represent an adequate solution for implementing a middleware that enables SNSs for users in the mobile network domain.

For future work we plan to measure scalability and time efficiency of the proposed middleware, and to perform a field trial of the MAgnet with a test group of users. Additionally, it is planned to add new services to the MAgnet middleware (e.g., modelling additional services such as virtual wedding/birthday gift lists, location-based services such as receiving newsletters from museums or theatres near the user) and to reuse existing social networking sites information for users who already have accounts in Facebook, Twitter etc.

ACKNOWLEDGEMENTS

This work was carried out within research projects 036-0362027-1639 "Content Delivery and Mobility of Users and Services in New Generation Networks", supported by the Ministry of Science, Education and Sports of the Republic of Croatia, and "Agent-based Service & Telecom Operations Management", supported by Ericsson Nikola Tesla, Croatia.

REFERENCES


