Agent-based Social Networking for Mobile Users

Rebeka Belavic, Marko Basuga, Vedran Podobnik*, Ana Petric, Ignac Lovrek
University of Zagreb
Faculty of Electrical Engineering and Computing
Department of Telecommunications
Zagreb, Croatia
{rebeka.belavic, marko.basuga, 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 an agent-based social network for users in the mobile network domain. The presented social network enables mobile users to define and customize their social relationships with other users, as well as to use those relationships for planning and managing group events. Additionally, users can also create and use wishlists.

Keywords: Agents, Software agents, Social networks, Social networking, Mobile users, Multi-agent systems

Author Bios

Rebeka Belavic is a Master’s Degree student at University of Zagreb, Faculty of Electrical Engineering and Computing. She received her Bachelor’s Degree in Telecommunications and Informatics from University of Zagreb, Faculty of Electrical Engineering and Computing. Her research interests include social networks, software agents and multi-agent systems.

Marko Basuga is a Master’s Degree student at University of Zagreb, Faculty of Electrical Engineering and Computing. He received his Bachelor’s Degree in Telecommunications and Informatics from University of Zagreb, Faculty of Electrical Engineering and Computing. His research interests include social networks, software agents and multi-agent systems.

Vedran Podobnik received his Dipl.Ing. (2006) degree in Electrical Engineering from the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. Currently, he is a Ph.D. student in Computer Sciences at the same Faculty. He works as a Research Assistant at the Department of Telecommunications of the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. His research interests include multi-agent systems, social networks, electronic markets, context-aware services, business process automation, and the semantic web. He is a member of the IEEE and KES International.

* Corresponding author – phone: + 385 1 6129748; email: vedran.podobnik@fer.hr; University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Telecommunications, Unska 3, HR-10000 Zagreb, Croatia.

Copyright © 2010 Inderscience Enterprises Ltd.
Ana Petric received her Dipl.Ing. degree in Electrical Engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia, in 2005. Since then, she has been employed at the Department of Telecommunications at the same University as a Research Assistant, where she is currently working towards her Ph.D. degree. Her research interests include procurement auctions, multi-attribute auctions, B2B electronic markets, supply chain management and intelligent software agents. She is a member of the IEEE.

Ignac Lovrek is with the Faculty of Electrical Engineering and Computing of the University of Zagreb, Croatia, where he works as a professor at the Department of Telecommunications. He authored and co-authored over hundred papers published in books, journals and conferences. His research interest includes call and service modeling and processing, software engineering and new soft technologies for telecommunications, and market of electronic services, including software agents theory and applications. He is a member of IEEE, ACM, GI - Special Interest Group on Petri Nets and KES International.
Introduction

Popularity of Web sites offering Social Networking Services (SNSs, e.g., Facebook\(^2\), MySpace\(^3\) or LinkedIn\(^4\)) 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 connections\(^5\). 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. These models can create value based on the results obtained from the social network analysis, i.e., telecoms and ISPs are able to provide personalized services or create and offer innovative group services while 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 due to the disadvantages placed before mobile users (e.g., battery consumption, screen size) who are 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 social network for mobile users based on MAgNet (Multi AGent system for enabling social NETworking for mobile users) middleware [6], a proof-of-concept prototype that enables SNSs for users in the mobile network domain. The remainder of the paper is organized as follows. An overview of used technologies is given in Section 2. The proposed MAgNet middleware is presented in Section 3 which contains a detailed description of the middleware implementation and

---

2 http://www.facebook.com
3 http://www.myspace.com
4 http://www.linkedin.com
5 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.

Copyright © 2010 Inderscience Enterprises Ltd.
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 phone, 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. Even though Internet access and data transfer from the mobile device is still more expensive than from the personal computer, the number of users who want to be connected all the time, no matter where they are is growing. This has sparked significant growth and progress in mobile Internet access and user security, which can today be classified as satisfactory by user standards. Furthermore, mobile devices are even more personalized than a personal computer. Due to physical characteristics of mobile devices (i.e., size mostly), users carry them along all the time which enables continuous interactivity and immediacy. Users do not just passively consume content provisioned through services, but want to interact with it, generate their own content and share it online.

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, ranging from their current location to learning about their behaviour patterns (e.g., tracking the duration of users’ calls). This enables development of location-based services which cannot be used by stationary users. A mobile device represents a significant source of information that is needed to create a user’s profile.

Advantages and disadvantages of mobile and stationary users are listed in the Table 1 shown below. It can be noticed from the table that, compared to the stationary users, mobile users’ main advantages are their mobility, possibility for device personalization, continuous interactivity and location awareness. Additionally, mobile devices have an advanced network support, although data transfer is still more expensive on mobile devices than it is on personal computers.

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Stationary</th>
<th>Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Personalization</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Network support</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Security</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Cost</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Location-awareness</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Continuous interactivity</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Immediacy</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 1. Comparison of the mobile and stationary users
Furthermore, with technological advancements mobile devices today have become more feature-packed and they perform various functionalities in tune with the modern lifestyle. Since recently, touch screen phones are becoming more and more widespread. The fact that they do not use standard keyboard enables them to have larger screens. This facilitates 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 amount of data. Taking all of this into account, mobile device represents a viable platform for implementing advanced services and transferring existing services for stationary users into the mobile domain.

We propose an agent-based social network for mobile users grounded on MAgnNet, a middleware based on software agents that enables SNSs for users in mobile network domain. Services provided by the MAgnNet middleware can be roughly divided into three groups:

1. creating and managing groups of users;
2. planning group events;
3. creating and using wishlists.

First two services are implemented while the third one is just modelled but not yet implemented within the MAgnNet middleware. Current set of implemented services is used as the MAgnNet middleware proof-of-concept prototype but it is possible to extend the middleware for provisioning more SNSs.

In the following subsections we describe the idea of social networking (which is enabled by the MAgnNet middleware), software agents (which are used to implement the MAgnNet middleware), user profiles (which enable agent-based personalization) and finally MIDlets (mobile application which provides graphical interface for users enabling them to access all middleware functionalities).

**Social networks**

A social network is a structure comprised of nodes which are 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 links in the graph (i.e., edges $E$) 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, the popularity of many SNSs is increasing since 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 or political 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 [7], 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). The explosive growth of social networking sites’ popularity continues – in 2010, only Facebook (the most popular social networking site) has over 400 million users.

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 [8][9][10][11] 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.

User profiles

Information about the user as well as user’s personal information is stored within the user profile [12]. Profiles managed by the MAgNet middleware are described by using Resource Description Framework (RDF) and Friend-of-a-Friend (FOAF) vocabulary [13], 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 [14]. 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.

```xml
<foaf:Person rdf:nodeID="6431287">
    <foaf:Phone>6431287</foaf:Phone>
    <foaf:Name>Annie Scott</foaf:Name>
    <foaf:givenname>Annie</foaf:givenname>
    <foaf:family_name>Scott</foaf:family_name>
    <foaf:mbox>annie@nomail.com</foaf:mbox>
</foaf:Person>
```

Listing 1. A simple FOAF profile
Title

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 *wishlist*. These elements were added to support features and services provided by the MAgNet middleware.

Mobile application

User’s communication with the MAgNet middleware is enabled by using MIDlets. MIDlet is a J2ME\(^6\) (Java 2 Micro Edition) application framework for the Mobile Information Device Profile (MIDP) that is typically implemented on Java-enabled mobile device or other embedded device or emulator (Figure 1) [15]. J2ME, a version of Java for small devices, defines two configurations: CDC\(^7\) (Connected Device Configuration) and CLDC\(^8\) (Connected Limited Device Configuration). Configuration defines core APIs (Application Programming Interface), while MIDP\(^9\) adds the APIs and specifications necessary to develop applications for a specific family of devices.

CLDC configuration is aimed for smaller devices like mobile phones, pagers, PDAs, and other devices of similar size. The CLDC is designed for devices with 160KB to 512KB of total memory, including a minimum of 160KB of ROM and 32KB of RAM available for the Java platform.

MIDP requires: a minimum of 256KB of ROM for the MIDP implementation (this in addition to the requirements of the CLDC), a minimum of 128KB of RAM for the Java runtime heap, a minimum of 8KB of nonvolatile writable memory for persistent data, a screen of at least 96×54 pixels, some capacity for input, either by keypad, keyboard, or touch screen and two-way network connection, possibly intermittent. MAgNet mobile application is developed for mobile devices supporting MIDP 2.0 and CLDC 1.0.

Figure 1: J2ME configurations and profiles [15]

---

\(^6\) [http://java.sun.com/javame/index.jsp](http://java.sun.com/javame/index.jsp)

\(^7\) [http://java.sun.com/javame/technology/cdc/](http://java.sun.com/javame/technology/cdc/)


Author

Related work

Besides the development of mobile versions of popular SNSs, the development of SNSs for mobile devices is also increasing. One of the projects which identified the importance of mobile users is the MobiSoc middleware [16]. It provides a common platform for capturing social state of physical communities learns patterns from the geo-social data and incorporates this new knowledge with the social state.

The Social Serendipity project [17] uses Bluetooth technology for detecting other nearby users. The service server contains user profiles and matchmaking preferences. It calculates similarity score between users’ profiles and behavioural data and uses it to cue informal interactions between nearby users who do not know each other. Although this project is not aimed at creating social services, it has recognized other potential uses for mobile devices.

Another project, the SAMOA framework [18], integrates a set of common management facilities for personalizing location-dependent social networks and for propagating social networks’ visibility up to the application level. Social networks created by the SAMOA framework are centred on a user and they are based on place visibility (i.e., the visibility of the user’s physical place) as well as on profile visibility (i.e., the visibility of place or user requirements and characteristics). The authors propose the use of their semantic context-aware social networks for various viral marketing scenarios.

The Whereabouts Diary [19] is a middleware service that supports location-aware activities of a mobile user by for logging the places visited by the user and labelling them in an automatic way, with descriptive semantic information. The use of the service results with a diary that describes the user daily life. The idea is to use the diary as a complement of the user profile and make available high-level knowledge about the context and the user activities for many other applications.

The Active Campus project [20] provides context-aware infrastructure for ubiquitous computing, exploring the challenges of simultaneously supporting extensibility and tight integration. Although many of these projects provide SNSs or use mobile devices for developing new applications as does the MAgNet middleware, the main difference between these projects and the MAgNet middleware is in the use of software agent technology for integrating SNSs and mobile devices.

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 three proof-of-concept services.
The middleware architecture

As shown in the Figure 2, the architecture of the MAgNet middleware can be roughly divided into three parts:

1. MAgNet Core;
2. Mobile application (MIDlet);
3. SIP Gateway.

Figure 2. The MAgNet middleware architecture
MAgNet Core enables all services provided by the MAgNet middleware. It is a MAS system that is comprised of two types of software agents: User Agent (UA) and Social Agent (SA). The UA is assigned to a single user and is designed to represent the user within the MAgNet middleware. The SA provides the support and coordination needed for providing SNSs within the developed MAgNet middleware. Unlike the UA (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 manages currently existing user groups and coordinates the message exchange between UAs in the system.

Mobile application (MIDlet) provides an interface between the user and her/his UA in the MAgNet Core: it basically enables the user to access middleware functionalities and to communicate with other users by presenting all relevant information and personal messages to the user via graphical interface.

In order to connect these two parts of the MAgNet middleware (i.e., MAgNet Core and MIDlet) and to support user mobility, SIP (Session Initiation Protocol) Gateway is used. MIDlet and the MAgNet Core communicate over a mobile network via SIP and SIP Gateway is responsible for enabling this communication. It consists of a servlet running...
on the Sailfin SIP server and two types of software agents: Send SIP Message Agent (SSMA) and Receive SIP Message Agent (RSMA). SSMA receives messages from the UA and forwards it to the accompanying MIDlet via servlet while RSMA forwards a message sent by MIDlet to the appropriate UA.

Agents’ position within the Next-Generation Network (NGN) architecture [21] is shown in Figure 3. Access and transport layers have the task to receive/send all data from/to mobile device while the function of transport control layer is to connect the various elements of the system via the access layer. Service control layer coordinates the access to various value-added services (i.e., MAgNet middleware services) which are located on the service layer.

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 user interface (GUI) of her/his MIDlet, shown in Figure 4.

Figure 4. Main MIDlet window

10 https://sailfin.dev.java.net/
Main MIDlet window displays information retrieved from the UA regarding the unread notifications received while the MIDlet was off. These notifications include: new friends (approved friend requests), received friend requests, new event invitations and event notifications. From this window the user can navigate to Friends or Events window which enable her/him to use services provided by MAgNet middleware, and to Planner window where the user can see her/his schedule for chosen date.

The middleware implementation

The MAgNet middleware was developed using Java Agent DEvelopment framework (JADE\textsuperscript{11}), a framework for developing agents and MASs in compliance with Foundation for Intelligent Physical Agents (FIPA\textsuperscript{12}) standards. Even though there is a modified version of the JADE platform - Light Extensible Agent Platform (JADE LEAP), which enables FIPA agents to execute on lightweight devices such as mobile phones running Java, this platform was not used to develop the MAgNet middleware. By using JADE LEAP user agents would have to be run on the mobile device. However, user agents are not developed to run on the mobile device, enabling them to be more autonomous, proactive and useful. There is a simple reason for this and it lies in the fact that user agent can move, act and complete tasks in the network even when the mobile phone is not turned on, allowing the user agent to work while its user is absent. Moreover, by separating the MAgNet middleware into three independent, but interconnected layers modularity to the entire system is added and the creation of mobile applications for various platforms as well as adding support for protocols other than SIP is simplified. Furthermore, by separating the mobile application from the user agent and the middleware core it is now possible to add support for various mobile platforms, even the ones without Java support, by developing the required mobile application while the system core itself remains platform independent.

In order to enable services provided by the MAgNet middleware, entities shown in Figure 2. have to communicate with each other. There are two types of communication within the MAgNet middleware:

1. Communication within the MAgNet Core (between the UA and the SA);
2. Communication between the MAgNet Core and the Mobile application (between the UA and appropriate MIDlet over SIP Gateway).

All agents within the MAgNet Core 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.

Mobile application (MIDlet) communicates with the MAgNet Core by sending and receiving SIP messages. Since all agents within the MAgNet Core communicate by exchanging ACL messages, SIP Gateway must perform the appropriate transformations. Each SIP message created by the MIDlet, besides carrying the required contact information, holds information necessary for creating associated ACL message. After receiving the SIP message, the SIP Gateway creates an ACL message by using the afore

\textsuperscript{11} http://jade.tilab.com/
\textsuperscript{12} http://www.fipa.org/
mentioned data and sends it to the appropriate UA. Similarly, when the UA sends a message intended for its accompanying MIDlet, the SIP Gateway transforms UA’s ACL message to a SIP message and forwards it to the MIDlet. These two cases are shown in the Figure 5:

![Figure 5: Communication within the MAgNet middleware: a) from MIDlet to MAgNet Core; b) from MAgNet Core to MIDlet](image)

Each agent has a number of tasks and each task is implemented as agent’s behaviour. Adding a behaviour to an agent tells the agent to execute a task which the behaviour represents. As stated earlier, there are three types of proof-of-concept services currently available within the MAgNet middleware: group management, planning group events and managing users’ wishlist which has not yet been implemented. Each of these three services 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 (e.g., communication between UA and SA). 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 or coordinating a selection of items off a user’s wishlist).
Figure 6. Application Profilko displaying: a) basic information; b) interests and wishlist; and; c) user’s groups.
Title

The middleware services

In order to use the MAgNet middleware services, a user has to create her/his user profile using the application Profilko shown in Figure 6. 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 middleware, 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. One of the concepts used in the MAgNet middleware to ensure user privacy is so-called block list, which will be explained in more detail later on.

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

Before describing the registration process, three important system features and their purposes will be explained: user profile, user group and block list.

1. User profile

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. Mobile phone number stored in the user profile is also used as the user’s ID and as the ID of user’s accompanying UA (i.e., user Annie Scott with profile defined in Listing 2. has an ID 6431287).
Listing 2. Sample User Profile

2. User groups

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 her/his Family group (i.e., user with ID 6431705 and user with ID 6342356) while other groups are empty. Every user can add users she/he wishes into her/his group when...
Title

ing in or modifying her/his profile. It is important to notice that the group which the user has created in her/his profile can differ from the real user group because whether the added user will actually be a member of the real group or not depends on the approval of added member (e.g. Alice will be in Bob’s Friends group only after she accepts his request to join his group).

3. Block list

In her/his profile the user can also define a list of users she/he wants to prevent in accessing her/his information – the block list. The block list is not like other four lists used in the system since it represents an extra measure, next to explicit user approval, for ensuring user’s privacy. The block list is a tool which constrains certain actions in the system in order to ensure user’s privacy and has to be checked before taking certain steps.

The registration process

After processing the entire user profile, the UA starts the registration process of user groups with the SA. Steps necessary to complete the registration of all members into a group are shown in Figure 7.

At the beginning of the registration process the Inviter’s UA sends a list of all members within the same user group to the SA. After receiving the list with all the members of the Inviter user’s group, the SA checks the AMS (Agent Management System) of the agent platform whether the UAs of the desired group members exist in the system (i.e., if those users even created their accounts). After this initial search the SA invites listed invited members to join the Inviter’s user group. Before sending invitations two additional actions are taken in order to create a list of all invited members (Figure 8).
Step 1: The block list check

As mentioned before, each user can define a list of users she/he wants to prevent in accessing her/his information. Before sending invitations to invited users, the SA checks the invited members' block lists to check whether the Inviter is listed in anyone of them. If the Inviter is in the invited user's block list, it means that the Inviter is actually blocked by the user she/he wishes to invite to her/his group. For example, Alice registers the
Title

group *Friends* with Bob as a new member. SA retrieves Bob's block list and checks if Alice is listed. If she is in Bob's block list, her request is automatically discarded. Otherwise, SA continues with the registration process.

Step 2: The corresponding user group check

The SA automatically approves all Friend Requests (FRs) between every pair of users that wish to add each other in the same group. This is done by checking the corresponding group of the invited member. 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 member is added to the list of invited members and is sent a FR.

Step 3: Sending Friend Requests

After these two steps are finished the list of invited users is created and FRs are sent to invited members. 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.

Belavíc, Rebeka; Basuga, Marko; Podobnik, Vedran; Petric, Ana; Lovrek, Ignac. 
Agent-based Social Networking for Mobile Users. 
Each user is assigned a status within the registered 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.

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 faster and the amount of messages exchanged within the MAgNet middleware is significantly decreased. The registration process is important because it is a prerequisite for using all other MAgNet middleware services such as planning a group event. Received FRs are listed to the user on her/his MIDlet as shown in Figure 9.

Number of approved FRs, new FRs and user defined groups are displayed on MIDLet’s Friends window. By selecting options listed in the Menu, user can accept or decline chosen FRs, view approved FRs and group members, as shown in Figure 10.
It is important to note that users who approved FR are automatically listed in the appropriate group.

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

The user is also offered the service which enables her/him to plan an event. This service includes two main functionalities:
1. creating and specifying a new event;
2. inviting one of her/his groups to attend the event, as well as two additional functionalities (changing her/his attendance status and cancelling the event).

Using the graphical interface of her/his MIDlet, the user can keep track of events she/he had created, events she/he is invited to, attendance status of other invited users at all times and can also change her/his own attendance status for a certain event.
Steps and actions necessary for planning a group event are shown in Figure 11 and described below.

1. Creating and specifying a new event
User can create a new group event by selecting a day in the calendar and inviting a desired group using the graphical interface which enables her/him 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 12).
2. Inviting a group to the event

After the user has specified the new group event, his UA sends the Event Request (ER) for planning an event to the SA. The ER contains 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 it is 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. Example of EI is shown in Figure 13.
Only users who have accepted the friend request (FR) from the user who created 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 the previous subsection are sent to all users listed in the registered group, EIs are sent only to those users from the list who have approved FRs and are therefore actual members of a certain group. The block list of each invited member is also checked before sending EIs (Figure 14).
Figure 14: Flowchart diagram for creating personalized Event Invitations
Additional feature 1: Changing the attendance status

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 when one of the invited group members has another member of the group on his block list. The blocked member will not receive and will not be aware of status updates about the user who blocked her/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 MIDlet (Figure 15).

![Figure 15: MIDlet windows showing: a) received event notifications, and; b) detailed event notification](image)

Additional feature 2: Cancelling the event

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 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.
3) Proof-of-concept service #3: Wishlist

Each user can create their own wishlist and make it available to her/his friends or certain groups of her/his choice. For example, Alice can create a birthday wishlist and make it visible only to her Friends group. Steps and actions necessary to enable the wishlist service are shown in Figure 16.

The user creates her/his wishlist in her/his profile so the wishlist is automatically stored in her/his accompanying UA. Twenty days before the user’s birthday, his UA checks the profile for members or user groups that the user listed in his profile and enables them to view the wishlist. UA sends the notification of the user’s upcoming birthday, wishlist and a list of members that should be notified to the SA. SA forwards this notification and the wishlist to the listed members (and begins monitoring the wishlist and the members’ responses).

Each user that received the notification can choose an item from the wishlist and mark it, which means he is going to give that item to the user as a birthday present. Whenever an item is marked, the message is automatically sent to the SA. SA marks the item and sends a notification to the remaining UAs that the item has been selected. The remaining UAs remove the item from the wishlist after receiving the notification. This way, it is ensured that two different users do not buy the same present.

It is important to note that none of the messages mentioned above are ever sent to the UA of the user whose birthday it is. That way the user does not know what exactly, if anything, off his wishlist he is going to receive.

13 The wishlist proof-of-concept service is not yet implemented within the MAgNet middleware, but just modelled. For future work we plan its implementation.

**Trial Results**

In order to evaluate the current design and functionality of the MAgNet system, a group of ten users was asked to test the system and then to answer a few short questions regarding it. The anonymous questionnaire contained six questions with grades from 1 to 5 offered to the examinees. The questionnaire and the results are presented below (Figure 17 to Figure 22).

**Question 1:**

*In a relation to other social networking software I have used, I found the MAgNet system to be:*

Very difficult to use 1 . 2 . 3 . 4 . 5 Very easy to use

![Figure 17: Results of the first question](image)

**Question 2:**

*MAgNet system functions are easy to find and corresponding menu items are well organized:*

Strongly disagree 1 . 2 . 3 . 4 . 5 Strongly agree

Belavic, Rebeka; Basuga, Marko; Podobnik, Vedran; Petric, Ana; Lovrek, Ignac. 
Agent-based Social Networking for Mobile Users. 
Question 3:
All expected functions are present in the menus:
Strongly disagree 1 2 3 4 5  Strongly agree

Question 4:
I found navigating around the MAgNet system screen to be:
Very difficult to use 1 2 3 4 5  Very easy to use
Figure 20: Results of the fourth question

Question 5:
I would recommend it to a friend:
Strongly disagree  1 . 2 . 3 . 4 . 5   Strongly agree

Figure 21: Results of the fifth question

Question 6:
My overall impression of the MAgNet system is:
Very difficult to use  1 . 2 . 3 . 4 . 5   Very easy to use
As the results of the questionnaire have shown, users found the current services offered by the MAgNet system and the design of the mobile application acceptable, but felt that improving the system by adding new services would make the system more useful.

**Conclusion**

Current social networking sites provide a web-based service which is predominately tailored for stationary Internet users. We believe that such an 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 social network for mobile users based on the middleware, called the MAgNet, which enables mobile users to use Social Networking Services (SNSs).

The MAgNet middleware 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. By adding the SIP Gateway to the MAgNet middleware, creation of mobile applications for various mobile devices, currently in the form of a MIDlet, was enabled.

Three proof-of-concept services showed that the MAgNet middleware enables mobile users to define and customize their social relations with other (mobile) users, to use established relations to plan and manage group events as well as to create and use birthday wishlists. 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 middleware with a test group of...
users. Additionally, we plan to add new services to the MAgNet middleware (i.e., described wishlist service, modelling additional services such as virtual wedding 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 on 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

Title

