The AMiGO-Mob: Agent-based Middleware for Group-oriented Mobile Service Provisioning

L. Vrdoljak, I. Bojic, V. Podobnik, and M. Kusek
University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia
{luka.vrdoljak, iva.bojic, vedran.podobnik, mario.kusek}@fer.hr

***Abstract*—In order to increase profit and remain competitive on the telecommunication market, telcos must introduce new types of services. Nowadays, basic mobile services are insufficient for an average user: advanced services include combination of entertainment and information content, and as such require greater personalization. The future of mobile communications is directed towards creating systems aware of user preferences, mobile device capabilities, and communication context. Such systems will enable telcos to use new approaches in service provisioning (e.g., dynamic user group formation can enhance provisioning of group-oriented services). This paper presents how using Semantic Web and software agent technologies can help improving telecommunication processes: we propose AMiGO-Mob middleware for enhancing group-oriented mobile service provisioning.**

I. Introduction

Simultaneous development of mobile devices and telecom operators' (telcos') infrastructure resulted in increasing complexity of mobile services [1]. The future of mobile communications is evolving from *linear services* (i.e., traditional services where the user cannot influence the predefined service provisioning procedure) towards new *non-linear services* (i.e., interactive services where the user participates in the service provisioning procedure, tailoring the service to his/her preferences, device and/or context) [2]. The non-linear services were available only for fixed network users until recently. In this paper we study special type of non-linear services: *group-oriented services* for *mobile users*:

- We define the *group-oriented service* as service in whose provisioning cannot participate just one user, but a set of users with certain similarities (e.g., similar preferences, devices and/or context);
- We define *mobile users* as users possessing mobile devices (e.g., mobile phone or PDA).

The main idea behind the group-oriented services is to group mobile users into clusters taking into account users' interests, their mobile devices' characteristics and the context in which they find themselves while requesting a service. To achieve that it is necessary to introduce a rather new approach in the service provisioning process: building implicit social networks of mobile users. Unlike explicit social networks (e.g., Facebook[1], MySpace[2] or LinkedIn[3]), implicit networks are built autonomously based on similarities of user profiles, without the interference of users themselves and in order to provide useful information for telcos [3].

Semantic Web technologies are rather novel but very amenable grounding for user clustering [4][5][6], while software agents have proven to be very suitable for user profile management [7][8][9] and telecommunication processes enhancements [9][10][12]. Semantic Web languages, such as *Resource Data Framework (RDF)*[4], *RDF Schema (RDFS)*[5] and the *Web Ontology Language (OWL)*[6], can be used to maintain detailed user profiles [13]. With the help of various query languages, based on *Structured Query Language (SQL)* syntax, it is possible to perform very efficient semantic profile matchmaking once the profiles have been created according to a certain standard. Such matchmaking enables us to perform clustering according to true, semantic similarities, rather than keyword matchmaking.

The paper is organized as follows. Section 2 describes the idea of three-layered middleware architecture on which AMiGO-Mob middleware is based. Section 3 presents the technologies used for implementation of AMiGO-Mob: the Semantic Web and software agents. Section 4 describes the implementation details of our middleware as a multi-agent system (MAS). In section 5 we elaborate upon a proof-of-concept group-oriented service enabled by AMiGO-Mob. Section 6 concludes the paper and proposes some directions for future work.

II. Three-Layered Middleware Architecture

Three-layered architecture used for modeling AMiGO-Mob is based on threefold view on the mobile users (see Figure 1) [14]:

- *physical layer*: this layer observes mobile users as humans physically situated in the mobile network environment and possesing mobile device;
- *ontology layer*: this layer observes mobile users through their semantic profiles;
- *social layer*: this layer observes mobile users through implicit social network – users are filtered by the telcos in aspect of a specific location, and are then included in the social network if their profile reaches a certain criteria (i.e., user and service profiles' similarity exceeds the given threshold).

---

[1] *http://www.facebook.com*
[2] *http://www.myspace.com*
[3] *http://www.linkedin.com*
[4] RDF specifications: *http://www.w3.org/RDF/*
[5] RDFS specifications: *http://www.w3.org/TR/rdf-schema/*
[6] OWL specifications: *http://www.w3.org/TR/owl-features/*

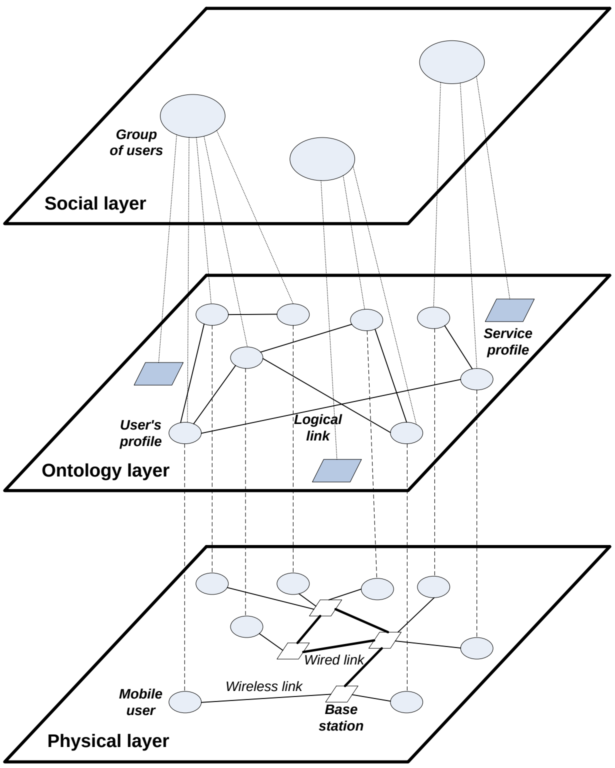


Figure 1. Three-layered social network

A. Physical Layer

Mobile users are connected to the telco's network via physical links. Mobile users possess mobile devices which are able to communicate with the telco's *base stations* via *wireless connections*. Mobile devices are also equipped with *Bluetooth* technology which enables *ad hoc* connections among users. The base stations are interconnected using *wired links* in the telco's core network.

B. Ontology Layer

Mobile users are represented through their semantic profiles. Additionally, telcos maintain profiles of all available services. Each service or user profile can be described using static and dynamic attributes. Mobile services defined in our ontology are described completely using static attributes. User profiles used by AMiGO-Mob are combined: they contain both static and dynamic attributes.

A single service profile (see Table I) consists of four sub-parts, namely: 1) *identification* (i.e., name of the service, *URI*); 2) *service capabilities* (i.e., bits per pixel, *QoS*); 3) *service requirements* (i.e., screen resolution, price); and 4) *service interface* (i.e., plain text, audio and video). Identification information provides information to name the service and identify its location. Service capabilities basically represent logical units of service functionality. Service requirements describe minimal mobile device or user capabilities necessary for adequate service consumption. Service interface defines the service delivery type.

TABLE I.
Service Profile

ID	Slumdog Millionaire - trailer
hasDescription	8 Oscar 2009 awards
hasDate	71225
hasDuration	7500
hasFramesPerSecond	15
hasHorizontalResolution	400
hasPrice	20
hasSize	200000
hasVerticalResolution	300
hasDeliveryType	Streaming
hasGenre	DramaMovie
hasInformationType	AdvancedSystemsFormat
hasLanguage	English

A user profile (see Figure 2) also consists of four static sub-parts: 1) *identification* (i.e., *IMEI*); 2) *user preferences* (i.e., *preferred service delivery type*, *preferred genre* for music or movies); 3) *mobile device hardware* (i.e., *screen resolution*, *available memory*); and 4) *software capabilities* (i.e., *Java version*, *Web browser*, *operating system*). As an addition, user profiles contain a dynamic component – *context information*. Context defines user's location, time, activity, etc.

C. Social Layer

Semantic description of user and service profiles enables similarity calculation (i.e., semantic matchmaking) between the desired service and user aspirations. The context part of the profiles enables telcos to calculate physical distance between users. Both semantic similarity and physical distance can be useful in order to determine the potential target users for specific services. Thus we can create a social network whose subnets are based on location and semantic similarity between user and service profiles.

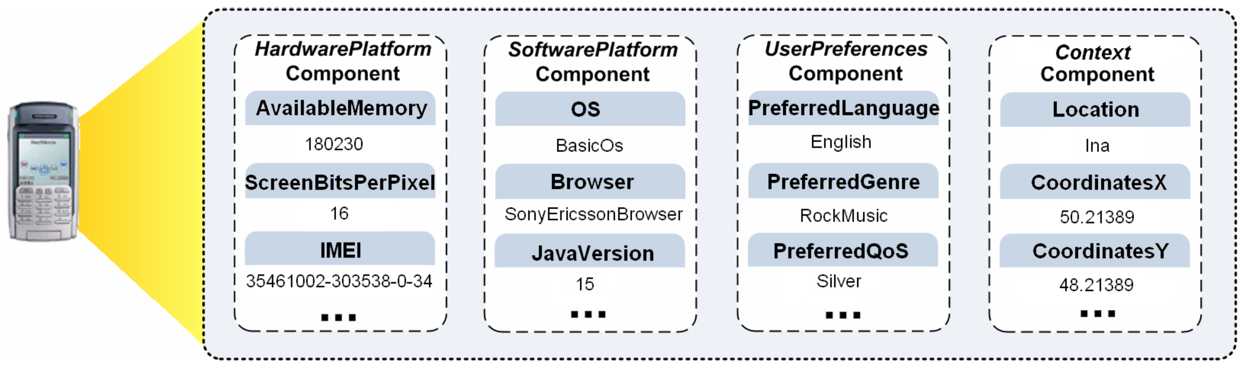


Figure 2. Mobile user profile

III. AMiGO-Mob Middleware for Group-Oriented Mobile Service Provisioning

This paper presents how using Semantic Web and software agent technologies makes it possible to dynamically cluster mobile service users. These technologies are used to model the AMiGO-Mob: system based on user and service profiles, semantic reasoning and self-organizing algorithms, enabling telcos new approach to provisioning of group-oriented services.

A. Semantic Web

The *World Wide Web (WWW)* has become the network of knowledge where most of the information is written in human-friendly form. The majority of search engines today are keyword-based. While keyword-based search engines are widely known and used on daily basis by lots of people, they still have a lot of room for improvement. The main problem with keyword-based search tools is high sensitivity to vocabulary because data is not stored in machine-interpretable form. Computers (i.e., computer programs) can retrieve this kind of data, or even perform spell checking and recognize nouns and verbs, but they lack the possibility to correctly interpret it. Computer programs cannot distinguish a context in which data is mentioned. The Semantic Web refers to idea of data that can be meaningfully processed by machines and it is considered that modern WWW will gradually evolve towards it. Semantically defined resources are still very rare but nevertheless a necessity if the use of software agents is ever to reach its full potential. The Semantic Web presents a vision in which knowledge will be organized in conceptual spaces consistent to its meaning, and keyword based searching will be superseded by query answering. Semantic Web is propagated by the World Wide Web Consortium[7]. It should enable users to discover resources available through Internet infrastructure by content, rather than just by tags or keywords. In order to do so, such resources must contain semantic information which will give the content a pre-defined meaning within a concept (ontology).

The idea of semantic reasoning has resulted in a number of languages, or better said, data models. Among these are *RDF*, *RDF Schema* and *OWL*[8]. Information retrieval from *RDF* and *OWL* is done through the various query languages. These languages are often loosely based on the *SQL* syntax, but are performed on a different data model; instead of relational database, the data being queried is written as a graph consisting of *subject-verb-object (SVO)* triples.

1) Ontology

In computer and information science, ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and can be used to define the domain. For example, ontology can contain knowledge about mobile service users, and their mobile devices. An example of ontology used by AMiGO-Mob is shown in Figure 3 [12].

[7] The World Wide Web Consortium: *http://www.w3c.org/*

[8] Semantic Web Stack: *http://www.w3c.tut.fi/talks/2003/0331umedia-on/slide7-0.html*

A profile instance has seven sets of attributes: 1) *user preferences*; 2) *Web browser*; 3) *hardware platform*; 4) *network characteristics*; 5) *push characteristics*; 6) *software platform*; and 7) *Wap characteristics*. User preferences consist of *preferred information type*, *preferred content type*, and *Quality of Service (QoS)*. For example, *preferred information type* can take on three different values: *text*, *audio* or *video*. The same can apply for *preferred content type* and *Quality of Service (QoS)*.

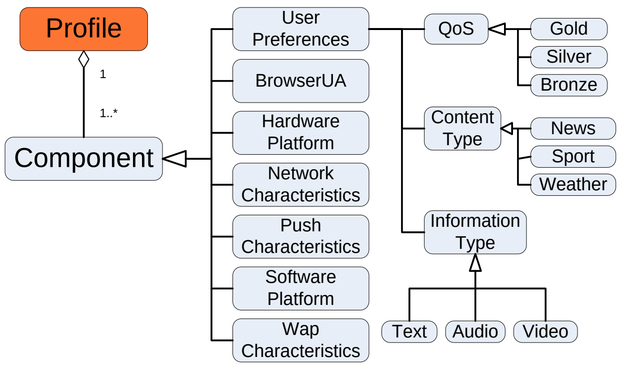


Figure 3. An example of user ontology

2) User Profiles: Related Work

User profiles shown in Figure 3 are based on the *Composite Capabilities/Preferences Profile (CC/PP)* and *User Agent Profile (UAProf)* standards.

The CC/PP[9] is a specification used for creating user terminal profiles. It defines a simple two-level structure consisting of components and associated attributes. The UAProf is a specification that uses CC/PP as a base for creating standard profiles for describing wireless devices.

The CC/PP describes device capabilities and user preferences. It is often referred to as a device's delivery context and may be used as a guide for the adaptation of content presented to the device. Profiles made in RDF describe user agent capabilities and preferences. As the number and variety of devices connected to the Internet grows, there is a corresponding increase in the need to modify the content depending on the capabilities of different devices. Some limited techniques, such as *HTTP (HyperText Transfer Protocol)* `'accept'` headers and *HTML (HyperText Markup Language)* `'alt='` attributes, already exist. As part of a framework for content adaptation and contextualization, it is necessary to create a general purpose profile format that describes the capabilities of a user agent and preferences of its user. CC/PP is designed to be such a format.

3) Semantic Matchmaking

Semantic matchmaking is the process of comparing two objects represented through semantic profiles, resulting with a number within a certain interval. In our case, AMiGO-Mob compares user profiles and rates their similarity with a number between *0* and *1*. Such comparison enables the telco (i.e., mobile service provider) to enhance the provisioning of group-oriented services.

[9] CC/PP specifications: *http://www.w3.org/TR/CCPP-struct-vocab/*

Semantic matchmaking relies on several different types of attributes. Most common attributes used for describing objects can be classified in one of the following types:

- *Interval*: An interval attribute is defined by a continuous linear scale divided into equal intervals (e.g., display resolution, available memory);
- *Ordinal* (or *rank*): An ordinal attribute has multiple states that can be ordered in a meaningful sequence. The distance between two states increases when they are further apart in the sequence and the intervals between these consecutive states might be different (e.g., in Figure 3 *Quality of Service (QoS)* could be qualified as an ordinal attribute with values *Bronze*, *Silver*, and *Gold*);
- *Nominal* (or *categorical*): A nominal attribute takes on multiple states, but these states are not ordered in any way (e.g., in the ontology shown in Figure 3 *preferred content type* is a nominal attribute).
- *Binary*: A binary attribute is a nominal attribute that has only two possible states (e.g., *data transfer type* can be *streaming* or *nonstreaming*).

Apart from mentioned basic types, attributes can also contain references to other objects within ontology. Each attribute type implies a different distance measure. For example, the distance between objects that are described by interval attributes can be described with *Manhattan*[10], *Euclidean*[11], or *Minkowski*[12] distance. Using distance measures we can approximate coordinates for each object in a two-dimensional plane. This procedure is called multidimensional scaling, and can be performed for two or more dimensions.

Table II.
Semantic Matchmaking

User		Service		Score
Attribute	**Value**	**Attribute**	**Value**	
Information Type	*Mp3*	Information Type	*Advanced SystemsFormat*	***0,20***
Information Service	*Madonna*	ID	*Slumdog MillionaireTrailer*	***0,25***
Language	*English*	Language	*English*	***1,00***
Genre	*PopMusic*	Genre	*DramaMovie*	***0,25***
Delivery Type	*NonStreaming*	Delivery Type	*Streaming*	***0,50***
Available Memory	*8000000*	Size	*200000*	***1,00***
Horizontal Resolution	*380*	Horizontal Resolution	*400*	***0,95***
Vertical Resolution	*320*	Vertical Resolution	*300*	***1,00***
			Similarity	***0,644***

[10] *Manhattan distance:* The distance between two points measured along axes at right angles. In a plane with p_1 at (x_1, y_1) and p_2 at (x_2, y_2), it is $|x_1 - x_2| + |y_1 - y_2|$.

[11] *Euclidean distance:* The distance between two points measured along the direct line connecting the two points. In a plane with p_1 at (x_1, y_1) and p_2 at (x_2, y_2), it is $[(x_1 - x_2)^2 + (y_1 - y_2)^2]^{1/2}$.

[12] *Minkowski distance* or *p-norm distance:* For a point $(x_1, x_2, ..., x_n)$ and a point $(y_1, y_2, ..., y_n)$, the *Minkowski distance* of order p is defined as: $\sqrt[p]{\sum_{i=1}^{n}|x_i - y_i|^p}$.

In our implementation we do not use standard distance measures for profile comparison, but novel approach based on semantic matchmaking. Table II shows how user and service profile matchmaking is made. Each attribute in the profile is asserted individually, while the end result is an arithmetic mean of the individual attribute scores. The details about semantic matchmaking procedure follow:

- *Common attribute types*: When comparing binary and nominal attributes the result is either *0* (if the values are not the same), or *1* (if the values are identical). When comparing ordinal attributes the result is a number between *0* and *1*, depending on the rank of each value. Number comparison result depends on preferred value (i.e., if the user's device supports screen resolution that equals or exceeds the one offered by the service provider the score is *1.0*, otherwise the score equals the ratio between the device's and service's resolution);
- *Attributes with object values*: Some attributes' values contain references to other class instances. Figure 4 shows how class hierarchy position is transformed into a real number that represents the similarity between two classes, or objects. Greater distance between two classes or objects should result in decreasing similarity between classes' instances: we can see that the Eurolegaue basketball match *MaccabiVsCibona* and the Premiership football match *ManUtdVsManCity* instances are separated by seven steps in the hierarchy. The similarity score is calculated by division of 1 with the number of steps (in this case 7), therefore we get the similarity score of 0,143. On the other hand, the Premiership football match *ManUtdVsManCity* and the FA Cup football match *ArsenalVsManUtd* instances are separated by four steps in the hierarchy and their similarity score is 0,25. If the same football matches were played in the same competition (e.g., Premiership), instances would be separated by only two steps in the hierarchy and their similarity score would be 0,5.

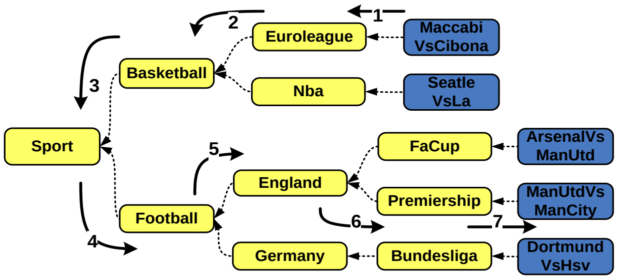


Figure 4. Class hierarchy

B. *Software Agents*

Software agents [15][16][17] are programs that perform complex information and communication actions over the *Web* for their users or owners. They reduce the time necessary to carry out certain personal or business tasks thus enhancing work efficiency [18].

Agent communication has to be regulated by a certain set of rules in order to effectively conduct a conversation. This necessity compelled *Foundation for Intelligent*

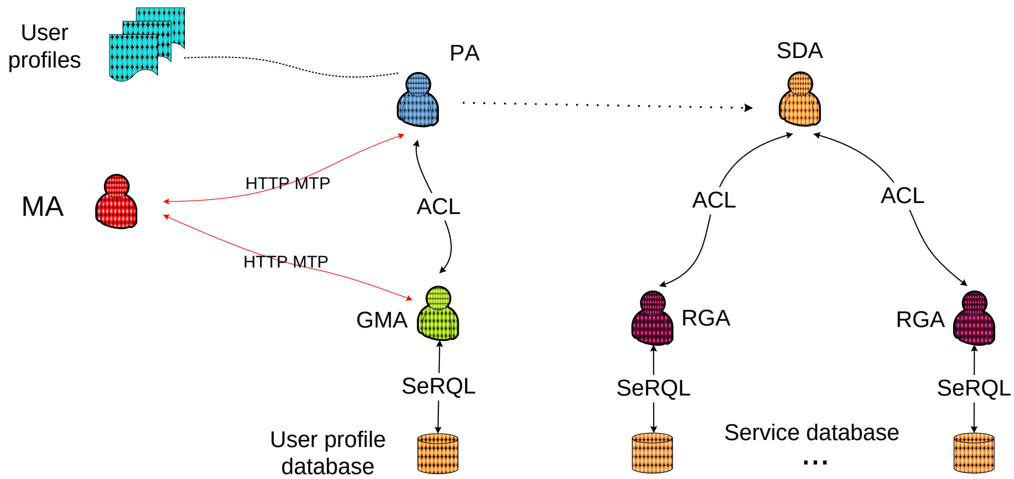


Figure 5. JADE part of AMiGO-Mob middleware

Physical Agents (FIPA)[13] to develop *Agent Communication Language (ACL)*[14]. The ACL defines universal message format and provides protocols for a great variety of possible situations.

Java Agent Development Framework (JADE)[15] is a software development framework for developing MAS (Multi-Agent System) conforming to FIPA standards for intelligent agents. It has been fully coded in Java and an agent developer, in order to exploit the framework, should code his/her agents in *Java* programming language.

JADE-LEAP (*JADE-Lightweight Extensible Agent Platform*)[16] is a modified version of the JADE platform that enables FIPA agents to execute on lightweight devices such as mobile phones supporting Java and Microsoft .Net Framework [19]. JADE containers and JADE-LEAP containers cannot be mixed within a single platform, but they can communicate as specified by FIPA (i.e., via HTTP).

In general the *JADE-LEAP* runtime environment can be executed in two different modes:

- Normal "*stand-alone*" execution mode where the complete container is executed on the device/host where the JADE runtime is activated, and;
- "*Split*" execution mode where the container is split into *FrontEnd* (actually running on the device/host where the JADE runtime is activated) and *BackEnd* (running on a remote server), both linked together through a permanent connection.

The AMiGO-Mob middleware uses the "*split*" execution mode for its lightweight nature. It is particularly well suited for resource-constrained and wireless devices taking into account the following:

- The *FrontEnd* is definitely more lightweight than a complete container;
- The bootstrap phase is much faster because all communication with the *Main container* required to join the platform are performed by the *BackEnd* and therefore are not carried out over the wireless link, and;
- Wireless links have different characteristics with respect to fixed network, such as high latency, low bandwidth, intermittent connectivity, and dynamic IP address assignment that must be taken into account properly.

IV. Multi-Agent System Implementing AMiGO-Mob Middleware

The AMiGO-Mob middleware consists of two basic parts. The first part includes the ontologies and profiles (user profiles and service profiles) stored in a semantic repository and JADE platform containing agents that manage the profiles and perform semantic matchmaking and group formation (see Figure 5). The second part is based on JADE-LEAP platform: it includes agents that represent specific users and a single agent that manages their requests (see Figure 6). These two platforms communicate using *HTTP Message Transfer Protocol (MTP)*.

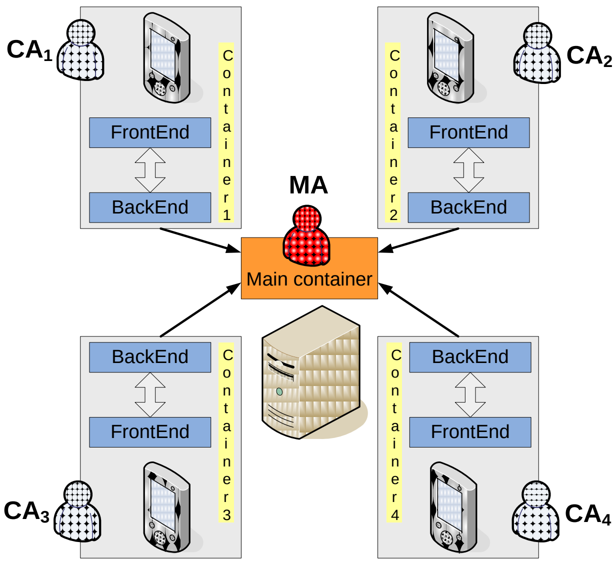


Figure 6. JADE-LEAP part of AMiGO-Mob middleware

[13] Foundation for Intelligent Physical Agents: *http://www.fipa.org*
[14] ACL specification: *http://www.fipa.org/specs/fipa00061/XC00061E.html*
[15] JADE: *http://jade.tilab.com/*
[16] JADE LEAP: *http://www.iro.umontreal.ca/~dift6802/jade/doc/*

A. *Profile Management*

The part of AMiGO-Mob middleware in charge of profile management is JADE MAS consisting of four different types of agents (see Figure 5). Each user has its own *Preference Agent (PA)* that manages its profile and communicates with the *Group Manager Agent (GMA).*

The GMA receives user profiles from PAs and stores them into a semantic database. Additionally, the GMA implements the algorithm for user group formation. *Service Discovery Agent (SDA)* keeps track of all available mobile services and provides the list on demand. *Repository Gateway Agent (RGA)* is an interface between a semantic repository with service profiles and the SDA.

B. *User Request Management*

The part of AMiGO-Mob middleware in charge of user request management is built on *JADE-LEAP* platform. There are two types of agents: a single *Manager Agent (MA)* in the main container, and *Client Agents (CAs)*, one for each registered user, and each in its own split container (see Figure 6). These containers are split into *FrontEnd* and *BackEnd*, as was mentioned before.

The MA communicates with the CAs using ACL, receives their requests, proposes new services, and keeps track of active and inactive agents. When one of the users requests certain (group-oriented) service the MA starts to communicate with the GMA (see Figure 5) which provides the information about the target group. The MA proposes the service to all the other users from the provided group.

The scenario of user request management is the following. At the very beginning, the MA and its main container are started. The next step is deploying the CAs, each in its own split container. After the CAs have been started, they send an ACL *"subscription"* message to the MA that keeps records of all *"living"* and *"non-living"* agents. After that, the whole middleware is on hold until one of the clients requires the certain (group-oriented) service. When such request occurs, the belonging CA_{req} sends an ACL *"request"* message to the MA. Afterwards, MA sends the CA_{req} an ACL *"confirm"* message, thus acknowledging its request.

V. Proof-of-Concept Service: Collaborative Download

Our proof-of-concept service demonstrates one possible usage of the AMiGO-Mob middleware. The following scenario should provide a more detailed insight into a service named *Collaborative Download*. The idea of this service is to identify a set of *n* mobile users which are physically near each other and they are interested in the same multimedia resource (e.g., watching movie trailer while waiting in the cinema queue). This is achieved by filtering users in approximately the same physical location (e.g., *Location* and *CoordinatesX/CoordinatesY* attributes in user profiles shown in Figure 2) and afterwards clustering those users into groups of similar interest. Clustering is achieved using following actions:

- calculating similarities between the service profile and each user profile within the required distance (see Table II);
- ranking users according to semantic similarities in a descending order (see Table III);
- possible candidates are determined by setting a similarity threshold (i.e., 60% of the highest value);
- enquiring users about their interest in the service (as explained later on);
- forming a group $G_i = (S_i, u_1, u_2, \ldots, u_n)$, with S_i being the service, and u_j users reaching the threshold and having interest in the service.

Table III.
User Ranking

Rank	User	Similarity	Status
1	$User_1$	0,754	☑
2	$User_2$	0,650	☑
4	$User_3$	0,480	☑
3	$User_4$	0,523	☑
-	$User_5$	0,315	X
-	$User_6$	0,285	X
-	$User_7$	0,332	X
-	$User_8$	0,402	X

Afterwards, each of *n* mobile users can download just a part of the movie trailer file via mobile network (e.g., GPRS or UMTS) and share it with other (*n-1*) mobile users in an *ad hoc* network via Bluetooth (see Figure 7). In such a manner all the mobile users can get the whole movie trailer, but save battery and money.

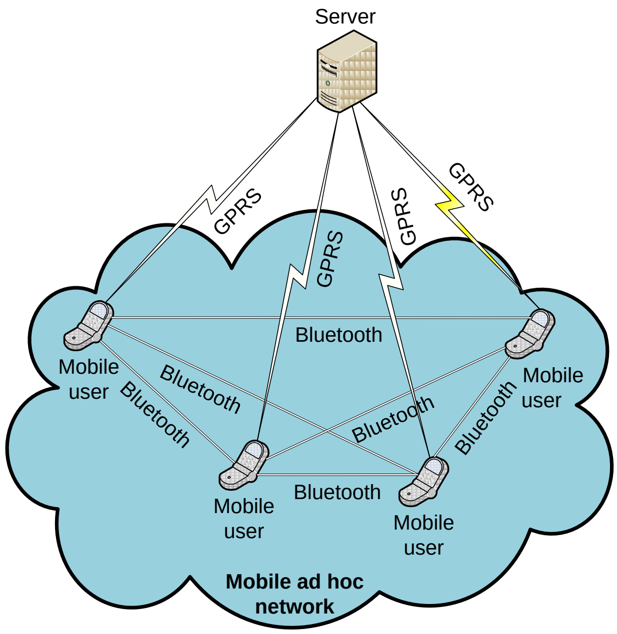


Figure 7. System architecture

A. *Scenario*

Supposing $user_2$ wants to see "*Slumdog Millionaire - trailer"* in a streaming mode (see service profile in Table I), the CA_2 sends an ACL *"request"* message to the MA containing its IMEI as its identifier and service's URI as its identifier (i.e., *http://www.tel.fer.hr/astorm/service.owl# SlumdogMillionaire-trailer*). The MA sends an ACL *"confirm"* message to CA_2, confirming the reception of the request, and then contacts the GMA in JADE platform via HTTP MTP. The HTTP request contains the client's IMEI and service's URI. Meanwhile, the PAs had sent their requests towards the GMA and the user profiles have been stored in the semantic repository. After having performed the location filtering and clustering, the GMA returns IMEIs of users in the same group as the requesting $user_2$, for it is most likely they would share its interest in service at hand. In this case those are $user_1$, $user_3$ and $user_4$ (see Table III). Figure 8 shows only messages between the MA and CAs.

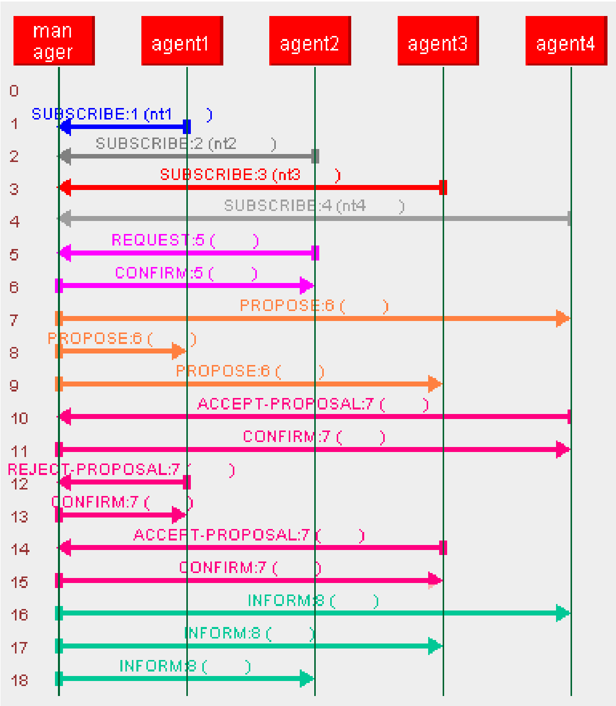


Figure 8. ACL message exchange

The MA sends ACL *"propose"* messages to all members of group that GMA has returned. It is then blocked until it gets all of the answers or *timeout* occurs. It is very important to set a timeout to prevent a deadlock.

Each CA can accept the proposition by sending back an ACL *"accept_proposal"* message, or refuse it by sending ACL *"reject_proposal"*. All messages MA did not receive before timeout occurred, are considered as *"reject_proposal"* messages. The MA then sends an ACL *"inform"* message to those CAs which confirmed their interest in the movie.

After that the download can be started. The MA sends to each CA which part it will download. After initial downloading the MA coordinates which CA will send its part to another CA via Bluetooth and also coordinates further download over mobile network using the principles of self-organizing algorithms (i.e., SOM algorithm). After getting all parts CA can join them and start playing movie trailer on its mobile device.

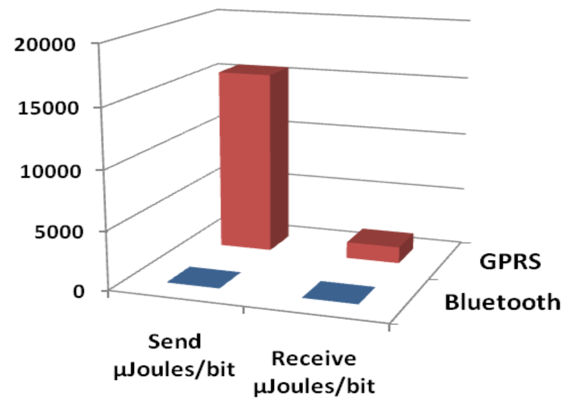
B. Ad hoc Network Architecture: Self-organizing Mechanisms

Our *ad hoc* network consists of mobile phones communicating via Bluetooth. Bluetooth network supports only up to 8 devices in one *piconet*[17], so our ad hoc network has to be reduced to 8 or less mobile devices.

Downloading strategy and coordination in the proposed ad hoc network are defined through a self-organizing algorithm called SOM (Self-Organized Market) [20]. Self-organization is an appropriate concept for building scalable systems consisting of a huge number of subsystems such as *ad hoc* networks. The primary objectives of self-organization are coordination and collaboration on a global level. Our main goal is to find an energy-efficient means for distribution of data parts between mobile users. SOM algorithm is market based algorithm. Each mobile user is simultaneously a buyer and a seller. All of them buy parts they miss, and sell parts they own. Market is controlled through auctions. The buyer is actually an auctioneer that starts an auction not for selling parts, but for buying them. Other mobile users bid with the price at which they are willing to sell their parts. Auctioneer gathers bids from the bidders and selects the lowest bidding user as the winner. Auctions take place in the second phase of SOM, while in the first phase of SOM all parts of the movie must be transferred from the server to mobile devices within the *ad hoc* network.

C. Benefits of AMiGO-Mob Middleware

Bluetooth network is more energy-efficient than GPRS network. The communication resources are characterized by the average amount of energy (measure in *Joules*) needed to communicate one bit of information. The specific energy characteristics of each network from [21][22] are shown in Graph 1.



Graph 1. Energy demand for mobile devices

Energy efficiency aside, Bluetooth network is virtually free of charge for mobile users unlike GPRS network. Also, transferring communication from the GPRS network to Bluetooth network releases GPRS channels which can then be put to further use by other users, thus decreasing the service cost both for telcos and mobile users.

Managing detailed user information raises certain privacy issues. However, user information is consciously provided to telcos by mobile users themselves because users' interest is to gain access to available services. Telcos encourage their users to provide as detailed profile as possible in order to receive a highly personalized treatment, but in no manner obligate their users to provide such data.

17 *http://developers.sun.com/mobility/midp/articles/bluetooth1/*

The reduction of the large number of users to a single piconet greatly simplifies the coordination. Moreover, this fact also makes the proposed middleware a scalable solution, along with the fact that the time necessary to include or eliminate a user from the potential user pool is less than 1/10 of a second.

VI. Conclusion and Future Work

The paper describes group-oriented service provisioning based on user preferences, mobile device capabilities and communication context. Mobile users possess mobile devices (e.g., mobile phone or PDA). They are firstly filtered out by location and their aspirations, while afterwards a group-oriented service is delivered to the group of users that reach the certain criteria.

Everything described is enabled by implementation of the AMiGO-Mob middleware. It is based on three-layered middleware architecture and grounded on Semantic Web and software agent technologies. A proof-of-concept group-oriented service enabled by AMiGO-Mob is *Collaborative download* service. Firstly, its implementation is explained, while afterwards it is used in scenario for delivering movie trailers to the users standing in the cinema queue.

Measuring performance of different delivering strategies and improving semantic matchmaking are the areas of the future work.

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

[1] M. Basuga, R. Belavic, A. Slipcevic, V. Podobnik, A. Petric, and I. Lovrek, “The MAgNet: Agent-based Middleware Enabling Social Networking for Mobile Users”, *Proceedings of the 10th International Conference on Telecommunications (ConTEL’09)*, Zagreb, Croatia, 2009, in press.

[2] C. Feijoo, J. Fernandez-Beaumont, J. L. Gomez-Barroso, A. Angeles Marin, and D. Rojo-Alonso, “The emergence of IP interactive multimedia services and the evolution of the traditional audiovisual public service regulatory approach”, *Telematics and Informatics*, vol. 24, no. 4, pp. 272-284, 2007.

[3] L. Vrdoljak, “Group-Oriented Service Provisioning”, *Proceedings of the Summer Camp 2008 (Ericsson Nikola Tesla & Faculty of Electrical Engineering and Computing, Zagreb, Croatia, 2008)*, pp. 152-187.

[4] G. A. Grimnes, P. Edwards, and A. Preece, “Instance based clustering of semantic web resources“, *Lecture Notes in Computer Science*, vol. 5021, pp. 303-317, 2008.

[5] N. Fanizzi, C. d'Amato, and F. Esposito: “Conceptual Clustering and Its Application to Concept Drift and Novelty Detection”, *Lecture Notes in Computer Science*, vol. 5021, pp. 318-332, 2008.

[6] S. Dietze, A. Gugliotta, and J. Domingue, “Conceptual Situation Spaces for Semantic Situation-Driven Processes”, *Lecture Notes in Computer Science*, vol. 5021, pp. 599-613, 2008.

[7] J. Bonnin, I. Lassoued, and Z. B. Hamouda, “Automatic multi-interface management through profile handling”, *Mobile Networks and Applications*, vol. 14, no. 1, pp. 4-17, 2009.

[8] C. Panayiotou and G. Samaras, “mPERSONA: personalized portals for the wireless user: An agent approach”, *Mobile Networks and Applications*, vol. 9, no. 6, pp. 663-677, 2004.

[9] F. Frkovic, V. Podobnik, K. Trzec, and G. Jezic, “Agent-Based User Personalization Using Context-Aware Semantic Reasoning”, *Lecture Notes in Computer Science*, vol. 5177, pp. 166-173, 2008.

[10] V. Podobnik and I. Lovrek, “Multi-Agent System for Automation of B2C Processes in the Future Internet”, *Proceedings of 27th IEEE Conference on Computer Communications (INFOCOM’08)* Workshops, Phoenix, USA, 2008, pp. 1-4.

[11] V. Podobnik, G. Jezic, and K. Trzec, “Towards New Generation of Mobile Communications: Discovery of Ubiquitous Resources”, *Electrotechnical Review*, vol. 75, no. 1-2, pp. 31-36, 2008.

[12] V. Podobnik, A. Petric, K. Trzec, and G. Jezic, “Software Agents in New Generation Networks: Towards the Automation of Telecom Processes”, in *Knowledge Processing and Decision Making in Agent-Based Systems*, L. C. Jain and N. T. Nguyen, Eds. Berlin Heidelberg: Springer-Verlag, 2009, pp. 71-99.

[13] F. Frkovic, “Agentski sustav za pronalaženje telekomunikacijskih usluga primjenom semantičkog rasuđivanja”, Thesis, University of Zagreb Faculty of Electrical Engineering and Computing, Zagreb, Croatia, 2008, in Croatian.

[14] J. J. Jung and J. Euzenat, “Towards Semantic Social Networks”, *Lecture Notes In Computer Science*, vol. 4519, pp. 267-280, 2007.

[15] C. E. Georgakarakou and A. A. Economides, “Software agents: an overview”, *Knowledge Engineering Review*, vol. 11, no. 3, pp.1-40, 1996.

[16] M. Fasli, *Agent technology for e-Commerce*. Glasgow: John Wiley & Sons, 2007.

[17] K. Jurasovic and M. Kusek, “Verification of Mobile Agent Network Simulator”, *Lecture Notes In Computer Science*, vol. 4496, pp. 520-529, 2007.

[18] V. Podobnik, A. Petric, and G. Jezic, “An Agent-Based Solution for Dynamic Supply Chain Management”, *Journal of Universal Computer Science*, vol. 14, no. 7, pp. 1080-1104, 2008.

[19] G. Caire and F. Pieri, *LEAP User Guide*. TILAB, 2006.

[20] I. Bojic and T. Lipic, “SOM: algoritam samoorganizacije u mobilnim ad hoc mrezama”, *Proceedings of 18th International Electrical Engineering Symposium (EIS’09)*, Sibenik, Croatia, 2009, in Croatian, in press.

[21] J. Fryman, C. Huneycutt, H.-H. Lee, K. Mackenzie, and D. Schimmel, “Energy Efficient Network Memory for Ubiquitous Devices”, *IEEE Micro*, vol. 23, no. 5, pp. 60-70, 2003.

[22] K. J. O'Hara, R. Nathuji, H. Raj, K. Schwan, and T. R. Balch, “AutoPower: Toward Energy-aware Software Systems for Distributed Mobile Robots”, *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, USA, 2006, pp. 2757-2762.