UDC:004.8
Original scientific paper

DATA ANALYSIS FOR AGENT BASED MOBILE SERVICES[1]

Marek Paralič
Technical University of Košice,
Faculty of Electrical Engineering and Informatics, Košice
Marek.Paralic@tuke.sk

Ján Paralič
Technical University of Košice,
Faculty of Electrical Engineering and Informatics, Košice
Jan.Paralic@tuke.sk

Abstract: *One of the high interesting areas in the distributed systems is the problem of building and running pervasive computing services for mobile users. The central point for building such a services is to appropriately model the physical and data environment. In this paper we concentrate on the data part of such an environment that would enable creating a general scheme for a category of flexible services for mobile users. We define the basic profile of the users and methods, how the services should deal with the profiles. For modeling the presence of the user in the data environment (even if he/she is off-line) the agent-based solution was chosen, so the distributed system is build as a multi-agent system.*

Keywords: *mobile services, agent based, multi-agent system, data analyses, context sensitive services.*

1. INTRODUCTION

One of the high interesting areas in the distributed systems is the problem of building and running pervasive computing services. Pervasive computing is an important research area whose challenges require a thorough rethinking and revision of conventional software design ideas. In reality, there is little consensus and very little basic understanding of the underlying issues and their interactions to produce useful solutions. The goal of our research is thus to contribute to this understanding. One of the perspective technologies in this area is agent-based solution, where the distributed system is build as a multi-agent system.

The term multi-agent system is now used for all types of systems composed of multiple autonomous components showing the following characteristics [12]:

- each agent has incomplete capabilities to solve a problem,
- there is no global system control,

[1] The described work is done within the projects Nr. 1/0176/03 and Nr. 1/1060/04 of the Slovak Grant Agency

- data is decentralized and
- computation is asynchronous.

Multi-agent systems (MAS) can become a fundamental enabling technology, especially in conditions where mutual interdependencies, dynamic environments, uncertainty and sophisticated control play a role. In the last years a lot of attention has been paid to the agent technology in the research. Despite many of relevant results, multi-agent systems have not become widespread as industrial and commercial applications. In order to bridging the gap between agent technology and methodologies or technologies accepted for real world applications some efforts have been done. A survey of agent-oriented methodologies can be found in [11].

The increasing number of mobile devices with some computational and communication power offers new possibilities for design and implementation of distributed services. Not only the changeable physical location can influence the architecture of the service but also the way, how the basic environment of the mobile user is modeled. The key idea behind our research is to utilize advantages of agent representatives of mobile users (i.e. users with mobile devices) in data environment to achieve context sensitiveness for proposed services. The paper is structured as follows. In section 2, the requirements for flexible mobile services are analyzed. In section 3 after short introduction of the Gaia methodology the architecture of the generic service is described. Section 4 briefly describes the MASF system – an implementation of the proposed architecture. Section 5 analyses the possibilities for extending proposed users profile to achieve advanced knowledge based mobile services. Section 6 summarizes the paper and sketches our future research in this area.

2. REQUIREMENTS FOR THE FLEXIBLE MOBILE SERVICES

First of all we want to specify how we understand and use the notions of flexibility and mobility. From one point of view these category of services are intended for mobile user so the services should be mobile by the usage. On the other hand the manner how to build such a service should be flexible enough to stay available even if only part of the system is accessible. And further, if we want to mark a service as flexible, the process of offering this service must somehow include the user of the service.

During the process of analysing the requirements for flexible mobile services the following were identified:

- high availability by support of accessibility from mobile devices,
- service should be distributed and organized as a set of independent communication nodes to enable the possibility of maintenance by the owner of the data (user) instead of the service,
- universal rules should be applied to the process of definition and manipulation with user data in order to create a scheme for whole category of services,
- automation of the routine tasks by manipulation with the user data to enable not only trivial tasks from mobile devices with restricted resources and accessibility,
- support of communities as “virtual environment” for offering and sharing the services.

To fulfil the requirement of organizing the service as a set of independent communication nodes, where the user in the role of node’s owner has control over the supported data, we proposed to maintain so called “user profile”. We use this notion to

identify input data for a category of services, which process this data to fulfil their task. Beside the aim of offering needed information for specific services, the profile should include information for building "data environment" for mobile user; therefore it has to contain some identification and social information. The created data environment supposes that information in the user profile is readable and understandable not only by human being, but also by computer processes. This problem is strongly related to effort by putting the semantics to the World Wide Web, what we also utilised by the testing implementation of the proposed kind of services.

From the content point of view, as a starting point for the generic service we identified the following items to be stored in the profile:

- identification information: name, e-mail (as an unique identifier), photography;
- additional information: telephone number, affiliation, publications, projects...
- social connections: list of known persons;
- other information (service specific): calendar, ownership relations...

From the architectural point of view, the service with proposed features should be decentralized and distributed. Modelling of mobile user presence in the system and at the same time offering the public information about the user even if he/she is off-line (to keep the social network), seems to be too strong requirement for current most popular distributed architectures – client/server or peer-to-peer. Therefore we decided to use the agent paradigm and model the system as a multi-agent application with static agents. The analyses and design of an architecture fulfilling the mentioned requirements (using the Gaia methodology) is briefly described in the next section.

3. THE ARCHITECTURE OF THE FLEXIBLE MOBILE SERVICE

Multi-agent methodologies try to overcome the fact that representing an agent as an object (i.e. as a set of attributes and methods) is not very useful because the representation is too fine-grained, operating at an inappropriate level of abstraction [1]. Agent paradigm is based on a significantly stronger notion of encapsulation than the object paradigm. Object model captures associations between object classes that model largely static dependencies and paths of accessibility, which are irrelevant in a multi-agent systems.

For design of our multi-agent system we choose the Gaia methodology [16] that proposes to break the design process into an *analyses phase* and a *design phase*. In both stages, designers are required to deal with models (roles model and interaction model for the analyses stage; agent model, services model and acquaintance model for the design stage). This way, designers and developers are progressively guided through specifying more and more details about the application and the system, being encouraged to follow a process based on organizational design.

In the first, *analysis phase*, we identified basic roles and interactions. The following 4 *basic roles* have been identified: Display (Table 1), UserRepresentative (Table 2), Service (Table 3) and ProfileManager (Table 4). Each of the identified roles is described using the role schema from the Gaia methodology.

Table 1: Scheme role: Display

Role Schema: `Display`
Description: Enables the contact between the user and the system. Displays the input requirements for the services and their results to the user. It is located on the mobile device.
Protocols and Activities: RequestMainPage, CallService, DisplayForm, RenderForm
Permissions: changes *form* (for the communication with the services)
Responsibilities: *Liveness*: Display = DisplayMainPage.(ServiceInteraction | DisplayMainPage)$^{\omega}$ DisplayMainPage = RequestMainPage.DisplayForm.RenderForm ServiceInteraction = CallService.DisplayForm.RenderForm *safety*: Communicates only with the responsible UserRepresentative.

Table 2: Scheme role: UserRepresentative

Role Schema: `UserRepresentative`
Description: Represents the user in the system, stores his/her messages and service results if the user is off-line.
Protocols and Activities: RequestMainPage, DisplayForm, CallService, RequestForm, PrepareMainPage, PrefillForm
Permissions: reads *requestForm*, changes *responseForm*, creates *mainPage*
Responsibilities: *Liveness*: UserRepresentative = (DisplayMainPage | ServiceRequest | ServiceResponse)$^{\omega}$ DisplayMainPage = RequestMainPage.PrepareMainPage.DisplayForm ServiceRequest = CallService.RequestForm ServiceResponse = ResponseForm.[PrefillForm].DisplayForm *safety*: Displays forms only to the responsible Display-role.

Table 3 Scheme role: Service

Role Schema: `Service`
Description: Service communicates with the environment using the input and output forms and as data source the user profile is used. **Protocols and Activities**: RequestForm, ResponseForm PerformService **Permissions**: reads *requestForm*, reads *profile*, creates *responseForm* **Responsibilities**: *Liveness*: Service= (RequestForm.[GetProfile*].PerformService. ResponseForm)$^{\omega}$ GetProfile = RequestForm. ResponseForm *safety*: true

Table 4 Scheme role: ProfileManager

Role Schema: `ProfileManager`
Description: Maintains the user profile and offers it to the services.
Protocols and Activities: ShowMenu, GetProfile, EditProfile, ViewProfile, AddKnown
Permissions: changes *profile*
Responsibilities: *Liveness*: ProfileManager = (ShowMenu | GetProfile | EditProfile | ViewProfile | AddKnown)$^{\omega}$ *safety*: Changes to the profiles are enabled only to its owner.

The *interaction model* describes in details four basic types of interaction between particular roles defined above. The following interaction types have been identified.

- Request for main communication screen with the system. This is the initialisation event when starting the client, as well as standard return screen after finishing of a service.
- Call of a service with particular parameters. As a result a form is returned to the user's display screen.

- User representative is asking for a form (empty or filled). As a result, service returns requested form (filled in case it is result of the service).
- Service asks for access to personal profile.

During the *design phase*, we firstly designed *agent model* consisting of 4 agent types (Figure 1). The *Display* agent type corresponds to the `Display` basic role identified in the analysis phase. *UserRepresentative* agent type is an aggregation of `UserRepresentative`, `Service` and `ProfileManager`. Last two agent types have been designed for the purpose of two simple applications. *Librarian* agent is aimed for supporting a service for books borrowing. *TimeCoordinator* agent realises the service for time planning of meetings.

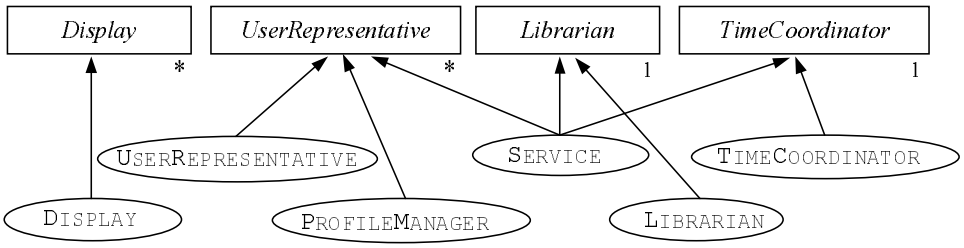

Figure 1: Proposed agent model.

Within the *service model*, for each type of agent defined above, a number of various basic services that will be provided by it, have been identified and described by means of standard UML attributes as name, input, output, pre- and post-conditions.
Finally, acquaintance model is quite simple (Figure 2). Each *Display* agent communicates only with its particular *UserRepresentative* agent. Only *UserRepresentative* agents are communicating with particular service providing agents, like e.g. *Librarian* or *TimeCoordinator* types of agents in our system.

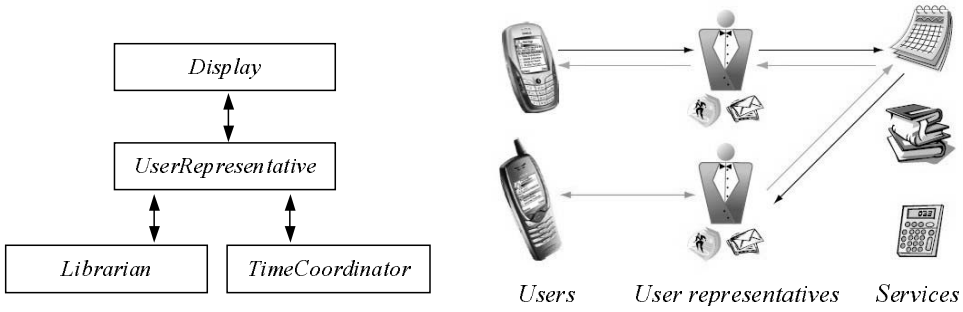

Figure 2: Acquaintance model (left) and architecture of the flexible mobile services system

4. MOBILE AGENT SERVICE FRAMEWORK

The architecture for flexible mobile services introduced in previous section was verified on an implementation called ***Mobile Agent Service Framework*** (***MASF***). The overall architecture of the designed system is presented in Figure 2. MASF has been implemented by means of the JADE (Java Agent DEvelopment Framework) [2] that is implemented in

Java. The JADE platform provides also a LEAP (Lightweight Extensible Agent Platform) [5] add-on module that can be used on a rich number of end devices such as PDA (Personal Digital Assistant) or mobile phone.

Detailed description of the chosen implementation technologies as well as the implementation itself can be found in [9] and [15]. On this place we will briefly mention the implementation technologies. XForms [7] W3C standard based on XML technology is used for communication purposes. User profile is defined by means of RDF, making use of the FOAF (Friend Of A Friend) dictionary [3]. Jena library [10] is used for processing of RDF models. For additional information presented in the user profile because of the special services also standardised tools are used. E.g. Dublin Core RDF syntax [8] is used for description of literature sources (books) to support the *Librarian* service, or iCal [6] (RDF standard for representation of calendar events) and RDF Geo [4] (for storing of information about the physical location of a person) is used for the *TimeCoordinator* service.

6. DATA ANALYSES FOR ADVANCED SERVICES

Presented MASF system is sufficiently general to incorporate new types of services based on existing or possible extended user profile. One example could be a kind of electronic marketplace, where specific goods or skills are offered and/or asked for. In case of borrowing (like above mentioned Librarian service example), no additional requirements are needed. In case of selling/buying scheme, some payment methods must be included. But in both cases, the *UserRepresentative* agent could perform some work on behalf of its owner - particular user - when negotiating, e.g. the final price or conditions of a transaction.

In case when particular user profiles could be collected on one place, more sophisticated data analyses techniques could be applied, e.g. by means of knowledge discovery process [14]. For example, clusters of users with similar interests may be identified [13], which could lead into suggestions for a particular user to contact somebody from the other community that is interested in similar things that given person. By means of association link analysis suitable contact person could be suggested.

Other possibilities would be to provide different view on user profile. A user may namely play different roles, e.g. having different professional and leisure interests. One could share part of her/his profile with one group of people (e.g. professionals in one domain area) and other part of the profile with different group of people (e.g. a kind of hobby) etc. Interest areas would be defined by means of ontology (with e.g. hierarchical structure of concepts). On one hand side each concept various types of resources may be associated (electronic or physical document, books, audio or video records, photos etc.). On the other side, to each concept a group of friend could be associated (e.g. if a person is associated as a friend to a concept, he/she could access all the resources associated to this particular concept, as well as all to underlying concepts. Based on the type of resource, only on some of the device types may be accessed. In case of the interest to access a particular resource, suitable way of accessing would be suggested by the electronic service. E.g. in case of a small audio record, or a photo suitable for viewing on a mobile phone screen, this would be directly downloaded and presented. In case of larger electronic document (which would cause unwanted increase of costs when downloading it to the mobile device), it could e.g. sent by e-mail, or prepared for presentation on other device, when user will access the service later on using suitable device. In case of physical resources (as books etc.), services like *Librarian* mentioned above, may be suggested and used.

7. CONCLUSION

In this paper architecture of the flexible mobile service as multi-agent system by means of Gaia methodology was introduced. Proposed architecture was implemented as Mobile Agent Service Framework (MASF) build on the JADE platform with LEAP add-on module that can be used on a rich number of end devices such as PDA or mobile phones. In such a way a general scheme for a category of flexible services for mobile users has been created, whereby as pilot applications two simple services were used - borrowing books and meeting arrangement. Moreover, some future possible extensions have been sketched. These may be achieved either by collecting the user profiles and analyzing them centrally (knowledge discovery approach), or to extend the profile by providing different views on a user (as playing different roles in life) and generalize access to user's resources by flexible services taking into account area of interest as well as available device type.

ACKNOWLEDGEMENTS

This work is done within the VEGA projects 1/0176/03 "Technological platform for creating multi-agent applications with hierarchical structure of mobile and static agents" and 1/1060/04 "Document classification and annotation for the Semantic web" of Scientific Grant Agency of Ministry of Education of the Slovak Republic.

REFERENCES

[1] Bauer, B.; Mueller, J.P.; Odell, J. (2001): *Agent UML: A Formalism for Specifying Multiagent Interaction*. Agent-Oriented Software Engineering, P. Ciancarini and M. Wooldridge eds., Springer-Verlag, pp. 91-103

[2] Bellifemine, F.; Poggi, A.; Rimassa, G. (1999): *JADE – A FIPA-compliant agent framework*. Proc. of PAAM'99, London.

[3] Brickley, D.; Miller, L. (2004): *FOAF Vocabulary Specification*. Mai 2004, available on-line: http://xmlns.com/foaf/0.1/

[4] Brickley, D. (2003): *An RDF Geo Vocabuary*. February 2003, available on-line: http://www.w3.org/2003/01/geo/

[5] Clair, G. (2003): LEAP User Guide 3.1. December 2003, available on-line: http://sharon.cselt.it/projects/jade/doc/LEAPUserGuide.pdf

[6] Connolly, D.; Miller, L. (2002): RDF Calendar Workspace. December 2002, available on-line: http://www.w3.org/2002/12/cal/

[7] Dubinko, M.; Klotz, L.L.; Merrick, R.; Raman, T.V. (2003): *XForms 1.0 W3C Recommendation*. October 2003, available on-line: http://www.w3.org/TR/2003/REC-xforms-20031014/

[8] Dublin Core Metadata Initiative (2003): *Dublin Core Metadata Element Set, Version 1.1, Reference Description*. June 2003, available on-line: http://dublincore.org/documents/dces/

[9] Hajduch, Tomáš (2004): *Agent based mobile services*. Diploma thesis, Dept. of Computers and Informatics, FEI TU of Košice, p. 78

[10] HP Lab Semantic Web Programme: *Jena Documentation*. Available on-line: http://jena.sourceforge.net/documentation.html

[11] Iglesias, C.A.; Garijo, M.; Gonzales, J.C (1998): *A Survey of Agent-Oriented Methodologies*. In Proceedings of the 5th International Workshop on Agent Theories, Architectures and Languages (ATAL'98), Paris, pp. 185-198

[12] Jennings, N.R.; Sycara, K.; Wooldridge, M. (1998): *A Roadmap of Agent Research and Development*. In: Journal of Autonomous Agents and Multi-Agent Systems, Vol. 1(1), pp. 7-38

[13] Machová, K., Klimko, I. (2004): Support of the adaptive WEB by means of machine learning. Proc. of the conf. Znalosti 2004, Brno, Czech republic, 2004, VSB-Technical University Ostrava, 218-225

[14] Paralič, J. – Bednár, P.: A Tool to Support of the KDD Process. In *Journal of information and organizational sciences*, Varaždin, Croatia, Vol. 22, No. 3-4

[15] Vaškovič, Pavol (2004): *Agent based mobile services – User Profile Management*. Diploma thesis, Dept. of Computers and Informatics, FEI TU of Košice, p. 87

[16] Wooldridge, M.; Jennings, N.R.; Kinny, D. (2000): *The Gaia Methodology for Agent-Oriented Analysis and Design*. Journal of Autonomous Agents and Multi-Agent Systems, Vol. 3(3), pp. 285-312