Classification of Agents-based Mobile Assistants

Fernando Koch[1] and Iyad Rahwan[2]

[1] Institute of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
fkoch@acm.org
[2] Department of Information Systems
University of Melbourne, Parkville, VIC, Australia
i.rahwan@pgrad.unimelb.edu.au

Abstract. We argue that agent-based approaches offer promising techniques for dealing with the challenges of service delivery to mobile users. We present a classification of the types of services agents can offer as mobile assistants. This serves both as a guide for analysing existing systems, and as a means for identifying new opportunities. We review a number of existing attempts to build agent-based mobile assistants applications and we place these within our classification.

1 Introduction

Pervasive computing [3], with its focus on users and their tasks rather than on computing devices and technology, provides an attractive vision for the future of computing. With the increase in use of powerful mobile devices it is expected that the market will demand highly personalized and context-aware services that make use of capabilities embedded in these devices [10]. The power of pervasive computing is unleashed when the application has the intelligence to process contextual information about the user and the environment in order to provide the user with the right information at the right time, and to possibly take action on behalf of the user. A framework for building these applications must provide the means to handle the distribution inherent in the environment and at the same time allow the easy mapping of human knowledge into computer applications.

Agent-based computing [5] is seen as a potential enabling technologies for second generation mobile services. The agent paradigm offers methodologies for creating distributed, intelligent, integrated and cooperative applications. No attempt has been done to date, however, to scope the applicability of the agent approach to mobile service provisioning. This paper attempts to provide such scoping.[3] In particular, we present a classification of the types of services agents can offer to mobile users. This serves both as a guide for analysing existing systems and as a mean for identifying new opportunities. We review a number of existing attempts to build agent-based applications for mobile assistants and we place these within our classification.

[3] Note *agent-based mobile services* and *mobile agents* are two different areas. The latter is concerned with software that moves between systems on a network [2].

Section 2 describes the requirements for mobile services and advocates in favour of using agents as an enabling technology for developing these applications. Section 3 presents a methodology for classifying agent-based mobile assistants, based on the application's *granularity*, *role* and *autonomy*. In section 4 we present two hypothetical mobile service scenarios and classify their variants using the proposed taxonomy. Finally, in section 5 we work out existing mobile assistant services into our classification system, then conclude in secion 6.

2 Requirements

Mobile service provision imposes two major challenges: the *infrastructure challenge*, which is concerned with building robust hardware and software technology that facilitate mobile connectivity, location-identification, service discovery, fault tolerance, etc. [4]; second is the *services challenge*, which is concerned with how we can use the infrastructure available in order to provide new and useful services, such as trip planning or and mobile commerce [12]. In this study, our focus is on the latter. In particular, we are interested in high-level *intelligent services*.

Satyanarayanan [12] distinguishes two fundamental missing features in pervasive computing: *proactivity* and *self-tuning* (or *adaptivity* as we shall refer to it). Roughly speaking, proactivity refers to the software's ability to anticipate user's intentions and other external events, and to act accordingly. Adaptation refers to the ability to automatically adjust behaviour to fit the changing circumstances.

Agent-based computing is becoming increasingly popular because it enables building modular software systems capable of operating in dynamic, unpredictable environments. An agent is an autonomous computer system that is situated in an environment, and is capable of flexible autonomous behaviour in order to meet its design objectives [5]. An intelligent agent is an agent that is:

1. *responsive/adaptive*: perceives the environment and responds appropriately;
2. *proactive*: is goal-directed and take initiative when appropriate;
3. *social*: able to interact with other agents or humans when needed.

The agent paradigm has produced a wide variety of concepts and tools for constructing sophisticated autonomous software, reasoning about context [11], and structuring high-level interaction patterns that facilitate cooperative behaviour [5]. In addition, a set of methodologies have been developed that enable system designers to distil domain knowledge and transform it into agent or multi-agent system specifications.

Hence, agents seem to offer a set of features that are very closely aligned with the requirements of service delivery challenge in pervasive computing.[4] Firstly, a mobile user is usually situated in some environment, which can be represented in terms of context information, such as the time, place, and task at hand. Secondly, the environment is dynamic, since users may move from one place to another,

[4] Agent-based methods have also been used to address the 'infrastructure challenge', as in the CRUMPET project [7].

and since their tasks may change based on their circumstances. Hence, agent concepts such as *situatedness* and *adaptivity* seem to offer promising tools of abstraction and computational methods for building software that operates in such environment. The concept of *proactivity* can potentially help build systems that reason about the user's goals and how they may be achieved Finally, in mobile settings, there is a need for applications to interact with other applications representing other users and/or service providers. For example, an agent working on behalf of an on-site engineer may interact with agents representing other engineers to sort out meetings and schedule joint tasks. In this regard, an agent's ability to conduct sophisticated forms of interaction can prove useful.

In summary, there is some coupling between the requirements of a pervasive mobile computing environment and the general features offered by the agent-oriented paradigm to software system development. So far, this coupling has been rather abstract. The next step is to make our argument more concrete by enumerating the services agents can offer.

3 Classifying Agent-based Mobile Assistants

In this section, we present a classification of agent-based services in general, and use this in order to provide a clearer account of what agents can offer for mobile assistance in particular.

There are several different dimensions along which one can analyse agent-based systems. We find the following classification, due to Jennings et al [5], useful for our purposes. Agent systems can be classified according to the *sophistication of the application* (e.g. pre-specified rules vs. predictive/proactive agents), *role of the agent*, and *granularity of the view* (single-agents vs. multi-agent systems). By correlating these classification aspects and their roles in mobile service applications, we came up with a taxonomy based on three key aspects: granularity, role and level of autonomy.

3.1 Granularity

In some applications the significant unit of analysis and design is the individual agent, whereas in other applications it is the society of agents that is key. The decision about whether to adopt a single-agent or multi-agent approach is generally determined by the domain and is similar in nature to decisions about whether monolithic, centralized solutions or distributed, decentralized solutions are appropriate. Single-agent systems are in a sense simpler than multi-agent systems, since they do not require the designer to deal with issues such as cooperation, negotiation, and so on. An agent-based mobile assistant systems can be classified on *granularity* as:

(a) single user support: The agent system is designed as a *single-agent systems* targeting to provide support to one user only;
(b) multi-user interaction support: The agent is able of communicate to other users' agents to coordinate activities, negotiate schedules, etc.

3.2 Role

Sheridan [13] presents a taxonomy of four main stages of complex human-machine tasks: (a) acquiring information, (b) analysing and displaying results, (c) deciding on an action or sequence of actions, and (d) implementing decided actions. A software system can offer varying levels of support at each of these tasks. We now discuss how agent technology can support these tasks:

(a) acquire information : The agent gathers information and saves it for future processing;
(b) analyse and display : The agent analyses collected information and displays it to the user, who then may use this information to aid his/her decisions;
(c) decide action : Rather than presenting raw information, the agent further analyses this information, based on the context and user's preferences, and suggests actions to the user;
(d) implement action : Here, the agent has the ability to carry out the actions required for the completion of some objective;

Note that the above features are not incremental. For example, an application that implements actions on behalf of the user does not necessarily have the ability to decide on that action. It may simply be executing a direct request by the user (e.g. an alarm does not 'know' why the user wants to wake up at the specified time).

3.3 Autonomy

Depending on the level of sophistication of the agent application they behave very basically, reacting to environment stimuli based on a set of pre-determined rules or user commands or act proactively considering the context and user preferences and behaviours. For mobile assistants, agent-based systems can be classified on *autonomy* as:

(a) reactive : These are agents whose actions are triggered by the user or the environment; the majority of current mobile service applications are examples of reactive agents;
(b) proactive : The agent does not simply act in response to their environment, but should be able to exhibit opportunistic, goal-directed behaviour and take the initiative where appropriate;

4 Mobile Computing Scenarios

In this section, we present two hypothetical scenarios of mobile use, and demonstrate how an agent-based approach could add value through the features classified earlier.

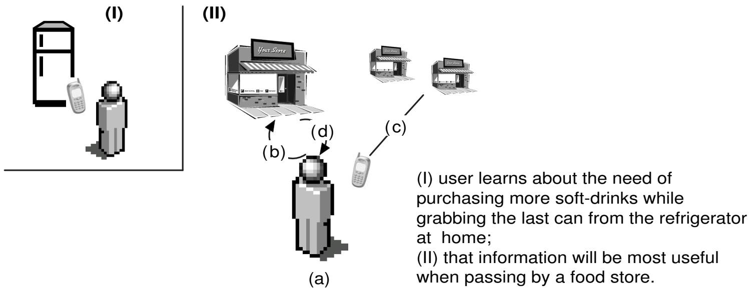

Fig. 1. Mobile-commerce and pervasive computing scenario: shopping assistant

4.1 Scenario 1: Mobile Commerce

The natural target for pervasive computing is data management – devices will accept information from the context it is originated and deliver this information in the context in which it is most useful. For example, the user learns he needs to buy more soft drink when getting the last can from his fridge and this information is added to his shopping list (Figure 1). However, it will be *more useful* to remind the user about his shopping list when he is nearby a supermarket (Figure 1(a)). Typically, this is a *single-user support application* for granularity. For *role* and *autonomy* possible configurations include:

– *acquire information, analyse and display* and *reactive*: in case the agent reacts to the user's command of requesting a quote for the shopping list from the near food store (Figure 1(b)) and once processed presents the quote back to user (Figure1(c));
– *acquire information, analyse and display* and *proactive*: For example, the agent infers the presence of a nearby food store and automatically contact it and collect a quote for user's shopping list; Moreover, the agent checks for quotes from other food stores the user frequents (Figure 1(c)) and presents the price comparison to user;
– *decide action* and *reactive*: The user delegates the act of bargaining to the agent; for instance, the agent collect the quotes from the near store and the remote ones; it also checks for alternative brands that the user configured as acceptable and compiles the best quote from every store, based on user preferences;
– *decide action* and *proactive*: The agent starts to bargain by itself when it detects the food store's proximity and infers from the user's schedule that there is enough time to stop by for some shopping (it wouldn't make sense to spend processing power and communication if the user has an important meeting in 15 minutes); The agent could also negotiate for the best deal;

A multi-agent variant of the last example is one where a *merchant* agent detects the user's proximity and provides a discount coupon valid for a period of time, teasing this customer to step into the supermarket.

4.2 Scenario 2: Organising Lunch

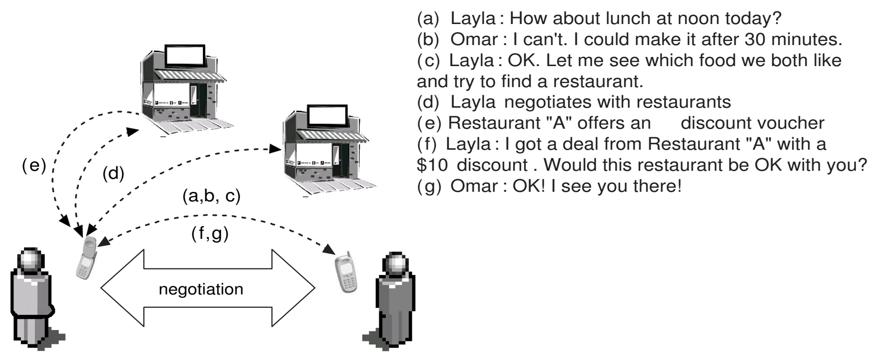

Fig. 2. Multi-user negotiation scenario: organising a lunch

Let's consider an example where two users – Omar and Layla – attempt to organise lunch on campus. Omar is at the library while Layla is at her study room, in different buildings. The interactions are outlined in the Figure 2.

The decision process involves: finding a commonly suitable time (Figure 2.a-c) and then agreeing on a conveniently located restaurant that satisfies Omar and Layla's food preferences (Figure 2.d-g). As an m-commerce advantage for the restaurant, once it becomes aware of the intention, it can provide e-gifts (discount vouchers) (Figure 2.e). Finally, the personal assistants would provide guidance to ensure they both get to the right restaurant at the appointed time.

For *role* and *autonomy* we could have the following example variations:

– *acquire information, analyse and display* and *reactive*: Omar's agent presents Layla's invitation, and Omar could confirm or decline. If confirmed, Omar's agent could present the available times and Omar would pick one. The agent would propose that time to Layla. Once Layla's agent gets Omar's confirmation, it requests Omar's food preferences; Omar's agent presents Omar with a list of options, which Omar ranks according to his preferences; this information is sent to Layla, who selects one restaurant and her agent sends a confirmation to Omar's agent which then adds the new appointment to Omar's agenda;
– *decide action* and *reactive*: for example, if for the previous scenario the user could delegate the actions of looking for a restaurant and bargaining; Layla's agent combines the food preferences, location and budge information from herself and Omar's agent and compile a list of conveniently located restaurants; then both agents negotiate with restaurant agents for best prices; once one is selected, Layla's agent would automatically forward the possibility to be confirmed or declined by Omar's agent – this action could happen automatically, based on Omar's private list of preferred restaurants!

– *decide action* and *proactive*: Layla's agent could ask for Omar's food preference and current location even before proposing the lunch – as Omar's location could make the meeting unviable;

5 State of the Art

In this section, we review a number of applications that utilize agent-based techniques to provide services to mobile users. Our aim is to provide the reader with a snapshot of current developments, and to give a feel for potential applications.

5.1 MyCampus

MyCampus [11] is a Semantic Web environment for context-aware mobile services in a university campus. The agent-based approach provides autonomous discovery of services on behalf of the user, based on domain specific rules, contextual information, and user-specific privacy preferences. Current implemented services include location-based movie recommendation and weather information.

Considering its features, the current state of MyCampus would classify as *single-user* support, as it does not involve intelligent support for inter-user interaction as far as we are aware. The agents roles are to *acquire information* and *analyse and display*, mainly in a *reactive* fashion.

5.2 AbIMA

AbIMA [8] is an agent-based intelligent mobile assistant that runs on a handheld device and assists the user through the execution of individual tasks. AbIMA uses a set of pre-programmed plans for executing different tasks. Based on the user's goals (e.g., attend a lecture, pick up a book from the library) AbIMA suggests a set of plans to the user. AbIMA implements the abstract agent programming language AgentSpeak(L) [9], which enables it to modify its plans in response to changes in the environment and user goals. For example, if a bus service gets cancelled during a trip to attend a lecture, AbIMA produces an alternative plan based on the new context.

AbIMA offers agent-based support for a single user. It gathers information from the user and the environment, performs analysis, and makes suggested decisions, but the user decides whether to implement the suggested actions. It exhibits proactive behaviour by providing advice to the user when things go wrong and initial plans cannot be executed. Moreover, AbIMA is able to adapt to changes in the user's goals, and modify the suggestions accordingly.

5.3 Paurobally and Jennings

Paurobally et al [6] proposed an agent-based framework for providing personalised mobile services. Consumers and producers of services, some of which are located on users' mobile devices, use negotiation as a mechanism for reaching agreement on the terms of a transaction in a mobile-commerce scenario. For example, a user attempting to organise a trip or browse film reviews while on a train may delegate the task of negotiating these services to his/her agent. The user agent may negotiate with service providers in order to obtain these services at an acceptable price (e.g., price of train ticket) and service level (e.g., video streaming quality). Agents adapt their negotiation strategies in response to the variations in time and resource limitations associated with mobility. For example, as the trip time draws near, the user agent negotiating train tickets concedes quicker in order to increase the speed of convergence to agreement. The resulting resource allocation is negotiated on-the-fly and are hence adaptive (as opposed to being pre-allocated and rigid).

As per our taxonomy, this infrastructure would be classified as *multiple-user* support, as it targets to support a community of users and its inter-operations with the environment; the role would be *implement action* and on autonomy it would classify as *pro-active*.

5.4 Electric Elves

The Electric Elves project [1] involves teams of agents that help users conduct routine, well structured tasks, such as organising meetings. Each person has their own proxy agent running on a mobile device. The agent keeps track of the user's location using their calendar, infrared communication within the building, and Global Positioning System (GPS). When the agent notices that the user is not able to attend a meeting (e.g., running late according to his/her current position), it asks the user if he/she would like to cancel or delay the meeting, or have it proceed without them. Based on the user's response, the agent takes appropriate action, such as notifying agents representing other meeting attendees. Once a week, an auction is used to identifying speakers for the research meeting. Potential speakers submit bids through their personal agents, and these bids are evaluated to decide who will be speaking. Finally, agents coordinate to order food from nearby restaurants based on their uses ratings.

The Electric Elves project is based on a *multi-agent* approach to coordinating multiple mobile users. Agents' roles range *acquiring* information, to *analysing* situations using decision-theoretic planning, to producing *suggested actions* and *displaying* them. Agents exhibit varying degrees of autonomy based on the type of situation at hand and learned user preferences.

6 Conclusions

We hope to have made a strong case for the potential of agent technology to provide intelligent services to assist individuals and groups of users perform their tasks on the move. We also hope that our classification would help future research in the area. Our classification could enable the identification of opportunities to extend existing efforts (e.g. by adding multi-user coordination support to existing single mobile user support systems). Furthermore, our classification could be used to identify opportunities for novel use of existing technologies for mobile assistance.

7 Acknowledgement

This work was conducted while Fernando Koch was a visitor to the Department of Information Systems, University of Melbourne. The authors are grateful to Liz Sonenberg and Connor Graham for discussions surrounding the paper topic.

References

1. H. Chalupsky, Y. Gil, C. A. Knoblock, K. Lerman, J. Oh, D. V. Pynadath, T. A. Russ, and M. Tambe. Electric Elves: Applying agent technology to support human organizations. In H. Hirsh and S. Chien, editors, *Proc. 13th International Conference of Innovative Application of Artificial Intelligence*. AAAI Press, 2001.
2. G. Di Marzo Serugendo, M. Muhugusa, and C. F. Tschudin. A survey of theories for mobile agents. *World Wide Web*, 1(3):139–153, 1998.
3. R. Grimm, T. Anderson, B. Bershad, and D. Wetherall. A system architecture for pervasive computing. In *9th workshop on ACM SIGOPS European workshop*, pages 177–182. ACM Press, 2000.
4. K. Henricksen, J. Indulska, and A. Rakotonirainy. Infrastructure for pervasive computing: Challenges. In K. Bauknecht, W. Brauer, and T. A. Mück, editors, *Informatik 2001: Wirtschaft und Wissenschaft in der Network Economy - Visionen und Wirklichkeit*, volume 1 of *Tagungsband der GI/OCG-Jahrestagung*, pages 214–222, Universität Wien, September 2001.
5. N. R. Jennings, K. Sycara, and M. J. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.
6. S. Paurobally, P. J. Turner, and N. R. Jennings. Automating negotiation for m-services. *IEEE Trans. on Systems, Man and Cybernetics (Part A: Systems and Humans)*, 33(6):709–724, 2003.
7. S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, and A. Zipf. Crumpet: Creation of user-friendly mobile services personalised for tourism. In *Proc. 2nd Int. Conf. on 3G Mobile Communication Technologies*, London, UK, March 2001.
8. T. Rahwan, T. Rahwan, I. Rahwan, and R. Ashri. Agent-based support for mobile users using AgentSpeak(L). In P. Giorgini, B. Hederson-Sellers, and M. Winikoff, editors, *Agent Oriented Information Systems*, LNCS. Springer, Germany, 2004.
9. A. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In W. V. de Velde and J. W. Perram, editors, *7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, volume 1038 of *LNAI*. Springer, 1996.
10. N. Sadeh. *M-Commerce Technologies, Services and Business Models*. John Wiley and Sons, Inc., USA, 1 edition, 2002.
11. N. M. Sadeh, T.-C. Chan, L. Van, O. Kwon, and K. Takizawa. Creating an open agent environment for context-aware m-commerce. In B. Burg, J. Dale, T. Finin, H. Nakashima, L. Padgham, C. Sierra, and S. Willmott, editors, *Agentcities: Challenges in Open Agent Environments*, LNCS, pages 152–158. Springer, 2003.
12. M. Satyanarayanan. Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4):10–17, 2001.
13. T. B. Sheridan. Rumination on automation. In *Proceedings of 7th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems*, Kyoto, Japan, 1998. Oxford: Pergamon Press. Plenary address.