Adaptive User Support in Agent Based Dynamic Environments

Jan M.V. Misker, Cor J. Veenman, Leon J.M. Rothkrantz

Delft University of Technology

Delft, The Netherlands

{j.m.v.misker, c.j.veenman, l.j.m.rothkrantz}@ewi.tudelft.nl

Abstract *In this paper the concept of an ad hoc agent environment as a way for users to interact with an ambient intelligent environment is introduced. Utilizing agents, users can interact on a level that best suits their needs and capabilities, leaving tedious chores to the agents. The user interacts with this environment as a whole, instead of interacting with individual applications on individual devices. Devices and services in the environment have to be more or less independent, which fits well with the notion that agents are autonomous. Intelligence enables the agents to learn about their user and adept the environment in such a way that the users cognitive resources are not overly addressed. A research application that demonstrates some of the issues involved when employing agents has been developed. The application shows that there is a tension between the amount of control a user can have over an agent and the autonomy of the agent.*

1 Introduction

This research is about user interaction in an interactive, dynamic environment. The typical environment we envision can be characterized by being a) able to be operated on a high level of abstraction, i.e. a level that has meaning to a typical user, b) context aware and c) offering a continuously changing set of available services, e.g. because a user moves around. The high level topic of interest to us is how a user interacts with such an environment as a whole. As a first step to make this question more tangible, we propose the use of agents as an intermediary level between the user interaction and the technical properties of the environment. Using agents provides a means of modelling the various information flows and interactions with the user and within the environment. Furthermore, when we look at the user interaction from an agent perspective we can, by adding knowledge about the system, reformulate some of the technical properties into interaction issues, for example discovery can be related to navigation.

The environment has a wireless, ad hoc network as the underlying infrastructure. Nodes have continuously changing connections with other nodes and information can be routed to nodes outside the direct radio range.

1.1 Methodology

We are building running prototypes that 1) demonstrate aspects of the concepts and issues involved in user environment interaction and 2) can be used to do user tests. The latter with the goal to find out whether the concepts we developed can be understood by users and how to set certain parameters. Two concepts that seem promising are agent autonomy and agent grouping; discussed in more detail below. The prototypes are built with the FIPA [4] compliant JADE [2] platform running on PDAs, which provides a realistic interaction experience.

1.2 Multidisciplinary approach

Various research fields are combined. Ambient Intelligence provides insights and concepts on what the user interaction should be like, UbiComp provides the technical infrastructure, possibilities and constraints and Multiagent Systems allow us to model the information flows.

Ubiquitous Computing is the research discipline concerned with bringing computing and networking capabilities to a users everyday environment, tasks and objects.

Ambient Intelligence, when contrasted with ubiquitous computing, is concerned with the applications that become available and how they can change the user experience and ultimately a users everyday life. A typical example of an ambient intelligent application is the GossipWall [6].

Multi-Agent Systems are systems consisting of multiple agents that deliver services and respond to queries. It should be noted that with agent we do not mean embodied agent or avatar. Our working definition of an agent is adopted from [7]: an agent is an artificial, computational entity that can perform certain tasks with a certain degree of autonomy or initiative whilst intelligent adaptation to its environment. Note that a human is not an agent in this definition.

2 Ad Hoc Agent Environment

We use the concept of an ad hoc agent environment as a research tool that allows us to model the interactions amongst the various parts of a dynamic, networked environment. Every device and/or service is modelled as one or more agents. This implies that not only the devices in the connected environment have an agent representing them, but also the devices and services that are associated with the user and more general services on a global scale like Internet web services.

2.1 Ad Hoc

The agent environment is ad hoc, meaning that connections between agents are made at run-time. There can be no centralized agents regarding naming and addressing of agents, but of course there has to be local infrastructure for this. There will also be a certain amount of common knowledge that the agents need in order to communicate about relevant topics, e.g. a message passing framework and ontologies.

2.2 Agent Groups

Within the agent environment there can be groups of agents that cooperate and/or share a purpose, for example: a) a group of agents that represent a users personal devices, b) the personal agents of a group of users or c) an information providing agent and a display agent. Within a group agents can communicate based on the *role* it performs in the group. The group concept can be used to abstract over technical properties regarding ad hoc combination of devices, services and content in the environment. From a user interaction viewpoint, it is expected that the notion of a group can help users gain a better grip on their interactive environment.

3 User Agent Interaction

Abowd et. al. [1] provide an excellent overview of user interaction in a UbiComp environment, showing the need for a more complete understanding of the way users perform tasks and activities. User interaction with agents provides several hooks that relate technical properties to user interaction:

- Autonomy: how much should a user know about the actions that an agent takes on his/her behalf?
- Time: what should happen when an agent message deadline is approaching whilst an agent is still engaged in an interaction with its user?
- Availability, discovery: how should agents let a user know they are available? How does a user convey his interest in a particular agent?
- Identity: for a user it will be of importance which agent performs a certain task, e.g. because an agent is tied to a certain physical device or because of trust considerations. Also agents will need to be aware for which user they are performing tasks.

The Electric Elves [5] project is a good example in which some of these issues were also investigated. An important result of that research was that it is very hard to separate general and task specific knowledge regarding agent autonomy.

4 Restaurant Negotiator Scenario

Based on the notion of an ad hoc agent environment we developed a demonstrator application that allows us to gain further insight into relevant research questions and provides a way to test competing answers to these questions. The application is based on the following scenario: a group of tourists are spread throughout a city they are visiting. They want to have dinner together but do not know which places are available. They are connected through a wireless ad hoc network, and have a device that maintains this connection and that enables user interaction. On every device a set of agents is running that can perform various tasks, including facilitating the restaurant scheduling.

Several issues can be investigated with this scenario and demonstrator implementation. These include:

- User attention: if a user is not actively interacting with his agent, deadlines might be missed. A possible solution to this is that the agent actively draws the user attention to the device/agent, for example with a sound signal. Still it might be the case that a user does not provide the required feedback in time, in that case the agent might intelligently construct the preferences himself, e.g. based on a user profile or on previous user input (which can be regarded as part of the user profile).

- User profile: sometimes certain information is available from a user profile. In this case the user location and eating preferences might be accessible through his/her profile.

- Agent autonomy: they negotiator agent can take decisions autonomously. Can users understand what the implications of this autonomy are for the result of the negotiation? Should they be able to see which actions their agent has taken on their behalf, or do they only need to see the result?

4.1 Related work

Planning support is an area in which agents are often used, some of them we discuss here, focusing on the differences with our approach.

4.1.1 RETSINA

Based on the RETSINA Multi agent system, Singh et al. [6] developed a meeting scheduler. Their system is primarily a demonstrator of the capabilities of the Semantic Web combined with intelligent agents. Users have a scheduler agent, called RCal, that reasons with knowledge stored and exchanged using the Resource Description Framework (RDF). Furthermore, the RCal agent can update its user's Microsoft Outlook calendar with new or changed meeting details.

Individuals wishing to make an appointment with a user of an RCal agent can do so by going to a webpage where they can enter their desired meeting details. This starts a negotiation between the RCal agent and the individual making the appointment, through the webpage. When a meeting is successfully scheduled, the RCal agent enters it into its user's calendar and sends an email to all participants.

The RCal agent is a good example of the possibilities of the Semantic Web enhanced with intelligent agents. However, from a User Agent Interaction viewpoint there are some issues. With respect to its user, the RCal agent is completely autonomous, which can give rise to for example the following problem: *Suppose a user makes an appointment himself, but forgets to put it in his calendar. The RCal agent schedules a meeting and puts it into the user's calendar, which reminds the user that he actually wanted to have another meeting at that time; a conflict is born.*
What it boils down to is that users may not be willing to give up control over their calendar. A possible solution could be to ask the user to give his approval for every appointment, but that would render the intelligent agent useless. Perhaps the user could specify that for certain periods he would let the RCal agent be autonomous, and for other periods ask his approval. This is an example of a case where the user controls the autonomy of the agent, which is one of the aspects of User Agent Interaction we study.

4.1.2 Electric Elves

The Electric Elves [5] is a project in which a human team used personal agents to help them in their teamwork. The activities the agents can assist the team in include: meeting (re)scheduling, arranging lunch and organizing the visit of external visitors. The agent system uses GPS, infrared, calendar information, computer activity, cell phones, PDA's etcetera.

What is of special importance in relation to our research is the notion of *Adjustable Autonomy.* Early on in the design of the Electric Elves system it was recognized that the personal agents cannot always act fully autonomous, furthermore it became apparent that even if a user kept control over certain decisions, their was still a need to act if a user did not provide input in time. The concept of adjustable autonomy was implemented using a decision-theoretic planning approach that used Markov Decision Processes (MDPs) to support explicit reasoning about team coordination, consisting of three steps: (i) Before transferring decision-making control, an agent explicitly weights the cost of waiting for user input and any potential team miscoordination against the likelihood and cost of erroneous autonomous action; (ii) When transferring control, an agent does not rigidly commit to this decision, but it instead flexibly re-evaluates when its user does not respond, sometimes reversing its decision and taking back autonomy; (iii) Rather than force a risky decision in situations requiring autonomous action, an agent changes its coordination arrangements by postponing or reordering activities to potentially buy time to lower decision cost/uncertainty.

The approach described above is tailored for team coordination, but some general ideas about adjustable autonomy can be distilled from it. (i) Agents should explicitly consider their autonomy, however this requires that the agents have a considerable amount of task-specific knowledge about the effect of their actions. (ii) Agent should not rigidly commit to decisions. This principle is not task specific, although not suitable for tasks in which the agent must make a commitment to another agent, e.g. booking a hotel room. (iii) Postponing and reordering activities to buy time can work in certain situations, but is also task specific.

4.2 Design and Implementation

The scheduling process is based on an iterated contract net protocol. Figure 1 provides an overview of the message flow. For the sake of simplicity some non-essential messages are not shown this figure. The main scheduler agent sends out a call for proposals to every participating agent, which respond by sending their user's preferences. The scheduler then attempts to make a plan that fits all preferences. If such a plan cannot be made, it sends a new call for proposals. When all preferences can be fit into one plan, a voting round is held that enables each agent to cast a vote on that plan. If the plan is dismissed, a new call for proposals is sent out, if the plan is accepted the scheduling is completed successfully. Every call for proposals or voting proposal is accompanied by a deadline.

The scenario has been implemented using the FIPA compliant JADE platform. The user interfaces were developed in Java AWT, making it possible to do the user interaction on PDA's as well as PC's.

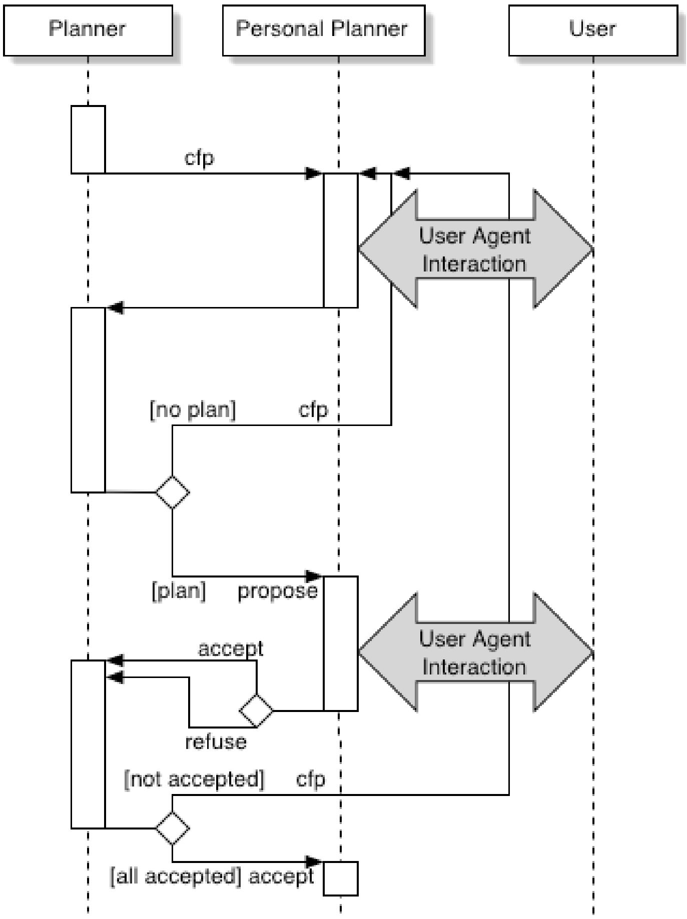

Figure 1: The flow of messages in the Restaurant Negotiator scenario, including the user interaction.

5 Conclusions and future work

We presented a novel approach to user interaction in an ambient intelligent environment using agents, enabling us to relate actual technical possibilities and constraints to user interaction problems and solutions. An important relation between technology and user interaction is the autonomy of an agent versus the degree of user control. We plan to continue developing demonstrator applications to further research these relations and design and test possible solutions. Testing will ultimately be done with test subjects in a realistic environment.

Acknowledgements

This PhD research, which started in October 2002, is part of the CACTUS project [3], which is a multi-disciplinary joint research project of Delft University of Technology (Faculty of Industrial Design Engineering and the faculty of Electrical Engineering, Mathematics and Computer Science) and research and technology organisation TNO.

CACTUS is part of the Freeband Impulse program that aims at the generation of public knowledge in advanced telecommunication (technology and applications). Freeband is based on the vision of 4G networks and services as developed in the report 'Vrijband: een breedbandvisie voor Nederland' (Concurreren met ICT Competenties, the Hague, August 2001), which describes the R&D challenges in telecommunications in the near future. It specifically aims at establishing, maintaining and reinforcing the Dutch knowledge position at the international forefront of scientific and technological developments, addressing the most urgent needs for research and novel applications in the present unfolding of new technology (UMTS, Ultra Wideband, Ad-hoc Networking, Pure Optical Networks, Home Networks, Web Services, IPv6, etc.). The Dutch Ministry of Economic Affairs is co-funding this programme as part of the policy plan "Concurreren met ICT Competenties".

References

[1] Abowd, G.D., Mynatt, E.D., and Rodden, T. (2002). The Human Experience. *IEEE Pervasive Computing*, 1(1), 48-57.

[2] Bellifemine, F., Poggi, A., and Rimassa, G. (1999). JADE – A FIPA-compliant agent framework. *Proc. PAAM 1999.*

[3] CACTUS website: http://www.cactus.tudelft.nl/

[4] FIPA, Foundation for Intelligent Physical Agents website: http://www.fipa.org/

[5] Scerri, P., Pynadath, D., and Tambe, M. (2002). Adjustable Autonomy for the Real World, in *Agent Autonomy*, Kluwer.

[6] Singh, R., Sycara, K., & Payne, T. (2002). "Distributed AI, Schedules and the Semantic Web" in *The XML Journal* , Vol. 3, No. 11, November, 2002, pp. 40-45.

[7] Streitz, N.A., Röcker, C., Prante, Th., Stenzel, R., and van Alphen, D. (2003). Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments. In Harris, D., Duffy, V., Smith, M., and Stephanidis, C. (eds.) *Humand-Centered Computing: Cognitive, Social and Ergonomic Aspects*, Lawrence Erlbaum.

[8] Wooldridge, M. (2002). *An Introduction to Multiagent Systems*, Wiley.