Adjusting Autonomy of Agent Systems

Amedeo Cesta
IP-CNR, National Research Council
Viale Marx 15
I-00137 Rome, Italy
amedeo@pscs2.irmkant.rm.cnr.it

Daniela D'Aloisi, Marcello Collia
Fondazione Ugo Bordoni
Via B. Castiglione 59
I-00142 Rome, Italy
dany,mcollia@fub.it

Abstract

This paper describes some mechanisms that could be useful to constrain the autonomy of interface agents. Such mechanisms should increase the acceptance and use of agents systems that several studies have demonstrated to be quite limited. Several users do not rely upon agents capabilities and competence as a result of limited trust in decision aids. Other problems, closer to a human-computer interaction view of the problem, are connected the possible loss of control on the agents actions that humans perceive as a negative feeling. Furthermore, users may perceive the system as something that limits their usual behavior constraining them from a free choice. In interactive applications, agents are more effective when associated with the human users to form an integrated mixed-initiative system. The paper describes some solutions realized in an agent system for meeting scheduling that give both agents and their users the ability to effectively perform tasks and to develop a trustful interaction. In particular, level of autonomy of the agent is decided by the user and can vary dynamically according to user needs and current operative context.

Introduction

This paper contributes to the topic of adjustable autonomy of agents with the point of view of personal agents' designers aiming at supporting users in time-consuming daily activities. The contribution is based on our experience in developing different agent systems (Cesta & D'Aloisi 1996; Amati *et al.* 1997) but in particular refers to experience gained developing an agent based meeting scheduler named MASMA (Multi-Agent System for Meeting Automation) (Brancaleoni, Cesta, & D'Aloisi 1997; Cesta, Collia, & D'Aloisi 1998; Cesta & D'Aloisi 1998). In such systems the user-agent interaction is managed according to mixed-initiative principles; attention is also given to control the agent, modify its autonomy according to internal and external constraints, to the current states of affairs, to user preferences, etc.

In designing personal agents we started from an objective analysis: agent-based systems were becoming more and more popular but very often their actual utilization was limited to enthusiasts and experts. "Normal" users did not rely upon agents capabilities and competence as a result of limited trust in decision aids (similar to the analysis done in (Muir 1987)). Criticisms attribute this failure (Whittaker & Sidner 1996; Shneiderman 1997) to the common view of an agent as a *completely autonomous* entity performing tasks on the user behalf and being completely independent from him. That induces a negative feeling in users who are worried of the possible loss of control. A further point is the fact that users may perceive the system as something that limits their usual behavior constraining them from a free choice (Friedman & Nissenbaum 1997). Indeed complete autonomy is not always the best solution. Besides actual problems in accepting artificial assistants, there are several real situations in which it would be better to control the agent initiative and to give it a variable, changeable autonomy.

In particular, in systems designed to assist users in critical assignments, "complete" task delegation to agents is not desirable. In contexts where agents manage personal data and take decisions that could be critical for the user, probably the user would rather prefer to maintain continuous control over the process since he is not likely to immediately trust how his agent could act on his behalf. Moreover, the context and the evolution of the agent-user interaction could require different and changeable level of autonomy.

Our claim is that, in interactive applications, agents are more effective when associated with the human users to form an integrated mixed-initiative system: indeed the couple agent-user is a multi-agent system that should avoid conflicts in order to work effectively. Characterizing the pair agent-user as a mixed-initiative system means to plug in mechanisms which give both the actors the ability to effectively perform tasks by respecting their role. A consequence is that the level of autonomy of the agent is decided by the user and can vary according to the current needs, tasks and context. The apparatus should allow the user to have the feeling of being continuously in control of the task, to incrementally increase his trust in the agent and then vary little by little the numbers of tasks he comfortably delegates to his agent. Mixed-initiative should also regulate the responsibility of an actor in a given phase of

the work: so there will be the possibility for the user to take over the agent in specific contexts, and for the agent to have a picture of the situations in which it has to refer to the user for taking decisions.

The idea of considering a user and his personal assistant as a mixed-initiative system is strengthened by the need for reconciling two contrasting issues: (a) taking advantage of agents problem solving activity; (b) favoring acceptance of the agent by the user.

The first is related to the fact that agent-based systems are regarded as useful tools, that can relieve users from their work overload by completely mechanizing repetitive and time-consuming functions. Furthermore it is worth observing that the need for actual agent autonomy is a function of the domain of application. In several domains the autonomy of the agent is unnecessary, while in others is very useful. Agent systems may be classified according to their behavior with respect to the user (Cesta & D'Aloisi 1998):

- *Low-interactive agents* carry out tasks that are quite complex and require very few interactions with human users. Both the softbots and the robotic agents fall into this category. In robotics, the aim is building completely autonomous robots that incidentally meet human beings in their environment. Also softbots mainly work by themselves and just occasionally they interact to receive new tasks or to give back results or operational reports. In both cases, the human-computer interface is not required to model the interaction but simply to study some form of information format and information presentation.
- *Highly-interactive agents* carry out expert tasks along with the user (as in the case of personal assistants) or play with him at some level (like in the case of synthetic characters for entertainment in virtual worlds). Personal assistants and believable agents have as a goal the interaction with the user. Because of their task, highly-interactive agents need "competence" about how to interact with the user, need to support forms of dialogue with him and need to relinquish the initiative during a conversation or collaborative decision making.

It is worth remarking that among highly-interactive agents a deep difference exists between personal assistants and synthetic characters. The former carry out their tasks in strict interaction with the user and take advantage of a mediated role of autonomy, while the latter have autonomy as a peculiar aspect (synthetic characters are engaging because they show independence from the interacting user).

Our conclusion is that in designing personal assistants we have also to design mechanisms allowing to modify and constrain agents autonomy. Our proposal is that the agent can follow basic protocols in which the flow of initiative depends on the particular task, the user preferences and profile, etc. All these variables differently influence decision points by assigning the initiative to one of the actors. The *amount* of initiative maintained by the agent measures its autonomy degree. In the first part of the paper, we describe our proposal in the context of the MASMA system that has been used as a laboratory for studying interaction between agents and users. In the second part, we try to highlight some general issues contained in our work concerning the interaction between users and personal agents.

MASMA: Variably Autonomous Agents

MASMA proposes a solution to the meeting scheduling problem in which the competence is distributed among a community of agents. Its architecture consists of a personal assistant for each user, the *Meeting Agent*, and other three middle (service) agents that guarantee common services: the *Server Agent*, the *Resource Agent* and the *Travel Agent* (Brancaleoni, Cesta, & D'Aloisi 1997).

The *Meeting Agent* contains most of the results of our recent efforts (Cesta & D'Aloisi 1998; Cesta, Collia, & D'Aloisi 1998). Because of the high interactivity of the task it is responsible of, the design of this agent is very crucial for the acceptance of the whole approach to the users. The design of the user-agent interaction presents all the problems related to human-computer issues. In addition, the interaction with an *artificial "potential" substitute* implies the following aspects:

- the user has to learn to trust the agents;
- the user can be scared by the autonomy and the initiative of the agent;
- the autonomy of the agent is problematic also for aspects connected to task delegation and particularly for situations in which the user feels the "loose of control" on the delegated task (Friedman & Nissenbaum 1997; Milewski & Lewis 1997);
- the agent deals with personal data (for instance, the user diary) and knows personal preferences (e.g., his attitude towards meeting a person).

Our goal was to build a personal agent whose autonomy could be "adjustable" according to the user profile and preferences, and such that the user could maintain control over the agent behavior. Basically we have tried to (a) ensure the user the possibility to trust the agent and hence to increase little by little its autonomy, (b) allow the agent different behaviors in different environments, (c) guarantee an autonomy related to the user preferences, i.e., an autonomy adaptable to the user way of solving a task.

The main task of MASMA is to support the management of possible dates for a meeting giving particular attention to cases in which it is necessary to arrange requirements of several participants. In such cases it is quite complex to find a middle course: so a negotiation process is engaged by the organizer agent in order to reach an agreement.

Negotiation Protocols. In an application like MASMA it is very important to have a basic mechanism to control interaction between different meeting

agents—during negotiation for scheduling a meeting—and between meeting agent and its personal user. This is done using multi-stage negotiation protocols, that essentially are finite state automata whose nodes represent actions for the agents and edges are weighted with possible outcomes from the actions.

The negotiation protocols define the standard behavior of a Meeting Agent. We have two standard behaviors in it, one for the role of meeting organizer and the other for the role of meeting attender. We have implemented the protocols as different reasoners that can be independently loaded according to current problem solving needs.

The protocols regulate the degree of interaction between the actors (both humans and agents) and so they influence the strategy applied by the system. Several mechanisms are provided to make the protocol more flexible according to current situation, as explained in (Cesta & D'Aloisi 1998) (basically the protocol interacts with the contextual knowledge introduced below).

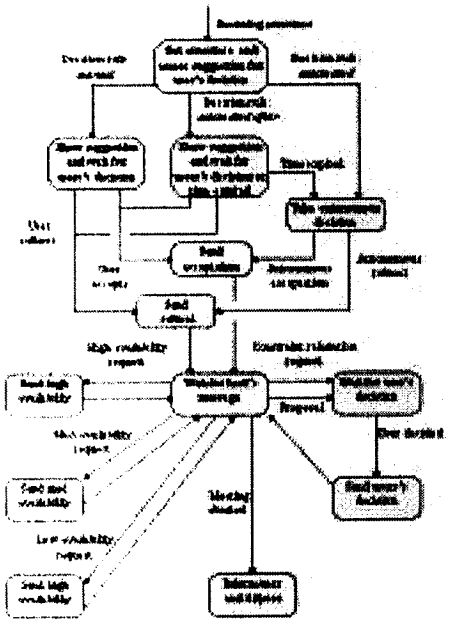

Figure 1: Attender protocol

A detailed description of the protocol is not given here, but an examples of a negotiation protocol is shown in Figure 1. The figure contains the basic protocol followed by a generic meeting attender.

The basic protocol is a sort of skeleton that is adapted to the current needs. Each decision or choice involved in the solution of the task is influenced by a set of values that depends on the user profile and on the rules set by the user. The protocols followed by the agents can influence the degree of interaction between the actors and so it can delineate the initiative strategy applied by the system.

A generic protocol can be modified according to user preferences. The interaction between the negotiation protocol and the user's profile allows MASMA to generate interactions that are customized to each single user. For example, in the first node of Figure 1 the agent can either wait for the user answer or decide by itself or decide after a delay. This setting can be modified at any time and so to influence the agent autonomy.

Controlling Autonomy and Initiative

As said before, the behavior of an agent is driven by a protocol adaptable to the current situation. The user influences the behaviors by putting constraints and rules. There are several ways to design the kind of agent the user wants:

- the user defines his personal profile and specifies preferences on how to solve the task and lead the negotiation;
- the user can define different environments and then set different profiles and preferences in each profile;
- the environments are hierarchically organized;
- the preferences can be dynamically changed;
- the user can inspect the agent behavior and modify it if necessary.

User Profile. Meeting Agents have a default behavior that allows them to organize meetings without additional information. MASMA allows the user to describe his profile in order to personalize the negotiation behavior according to the user preferences. All the preferences are translated as internal rules that are used to activate certain knowledge during negotiation. User profile contains both information that holds in any negotiation (*temporal availability rules*) and information that is dependent from the particular context in which a negotiation take place. This second possibility (*negotiation environment*) allows for a more fine grained control of the interaction attitude of the agents, but both aspects influence and control the shift of initiative when actual problem solving activity goes on. It is to be noted that the rules are a part of knowledge concerning the interaction: they can be used to diminish the interactions with the user, to increase the level of task delegation, to shift control only when strictly necessary, etc. Hence, they are a way to model the interaction and to allow exceptions to the rigidity of the weight mechanism.

Temporal Availability Rules. This part of the user profile allows the user to communicate his agent *his availability on different time intervals.* A preference value can be assigned to each hour interval chosen in the set $\{high, medium, low, nil\}$. These values can be manually set by the user directly on his agenda or deduced by the agent from the preference rules the user can define. The rules—specifically *MonthDay-rule*, *WeekDay-rule*, *Near-rule* and *Holiday-rule*—allow the user to gen-

eralize his time preference on temporal range. For example, he can specify a recurrent appointment occurring at the same time, e.g., from 9 a.m. until 11 a.m., on the same day, e.g, the 15th, of each month on a specific stretch of time, e.g., from April 1998 to April 1999. This information allows to define not only specific requirements but also general attitudes of the user on time intervals (e.g., rules of the kind "I never like to meet people early in the morning").

Negotiation Environments. This part of the profile strongly influences the level of autonomy of the agent and allows the user to ascribe different behaviors to separate contexts. To partially automate the decision processes and to frame the agent behavior into a context and the correspondent preferences, MASMA allows a user to define his profile partitioned in *environments*. Environments are defined using a hierarchy starting from a generic default environment, called "Standard". The hierarchy is used to introduce sets of rules that describe the behavior of the user in given contexts. For example, initially the user can describe the rules that always hold and insert them in the environment "Standard"; then the rules that hold when he is at work are inserted in the environment "Working"; then the rules holding during his free time are inserted in the environment "Entertainment", etc.

In each environment two kinds of information can be defined:

- *the user attitudes towards meeting in that environment.* This part of the user's profile represents a powerful way to inform the agent on the meeting evaluation criteria, concerning a specific context. For example, the *Host-rule* specify the user interest in meeting a specific person in a the range {0-10}. The values 0 and 10 force the agent to refuse or accept the meeting giving it complete autonomy. Other rules, e.g., *Subject-rule*, *Priority-rule*, *Place-rule*, *Length-rule*, etc., allow the user to express his preferences of other aspects of the meeting. It is to be noted that different values lead the agent to different behaviors with different degrees of autonomy. Moreover, the user can define the relevance value of these rules in assessing the importance of a meeting and then the threshold this value must be compared with. Through the application of this set of rules, the user is able to define rules both of social and organizational kind and so on; for example the user could give high or absolute priority to meeting about a certain project (organizational kind) and/or to any request coming from his boss (social kind).
- *the default level of autonomy the user allows his agent.* The user explicitly specifies the level of freedom granted to the agent in taking a decision in specific points of the problem solving process. The agent can *Wait For User Decision*, and so the user informs the agent about his intention to be in charge of decision. A suggestion, based on the previous set of rules, will be available for the user. The second option is the *Automated* rule that leaves the agent free to decide autonomously. The *Automated After* option represents a compromise between the two previous scenarios. The agent provides a suggestion that will evolve in an autonomous decision only after a time interval specified by the user. In this time interval the user can take over the agent.

The negotiation environments represent the basic mechanism to influence the behavior of negotiation protocols. Moreover these environments endows the system with a further level to adjust the agent autonomy that can change according to the specific context.

Control of the Initiative and Inspection. The system allows the user to define the rules for relinquishing control during standard problem solving activity. This initiative flow can be modified by the specific information in the user's profile and by the direct intervention of the user. To this aim, we allow the user to observe the current negotiation going on and eventually to directly or indirectly influence the negotiation behavior of the agents. This feature has been designed to increase the possibility for the user to control its agent, and consequently to become confident with the behavior of the agent.

These mechanisms are aimed at allowing the user to alter the usual flow of information, according to contingent needs. These requests are independent from the knowledge in the user's profile and from the current step of the negotiation protocols.

The user can inspect the agent activities through an inspection window containing information about the ongoing processes. Moreover, the user is simultaneously able to modify values and parameters to influence the current state of affairs: this would be a way of constraining the dynamic behavior, a "dynamic" possibility of influence. In particular the user can view the current negotiations, and for each of them, a summary of meeting data, time slot in use, the ones already tried, and the ones to be tried ordered on the basis of their convenience values. The user can influence implicitly the organization and negotiation processes by dynamically modifying preference and availability values. He is also given the possibility to explicitly interfere by altering the order in which the slots are arranged. The inspection window does not interrupt the agent activity that goes on with the negotiation process. The change of preferences may happen at any time independently from the achieved results: the job of re-contract is left to software agents not to humans. The inspection window is useful to verify, but it is also an instrument at user's disposal to interfere indirectly or directly in the process and to take back control and influence the degree of agent autonomy.

An Example

The following example involves three people named Amedeo, Daniela and Marcello each of them having a personal agent respectively named $\mathbf{Agt_A}$, $\mathbf{Agt_D}$, and $\mathbf{Agt_M}$.

Personalizing the Meeting Agent. Marcello has been recently hired in the firm "XYX" where MASMA is used as a decision support tool and he immediately sets up his personal agent.

Amedeo is the boss of the firm. Marcello is involved in the project "MASMA" which also Daniela is working on as a head-designer. He is also a member of a fishing-club and every now and then he attends social meetings. These are the scenarios, that Marcello wants his agent $\mathbf{Agt_M}$ to be acquainted with.

First Marcello defines an environment, representative of the firm, called "XYX" and, inside it, he sets his own profile. Since Amedeo is the boss, any request coming from him will have to find Marcello available. Such a behavior, which represents an interpersonal relation based on 'social hierarchy", can be modeled by the application of HOST-RULE introducing Amedeo as a HOST and awarding him the value-max (10) in priority, so forcing $\mathbf{Agt_M}$ to accept any request of HOST Amedeo with no consideration for the evaluation of the remaining features of the meeting.

Marcello has now to represent his involvement in the project "MASMA". Two are the possible options: the first one, a consequence of the fact that in Marcello's view the behavioral rules hold as well as in the "XYX" environment, is to define, inside this environment, a new subject of interest called "MASMA" and award it the desired value by the application of SUBJECT-RULE. If, on the contrary, inside the project "MASMA" different behavioral rules are in force, Marcello can define a new environment "MASMA" by specializing the environment "XYX" and modifying some preferences and adding new ones. Suppose, for example that Marcello, must recognize the hierarchy of Daniela within the project, but not within the firm, then in "MASMA" the HOST Daniela will be awarded the max value so forcing $\mathbf{Agt_M}$ to accept any request coming from her, whereas within "XYX" HOST Daniela will be awarded a value in the range {1-9} so that her requests are evaluated globally and not for just coming from her. In fact, both value-min (0) and value-max (10) "force" the agent to respectively reject or accept requests coming from her. Let us suppose that Marcello chooses the first option. It is worth remarking that the simple HOST-RULE allows the user to tune the agent autonomy in different situations.

The last step Marcello can perform to personalize the $\mathbf{Agt_M}$ behavior in "XYX" environment is to fix borders to the agent autonomy. It is clear that Marcello's choice will depend on his overall evaluation of the criticality of the decision in that environment and on how trustworthy he thinks his agent is. As a consequence we can suppose that initially he sets the autonomy level to *Wait For User Decision* in order to continuously maintain control over "not forced" decisions.

Furthermore, Marcello can define any other environment in the same way, for example the environment concerning fishing, where other behaviors are in force, with no consideration for social hierarchies (at least we hope!) but ones which can be expressed too using a combination of the rules described in the previous section. As well as HOST and SUBJECT preferences, Marcello can inform the agent about his preferences concerning length, place of meeting and so on. Marcello can now tune the weight of each previous rule as to relevance evaluation of the meeting and fix a threshold value for it.

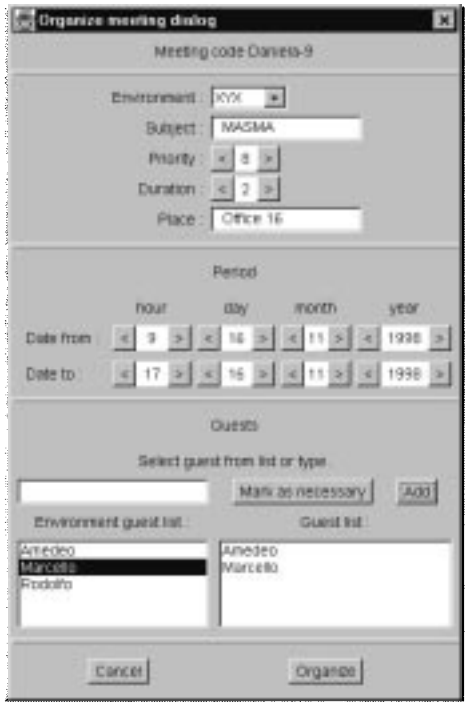

Figure 2: The meeting organization window

Organizing a Meeting. Suppose now that Daniela wants to organize a meeting whose subject is "MASMA", within the firm, on November 16th. Then, she starts using the dialogue window provided by her personal agent that is set as shown in the Figure 2. Since the meeting is a technical one, Amedeo, the boss, will be invited according to business rules, but his attendance will not represent a constraint for the happening of the meeting; as a consequence, Daniela declares Amedeo's attendance not necessary, whereas Marcello, being a project scientist, will be a necessary invitee. At this point, Daniela's agent ($\mathbf{Agt_D}$) will send the meeting announcement to Amedeo's agent ($\mathbf{Agt_A}$) and Marcello's $\mathbf{Agt_M}$.

$\mathbf{Agt_A}$ evaluates the meeting importance and finds out that he is not interested in it and, being set on *Automated After* decision, after five minutes, undergoing no change from Amedeo, sends $\mathbf{Agt_D}$ its rejection. $\mathbf{Agt_M}$ instead, evaluates the meeting positively and, according to its settings, turns to Marcello for decision providing him with a suggestion. Marcello accepts the suggestion and $\mathbf{Agt_M}$ sends confirmation to $\mathbf{Agt_D}$.

At this point the negotiation begins between $\mathbf{Agt_D}$ and $\mathbf{Agt_M}$. $\mathbf{Agt_D}$ asks $\mathbf{Agt_M}$ for high availability, if no satisfactory interval is found, it will ask $\mathbf{Agt_M}$ for medium availability and afterwards for low. As no agreement is reached $\mathbf{Agt_D}$ asks for the relaxing of constraints. $\mathbf{Agt_M}$ cannot decide whether or not to relax constraints since its protocol involves a specific request to the user. It is in fact a critical decision which might cause the cancellation of a meeting previously fixed and a consequent costly reorganization, let alone the disappointment of the invitees. The agent reports to Marcello that the date referred to for Relax-Constraints was not available because it is next to a meeting which, however, takes place in the same building and he decides to accept the relaxing of constraints so that the meeting is fixed. At this point $\mathbf{Agt_D}$ sends confirmation and $\mathbf{Agt_M}$ informs Marcello and updates his agenda.

Mixed-Initiative and Agent Systems

The study of agent-user interaction is a really relevant point in designing software agents that turn out to be actually accepted and used. Experiments have shown that user is not comfortable in dealing with a completely autonomous agent (Whittaker & Sidner 1996). A closer investigation assesses several motivations, strictly connected with task delegation that is often useful but that requires sophisticated communication, control mechanisms and trust. Similar problems are also found in case of human–human interaction. In (Milewski & Lewis 1997) design issues connected with delegation-based interfaces are examined. Authors' analysis summarizes the following critical points:

- the tradeoff between benefits and costs of delegation must be positive;
- the effort in communicating and explaining tasks must not be excessive;
- delegation requires that users trust agent reliability;
- users should have the opportunity to control the agent performance;
- user's personality and culture influence delegation.

From this analysis, we have figured out at least three classes of problems that need to be carefully considered when designing an agent:

- the nature of trust and the development of a trustful relationship between agents and users;
- the user's concern about a possible loss of control over the agent decisions;
- task delegation seems to be a good solution in several situations, but it is not always a natural and well accepted form of human behavior.

Although an agent is mostly defined as an autonomous entity, designed to act on behalf of the user, the previous considerations have led us to moderate its independence with measures suitable to increase its acceptance and use. In our view, the <agent, user> pair is a mixed-initiative system in which the agent can act autonomously but the user can control it (the agent) and decide how, when and why to relinquish the initiative to it. In this system the initiative moves from one actor to another (Cesta & D'Aloisi 1998).

Typical examples of mixed-initiative systems are dialogue systems (Walker & Whittaker 1990), in which the turns of conversation, along with an analysis of the type of utterances, determine which actor has control of the dialogue. Also some planning systems (Ferguson, Allen, & Miller 1996) follow mixed-initiative principles in making agent and user collaborate during problem solving. We claim here that also in most agent-based systems the interaction between users and agents can be modeled as a mixed-initiative interaction in which the autonomy of the agents can change according to:

- user decisions, preferences and model,
- the nature of the relationship that can typically vary from collaborative to slave,
- the nature of the task,
- the context,
- the agent's competence, knowledge and expertise.

The user–agent interaction is not the only possible in distributed systems. Let us suppose an environment in which several agents are present that collaborate (or simply dialogue) among them. The agents can be either personal assistants, that directly supports (or even substitutes) a single user, or service agents, that can be shared by a community of users and agents (both personal or service). It is possible to identify different types of interactions in which different actors are involved:

- a user and his personal assistant,
- two personal assistants,
- a personal assistant and a service agent,
- two service agents.

Concerning autonomy, each of the previous cases requires a specific solution influenced by the type of interaction and by the followed mixed-initiative principles.

The autonomy of a service agent can be either restricted or total according to the scenario in which the agent works. For example, the *Travel Agent* could propose solutions, and hence be completely autonomous in planning a travel, but then the final decision will be up to the user or to his personal assistant. In a scenario in which a scheduling agent devoted to schedule the use of a complex spatial probe, the task to carry out is quite

complex and requires autonomy also because a human control on the possible solution is almost impossible.

The interaction between personal assistants, representing human users, is driven by relationships between their users and by the specific relationship between a personal assistant and its user.

The representation of the agent-user pair as a mixed-initiative system allows for the modeling of the different roles the two actors play in the interaction. Moreover, the reasons underlying decisions and actions may be described; the interaction model will be clarified and the control mechanisms will be better identified. The apparatus should allow the user to have the feeling of being continuously in control of the task: as a consequence, he could gradually increase his trust in the agent and vary little by little the number of tasks he comfortably delegates to the agent.

In using an artificial personal assistant, the user is interested in getting acquainted with the agent and exploiting its competence; the user wants the agent to work on his behalf, and the agent has the implicit goal of satisfying its user. That implies that the agent is autonomous, although the user can delegate it more or less autonomy also according to its reliability. The user evaluates the agent's behavior to decide the amount of autonomy to give it. Even when the agent results particularly reliable, the user may prefer to maintain control of the system. Of course, maintaining control does not necessarily involve a continuous check on what the agent is doing.

Furthermore, the need to solve the user's problem will imply taking decisions according to the current constraints and the model of the *system*, considered as the user–agent pair. This requires accounting for different *system* features, as user preferences, status of the user involved, current state of affairs, etc. Moreover, the distinctive features of the interaction can change in the long run; the user modifies his preferences and attitudes, and also the environment can undergo change. In other words, the user's model can evolve and, as a consequence, several consolidated standpoints need to be revised.

In the paper we have used terms like initiative, decision and control that are worth being explained. In our model, initiative and decision often coincide, since who decides has the initiative although not necessarily is carrying out the action. The control is held by the user even when he does not have the initiative. It could be possible to hypothesize a scenario in which the agent maintains the control of the system, for example in space probe where the user is not able to directly verify what is going on.

The definition of interaction protocols has allowed us to find a compromise among initiative, decision and control. The point was how to generate an initiative flow between a user and his agent by satisfying the contrasting needs of the user. On the one hand the user wants to delegate a complex or boring or time–consuming task to the agent; on the other hand, he wishes to control how the agent accomplishes its duties and be sure that the problem is correctly solved.

The interaction protocols drive the initiative flow according to:

- the user's profile and general preferences,
- the user's preferences on the methods to solve the task,
- a list of constraints on the degrees of freedom allowed to the agent,
- the criticality of a decision or choice,
- an analysis of the current situation and the past history of the interaction (when possible).

The protocol has to specify—following the directions extracted by the list above—who can take the decision at a certain moment. This implies that the agent will not continuously ask its user: "Can I take the initiative? I think I can decide on this point." The protocol will help the agent to determine whether it is able to decide by itself. For example, when an agent is requested to attend a meeting, the protocol may suggest it to verify its knowledge base concerning rules about the organizer and the subject of the event. For certain sets of these values, the agent can be authorized to proceed, otherwise it has to wait for the user's suggestion. For example, let us suppose the user has defined a rule stating he does not want to attend meetings about "agent-based systems" unless the organizer is "Red Brown". Let us also suppose the agent receives two requests for two meetings about "agent-based systems". The organizer of the first meeting is "John Smith", so the agent decides for a negative answer without waiting for the user's suggestion. The organizer of the second meeting is "Red Brown", then the user is asked by his agent.

In the user–agent case, the analysis of a behavior protocol allows for establishing who takes the initiative (not necessarily who is performing the current actions). Such an analysis has to be integrated with an analysis of the current state of affairs and the features of the user involved.

The evaluation of how many times the agent takes the initiative gives a measure of the agent autonomy. A completely autonomous agent will never relinquish the initiative, while a *slave* agent will simply be able to execute the user's directions. Most implemented software agents—along with MASMA—exhibit an intermediate behavior.

Conclusions

Adjusting and controlling the level of autonomy of an agent is a relevant problem in agent system design. This paper discussed this issue in the framework of MASMA, a multi-agent system for meeting scheduling. In MASMA a personal agent and its user share the task of managing agenda and exchange responsibility of decision according to the type of currently active context. Focus of our effort has been preserving user control over the whole

operation by: (a) allowing him to personally define the different environments; (b) creating possibilities to inspect the agent behavior; (c) allowing dynamic changes to the user preferences at run time.

MASMA can also be seen as an example of mixed-initiative system and we have discussed here some aspects related to autonomy of agents in such a scenario.

Acknowledgments

Amedeo Cesta's work is partially supported by CNR Committee 12 on Information Technology (Projects SIAD and SCI*SIA). Daniela D'Aloisi and Marcello Collia carried out their work in the framework of the agreement between the Italian PT Administration and the FUB.

References

Amati, G.; D'Aloisi, D.; Giannini, V.; and Ubaldini, F. 1997. A Framework for Filtering News and Managing Distributed Data. *Journal of Universal Computer Science* 3(8):1007–1021.

Brancaleoni, R.; Cesta, A.; and D'Aloisi, D. 1997. MASMA: A Multi-Agent System for Scheduling Meetings. In *Proceeding of the 2nd International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London (UK)*, 31–50.

Cesta, A., and D'Aloisi, D. 1996. Active Interfaces as Personal Assistants: a Case Study. *SIGCHI Bulletin* 28(3):108–113.

Cesta, A., and D'Aloisi, D. 1998. Mixed-Initiative Issues in an Agent-Based Meeting Scheduler. *User Modeling and User-Adapted Interaction.* To appear in the special issue on "Computational Models of Mixed-Initiative Interaction".

Cesta, A.; Collia, M.; and D'Aloisi, D. 1998. Tailorable Interactive Agents for Scheduling Meetings. In Giunchiglia, F., ed., *Artificial Intelligence: Methodology, Systems, and Application*, Lecture Notes in Artificial Intelligence, LNAI 1480. Springer.

Ferguson, G.; Allen, J.; and Miller, B. 1996. TRAINS-95: Towards a Mixed-Initiative Planning Assistant. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96), Edimburgh (Scotland)*, 70–77.

Friedman, B., and Nissenbaum, H. 1997. Software Agents and User Autonomy. In *Proceedings of the First International Conference on Autonomous Agents*, 466–469.

Milewski, A. E., and Lewis, S. H. 1997. Delegating to Software Agents. *International Journal of Human-Computer Studies* 46(4):485–500.

Muir, B. M. 1987. Trust Between Humans and Machines, and the Design of Decision Aids. *International Journal of Man-Machine Studies* 27(4):527–539.

Shneiderman, B. 1997. Direct Manipulation Versus Agents: Paths to Predictable, Controllable, and Comprehensible Interfaces. In Bradshaw, J. M., ed., *Software Agents.* AAAI Press/The MIT Press. 97–106.

Walker, M., and Whittaker, S. 1990. Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation. In *Proceeding of the 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh (PA)*, 70–78.

Whittaker, S., and Sidner, C. 1996. Email Overload: Exploring Personal Information Management of Email. In *Proceedings of CHI '96 Human Factors in Computing Systems (ACM), Vancouver (Canada)*, 276–283.