Mixed-Initiative Aspects in an Agent-Based System

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

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

Rodolfo Brancaleoni
Fondazione Ugo Bordoni *
Via B. Castiglione 59
I-00142 Rome, Italy
gheo@fub.it

Abstract

This paper describes a multi-agent system, called MASMA, that manages the meeting schedule of a set of users. Masma is a mixed-initiative decision support system based on agent technologies that addresses various aspects of the agenda management problem: in particular it is the result of an investigation on several issues concerning the acceptability of the agent approach by human users.
The paper is focused on how the initiative moves between the different actors involved in the system: human users, personal interface agents and service agents. Two classes of control mechanisms are introduced and explained that coordinate continuous interaction among the actors. The first group concerns negotiation protocols and personalization of agents that are used as standard tools to model strategies of communication. In the second class inspection windows and heuristics are applied to avoid continuous question-answering in order to increase the acceptability of the whole system.

Introduction

Intelligent agents are generally designed as active entities capable of taking decisions, yielding changes to the environment, supporting or even taking over a user in different types of activities. Their degree of autonomy depends on the tasks they carry out but in their design great attention should be paid to the *task delegation* issue in order to control their performance. It happens to turn out that task delegation is not necessarily a good solution even in case of repetitive and tedious tasks. Moreover the possibility that the user could not completely control the agent is considered as negative from a human-computer interaction (HCI) point of view (see for example (Whittaker & Sidner 1996)).

In the debate on the differences between having an autonomous system or a "slave" system we assume a somehow intermediate position: we are working at the development of active, cooperative and adaptive systems but contemporaryly we consider the need for the user to maintain the agent under control. In many cases the acceptance of the system strictly depends on the effective intertwining of the user-agent system as a whole. Our proposal is to consider the pair user-agent as a mixed initiative system in which the control shifts between the two according to the current situation, the shared knowledge and the user profile. Due to the criticality of some decisions, it is convenient to maintain the possibility of inspecting and modifying the underlying processes and decision steps.

A strategy based on a mixed initiative control has been applied in Masma (Multi Agent System for Meeting Automation), an agent-based system for the mechanization of meeting organization. In the system Masma, the agents on one side automate low level functionality (e.g., avoid the user to make quite a number of calls to organize the usual activities involved in a meeting) on the other side strictly interact with the user when critical decisions have to be taken. Furthermore functionalities are provided that allow the user to maintain complete control on the agenda management.

Several aspects that emerge in this system are presented in the paper as a contribution to the general discussion. It should be noted that we do not consider our current work as a definite answer to the various problems involved but as a step in the direction of general acceptability of agent-based support systems.

Mixed-Initiative Issues

The aim of our project is to develop agents to support users in different types of activities with the possibility of taking him over in some decisions (Cesta & D'Aloisi 1996). The agents are cooperative entities endowed with the following features: pro-activeness, autonomy, adaptability. In this context the type of interaction

* Work carried out in the framework of the agreement between the FUB and the Italian PT Admnistration.

† Partially supported by Esprit III BRWG 8319 Model-Age and by CNR Committee 12 on Information Technology (Project SARI).

and communication between the user and the system is crucial.

The interaction must be bidirectional, that is the initiative should move from the user to the agent according to an analysis of the current state of affairs. Also the control can shift between the two actors based on who is in charge of solving the current task. The resulting system is characterized by mixed initiative aspects because of the nature of the agents and the need for the user to maintain the system under his control. Even in the case that eventually the agent is able to work autonomously, it will be necessary a phase in which the user instructs the agent and verifies the correctness of its decision. Then the control will move more and more from the user to the agent. In any case the user must be able to take the control back whenever he wishes.

Another mixed initiative aspect of our framework derives from the multi-agent perspective followed in designing the internal architecture of the systems. Often a single agent cannot afford the accomplishment of too complex tasks since that would require composite capabilities, for instance to deal with several kinds of knowledge or to exhibit different problem solving aptitudes. In Masma, as in other agent-based system we worked at, different competences are assigned to different agents that work together to accomplish a final common task. That implies that the control and the initiative can shift from an agent to another.

From what said above, in our approach at least two mixed initiative aspects emerge:

- The control moves from an agent to another one. In some cases it can be a simple help request but it can also imply more complex kinds of interaction, for example cooperating to accomplish a goal, common problem solving, etc.
- Each agent interacts with one or more users. In this case the control shifts from the agent and the user(s) according to the current state of affairs. If the agent is a personal one then the interaction is generally not "static", that is the relationship between the two changes and adapts to the level of reciprocal knowledge and confidence.

The two aspects can also appear separately as in the case of single agent systems or distributed architectures. In our view the two mixed initiative aspects present differences, but it is beyond the aim of this paper to discuss them as well as deeply examining the relationship between agents.

The shift from an actor to another can be driven by rules or by protocols that state behavioral models. In Masma the agents follow specific protocols: each decision is based on the current situation, the user profile and given rules. The reason for that is that the agents are supposed to reach a common decision by respecting given constraints: that implies a negotiation process in which the control moves from an agent to another. Moreover each agent needs to interact with its user: in this case the control returns to the user.

The user-agent relationship is also conditioned by HCI issues (Norman 1994). HCI people have criticized agent-based methodologies that seem to produce systems not easily accepted by the user: one of the main reason is the autonomy of the agents that can cause a loss of control by the user. Hence we have emphasized aspects that can increase acceptability as: give the user time for trusting his agents, let him free of taking the control over whenever he decides to and leave him the possibility of verifying what the agent is doing. Obviously the user can freely decide his level of intervention.

There are other HCI issues particularly relevant in Masma since certain agents manage private information of the users agenda, and so they are supposed to take decision quite crucial for the user activity.

The application area of Masma is the meeting scheduling for a set of users. The problem has been chosen due to the high numbers of actors involved. In fact organizing meetings generally requires a massive organizational effort, complex negotiation strategies and a huge numbers of communication acts, e.g., e-mail messages, phone calls, faxes, etc. Moreover it is also necessary to find a compromise among the different users' constraints, the availability of the resources and the need of satisfying the highest number of people. An efficient and optimal scheduling process could bring benefits in term of saving time and costs and of optimizing the information flow. The problem can be seen as a particular application domain of distributed scheduling among a set of agents (Distributed Task Scheduling). In this context the tasks are the appointments to be fixed and the resources are the participants. To each resource (person) an agent is associated managing his agenda that organizes meetings by negotiating with other agents. It knows the user's preferences concerning the available dates, the information to be exchanged with the others, the agents to negotiate with, the user's interests and so on. Each meeting is organized by an agent, involves a number of other agents and is characterized by a set of constraints. The organizer agent leads the negotiation process proposing a set of possible time intervals, gathering the invitees' answers and figuring out a common solution. If necessary the process is repeated until an agreement is reached.

MASMA: An Agent-Based System

The system Masma (Multi-Agent System for Meeting Automation) is an agent-based and multi-agent system (MAS) designed to support users in managing their personal agenda and to organize and schedule appointments, meetings, small conferences and seminars. Masma consists of a set of agents that cooperate among them, each devoted to deal with a particular task or set of tasks. In developing Masma the practical usability of MASs and their effectiveness in interacting with users assumed a fundamental importance. In developing the system we first designed a generic agent architecture, then we instantiated it to build up different kind of agents.

A Generic Agent Architecture

The proposed general model to design agents is flexible and adaptable enough to guarantee an incremental and modular development of the whole framework.

The architecture follows a *Body/Head/Mouth* metaphor (Steiner, Mahling, & Haugeneder 1990) and it is described in (Cesta & D'Aloisi 1996; Cesta, D'Aloisi, & Brancaleoni 1997). It consists of three components:

- The *body* is in charge of task execution; it carries out the specific tasks of the agent in the application domain. A task can consist of pre-existent software, public-domain programs, on-purpose software or hardware components. That guarantees a high adaptability since it allows for incorporating and/or using any old part without developing from scratch.

- The *head* is devoted to coordinate the different functionalities; to manage the representation of the external world, of the agent and of the current state of affairs; to reason, to solve problems and take decisions; to coordinate and negotiate with other agents. In turn, it consists of four components, the controller, the reasoner, the knowledge base and the working memory. The controller continuously checks for new information status and activates a task of the body to execute next. In case of problematic decision, the controller gives responsibility to the reasoner to perform more high level functionalities in order to decide future plans.

- The *interface* is in charge of the communication with the user, other agents and the environment. It consists of: a *KQML Message Manager* to support the interactions between agents; a set of *Sensors* and *Actuators* to exchange data with the environment; a *User Interface Manager* to communicate with the users.

The Multi-Agent System

Masma proposes a solution in which the competences are distributed among different types of agents: moreover, inherently distributed problems are faced in a distributed way and centralized problems are faced in a centralized way. Its architecture consists of a personal assistant for each user, called *Meeting Agent*, and other three service agents that are shared among a community, the *Server Agent*, the *Resource Agent* and the *Travel Agent*. Each agent is an instance of the general model described in the next section.

A *Meeting Agent* is associated to each user and behaves as a personal assistant specialized in meeting organization. It carries out two main tasks: managing the user's profile and taking part in the meeting organization. The profile contains information concerning the personal agenda and the preference values assigned to the different dates and times: moreover it also maintains data about the user's general interests. The agent is already connected with the user's diary from which can extract and to which can communicate information. In the organizational process the Meeting Agent represents the user according to his profile. Usually this process needs a negotiation phase to decide the date and the place of the event, the duration of which depends on the imposed constraints. The Meeting Agent can play the role of organizer or attendee by applying the correspondent negotiation protocol: it is possible to include more than one protocol and to base the choice on the context and current status. It is also possible to envision very sophisticated protocols that take several issues into account, for instance the role of the users in the community, the weight of previous engagements, the type of event, etc.

At present the agent is designed to maximally mechanize the decision process and to limitate the interaction with the user. In the decisional process, the control moves according to the content of the profile and to the role played by the agent. Looking at the whole task of scheduling a certain event, the organizer's Meeting Agent—that leads the play—and the other Meeting Agents involved follow a high-level protocol: in this case the control moves from an agent to another. Instead in case of a "local" decision the control shifts from the agent to its user.

A Meeting Agent can also interact with a service agent to obtain information. In this case the control moves temporarily to such a service agent. The three service agents work as specialized knowledge servers to which some common tasks have been delegated. They could also perform their job autonomously without any connection with the Meeting Agent.

The *Server Agent* is in charge of managing the net-

work addresses and of maintaining a knowledge base with the users' addresses. In case of new users, it is able to get the addresses by querying an external server. It also manages a database containing the interest areas associated with the users so that it can help the agent organizer to spread announces out in a selective way without bothering all the connected users.

The *Resource Agent* adopts a centralized administration of the common resources to avoid conflicts in selecting one of them. The Congress Centers or Universities or other similar sites are crucial resources in a meeting organization. Each site is characterized by $< attribute - value >$ pairs that describe it. The *Resource Agent* maintains the databases and furnishes to the Meeting Agent a list of structures satisfying the problem constraints. When a decision is taken, the agent carries out the operations necessary to reserve the place.

The *Travel Agent* helps the user to mechanize the last step in organizing a meeting, the lodging and travel decisions. The agent can connect the user to train and flight timetable, decide the best path between to places, inform him about prices, show a list of possible hotels. It could also furnish a reservation service. At present it work on local databases, but it could also be connected to external information servers specializing the agent to understand database management languages.

Masma is based on a mixed-initiative approach in which each step in any decision process is carefully considered: if the user prefers then any crucial decision remains up to him. Moreover he can always maintain the control on his agent and interfere and influence its actions. The user can dynamically influence the negotiation process by changing the constraints on line. After a testing phase, the user can decide to leave more and more decision steps to his agent although the possibility remains of inspecting and interfering in its behavior.

Mixed-Initiative Aspects in MASMA

In a system like Masma there are several aspects in which the control of the initiative is relevant. As partially noted before, there are at least three different types of interaction: human user *vs* (personal) meeting agent, meeting agent *vs* meeting agent, meeting agent *vs* service agent.

The approach followed for distributing the initiative and the devised solutions are characterized by a part influenced by the specific application and by a part devoted to general issues related to delegating a task to an agent.

The most significant aspects derived from the attempt of solving the specific problems are the following:

- Task driven control of the initiative: the negotiation protocol followed by the agents to reach an agreement about the meeting actually establishes activity turns between the user, his agent and the other agents.
- User centered control through personalization: strictly speaking also the possible personalization of agents' behavior in meeting scheduling can represent an indirect means of constraining the interactions, e.g., by using default knowledge as shown below.

Other two aspects are instead connected to the task delegation towards the agents:

- Checking mechanisms available to the user to take the control: the user can inspect what the agent is doing.
- Heuristics and mechanisms to verify the opportunity for the system to give the initiative back to the user: general strategies of how and when interacting are relevant to this issue.

These four points will be respectively analyzed in the following subsections.

The Basic Negotiation Protocol

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.

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. The attendants' agents try to protect their users and they in turn apply a strategy to safeguard their privacy and to avoid the relaxation of important constraints. At present, the strategies are fixed—one for the organizer and one for the participants—but it is possible to differentiate them. The agent organizer has the goal to look for an optimal solution: obviously the optimality depends on the selected criterion. At present the solution is to maximize a common utility function and to minimize the requests for constraints relaxation.

The description of the protocol will point out how the control and the initiative move between the agents and the users. In any case it is the organizer that guides the game and takes the final decision.

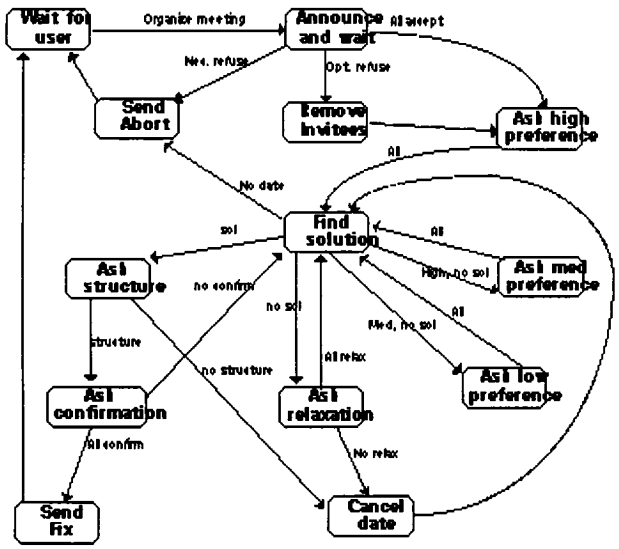

Figure 1: The organizer's negotiation protocol

Once the user has defined the features of an event, his Meeting Agent sends the announce to the interested users through the Server Agent. At this point the negotiation begins, and the organizer agent behaves according to the specified protocol (Figure 1).

The agents receiving the announce calculate the availability of their users concerning the proposed date(s) and then they can ask for confirmation to them: this step could also be mechanized. It is possible that some of the invitees reject the request: according to the current situation either the negotiation goes on or the organizer can decide to interrupt the process and to re-arrange the event by changing its features. If the first case, the organizer agent asks the other Meeting Agents for the time intervals with an high preference values: if it does not find any possible intersection, it asks for lower preference values. If it is still not able to find any possible data, it starts asking other agents for relaxing some dates. In order to fasten the converging of the decision process, the agent suggests the interval with the highest number of adhesions and with the highest total preference values: it asks to relax it only to the agents having previously rejected that date. If they accept, then they supply a new preference value for that time. Whether at least one of the them rejects the proposal, the step is applied again with the successive intervals until an agreement is reached. It is also possible that some of the participants decide not to participate any more, or that the organizer asks for a decision to the agents obstructing the negotiation. It is be noted that the process can cause side effects: a Meeting Agent can be constrained to cancel a previous engagement and so it has to start a new dialogue with the other agents involved in order to re-schedule it. Before taking a so critical decision, the agent asks for confirmation to its user. Also this step can be mechanized: for example, the user could have instructed his agent (or it can have learned such a rule) that the request of his boss have absolute priority with respect to any other appointment including personal engagements. We are starting to integrate this kind of features in Masma. When the organizer agent has a date, it inquires the Resource Agent to verify the availability of the sites. It could be necessary to change some features if the chosen place is not available. It could also be necessary to look for a new date. When the place has been booked, the Meeting Agent confirms the event to all the participants. They can interrogate the Travel Agent to decide how to get the place and the available hotels: the agent may reserve for them.

As often happens in multi-agent systems, also in Masma the control of interaction is delegated to the execution of a protocol that drives critical phases by avoiding conflicts.

Personalizing Behavior through User Profile

In order to mechanize the decision process and to limitate the interaction with the user, the agent maintains a user's profile with his availability and preferences and supplies tools to define and update the profile itself.

The profile controls the interaction user-agent by means of personalized information. Each meeting agent makes available to the user the possibility of setting a general profile that is considered as his standard behavior in meeting scheduling. The user's profile specifies the level of availability concerning the different time intervals: these values can be manually set by the user or deduced by the agent from the preference rules the user can define. The rules are described by a formal language that is hidden to the user by a graphical interface. Eventually all the profile is translated into rules. A first set of rules concerns user's habits respect to his agenda, while a second group regards his behavior respect to a person or groups of people. The rules are a way to personalized the agent: the rules described below are simple, but it is possible to define different types—for instance domain-independent rules—in order to better model the interaction and to restrict the passages of initiative from the agent to the user.

The first group of rules allows the user to assign a preference value {*high, medium, low, nil*} to each hour interval: the agent sets automatically to *nil* the dates in which appointments have already be fixed.

The agent can take over the calendar system in the environment, in our case the calendar manager of the UNIX systems. For the hour intervals the user does not have any appointments, the agent figures out the value according to the preference rules. At present, there are three different of rules it can apply:

1. The `Holiday-Rules` associate a preference value to any weekend day and holiday. So if the user likes to work on Sunday morning, he can assign the value "high" to this interval. The holiday are automatically extract by the calendar, but the user can also set his own holiday time.

2. The `Proximity-Rules` allow the user to give a preference value to a time interval around a previous meeting. For example, it is possible to specify the time distance between two different appointments in the same town, e.g., 2 hours, or in different towns, e.g., two days.

3. The `Fixday-Rules` can be used to define an availability value for the same day(s) of the week or for the same day(s) of a month in a specified time interval: for example, the user can be busy every Monday at 9 until 11 from June 1996 to July 1997 or every 15 in the same time interval.

Starting from the user's preferences, the calendar setting and the rules, the agent calculates the final value by applying an algorithm that also accounts for inconsistencies introduced by the user himself.

A second aspect concerns the possibility of forcing automated behavior when certain kind of meetings take place. We are currently addressing this aspect and so we give here the direction we are following. The idea consists of endowing the user profile also with information concerning given situations or given people. For example a rule could give absolute priority to meeting about a certain project or to any request coming from the boss.

The problem with this kind of rules is that are subjective: at present we are studying how to allow such a personalization but more experiments are needed. Even it could seem that there is an indirect connection with mixed initiative aspects, 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 move control only when strictly necessary, etc.

These mechanisms influence the "static" or usual evolution of the system, while how to constrain the dynamic behavior during a particular session is explained in next section.

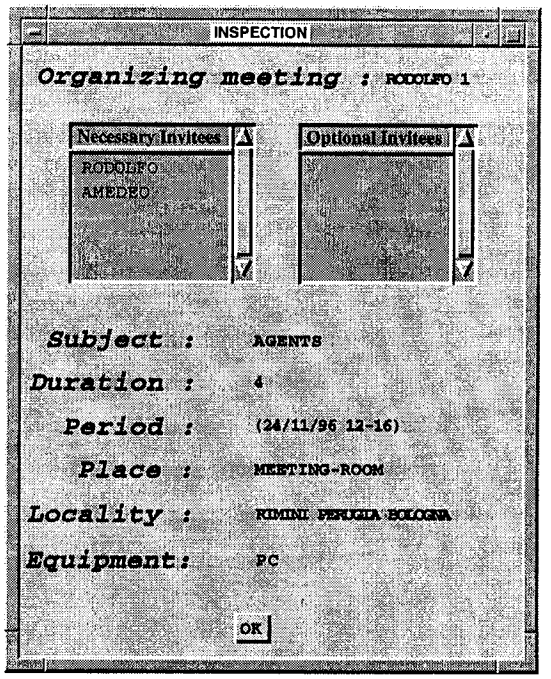

Figure 2: The Inspection Window

Allowing Control to the User

In Masma particular attention has been paid to issues that are essential to the user's acceptance of the system like: the possibility of inspecting the agents, the respect of the data privacy, the simplicity of utilization of the whole tool. A major drawback that the HCI community ascribes to agent technology is the user's loss of control about the activities of software agents. Indeed some reasons for this complain exist because of the limited research dedicated to examining the role of a software agent tool as a mixed-initiative interactive system opposite to a completely automated system. Although exciting, the possible autonomy of software agents should be held under control in systems that continuously interact with the user, in spite of the requested autonomy for agent-based active interface. An agent should be endowed with the capability of acting and autonomously proposing solutions according to the current problem, but the user must have the possibility of controlling and inspecting the agent decisions. To this purpose we have investigated on the *possibility of inspection* of the agent behavior from the user.

The agent-based system should be endowed with an inspection mode about its activities, should take the user into account in the most important decision processes and should be autonomous in the less critic decision steps: that will involve a major level of trust by the user.

Masma allows in its main dialogue window to verify the status of the agent, and then to inspect the details of the current activity. When no negotiation is running the agent is free, and it is busy when it is involved in a organizational process. In this case, a button is active that displays a window (Figure 2) allowing the user to analyze the running processes, to verify the information and data at agent's disposal and to interfere or take over the agent if necessary.

Currently the window shows only information concerning the being organized event; a new version is under way that displays also a graphical representation of the negotiation process. A graph will depict the agent's actions and decisions that have been undertaken to reach the current status.

The user can influence the organization and negotiation processes by dynamically modifying the preference and availability values. This is an implicit way of modifying the running process, but a more explicit and effective method will be available with the new version. The user and the agent will be able to engage a dialogue: the user communicates his problems to the agent that can suggest him the better strategy to accomplish his goals.

The inspection window does not interrupt the agent activity that goes on with the negotiation process. The change of the preferences can happen at any time independently from the achieved results: the job of recontract is left to the 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 the control.

Aspects Relevant to the User's Trust

From the point of view of user involvement, attention should be paid to the decision about *how* and *when* to interact with the user. Concerning *how* to interact, the system must be *easy to use.* The agents dialogue with their users by means of simple and effective windows that can be adapted to the current needs and that hide the complexity of the system. To be able to relieve the user's work, the interfaces should be few invasive and really discreet. Masma's interfaces satisfy these requirements.

The second point concerns *when* the agent calls for its user. The agent does not have to be invasive but at the same time it is active and takes decisions that could be critical. The personal agent is supposed to always involve the user in decisions but a system that continuously asks for confirmation would fail in its main task. That can happen independently from the possibility of taking over the agent. The meeting agent interrupts its user only for the most important—in user's view—decisions: this means that is the user who decides when to leave decisions to the agent. In the initialization phase, the agent describes to the user its different activities with the types of involved decision steps. The user can assign a weight to each step that measures the criticality of the connected decision in his view. Then a threshold is fixed the suggests the intervention level for the agent. It is to be noted that the most important decisions can be put outside the measurement to point out that the user is the only in charge of deciding.

Conclusions

In this paper we presented aspects of Masma, a multi-agent system devoted to interact with a set of users to help organizing meetings. We have focussed our attention on the interaction with the user in order to obtain an effective mixed-initiative system.

Starting from the fact that complete task delegation was not what we wanted from the multi-agent system, we developed strategies for: a) allowing the user to personalize the default behavior of the personal agent; b) subdividing tasks between user and personal meeting agent according to criticality of decisions; c) allowing the user to maintain control over the negotiation mechanism by inspecting his agent and dynamically modifying personal data.

Although solutions were developed in a particular system, the underlying problems may be considered general enough to be further investigated to achieve a methodological perspective.

References

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

Cesta, A.; D'Aloisi, D.; and Brancaleoni, R. 1997. Mixed-Initiative Agenda Management with Software Agents. In *Proceeding of the First International Conference on Autonomous Agents.*

Norman, D. 1994. How Might People Interact with Agents. *Communication of the ACM* 37(7):68–71.

Steiner, D.; Mahling, D.; and Haugeneder, H. 1990. Human computer cooperative work. In H.Huhns., ed., *Proceedings of the International Workshop on Distributed Artificial Intelligence.*

Whittaker, S., and Sidner, C. 1996. Email Overloading: Exploring Personal Information Management of Email. In *Proceedings of Conference on Human Factors in Computing Systems (CHI 96)*, 276–283.