User Modeling in the Design of Interactive Interface Agents

Michael Fleming and Robin Cohen*

Department of Computer Science,
University of Waterloo,
Waterloo, ON, Canada

Abstract. This paper presents a model for more interactive interface agents. This more interactive style of agents aims to increase the trust and understanding between user and agent, by allowing the agent, under certain conditions, to solicit further input from the user about his preferences and desires. With the user and agent engaging in specific clarification dialogues, the user's input is employed to adjust the agent's model of the user. Moreover, the user is provided with an ability to view this user model, under certain well defined circumstances. Since both the agent and user can take the initiative to interact, basic issues regarding mixed-initiative systems arise. These issues are addressed in our model, which also takes care to restrict the agent's interaction with the user, to avoid bothering the user unduly. We illustrate our design for more interactive interface agents by including some examples in the domain of electronic mail.

Keywords: user modeling agents, personalized and adaptive information assistants, mixed-initiative interaction

1 Overview

In recent years, the area of intelligent agents has been one of the most prevalent fields of research in the AI community. This paper deals with one specific type of agent, the interface agent, which is a program that acts as a personal assistant to a user dealing with a particular computer-based application, and which is able to "view" and act upon the application interface just as a human user might. Previous designs of interface agents can be broadly classified into two categories: autonomous agents (*e.g.*, Maes, 1994), which attempt to automate certain actions on behalf of the user, and collaborative agents (*e.g.*, Rich and Sidner, 1997), which are more equal partners with their users, working together on a joint plan and participating in a dialogue in order to determine an appropriate course of action.

We argue that there is a middle ground to be covered. Using autonomous learning interface agents as a starting point, we propose a model which makes these agents more interactive, allowing them to take the initiative to solicit further input from the user, toward improving their overall performance.

* This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

2 Background

In order to develop our model, we have used as a starting point the learning interface agent architecture developed by the Software Agents group at MIT. The following is a very brief description of how these agents operate; see Maes (1994) for more detail. The MIT agents act primarily by observing their users, and by using a form of learning called memory-based reasoning (Stanfill and Waltz, 1986). For each new situation that arises, the agent computes the distance between the current state and each of the past situations it has stored in its memory, using a weighted sum of several relevant features. According to the actions taken by the user in the most similar past situations, the agent selects an action for the current situation, and calculates a corresponding confidence value (Kozierok, 1993). According to "do-it" and "tell-me" thresholds established by the user, the agent determines whether to automate an action on the user's behalf, to suggest an action, or to do nothing at all. Figure 1 shows a simplified description of these learning agents.

PRIOR TO OPERATION: The user has set the tell-me and do-it thresholds, has indicated how many past situations the agent should look at during its action selection, *etc.*

INPUT: A signal that there exists a new situation to be addressed (*e.g.*, in the e-mail domain, a new mail message arrives, the user has just finished reading a message, etc.)

OUTPUT: The agent has completed an action on the user's behalf, has suggested an action, or has decided to do nothing for the current situation.

Select action A via learning techniques and assign confidence value C.
if $C >$ do-it threshold **then**

- perform action A and add it to a list of automated actions for user to examine at his own leisure
- if user indicates that action was incorrect, ask user to adjust priority weightings for the various features which contribute to calculations

else if $C >$ tell-me threshold **then**

- suggest action A

else

- consult other agents for help, establish suggested action A' and compute new confidence value C'.
- **if** $C' >$ do-it... (as above)
- **else if** $C' >$ tell-me... (as above)
- **else** do nothing

Figure 1. High-level algorithm for the behaviour of learning interface agents

3 More Interactive Interface Agents

While the MIT design has many strong points, several shortcomings can be identified (Fleming, 1998). In particular: (i) these agents do not deal very well with situations that are somewhat ambiguous; (ii) the lack of communication between agent and user makes it difficult for a user to understand and to trust such an agent; (iii) memory-based learning can be quite slow because it may require an examination of a large number of previous situations.

We address these issues and others, by presenting a variation on the architecture of the MIT learning agents. This new model allows for an agent which is still more or less autonomous, but which recognizes opportunities for asking the user for further information, with the goal of improving the agent's overall performance. The information which is solicited then becomes part of the agent's user model, to be used in future interactions with the user. A very high level algorithm for our semi-autonomous agents is shown in Figure 2. A few major points of the algorithm are explained in this paper, illustrated for the domain of assisting users with e-mail. It is important to note that the algorithm is general enough to operate in a variety of application areas. The central decisions made are when to solicit input from the user and how to manage the agent's rule base in such a way that the user can contribute to its specification.

PRIOR TO OPERATION: The user has set the tell-me, do-it and bother thresholds, has indicated how many past situations the agent should look at during its action selection, *etc.*

INPUT: A signal that there exists a new situation to be addressed (*e.g.*, in the e-mail domain: a new mail message arrives, the user has just finished reading a message, *etc.*)

OUTPUT: The agent has completed an action on the user's behalf, has suggested an action or has communicated to the user that it can do nothing for the current situation.

(0) Consult rule database for applicable rules previously created by the user (with or without the agent's help). If a single rule is found to apply, then use that rule. If two or more conflicting rules are found, initiate rule conflict dialogue with user. If no rules are found to apply, then proceed with step 1.

(1) Use learning techniques to get possible actions A_1, ..., A_n

(2) **if** choice of action A is clear[a] **then**

> (3) Compute confidence value C (as in the MIT agents – see Kozierok (1993), for example)
>
> (4) **if** $C >$ do-it threshold **then** perform action A and indicate that there is a proposed rule for the user to approve/reject/edit
>
> (5) **else if** $C >$ tell-me threshold **then** suggest action A

(6) **else** //choice unclear because two or more actions have similar scores

> (7) **if** peer agents exist and are able to provide trustworthy advice **then** automate/suggest recommended action
>
> (8) **else** // choice still unclear
>
> > (9) Compute clarification factor CF.
> >
> > (10) **if** $CF >$ user-defined bother threshold **then** initiate dialogue with user.

[a] The choice is considered clear if the score computed for the highest-scoring action exceeds the score of the next best choice by a constant difference threshold (say, 10%).

Figure 2. High-level algorithm for our more interactive interface agents

3.1 Ambiguous Situations

A key circumstance which suggests the value of user input is that of ambiguous situations: cases where the agent, via its learning methods, is unable to select one course of action as being a clear winner. (See steps 6-10 in the algorithm.) For example, in the e-mail domain, suppose an agent has successfully learned that all messages from David Fleming should be filed in the *David* folder and that all messages with subject "Hockey pool" should be filed in the *Hockey* folder. What will the agent do with a message from David Fleming with subject "Hockey pool"?

Suppose a message with the following feature values has just been read:

Feature	From	Cc	Date	Subject
Value	David Fleming	None	October 26	Hockey pool

Suppose also that the agent has assigned the following weights to each of the relevant fields, based on how well the current situation's value in each of those fields has typically predicted the action taken (as in Kozierok, 1993).

Feature	From	Cc	Date	Subject
Weight	0.90	0.08	0.01	0.88

Finally, suppose that the following four messages were found to be the most similar to the current situation, with the distance between the value in the current situation and the corresponding value in the past situation shown in the third row. The overall distance between two situations (shown in the fourth row) is computed by taking the sum of the products $d_i w_i$, where d_i is the distance between the values of field i and w_i is the weight assigned to field i.

Feature	From	Cc	Date	Subject
Value	David Fleming	None	October 11	Habs
Distance	0	0	0.90	0.98
$\Delta(s_{new}, s_1)$	0.8714			
Action	File under David			

Feature	From	Cc	Date	Subject
Value	David Fleming	None	October 3	Hi
Distance	0	0	0.92	1
$\Delta(s_{new}, s_2)$	0.8892			
Action	File under David			

Feature	From	Cc	Date	Subject
Value	Owen Barnhill	None	October 7	Hockey pool
Distance	1	0	0.86	0
$\Delta(s_{new}, s_3)$	0.9086			
Action	File under Hockey			

Feature	From	Cc	Date	Subject
Value	S. Fillmore	None	October 23	Hockey pool
Distance	1	0	0.90	0
$\Delta(s_{new}, s_4)$	0.9090			
Action	File under Hockey			

In such a situation, MIT's *Maxims* (Metral, 1993) e-mail agent would compute scores for each of the two candidate actions (*File under David* and *File under Hockey*), would choose the action with the higher score and would calculate a confidence value. In this case, the scores for the two actions would be very close together; the agent would choose filing the message in the *David* folder but would have a very low confidence value. As a result, this agent would likely do nothing in such a situation. It would be the responsibility of the user to realize that nothing had been done, and to perform an appropriate action himself. It is important to note that the autonomous agents (as in Maes, 1994) will not even suggest an action, if there is low confidence. A user has the responsibility of performing any required actions which are simply left unaddressed by the agent.

Our more interactive agent, on the other hand, would examine the same situation and recognize that two candidate actions have similar scores. Based on how close together the scores are, along with a number of other factors,[1] the agent will compute a *clarification factor*. This clarification factor is then compared to a user-defined *bother threshold* to determine whether or not to initiate a clarification dialogue with the user. The goal of such a dialogue is to find out which action is most appropriate in this situation and to attempt to generalize this into a rule. Possible actions are provided, along with explanations, which serve as an encapsulation of the learning algorithm which led the agent to consider these actions. These explanations essentially provide the user with an understanding of the underlying user model which the agent is proposing – they show what the agent has determined to be the user's preferences, based on past actions. An example screen is presented below:

Situation: `The following message has just been read.`

From	Cc	Date	Subject	...
David Fleming	None	Oct. 26	Hockey pool	...

Possible actions:

Action	Score	Explanation
File under David	2.272	In past situations in which the sender was David Fleming, the action taken was *File under David* in 95% of cases.
File under Hockey	2.201	In past situations in which the subject was ''Hockey pool'', the action taken was *File under Hockey* in 100% of cases.

`Please click on the action you wish to choose, or click` `Cancel` `to conclude this interaction.`

---

[1] These factors include how "important" the agent considers the candidate actions to be (based on the do-it thresholds (Maes, 1994) established by the user for those actions) and how often the user has been bothered recently. We omit the presentation of the actual formula in this short paper.

If the user were to choose the action *File under Hockey*, for example, the agent would proceed to propose two rules, as seen in Figure 3. The first states specifically that when the subject line is "Hockey pool" and the message sender is David Fleming, the message should be filed in the *Hockey* folder. The second rule is more general, and states that any messages with subject line "Hockey pool", regardless of the sender, should be filed in the *Hockey* folder. The user has the option of accepting or editing either of these rules, or of cancelling the interaction entirely if neither rule is appropriate. When the user approves a rule, this rule is then employed by the agent in future interactions and the agent updates the model of the user's preferred actions.

Possible rules:

`1`	**Subject:**	`Hockey pool`	**Action:**	`→ Hockey`
	From:	`David Fleming`		`ACCEPT`
				`EDIT`
`2`	**Subject:**	`Hockey pool`	**Action:**	`→ Hockey`
	From:	`*`		`ACCEPT`
				`EDIT`

`REJECT ALL RULES`

Figure 3. Agent's proposal of possible rules

Even in cases in which the user is not immediately bothered by the agent (*i.e.*, the clarification factor does *not* exceed the bother threshold), the agent can indicate that it has a question for the user without actually requiring the user to deal with it immediately. To achieve this interaction, we propose having the agent maintain a "question box" where it would store information about situations with which it could benefit from the user's help, but for which it chose not to interrupt the user immediately due to a low clarification factor. This question box would appear in the interface as a small box in the lower left corner of the screen, indicating how many questions the agent currently had. The user could choose to click on this box at his own convenience, in order to initiate dialogues of the form presented earlier.

This feature is incorporated into our model to explicitly allow both the user and the agent to initiate interactions. The user is essentially provided with an opportunity for finding out more about the proposed actions of the agent and the underlying user model which leads to these proposals, at a time which is convenient to the user.

3.2 Rule Base

Another novel aspect of our algorithm, as compared to the learning interface agents developed at MIT, is its incorporation of truly hard-and-fast rules into the agent's behaviour. An example of such a rule, from the e-mail domain, might be "If a message arrives with subject line 'Make money fast', then delete it." Rules can either be programmed by the user, or developed and proposed by the agent when it has high confidence in a prediction (as in Step 4 of Figure 2). Although the MIT group does provide "rules" for its agents, these rules are simply represented as hypothetical situations, and are treated just as though they were situations the agent had actually observed in the past. In any new situation, an agent would still have to examine each past situation in its memory and go through a series of calculations. Our proposal is for the agent to maintain an entirely separate database of rules, which can be fired immediately whenever an appropriate situation is encountered.

We believe that the incorporation of rules is a necessary addition for two main reasons: (1) it will likely speed up the agent's performance[2] in situations where it can simply apply a rule, rather than going through a series of complex calculations involved in the agent's learning algorithm; (2) because rules are more explicit and concrete than the calculations involved in learning techniques, having a separate rule base which is always available to inspect would help to provide the user with a better understanding of, more trust in, and a better sense of control over, the agent's behaviour. Our agents also allow for agent-user communication in the event of conflicts occurring in the actual rules programmed by the user (Step 0). This communication is not through natural language, but rather via dialogue boxes, menus and buttons in a graphical user interface. Fleming (1998) presents examples illustrating dialogues to address such rule conflicts.

4 Reflecting on Initiative

The design outlined in Section 3 allows for both the agent and the user to take the initiative and can therefore be classified as a mixed-initiative AI system. Allen (1994) and Burstein and McDermott (1996) identify several important issues which must be addressed when designing mixed-initiative systems, including: (i) specification of when exactly the system and user should communicate, and what that communication should look like; (ii) registration of context when one party interrupts the other; (iii) ensuring that both parties share the responsibilities involved in the task, and are fully aware of the responsibilities of each party.

For the particular application of interface agent design, our model addresses each of these issues. An algorithm is presented for determining when an agent should choose to initiate communication with the user, and details are given about the format of this interaction. Registration of context is also taken into consideration in the model. Whenever the agent interrupts the user, it must take care to set the stage for what exactly it wishes to ask the user. For instance, in the example presented earlier, the agent registers the context by establishing that the user has just finished reading a message which the agent does not know how to treat, and by providing the user with the exact features of that particular message. In our model, the agent and user share responsibilities quite well, and should always be aware of who is responsible for what tasks. Upon

[2] Note that, in practice, the actual gain in performance by using a rule-based approach would depend strongly on the size of the rule base and on the format used to represent rules.

encountering any new situation, it is understood that the agent will attempt to do whatever it can to perform an action for the user (or to make a suggestion) using the knowledge it has previously acquired. If it has insufficient information to do anything, it will still be able to inform the user by adding messages to the question box discussed earlier.[3]

Among other things, the agent in our model can take the initiative to clarify ambiguous situations, to ask for contradictory rules to be clarified, to propose generalizations of rules specified by the user, to propose rules when its confidence is high enough, to maintain the question box and to indicate to the user when new items have been added. The user can take the initiative to set threshold values, to react to proposed rules from the agent, to accept or reject the agent's proposed generalizations of rules, and to click on the agent's messages in the communication column or in the question box in order to initiate dialogues.

5 Discussion

This research has obvious comparisons to previous work on interface agents, which was drawn out somewhat in Sections 2 and 3. In situations where other agents (*e.g.*, Maes, 1994) would be unable to propose an action for a user (and would therefore rely on the user to act), our agents would engage in clarification, resulting in an action taken for the user. As the user communicates with the agent, a better understanding of the agent's operations is gained. Yet, there is still an opportunity for autonomy on the part of the agent, so that there is less burden on the user to "program" the agent's every move. There is also important value to mapping out the circumstances under which agent and user can take the initiative to act, as discussed in Section 4.

Our model also suggests some valuable new directions for user modeling. The user model in our kind of application is simply a record of the user's past actions (which can be surveyed at any time, to find possible patterns of similarity with the current situation), together with a critical rule base which captures the general rules the agent has developed, to characterize the user's preferences, from previous learning episodes. The clarification dialogues which are introduced essentially provide the user with the opportunity to view the system's user model and to directly propose changes to that model. Typically, user models have either been acquired implicitly (by inference) or explicitly (from some kind of interview process) and have changed on the basis of observation of the user (as discussed in Kobsa and Wahlster, 1989). The style of interaction which we have developed allows the user a more active role in the ongoing maintenance of the user model. Providing users with this role as an option, carefully administered so as not to overburden, is the best method of engaging the user, in our opinion.

Other work which has investigated the use of user models in interfaces includes Thomas and Fischer (1996). Here, a user model is maintained to assist users in browsing the Web. However, the user model is essentially acquired implicitly, on the basis of the user's actions. Our approach is somewhat more in line with that of McCalla et al. (1996), which allows users to change a case library, to influence the user model which is maintained, for applications of information filtering. In our model, the opportunities for the user to influence the agent are well specified and

[3] Fleming (1998) discusses other methods for communicating with the user as well. For example, it is possible to use a separate "communication column" in the display of all e-mail messages in a mailbox, which records the current status of that message with respect to the agent's processing.

constrained to a clarification dialogue, so that both parties are aware and can build up trust and understanding between them.

Cesta and D'Aloisi (1998) have also discussed the value of mixed-initiative interaction between users and agents. Their MASMA meeting scheduler is in fact a multi-agent system, where users define and maintain their own user profile (so that this information is not learned by the agents). Then, depending on the criticality of the task, agents may interact further with users. Users are also able to control and inspect their agents, at any time. This work therefore suggests a somewhat different role for users, but reinforces the hypothesis that it is important for users to know and trust their agents.

In a similar vein, in Akoulchina and Ganascia (1997), the user is allowed to create hypothetical rules to direct the agent, which is also a part of our model. However, the user is required to make all the final decisions, so the agent has less opportunity for autonomy, compared to our agents.

For future work, in our model, it may also be useful to track which rules the agent and the user have discussed, to possibly influence the form of future communication about these rules. For application areas such as recommending Web pages (see Fleming and Cohen, 1998), it may be more critical to track previous interactions. A useful reference here is Maglio and Barrett (1997), which suggests displaying the user's past interactions in a condensed representation, to facilitate the user's understanding of the agent's user model.

In summary, we have presented a model for designing autonomous, interactive agents. These agents make an effort not to bother indiscriminately, but provide their users with a view of the user model which underlies their operation and, in so doing, offer increased reliability.

References

Akoulchina, I., and Ganascia, J.-G. (1997). Satelit-agent: An adaptive interface based on learning interface agents. In *Proceedings of the Sixth International Conference on User Modeling*, 21–32.

Allen, J. (1994). Mixed-initiative planning: Position paper. Presented at the ARPA/Rome Labs Planning Initiative Workshop. Available on the World Wide Web at http://www.cs.rochester.edu/research/trains/mip.

Burstein, M., and McDermott, D. (1996). Issues in the development of human-computer mixed-initiative planning. In Gorayska, B., and Mey, J., eds., *In Search of a Humane Interface*. Elsevier Science B.V. 285–303.

Cesta, A., and D'Aloisi, D. (1998). Mixed-initiative issues in an agent-based meeting scheduler. *User Modeling and User-Adapted Interaction (to appear).*

Fleming, M., and Cohen, R. (1998). Interactive interface agents as recommender systems. In *Papers from the AAAI 1998 Workshop on Recommender Systems*, 42–46. AAAI Press.

Fleming, M. (1998). Designing more interactive interface agents. Master of Mathematics thesis, University of Waterloo, Waterloo, Ontario.

Kobsa, A., and Wahlster, W., eds. (1989). *User Models in Dialog Systems*. Springer-Verlag.

Kozierok, R. (1993). A learning approach to knowledge acquisition for intelligent interface agents. Master of Science thesis, Massachusetts Institute of Technology, Cambridge MA.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM* 37(7):31–40.

Maglio, P., and Barrett, R. (1997). How to build modeling agents to support web searchers. In *Proceedings of the Sixth International Conference on User Modeling*, 5–16.

McCalla, G., Searwar, F., Thomson, J., Collins, J., Sun, Y., and Zhou, B. (1996). Analogical user modelling: A case study in individualized information filtering. In *Proceedings of the 5th International Conference on User Modeling*, 13–20.

Metral, M. (1993). Design of a generic learning interface agent. Bachelor of Science thesis, Massachusetts Institute of Technology, Cambridge MA.

Rich, C., and Sidner, C. (1997). Collagen: When agents collaborate with people. In *First International Conference on Autonomous Agents*, 284–291.

Stanfill, C., and Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM* 29(12):1213–1228.

Thomas, C., and Fischer, G. (1996). Using agents to improve the usability and usefulness of the World-Wide Web. In *Proceedings of the 5th International Conference on User Modeling*, 5–12.