A User-Adaptive Interface Agency for Interaction with a Virtual Environment

Britta Lenzmann and Ipke Wachsmuth

University of Bielefeld
Faculty of Technology
AG Knowledge-Based Systems
D-33501 Bielefeld, Germany
{britta,ipke}@techfak.uni-bielefeld.de

Abstract. This paper describes an approach to user adaptation realized in a multiagent interface system for interaction with a virtual environment. The interface agency adapts to users' individual preferences by learning from direct feedback. The core idea is that agents that were successful in meeting the user's expectations are given credit while unsuccessful agents are "discredited." Communicating credit values, agents organize themselves so that the overall behavior of the interface agency gradually adapts to the individual user as the session is proceeding.

1 Introduction

Agent systems have proven especially useful in the design of more intelligent user interfaces. By allowing more human-like communication forms, they can add comfort in human-computer interaction [Laurel, 1990]. Beyond this, agent systems may act as mediator between the user and the application system [Maes, 1994]; [Wachsmuth & Cao, 1995]. The user can instruct the application system by way of abstract commands (virtual interaction), and the interface agency interprets them (intuitive communication) and transmits the results to the application system via technical communication (see figure 1).

While communication between humans is situated naturally, the interface system must be able to meet varying conditions to enable an effective human-computer interaction. Thus, incorporating adaptation facilities in the agent system becomes essential. In our work, we distinguish two aspects of adaptation: adaptation in respect to individual differences across users and adaptation to varying situation circumstances. In this paper we focus on the first aspect, i.e., user adaptation.

A prominent approach is to build user-adaptive interface agents by applying machine learning techniques. For example, [Maes & Kozierok, 1993] use techniques such as learning by observing the user, learning from direct or indirect user feedback, or learning from examples given by the user. Applying these techniques, [Maes, 1994] describes learning personal assistants, e.g., for electronic mail handling and electronic news filtering which accumulate knowledge about tasks and habits of their users to act on their behalf. Similarly,

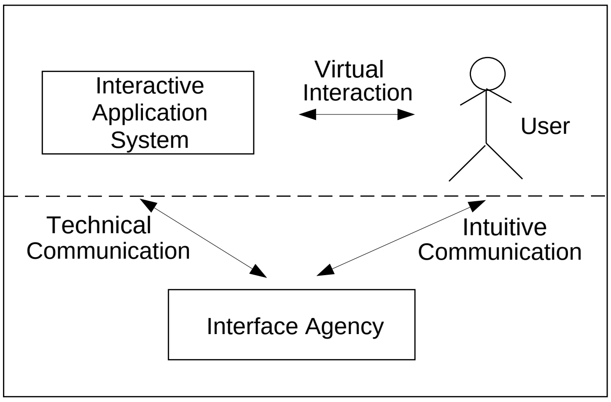

Fig. 1. Agent-mediated interaction: Software interface agents free the user from the burden of knowing and communicating technical detail.

[Mitchell et al., 1994] have built learning apprentices for calendar management, electronic newsgroup filtering, and email negotiation which automatically customize to individual users by learning through experience. In these applications, a visual agent gives advice to the user by expressing facial emotions or by prompting suggestions. Common to these approaches is that a solitary interface agent is used which adapts to individual preferences of its user by aquiring user data and changing its internal functionality accordingly.

In our approach, we consider a system of *multiple* interface agents which adapts to user preferences by learning from direct feedback without explicit acquisition of user data. The user gives an implicit feedback by way of correcting solutions offered by the agency until the agent generating the preferred solution becomes dominant in the system-user interaction. Thus, internal functionalities of agents can remain unchanged but individual agents are preferred from among a variety of heterogeneous interface agents. The core idea is that agents which were successful in meeting the user's expectations are given credit while unsuccessful agents are "discredited." By this, the overall behavior of the interface agency gradually adapts to the individual user as the session is proceeding.

Our approach is realized in a multiagent interface system for interaction with a virtual environment, and is carried out in the VIENA project. We start with explaining the VIENA system, describe then our approach to user adaptation by a multiagent system and, in concluding, discuss our results and sketch future work.

2 VIENA: Interaction with a Virtual Environment by a Multiagent System

VIENA[1] ("Virtual Environments and Agents") is a research project concerned with the interactive manipulation of high-quality 3D graphical scenes by way of natural language input [Wachsmuth & Cao, 1995]. A multiagent system translates qualitative verbal communications of the user to quantitative, technical commands that are used to update the visualization scene model. To this end, the multiagent system has to solve different tasks which are distributed among a number of specialized agents (figure 2). A parser translates an instruction to an internal deep-level representation which outputs to the mediating agents. Agents take special responsibilities in mediating an instruction. For example, a space agent translates qualitative relations such as 'left of' to appropriate scene coordinates. A bookkeeping agent is authorized to access and modify the augmented data base to supply current situation information to agents on request. Some of these agents are actually agencies, that is, they incorporate agents of the same type but slightly different functionality. For instance, this is the case for the space agent discussed in section 3. Typically, such agents compete in the allocation of sub-tasks.

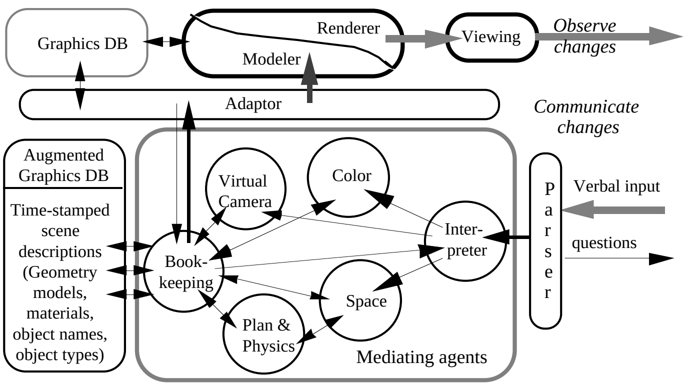

Fig. 2. The architecture of the VIENA system (after [Wachsmuth & Cao, 1995])

For computing the entire solution of a user's instruction agents have to communicate and cooperate with each other. Defining agents as autonomous, heavy-weight processes which can be installed on different computers in the network, communication is realized by message exchange. A Client-Server concept

[1] The VIENA project is part of the "Artificial Intelligence and Computer Graphics" research program at the University of Bielefeld and is partly supported by the Ministry of Science and Research of the Federal State North-Rhine-Westphalia under grant no. IVA3-107 007 93.

was designed which formally defines the agents as clients, and a communication subsystem as server to handle the messages. The cooperation method is basically characterized by a negotiation process similar to the contract-net approach [Davis & Smith, 1983]. Each agent can take on the role of a contractor as well as the role of a bidder. In detail, the process consists of a sequence of message passing operations which are: the posting of tasks, the generation of bids, the allocation of tasks or the rejection of bids, resp., and the return of computed results. In addition, a master-slave and a blackboard type of behavior can be modeled by allocating tasks directly, or, resp., by addressing tasks to groups of agents simultaneously. A more detailed description of the VIENA multiagent system is given in [Lenzmann et al., 1995].

The multiagent system is used in a prototype scenario with various items of furniture as well as color and light impressions of a virtual office room. A sequence of possible interactions with the VIENA system is shown in figure 3. The inputs shown allow interactive modification of the visible scene. Furthermore, changes of the viewpoint and the processing of simple deictic instructions are possible[2].

1. move the chair to the left .
2. a bit less .
3. turn the chair left .
4. put the desk behind the chair .
5. put the palmtree on the desk .
6. move the palmtree to the left .
7. put the bowl on the desk .
8. put the chair on the table .
9. move the chair to the window .
10. turn the desk right .
11. move the chair to the front of the desk .
12. put the bowl on the ground .
13. put the palmtree on the floor .
14. show the palmtree .
15. show the shelf .
16. move the chair to the right of the desk .
17. move the chair here .
18. put the plant between the desk and the chair .
19. display the plant .
20. point to the plant .
21. put the bowl there .

Fig. 3. Sample sequence of interactions with the VIENA system

Since the computed solutions do not always meet the expectations of the user, the VIENA system accepts corrections such as 'a bit less' which modify the previous solution accordingly by inspection of the current and the previous scene models which are stacked for this purpose. In this way, the semantics of

[2] Such example interactions are part of a demo video presented in the IJCAI-95 videotape program [Cao et al., 1995].

instructions can be negotiated in the system-user discourse; we refer to this as "negotiated semantics."

However, frequent corrections are uncomfortable for the user. Moreover, the practical experience with the system has shown that variations of individual preferences exist across users which call for expanded internal functionalities of the interface agents. This gave rise to the idea of incorporating adaptation facilities in the interface agency. In section 3, we explore this idea in more detail and describe an approach where user adaptation is achieved by learning from direct feedback.

3 User-Adaptation in a Multiagent Interface System

Our aim is that user adaptation be achieved without the need to acquire knowledge about tasks and habits of the user or to accumulate explicit user models. Avoiding explicit user modeling seems a desirable goal in several respects. Explicitly acquired information is less likely to change on the short term whereas implicit information is more dynamic as it is usually acquired incrementally during the course of a session, and dynamic models are more useful as they adapt to changing characteristics of users [McTear, 1993]. Moreover, there are social and legal issues with explicit user models since they will contain a great deal of personal information; for this reason, explicit user models have found critique with respect to privacy of user data [McTear, 1993]; [Norman, 1994].

The core idea of our approach to *implicit user adaptation* is that agents of the same type but slightly different functionality — corresponding to possible variations of users' preferences — organize themselves to meet the preferences of the individual user. Getting positive or negative feedback from the user, agents increase or decrease their amount of selfconfidence, so that successful agents become dominant in the ongoing session. In detail we present the learning technique in section 3.2. Before that, we describe the problematics of using different spatial reference frames when transforming objects in the virtual scene as an example application of our approach to user adaptation.

3.1 Users' Preferences for Different Spatial Reference Frames

In VIENA spatial transformations of scene objects are communicated by way of qualitative verbal instructions, as in 'move the palmtree to the left' (cp. instruction 6 in figure 3). The semantics of such spatial instructions may depend on different perspectives [Retz-Schmidt, 1988]: from the user's point of view (deictic perspective) or from the point of view of an object which has a prominent front (intrinsic perspective). In addition, users' preferences for spatial reference frames may depend on the orientation of the desk given in the actual situation. This 'situated' aspect of preferring one spatial reference frame over the other is a further research subjective but not the focus of this paper. Figure 4 illustrates the two alternative solutions when an object located on a desk is to be moved to the left.

Fig. 4. Example scene from the VIENA test application: The palmtree located on the desk can be moved to the left from an intrinsic perspective (1) or from a deictic perspective (2).

The first VIENA prototype system just offered the possibility of transforming objects from the deictic perspective. When we demonstrated the system to a number of test users, some of them mentioned that they expected the palmtree to be moved to the left from the intrinsic perspective. We then assumed that, depending on their individual preferences, users may choose one of either perspective. For verification of this assumption, we carried out an empirical study. A total of 64 probands were asked to perform the instruction 'move the object to the left' in a simplified setting of the one presented in figure 4. The results showed that 36% of the probands used the intrinsic perspective (solution 1), whereas 64% used the deictic perspective (solution 2). Hence, we could substantiate that designing space agents able to adapt to users' preferences for different spatial reference frames is a useful goal.

Consequently, we conceptualized two instances of a space agent that are similar in the way they compute spatial transformations, and different with respect to the reference scheme they take on. More concrete, we have implemented one space agent embodying the user's egocentric reference frame (deictic reference) and one space agent embodying an externally anchored reference frame (intrinsic reference).

Instructing the system with a spatial transformation, the space agent which is currently dominant, e.g., the deictic agent, offers a possible solution. In case the visualized solution does not meet the user's expectation, the user can correct the system by stating 'wrong'. The negative user feedback leads the agents to reorganize themselves in a way that the intrinsic space agent can now generate an offer which modifies the previous solution. By this, adaptation to a user's preferred reference scheme is achieved by direct user feedback.

3.2 Learning from direct feedback

Extrapolating from this example application of spatial transformations, the interface agency as a whole, including the space agency, adapts to users' preferences by learning from direct feedback. In more detail, direct feedback is derived from implicit positive or explicit negative feedback. Implicit positive feedback is given when a user's instruction is followed by any instruction which does not decline the previous one. Explicit negative feedback is given when the user corrects the visualized solution computed by the interface agency. Consider the following sequence of example instructions:

1. *put the palmtree on the desk.*
2. *move the palmtree to the left.*
3. *wrong.*
4. *turn the desk right.*

Since the second instruction does not directly refer to the first one it can be interpreted as positive feedback regarding the previous instruction. The correction within instruction 3, on the other hand, indicates a negative feedback in reference to the second one. While user feedback regarded from the point of view of the entire interface agency happens more or less directly, single agents learn by indirect feedback since users' instructions are decomposed in subtasks and distributed among the agents.

From the system internal point of view, the adaptation process is achieved by a form of *reinforcement learning* [Kaelbling, 1993]. Learning is realized in a way that the system will take actions that maximize the reinforcement signals received from the environment. In our approach, this means that users' instructions (or corrections, resp.) represent reinforcement signals which are interpreted and encoded by the interface agency in the form of *credit values*. Each agent stores a credit value corresponding to its quality ("strength") at discrete periods of time. Learning is achieved by adjusting agents' credits in correspondence to the users' feedback and assigning those agents which are eligible for the task in question and have maximal credits.

In more detail, the process consists of several steps. When the user gives an instruction, the interpreter agent (cp. figure 2) determines which agents (subagency) are eligible to solve the task and informs the corresponding ones by sending a task posting. Depending on the task description, each of these agents generates a bid which includes its actual credit value and sends it to the contractor. All received bids are pooled and evaluated by comparing their associated credit values. The agent offering the best bid, that is, the bid with the highest credit, gets the task whereas the bids of the other agents are rejected. Figure

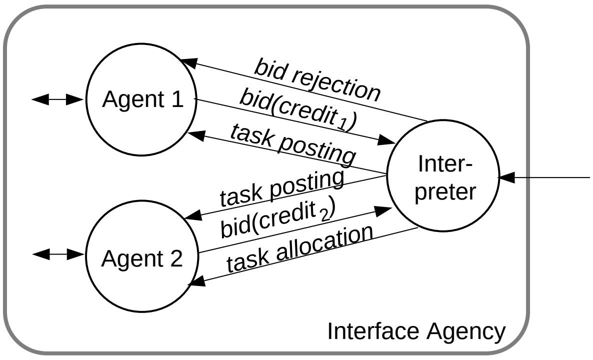

Fig. 5. A detail of the VIENA coordination strategy: both agents make offers qualified by their current credit values; agent 2 generating the better bid gets the task, whereas the bid of agent 1 is rejected.

5 illustrates a detail of the VIENA coordination structure where two agents compete with each other to compute a solution regarding the user's instruction.

For adapting to users' preferences, agents have to adjust their credit values dynamically while the session is going on. Adjustment takes place if a user corrects the system (cp. the example instructions above). A correction can come about because a different user now works with the system, or the same user undergoes short-term change of his/her preferences. In this case, the interpreter informs the corresponding agents about the correction by generating a task posting which includes a label indicating a unsatisfactory solution. Receiving this message, each potential bidder checks to see if it has worked out the previous task and because of that has caused the unsatisfactory solution. The agent which has worked out the task then makes the bookkeeping agent reset the database and reduces its own credit value whereas the other agents increase their credits (cf. section 4). Having modified their credit values, each of these agents generates a bid with changed credit value. Again, the agent with the currently best bid gets the task whereas the bids of the other agents are rejected (cp. figure 5). In this way, adaptation to user preferences, even those dynamically changing, can be realized.

Speaking metaphorically, increasing or reducing credit values induced by being successful/nonsuccessful in meeting the user's expectations corresponds to agents being more or less selfconfident. Based on their selfconfidence, agents are able to organize themselves in the way individual users' preferences call for without the need to accumulate explicit user models.

4 First Results

A prototype version of the adaptation method described above has been implemented and tested for the case of users' preferences for different spatial reference frames. As described in section 3.1, we have implemented two space agents, the deictic and the intrinsic space agent, which correspond to the possible variations of users' preferences regarding spatial reference frames.

In implementing our approach, we had to decide how to initialize and how to modify credit values of both space agents. In our first implementation, the following simple heuristics is used:

1. The deictic space agent has a initial credit value of *two* whereas the intrinsic space agent has a value of *one*.
2. On negative feedback, both agents modify their credit values in the way that the agent which worked out the previous task decrements its value by *one* whereas the other agent increments its value by *one*. On implicit positive feedback, credit values remain unchanged.

Regarding the first aspect, initial credit values of both agents are chosen differently, that is, biased. Same values would imply that either agent could be elected at random (depending on network communication speed) whenever various interactions without negative feedback are given in the preceding session. The decision on initializing the credit value of the deictic space agent higher than the one of the intrinsic agent is based on results of the empirical study (cp. section 3.1) where probands used the deictic perspective more frequently than they used the intrinsic perspective.

Credits are modified as explained in the second aspect because the interface agency can, in this way, immediately alter its bias for reference frames. The modifications of credit values can be illustrated by considering the second and third instruction of the example interactions presented in section 3.2. Assuming the deictic space agent is the dominant one in this situation, both space agents generate the bids illustrated in figure 6 when receiving the task posting induced by instruction 2.

Message-ID: <M2> <20>
Sender: <Viena> <Space> <1>
Recipient: <Viena> <Interpreter> <1>
Type: <Bid>
Reference: <M2> <19>
Time-stamp: <607>
Solution-time: <10>

<credit> <2> <ADR> <Camera>
<COM> <move> <REF0> <palmtree> <LOC> <left>

(a)

Message-ID: <M2> <21>
Sender: <Viena> <Space> <2>
Recipient: <Viena> <Interpreter> <1>
Type: <Bid>
Reference: <M2> <19>
Time-stamp: <610>
Solution-time: <10>

<credit> <1> <ADR> <Camera>
<COM> <move> <REF0> <palmtree> <LOC> <left>

(b)

Fig. 6. Each space agent sends a bid message comprising its current credit value and the task description. Whereas the deictic agent (a) offers a task solving qualified by a credit of *two* <2>, the intrinsic agent (b) makes an offer with a credit of *one* <1>. Note that the interpretation of <left> will be different depending on which agent is allocated the task.

Indicating a difference between the solution produced and the user's preference, instruction 3 induces a modification of credit values. Figure 7 shows the bid messages sent by each space agent to the contractor after updating its credit value.

```
Message-ID:  <M3> <33>
Sender:      <Viena> <Space> <1>
Recipient:   <Viena> <Interpreter> <1>
Type:        <Bid>
Reference:   <M3> <32>
Time-stamp:  <848>
Solution-time: <10>
-------------------------------------------
<credit> <1> <ADR> <Camera> <COM> <wrong>
```

(a)

```
Message-ID:  <M3> <35>
Sender:      <Viena> <Space> <2>
Recipient:   <Viena> <Interpreter> <1>
Type:        <Bid>
Reference:   <M3> <32>
Time-stamp:  <854>
Solution-time: <10>
-------------------------------------------
<credit> <2> <ADR> <Camera> <COM> <wrong>
```

(b)

Fig. 7. Having received negative user feedback, both space agents generate bids including modified credit values and a label indicating an unsatisfactory solution. Whereas the deictic agent (a) has decreased its credit value to *one* $<1>$, the intrinsic agent (b) has increased it to *two* $<2>$.

By this simple procedure, adaptation to varying users' preferences for different spatial reference frames can be achieved. An alternative approach to be investigated and implemented next is to increase credit values incrementally whenever the user gives a positive feedback. This corresponds to agents becoming more selfconfident by having solved tasks in the expected manner. Furthermore, a more complex relationship between agents and subagencies could be taken into account. However, alterations of users' preferences have to be realized in a slightly different way: agents decrease their credit values, on the one hand, but, inform their contractor of working out a task corrected by the user in the preceding interaction, on the other hand.

5 Discussion and Future Work

In this paper we presented an approach to user adaptation by a multiagent system for interacting with a virtual environment. Learning from direct feedback is used to adapt the interface agency to users' individual preferences. While the session is continuing, more successful individual agents within the agency are preferred. Agents organize themselves by communicating credit values which represent their amount of selfconfidence. Depending on their success in the preceding session, agents adjust their credit values dynamically to meet the user's expectations. The system's knowledge of user preferences is expressed in credit adjustments of agents and is distributed across agents. Thus, user adaptation is achieved without accumulating explicit user models.

We illustrated our approach by focusing on users' preferences for different spatial reference frames. As further preferences which the agent system could adapt to, differences in color perception as well as differences in strength regarding transforming or scaling objects will be investigated. Furthermore, we have considered the interaction with a virtual environment as an example application but, in our view, the approach seems also applicable in other scenarios where a multiagent interface is used to mediate between users and an application system.

Besides of verifying our techniques by considering other kinds of preferences, we think of more global measures to optimize the adaptive behavior of the interface agency. On the one hand, frequent corrections at starting time of a session could be avoided by appropriate initializations of agents' credit values. Our idea is that "anonymous user profiles" (replacing the idea of stereotypes [Kobsa & Wahlster, 1989]) could evolve, and be pooled, as the system becomes experienced. Resembling correlations among agents' credits, such profiles could be used to enhance adaptation speed by more global adjustments which, implicitly, follow stereotype preferences of groups of users.

On the other hand, we plan to investigate in which form users' individual preferences depend on actual situation circumstances. As mentioned in section 3, users' preferences for spatial reference frames may depend on the orientation or position of objects given in the actual situation. Therefore, actual scene data would have to be integrated in the adaptation process. By this, we envisage adaptation to users' preferences as well as adaptation to varying situation parameters to be realized by multiagent systems.

References

[Cao et al., 1995] Y. Cao, B. Jung, I. Wachsmuth. Situated verbal interaction in virtual design and assembly. IJCAI-95 Videotape Program. Abstract in *Proceedings of IJCAI-95*.

[Davis & Smith, 1983] R. Davis and G. Smith. Negotiation as a Metaphor for Distributed Problem Solving. In A.H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 333–356. Morgan Kaufmann, 1983.

[Kaelbling, 1993] L.P. Kaelbling. *Learning in Embedded Systems*. The MIT Press, 1993.

[Kobsa & Wahlster, 1989] A. Kobsa and W. Wahlster (eds.). *User Models in Dialog Systems*. Springer Verlag, London, 1989.

[Laurel, 1990] B. Laurel. Interface agents: Metaphors with character. In B. Laurel, editor, *The art of human-computer interface design*, pages 355–365. Addison-Wesley, 1990.

[Lenzmann et al., 1995] B. Lenzmann, I. Wachsmuth, and Y. Cao. An intelligent interface for a virtual environment. KI-NRW (Applications of Artificial Intelligence in North-Rine Westphalia) Report 95-01.

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

[Maes & Kozierok, 1993] P. Maes and R. Kozierok. Learning interface agents. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93*, pages 459–465. AAAI Press/The MIT Press, 1993.

[McTear, 1993] M.F. McTear. User modelling for adaptive computer systems: a survey of recent developments. *Artificial Intelligence Review*, 7:157–184, 1993.

[Mitchell et al., 1994] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski. Experiences with a learning personal assistent. *Communications of the ACM*, 37(7):80–91, 1994.

[Norman, 1994] D.A. Norman. How might people interact with agents. *Communications of the ACM*, 37(7):68–71, 1994.

[Retz-Schmidt, 1988] G. Retz-Schmidt. Various views on spatial prepositions. em AI magazine, 9(2):95–105, 1988.

[Wachsmuth & Cao, 1995] I. Wachsmuth and Y. Cao. Interactive graphics design with situated agents. In W. Strasser and F. Wahl, editors, *Graphics and Robotics*, pages 73–85. Springer, 1995.