Context-Aware Office Assistant

Hao Yan, Ted Selker
MIT Media Laboratory
20 Ames St., E15-320N
Cambridge, MA 02139 USA
{yanhao, selker}@media.mit.edu

ABSTRACT

This paper describes the design and implementation of the Office Assistant – an agent that interacts with visitors at the office door and manages the office owner’s schedule. We claim that rich context information about users is key to making a flexible and believable interaction. We also argue that natural face-to-face conversation is an appropriate metaphor for human-computer interaction.

Keywords

Human-computer Interaction, User Interface, Agent, Office Automation

INTRODUCTION

This paper describes the design and implementation of the Office Assistant – an agent that interacts with visitors at the office door and manages the office owner’s schedule. The agent is installed at the threshold of a high-traffic office in a noisy environment where the office owner prefers to have the door closed. The agent's goal is to facilitate informal, everyday office communication in an unobtrusive manner.

One of the primary tasks of the office assistant agent is to interact with visitors, give them proper information about their appointments or instructions to set up appointments, and update calendar entries to reflect recent interactions. The agent can also access and manage the office owner's schedule. For example, if the owner is in a conversation with another person when a visitor comes to the threshold, the agent will decide whether to interrupt according to whether the visitor has an appointment with the owner.

The Office Assistant is able to create adaptive user and task models based on information from the office owner, visitors, and the interaction history, and thus change its interaction behavior. It uses different technologies to collect context information, such as whether the owner is busy, whether there is a person arriving or leaving, the owner’s available time slots, the visitor’s identity, etc.

In the next section, we briefly describe our motivation. We then review some background issues and present the design and implementation of the Office Assistant. Two sample interactions follow. Finally, we conclude and discuss some possible future directions.

MOTIVATION

Commonly, people use schedule books, personal digital assistants, or scheduling software to organize their time and tasks. Unfortunately, we still have trouble scheduling events and finding people. In fact, one of the difficulties of working with other people is to know when it is appropriate to intrude on their space. The office threshold is a social demarcation, the separation of private and public space. It is where we obtain subtle information about someone's availability or their willingness to be interrupted. Its role also changes depending on the relationship between the office owner and the visitor. This work extends other scheduling metaphors by making an assistant agent, which is human-like and able to have simple conversations, interact with visitors at the threshold.

An important purpose of using the system is to decrease the number of interruptions to the office owner. Many visitors just want to leave a short message or check the status of the office owner. They do not necessarily want to talk with the office owner. In this case, the Office Assistant agent acts like an information filter. The system is also designed as an investigation in changing people’s social behavior. It is easier for the office owner to avoid seeing people she does not want to see, since the door is closed and visitors cannot see the office owner. It is also helpful for people who are too shy to interrupt others’ conversation. For those people, it is much easier to interact with an intelligent agent instead of a real person.

BACKGROUND

There are many challenges in making intelligent user interfaces like the Office Assistant. These range from design issues such as making choices about interface metaphors to technical issues such as improving speech recognition accuracy.

Interface Metaphors

One big challenge in human-computer interface design is how to convey system functionality through the interface in an efficient and easy-to-learn way. An intuitive yet powerful technique is to use metaphors of real world objects or activities to represent the properties or operations of the system. However, it does not always produce appropriate interfaces. If the majority of the features of the metaphor used when designing user interfaces are inappropriate to the context of system usage, the user will have wrong expectations of the system and therefore will have difficulty to use it. So, it is also important to minimize the number of features of the real world entity that do not apply in the system in design when selecting metaphors [1]. Two particular methods are suggested [1]: 1) select a real world metaphor with restricted scope. 2) expand the scope of the system to map more features of the real world metaphor.

How to Enter an Office

Anderson and Alty studied people's behavior at the office door [2]. The result shows that people's actions at the threshold are determined by the state of the office door (open, half open, and closed), and by their status or role with respect to the office owner. When the door is closed, most visitors choose the action "knocking and waiting", "checking status", or "leaving a note". The action "walking in" and "knocking and walking in" are much less frequent. These results suggest that an information assistant at the threshold level is necessary.

Context-Aware Computing

Context-aware applications are those that change behaviors according to context [4]. The context information in those applications is usually not specified as explicitly required input. It could be background knowledge such as the geographical location of a grocery store, or knowledge about a specific user's preferences. Using these information as cues to change behaviors, context-aware applications are likely to have a more personalized and believable interface to the user. An example of the context-aware applications is the "Forget-me-not" system, which runs on a handheld device that constantly collects context data and helps users to recall past events [6].

Previous Scheduling Systems

In Office Monitor [9], a lifelike mannequin is placed inside an office to enable office visitors to leave quick messages and to enable office owners to reveal their schedules and location information. The system detects the presence of visitors using a motion sensor and interacts with visitors via a speech interface. The user study of the system showed that inside an office is not a good place for the agent, because people are reluctant to go into others' office in the owner's absence, even if the office door is open. CAP [8] is a calendar-managing agent that can learn users' scheduling preferences from experience. However, it is not able to obtain other people's scheduling information.

OFFICE ASSISTANT

Interface Design

Two major metaphors are employed in the interface design: threshold and face-to-face conversation. Threshold in the real world has very simple functions – it is a device at the office door that people step on and cross. People would not have high expectations in the functions of the threshold. The threshold is not a new place to put sensors. People have been using doorbells for years. However, by incorporating an agent and visitor detector sensor to the threshold, we give the threshold some social functions. Face-to-face conversation is used as the major metaphor of the visitor-agent interaction in the system. People have conversations with each other every day, so they already know how to use a system that has a conversational interface. Moreover, studies show that users were more likely to be engaging and cooperative with an interface agent when it had a human face [3,5].

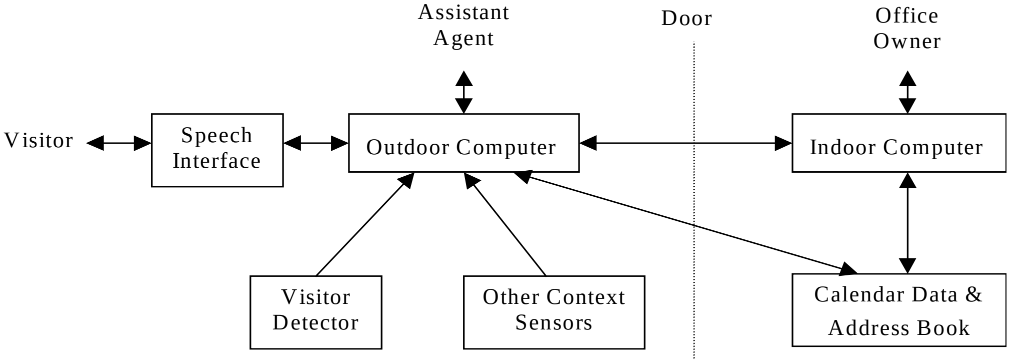


Figure 1. Office Assistant System Architecture

The agent-visitor interaction is conducted through speech dialogue. The interaction between the office owner and the notification program is simply based on keyboard or mouse so that it is less intrusive and quick to respond.

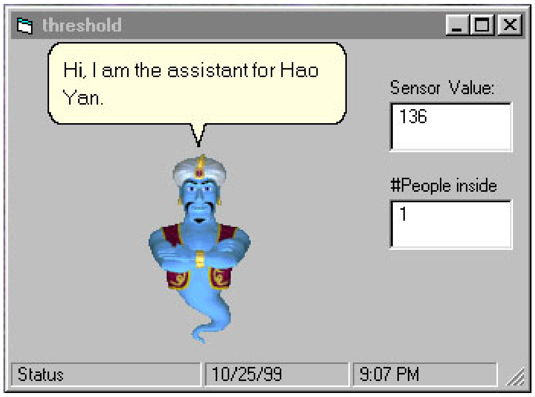
System Architecture

The architecture of the Office Assistant system is shown in figure 1. The agent runs on the outdoor computer, which collects context data and interacts with visitors. Another program runs on the office owner's working computer, which interacts with the office owner and communicates with the outdoor program through a TCP/IP connection. The agent can access and modify office owner's calendar data, while the office owner can also use her familiar calendar software to view and change these data as normal.

Implementation

The animated agent and the speech dialogue interface are implemented using Microsoft Agent, which provides basic components of agents such as animation of different behaviors. We selected Microsoft Outlook – a popular personal information managing software - as the calendar program running on the office owner's PC. It provides great examinability [7] to agent developers. IBM ViaVoice and Microsoft text-to-speech are used as underlying speech engines. The whole system is implemented using Visual Basic, which can easily access and manipulate Outlook and Agent objects. Figure 2 is a screenshot of the agent.

The visitor detector consists of two pressure-sensitive mats placed on both side of the office door. They are connected to the same parallel port of the computer outside the office. Both mats are big enough so that it is hard not to step onto them in sequence when a person enters or leaves the office. By detecting the state changes of the mats, the agent is able to figure out whether the person is entering, exiting, or standing on the threshold, and to keep track of the number of people in office. The two mats look similar to normal carpets, so that people will likely not notice the sensors.



Hi, I am the assistant for Hao Yan.

Sensor Value: 136

#People inside: 1

Status | 10/25/99 | 9:07 PM

Figure 2: Office Assistant Agent

A monitor and two stereo speakers are installed beside the office door. A noise-canceling microphone is mounted on top of the monitor for picking up user's voice. Also, there are keyboards and mouse in front of the monitor as backup devices if speech recognition fails; visitors can always type in their names and use the mouse to make choices.

Several context-free grammars are used in the speech recognition engine to increase the recognition accuracy and to avoid linguistic parsing of speech input (by specifying speech act tags in the grammar). Based on context, the system dynamically activates and deactivates appropriate grammars. Each grammar is rich enough to reflect most variations of speech visitors can say in a certain scenario. Currently, there are four types of grammars: name query, making appointments, leaving messages, and confirmations. In addition, there are utterances that visitors can always say, such as greetings and farewells. These utterances are made global in all grammars.

Context Information Processing

Different context information is collected and used to alter the agent's interaction with visitors. For example, if a visitor comes to the office without an appointment, depending on the office owner's busy status, the assistant agent will behave differently. If the office owner is engaging in a conversation (more than 1 person in office), the agent will ask the visitor to leave a message or arrange an appointment. However, if the office owner is alone, the agent will ask the office owner to decide whether to see the visitor immediately or later. The major context information used in the system includes:

1) The identity of the visitor. Currently, this is obtained through a question/answer process.
2) The office owner's schedule status. This includes appointment time, appointment content, and available time for appointments, etc.
3) The office owner's busy status. It can be obtained in various ways, such as checking busy tags in calendar data and checking number of people in office
4) The office owner's willingness to see the current visitor. Obtained through the notification program.

SAMPLE INTERACTIONS

In this section, we present two real interactions between users and the Office Assistant system. They are annotated with important screen displays, switching of speech recognition grammars, and changes to the office owner's calendar data.

Scenario 1: New Visitor

Tom came to see Professor Brown and did not have an appointment with him. When Tom stepped on the threshold, the agent noticed his presence and started greeting:

Agent: "Hi, I am the assistant for Professor Brown, what is your name?" (The agent activated the speech recognition grammar for name query)

Tom: "I am Tom."

Agent: "Hi, Tom. You don't have an appointment with Professor Brown. But I will inform Professor Brown that you are here. Please wait". (In the office, a dialog box appeared on Professor Brown's computer screen with a notifying sound, displaying "Tom is outside. He does not have an appointment with you. Could I let Tom in?" with three buttons, "Let In", "Wait a minute", and "come later". Professor selected "Wait a minute".)

Agent: "Professor Brown is finishing his current work. He will see you in a couple of minutes. Please wait!"

Tom: "OK!"(A minute later, the same dialog box appeared on Professor Brown's screen with a notifying sound played. This time, Professor selected "Let In".)

Agent: "Professor is ready to see you. Come on in!" (Tom entered the office. The agent increased the number of people in office by one.)

Scenario 2: Rearrange Appointment

Jerry had an appointment with Professor Brown at 3pm. He went to Professor Brown's office at that time. Professor Brown was in a conversation with Tom (greeting omitted).

Agent: "Hi, Jerry, you have an appointment with Professor Brown from 3pm to 3:30pm. I will inform professor Brown that you are here, please wait." (In the office, a dialog box appeared on Professor Brown's computer screen with a notifying sound, displaying "Jerry is outside, he has an appointment with you right now, could I let Tom in?" Professor Brown selected "Come later".)

Agent: "Sorry Jerry, Professor Brown is really busy, he wish to rearrange the appointment with you. Do you want me to arrange another time or just leave a message?" (agent activated the conformation grammar.)

Jerry: "Please arrange another appointment!"

Agent: "OK!" (agent activated the appointment grammar.)

Agent: "How about tomorrow at 9am?"

Jerry: "I will be busy then"

Agent: "Then, would tomorrow at 2pm work for you?"

Jerry: "OK!"

Agent: "Good, see you then!"

Jerry: "Bye" (a new appointment entry with proper description was added to Professor Brown's calendar).

CONCLUSION

Recognizing the role of an office door and installing simple sensors allow a system to monitor and facilitate simple social behaviors at an office threshold. Context information about the visitor, the status inside the office, and the office owner's scheduling data, are collected from different source and can consequently alter the office assistant agent's decision about interaction style. Further, the use of face-to-face conversation as a metaphor in the design of the Office Assistant system makes the interface easy to learn and use. Quantitative user studies about the effectiveness of the system still need to be done. We are already encouraged by the fact that context-awareness can greatly improve the performance of a human-computer interaction system.

FUTURE DIRECTIONS

Future work on the system includes incorporating more sensor technologies to detect richer context information. For instance, a computer-vision-based automatic identification detector can be added such that the system knows the visitors' identity when they approach the door and avoid the relatively boring name-asking process.

The appointment arrangement algorithm needs to be improved. Currently, the agent checks all available time slots for the office owner and asks the visitor about the appointment time one by one until the visitor agrees on a time slot. An improvement to this paradigm could be letting the user propose a time or select from a table of available time slots.

Certainly, improving the adaptive user and task models is also important. For example, the boss of the office owner should always have higher priority to enter the office. The agent could behave differently if the coming person is a frequent visitor to the office. It is also possible that the system observes the action-selection style of the office owner and adjusts the frequency of interrupting the office owner.

ACKNOWLEDGMENTS

Thanks to Vadim Gerasimov for the help in the use of pressure-sensitive sensors and PC parallel port. Thanks to Dr. Nicole Yankelovich, Dr. Henry Lieberman and Bradley Rhodes for giving valuable designing advice. Thanks to Mike Ananny, Lee Campbell and our anonymous reviewers for reading this paper and giving valuable comments.

REFERENCES

1. Anderson,B., et al., Minimising Conceptual Baggage: Making Choices about Metaphor, *People and Computers IX*, Proceedings of HCI '94, Cockton, G., Draper, W., and Weir, S. Eds., Cambridge University Press, 1995.
2. Anderson, B., and Alty, J., Everyday Theories, Cognitive Anthropology and User-centered System Design, *People and Computers X*, Proceedings of HCI '95, Kirby, M, Dix, A, and Finlay, J. Eds., Cambridge University Press, 1995.
3. Andre, E., Rist, T., Mueller, J. Integrating Reactive and Scripted Behaviors in a Life-Like Presentation Agent. *In Proceedings of Agents '98* (Minneapolis/St. Paul, May 1998), ACM Press.
4. Brown, B., Bovey, J., and Chen, X., Contex-Aware Applications: From the Laboratory to the Marketplace, *IEEE Personal Communications*, October 1997.
5. Koda, T., and Maes, P. Agents with faces: The effect of personification. *Proceedings of the Fifth IEEE International Workshop on Robot and Human Communication (RO-MAN '96)*. 189-194.
6. Lamming, M. and Flynn, M., "Forget-me-not" Intimate Computing in Support of Human Memory, *Proceedings of FRIEND21, '94 International symposium on Next Generation Human Interface*, Meguro Gajoen, Japan, 1994
7. Lieberman, H., Integrating User Interface Agents with Conventional Applications. *Knowledge-Based Systems* 11, 1998.
8. Mitchell, T.M., Caruana, R., Freitag, D., McDermott, J. and Zabowski, D., "Experience With a Learning Personal Assistant", *Communications of the ACM*, Vol. 37, No. 7, pp. 81-91, July 1994.
9. Yankelovich, N. and Mclain, C.D. Office Monitor. *CHI '96 Conference on Human Factors in Computing Systems*, Vancouver, British Columbia, Canada, April 14-18, 1996