Agents in their Midst: Evaluating User Adaptation to Agent-Assisted Interfaces

Tara Gustafson, J. Ben Schafer, Joseph Konstan
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455 USA
+1 612 625 4002
{tara, schafer, konstan}@cs.umn.edu

ABSTRACT

This paper presents the results of introducing an agent into a real-world work situation - production of the online edition of a daily newspaper. Quantitative results show that agents helped users accomplish the task more rapidly without increasing user error and that users consistently underestimated the quality of their own performance. Qualitative results show that users accepted agents rapidly and that they unknowingly altered their working styles to adapt to the agent.

Keywords

Agent-assisted interface, user studies, user-centered interface design, learning agents, online newspaper production, electronic publication.

INTRODUCTION

While much research has been done in the design and implementation of agents, there has been little work done to study how users adapt to agent-assisted software. Donald Norman asks "How will intelligent agents interact with people and perhaps more important, how might people think about agents?" [13]. In addition, Riecken questions if people will want agents [16]. Studies have included the use of agents in the domains of Usenet newsreaders, calendar scheduling, air travel planning, and email systems [1,3,9,10,11,12,15]. However, these studies focus on the feasibility of agent-aided systems and the architecture for these agents. User studies, if any, were inclined to use test subjects in a controlled setting, rather than real users in their work context. The study by Lester et. al [8] used elementary school students in a controlled setting to examine how people perceive life-like agents. Now that agents have been proven effective in a laboratory setting, it is incumbent upon us to study how users in a real-world situation adapt to agent assistance in their software.

We studied the acceptance of agents by the online production staff of a daily campus newspaper [5]. Two phases of user testing – with and without an agent assistance module – explored several broad-based questions.

- How do users accept and adjust to agents?
- How do user's perceptions of efficiency and accuracy compare with reality?
- Can a successful agent be designed for this domain?

BACKGROUND

The Environment

The *Minnesota Daily* is the nation's largest college daily newspaper, distributing 23,100 copies a day. In 1995 it became one of the first campus newspapers to publish on the World Wide Web. This online edition is viewed by over 800 people a day nationwide.

To produce this digital edition the nightly production staff follows a four-step procedure. The process begins with a transfer to the web server of the QuarkXPress[1] files that comprise the print version. These files include not only the text of the stories, but also story elements such as headlines, photos, cutlines (photo captions), and graphics. In the second step the user arranges the stories into departments (sports, for example), sorts them into the correct order within the department, and assigns each story its elements. Next, the formatting program creates the HTML code by translating and combining the QuarkXPress files. Finally, this finished edition is added to a searchable database for archive purposes.

Although the print version of the paper is put together digitally, online production is still a labor intensive procedure. There are several factors that complicate this procedure. First, the online edition does not have the same structure as the print version of the newspaper. The

IUI 98 San Francisco CA USA
Copyright 1998 ACM 0-89791-955-6/98/ 01..$3.50

[1] QuarkXPress is a page layout program. It is a powerful tool for formatting and manipulating text on a page, especially when using multiple columns across several pages.

online editíon is organized by department (news, sports, etc.), whereas the paper publication is organized spatially with articles from different departments often sharing the same page. Because the department of a story is never explicitly stated in the print version, human intervention is necessary to assign each story to its department. Second, QuarkXPress views the story text, headlines, photos, and other story elements as separate entities. It requires human input to group these entities into stories. The paper production process only implicitly groups story elements through spatial layout; each picture is adjacent to several text elements.

The Users

During the design of the program, the online edition was single-handedly produced each night by the online production manager. While designing the study, it was assumed that this situation would continue. On the first night of the trial, the online manager presented a surprise – his three online interns[2] would take over the nightly production duties. While this development was unexpected, it reflects the dynamics of a real world study. Though the experimental design had to be revised, this change was quite a bounty. Now the interface could be tested with novice users and data gathered on agent adaption for four users, instead of only one.

	PREVIOUS *DAILY* EXPERIENCE	PRIOR COMPUTER EXPERIENCE
User A	edit writer, staff reporter, opinions editor, online intern	Word processing, QuarkXPress
User B	online intern	Word processing, intranet mail and file transfer, PowerPoint
User C	freelance, online intern	Mathematica, C, HTML, word processing
Production Manager (PM)	online production manager	Word processing, PERL scripting, HTML, QuarkXPress

Table 1: User Backgrounds

All four users were University of Minnesota undergraduates. As Table 1 shows, their newspaper and computer experience was varied. They also came from different academic backgrounds. The production manager served as the paper's representative to the interface design process and trained each of the interns. By the implementation and research stages the three interns had just completed their training in paper and online newspaper production.

THE INTERFACE

While most of the online production process is automated, the second step – grouping elements by story, arranging the stories into department, and ordering the stories within the departments – requires substantive user involvement. Before this project, the staff used a web-based, CGI-driven program that was very difficult to use. In addition to having few error correction capabilities, the program forced users to perform their tasks in a rigid and specific order. Once an incorrect selection was made there were frequently only two choices – start over from the very beginning, or complete the program and modify the generated HTML manually. This process proved to be both time-consuming and highly stressful for the users. Users reported averaging over half an hour using the program each night, as well as an additional half-hour manually correcting the output.

It was quite clear that a new interface was needed - one that was more forgiving of mistakes and less rigid in its flow structure. The task centered user interface design process [7] was used to develop a new interface in Tcl/Tk [14]. The interface was modeled after WinFTP, a program used in a prior step of the process, to provide a familiar look and feel. Figure 1 shows the arranging and ordering screen. It consists of two listboxes separated by buttons with directional arrows. The left-hand list initially contains all of the stories for the edition in progress, while

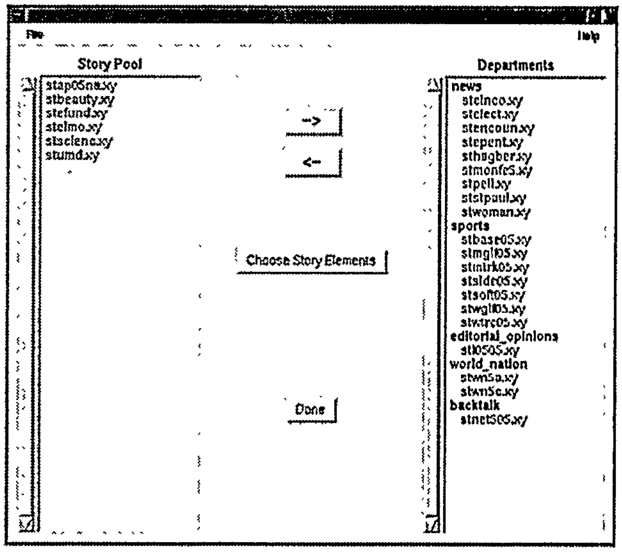

Figure 1: The story sorting and ordering window

[2] Although the interns were of mixed gender, all interns will be referred to as "she" in this document for the sake of the users' privacy.

the right-hand list contains the five departments used by the paper (News, Sports, Editorial Opinions, World and Nation, and Backtalk). Users select stories and transfer them to the department list. Users drag and drop stories to sort them within each department. Furthermore, at some point, users must associate each story with its elements (headlines, photos, etc.).

THE AGENT

To focus this study on user adaptation versus agent technology, a simple but effective agent was designed, consisting of several cooperating algorithms. As Figure 2 shows, the agent combines programmed knowledge with a scoring system based on programmed and learned syntactic attributes. The results of each algorithm are weighted, and if the agent feels that the combined confidence level is high enough it places the story in the recommended department.

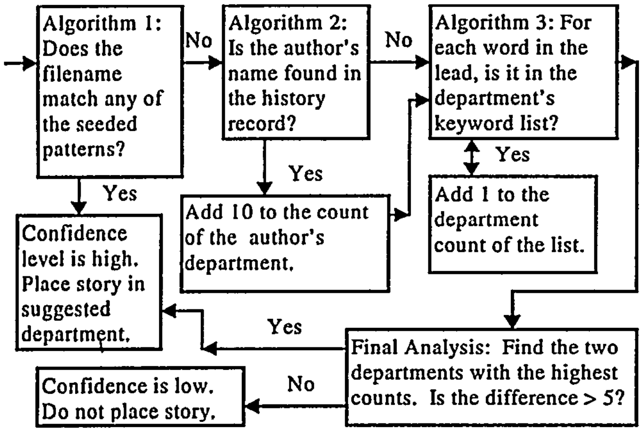
Algorithm 1: Does the filename match any of the seeded patterns?
No
Algorithm 2: Is the author's name found in the history record?
No
Algorithm 3: For each word in the lead, is it in the department's keyword list?
Yes
Yes
Yes
Confidence level is high. Place story in suggested department.
Add 10 to the count of the author's department.
Add 1 to the department count of the list.
Yes
Final Analysis: Find the two departments with the highest counts. Is the difference > 5?
No
Confidence is low. Do not place story.

Figure 2: Agent Algorithm

The first algorithm is designed to make recommendations based on the filename of the story. For example, there should be a file each night named 'stl####.xy' (where the #### is the month/day combination). This file contains the "letters to the editor" and therefore can safely be assigned to the Editorial Opinions department (regardless of the author(s) or keywords contained in the body of the file).

The second algorithm extracts the "by-line" from a story's body to determine the author of the article. Because reporters for the *Minnesota Daily* tend to write articles within a single department, each author's history can be used to make a prediction. The algorithm updates its author association rules each night by examining the finished, user verified online edition.

The third algorithm searches the lead of a story (the first paragraph) for keywords. Due to time constraints, a non-learning search was used [6]. Lists were compiled by manually searching the archives for words common to a given department. The algorithm compares words found in a story's lead to these lists. Each "hit" (a word found in both the lead and a departmental list) adds more weight to that department. Thus, words like "basketball," "player," and "court" found in the lead of a story probably suggests a sports story. This algorithm could be extended to update the keyword lists based on each night's stories.

RESEARCH DESIGN

A six-week, two-phase investigation was designed. Phase I consisted of introducing the program (without agent assistance) into nightly use at the *Minnesota Daily*. Phase II incorporated the agent assistance module into the program. Users were not informed about the pending agent assistance module prior to the study.

We gathered data in three ways. First, the program included code to trace and record user actions and timing information. These records provided a timeline of each user's actions, allowing an analysis of user performance and working style. Second, live observations provided an understanding of the context in which users work each night and how errors occur. Finally, user interviews provided insight into each user's thought process and how she perceived her own performance.

Finally, we needed a criterion to evaluate the third goal – the success of the agent. Success was defined as a 20% reduction in sorting[3] time per story with no significant increase in the error rate. Data about the agent was gathered using the tracing mechanism defined above.

METRICS

To aid in defining the study and analyzing the results, the following metrics were defined:

- sorting time: the total time spent sorting, where sorting is moving stories to a department and ordering the stories within that department.
- placement rate: the percentage of stories placed by the agent.

Because this study occurred in a real-world situation, users often interrupted production to tend to other tasks, socialize, or correct prior production errors, such as files absent from the system. Therefore, times had to be adjusted to account for serious discontinuities in production. All resulting measurements should be considered approximate.

We also defined three error metrics. These indicate wasted time either by human or agent misplacement.

- user error: the number of times that a user placed a story in a department and later moved the story to a different department.
- agent error: the number of times that the agent placed a story into a department and the user moved it to a

[3] We use time spent sorting rather than total time as our measure because it is the area where the agent's effects are concentrated.

different department.

- significant increase in the error rate: this is a difficult metric, because the "error rate" includes correcting agent errors and the number of user placements is reduced when the agent placed stories. A decision was made to use subjective user satisfaction as the primary measure. Also, the change in user error was used to assess whether the users were less careful with unplaced stories.

The study did not take into account uncorrected errors, where a story was in the wrong department when the user finished using the program.

PHASE I

Goals

Phase I consisted of introducing the agentless program into nightly use at the *Minnesota Daily*. There were three goals for this phase.

- Track implementation of the program.
- Compare user perceptions with reality.
- Establish a baseline for phase II.

Results

At the end of phase I, we asked users to estimate the time spent using the program and the number of errors made each night. As Table 2 shows, in most cases users underestimated their own abilities. Users frequently reported times 50-100% higher than their actual performance times. They also tended to report higher error rates than those recorded.

USER	PHASE 1			
	AVG TIME (minutes)		AVG ERRORS	
	EST	ACTUAL	EST	ACTUAL
User A	10-15.	14.56	1	0
User B	30	19.49	3 – 4	2.67
User C	20	11.00	1	1

Table 2: Phase I data: estimates by users and the actual figures.

The users[4] quickly developed individual styles:

> User A developed the habit of sorting stories department by department. She preferred to select all of the stories for the first department and transfer them as a unit. Then she arranged them in the correct order. This was repeated for each of the remaining departments.
>
> User B was inclined to transfer stories one at a time in the order in which they would be presented in the final product. She started with the first story of the news department, and worked her way through the news department. Then she moved on to the other departments.
>
> User C was less consistent, but she tended to work through the alphabetical story list one story at a time transferring each story to its department. After all stories were transferred she sorted each department's stories into proper order.

All three users completed the task of sorting all of the stories before continuing with the task of associating each story with its elements.

Despite never having used the program before, the users became quite proficient at the online production process. In order to complete the program, users spent an average of 41 seconds per story [Table 3]. More specifically, the users spent an average of 11 seconds sorting each story and made approximately one error per night.

PHASE II

Goals

Goals in phase II were natural extensions of the goals in phase I.

- Examine how users adapt to agent-assisted software.
- Compare user perceptions with reality.
- Determine if placement agents can improve overall performance.

Results

The first time each user encountered the agent, the typical reaction was "cool, it placed some of the stories for me." Surprisingly, the users did not ask the observers why some of the stories were being moved. Instead, they simply returned to the task of completing online production.

With the agent in place, users settled into new working styles. Users A, B, and the production manager (PM) adopted the style of moving all unplaced stories to the correct department and then ordering the stories within the departments. However, user C adopted the nearly opposite style of ordering the stories department by department and moving unplaced stories over as needed.

At the end of phase II, users were again asked to estimate their time and error rates. They were also asked to estimate the agent's placement and error rate. Table 4 shows that users continued to underestimate their own performance. They all overestimated the agent's placement rate, while also overestimating the number of errors that the agent made each night. When asked to describe the performance of the agent all reported placement rates in the 80-90% range. In fact, placement rates were closer to 70%. Users estimated the number of

[4] Due to the production schedule, the production manager did not contribute any data to this phase of the trial.

	USER	# OF STORIES	TOTAL TIME PER STORY (SECONDS)	SORTING TIME PER STORY	ERRORS BY USER
Day 1	User A	27	0:49.9	0:11.9	0
Day 2	User B	24	0:47.5	0:11.2	1
Day 3	User A	27	0:24.5	0:05.9	0
Day 4	User C	22	0:17.2	0:04.5	0
Day 5	User C	18	0:36.2	0:12.1	0
Day 6	User A	23	0:20.1	0:08.3	0
Day 7	User B	24	0:56.5	0:14.1	5
Day 8	User A	21	0:29.6	0:15.9	0
Day 9	User A	29	1:03.1	0:09.1	0
Day 10	User B	15	1:11.4	0:16.8	2
Day 11	User A	16	0:28.4	0:06.8	0
Day 12	User C	19	0:49.9	0:16.4	3
Averages		22	0:41.2	0:11.1	0.92
User errors per story placed	0.04				

Table 3: Phase I Results

USER	PHASE II									
	TIME (minutes)		USER ERROR		AGENT PLACEMENT		AGENT ERROR		DID AGENT IMPROVE SPEED?
	EST	ACTUAL	EST	ACTUAL	EST	ACTUAL	EST	ACTUAL	
User A	10 - 20	7.55	< 1	0	75%	69%	3 - 4	2.33	No change
User B	30 – 40	13.03	3 - 4	0.5	75%	65%	2 - 4	2.00	Yes
User C	25	10.30	< 1	0	85%	70%	3 - 4	0.67	Yes

Table 4: Phase II data: estimates by users and the actual figures. The last column contains the users' responses when asked if the agent improved their speed.

errors made by the agent to be approximately 2 to 3 each night. In reality, the agent averaged 1.5 errors each night.

Overall completion times became far more stable than those observed in phase I. Users quickly settled into production times of less than 10 minutes [Table 5]. Sorting times averaged 8.9 seconds per story in phase II. The users made an error only once during the entire phase.

ANALYSIS

Adapting to Agents

While examining the users' working styles, an interesting phenomenon was observed. In phase II, the users reversed their sorting strategies. In phase I, users A and B chose to assign and order all the stories for a single department before moving on to the next department. In phase II, where the agent placed many of the stories, both users changed their strategy to move all unplaced stories into departments before ordering them. In phase I, user C tended to work through the alphabetical story list one story at a time, transferring each story to its department. In phase II, she adopted the style of ordering the stories department by department and moving stories into the departments as needed.

Comparing phases I and II at the user level revealed odd results. For example, while User A improved her overall time by 47%, her sorting time (which should show the agent's effects) didn't change at all. User C, on the other hand, experienced the opposite phenomenon. Her overall time decreased only slightly, but her sorting time dropped

	USER	# OF STORIES	TOTAL TIME PER STORY (SECONDS)	SORTING TIME PER STORY	ERRORS BY USER	ERRORS BY AGENT	PLACED BY AGENT	PLACED CORRECTLY BY AGENT
Day 1	User A	21	0:29.2	0:09.6	0	4	19/21	15/19
Day 2	User B	21	0:48.2	0:13.6	1	1	11/21	10/11
Day 3	User C	22	0:52.1	0:13.8	0	1	13/22	12/13
Day 4	User C	15	0:30.3	0:07.6	0	1	13/15	12/13
Day 5	User B	19	0:29.2	0:05.6	0	3	15/19	12/15
Day 6	User A	21	0:20.0	0:07.8	0	1	12/21	11/12
Day 7	User C	17	0:31.8	0:07.8	0	0	12/17	12/12
Day 8	P.M.	13	0:32.5	0:07.9	0	0	10/13	10/10
Day 9	User A	20	0:19.5	0:11.3	0	2	12/20	10/12
Day 10	P.M.	22	0:13.2	0:03.7	0	2	15/22	13/15
Day 11	User C	19	0:34.1	0:07.0	0	1	11/19	10/11
Day 12	User C	18	0:20.0	0:05.6	0	0	15/18	15/15
Averages			0:30.0	0:08.4	0.08	1.33	158/228	142/158
User errors per story placed	0.01							

Table 5: Phase II data

by 24%. A consistent explanation for these results has not been found.

However, one explanation may lie in the production manager's observation that more than just their style changed – so did their attitude. While admitting that as a user he preferred the agent-assisted version, he expressed concerns as a supervisor. He felt that users put too much confidence in the agent's decisions and unconsciously extended the agent's abilities into areas where they did not exist. For example, he felt that users paid less attention to the order of the stories within a department. This is hard to verify, because this was not a controlled experiment with exact right and wrong answers. Short of asking the production manager to produce an "answer key" for each evening's edition, it is very difficult to establish correctness. What is published online may or may not be the "correct" edition.

Perceptions Versus Reality

An interesting pattern emerged when users were asked to estimate their performance using the program. In nearly all cases users underestimated their own abilities. Users frequently reported times 50-100% higher than their actual performance times. They also tended to report higher error rates than those recorded. We speculate that the users may have recalled a night where they made several errors and reported that as their average, forgetting the fact they also had several nights without any errors. User estimates of the agent's error rate confirms this speculation. Each reported an agent error rate of several mistakes each night, when the agent averaged only 1.5. Users did, however, overestimate the number of articles placed by the agent.

During the interviews, the users were asked how the agent affected their use of the program. All reported no change in their work patterns. Yet, as discussed above, each of the users significantly modified her style. It may be that the style modifications each user made was so natural to her that she did not recognize that her style had changed.

Two of the users made an interesting observation. They felt that the agent improved the "thought-flow" of the program. In general, they felt this program has two separate mindsets – first placing stories and then error checking. Users start in the first mindset. Once all stories are assigned to a department, the user must switch to the second mindset and make sure the stories are in the right order within the correct department. Users then check to make sure that the right elements have been assigned to each story. When the agent places a majority of the stories, users may bypass the first mindset and sort the unplaced stories while error checking.

Agent Evaluation

Recall that there were two criteria established to measure the success of the agent: a 20% reduction in time, and no significant increase in the error rate. When we examine the time spent running the program, we see that the average declined from 41.2 seconds per story to 30.0 seconds, a decrease of 27.2% [Table 6]. A more accurate and realistic measure of success is an examination of the "sorting" time. Because the agent aids the user in assigning stories to a department, the agent's actions are concentrated most clearly in the sorting domain. Sorting dropped from 11.1 seconds per story in phase I to 8.4 seconds per story in phase II, an overall reduction of 23.7%.

USER	TOTAL TIME PER STORY (SECONDS)	SORTING TIME PER STORY (SECONDS)
Group	-27.2%	-23.7%
User A	-41.4%	-0.9%
User B	-33.8%	-31.4%
User C	-2.3%	-24.0%

Table 6: Percentage change between phase I and II.

The second criterion of success was no significant increase in the error rate. While the error rate increased by 54.5% between the two phases, this is only an increase from slightly less than one error to one-and-a-half errors per evening. Since user error rates dropped (in fact there was only one user error in all of phase II) and users reported no difficulty in catching the agent's mistakes, this was viewed as a non-significant increase.

Looking at the number of errors each night may be misleading since the number of stories placed by the user differed between phases. In phase I, users average .04 errors per story placed, whereas in phase II, users averaged .01 errors per story placed.

The numbers behind this project suggest that a successful agent can be built for this domain. However, history teaches us that no matter what the numbers may show, it is the users' perceptions and feelings that determine the overall success of a system [2].

Both in observations and in interviews, it was clear that users appreciated the agent assisted version. Despite the agent's imperfections, it quickly gained the confidence of the users. All users reported that they were comfortable with the agent and voiced a resounding preference for the agent assisted version. None of the users reported concern with how or why the agent was making its placement decisions. This may stem from the fact that production frequently occurs between the hours of 11:30 PM and 2:15 AM. At this late hour, users' attitudes are "I don't care what happens or how it happens, as long as it gets me out of here early."

FUTURE WORK

This project is a work in progress. We continue to collect data from both versions of the program. It is hoped that this will help determine whether the difference between phases I and II reflects the agent or was skewed by the users' learning curve. We also look forward to seeing if user styles revert when the agent is removed from the program.

We are also interested in studying these questions:

- effect of agent accuracy and placement rates on user performance
- effect of slow/fast increases/decreases in agent performance on the user's ability to detect errors
- nature of user centered design in agent assisted applications [15]
- the value of filename and keyword analysis for this domain

AFTERWORD: User Ownership and Acceptance

During the task-centered design process, we worked closely with the online production manager. He was very excited about the development of the program and his input was invaluable during its design. By the time the online interns were ready to perform production, the program's design was complete. Though the interns did not directly contribute to its development, they began to feel that the program was being developed for them and that they could influence its path [4]. We did not anticipate this sense of ownership, since the interns were never directly involved in the program's design. This feeling of ownership may be attributed to the online production manager's involvement in the project, his relationship with the interns, and his enthusiasm for the program.

Perhaps due to this feeling of ownership, the interns were willing to overlook early bugs in the program that prevented them from completing production, often trying three or four times before turning to the previous production program. It may have also contributed to the ease with which users adapted to the agent.

ACKNOWLEDGMENTS

We would like to thank Joanne Holroyd for her involvement as a team member in the early stages of this project and the staff of the *Minnesota Daily*; in particular Patrik Dousa who let us into the late night world of online production and never once complained when our bugs meant he or his staff had to attempt production multiple times.

REFERENCES

1. Armstrong, R., Freitag, D., Joachims, T., Mitchell, T., Web Watcher: A Learning Apprentice for the World Wide Web, *AAAI Spring Symposium on Information Gathering*, Stanford, CA, March 1995.

2. Baecker, R., Grudin, J., Buxton, W., and Greenberg, S., *Readings in Human Computer Interaction: Toward the Year 2000.* Morgan Kaufmann, San Fransisco CA, 1995, 49-52.

3. Barrett, R., Maglio, P., Kellem, D., How to Personalize the Web, *Proceedings of CHI '97* (Atlanta GA, March 1997), Addison-Wesley, 75-82.

4. Bass, L., Kasabach, C., Martin, R., Siewiorek, D., Smailagic, A., Stivoric, J., The Design of a Wearable Compter. *Proceedings of CHI '97* (Atlanta GA, March 1997), Addison-Wesley, 139-146.

5. Bellotti, V., Rogers, Y., From Web Press to Web Pressure: Multimedia Representations and Multimedia Publishing, *Proceedings of CHI '97* (Atlanta GA, March 1997), Addison-Wesley, 279-286.

6. Gennari, J., Langely, P., Fisher, D., Models of Incremental Concept Formation. Elsevier Science Publishers B. V., North-Holland, 1989, 11-61

7. Lewis, C., Rieman, J., Task-Centered User Interface Design, ftp.cs.colorado.edu

8. Lester, J., Converse, S., Kahler, S., Barlow, S. T., Stone, B., Bhogal, R., The Persona Effect: Affective Impact of Animated Pedagogical Agents, *Proceedings of CHI '97* (Atlanta GA, March 1997), Addison-Wesley,359-366

9. Lieberman, H., Autonomous Interface Agents, *Proceedings of CHI '97* (Atlanta GA, March 1997), Addison-Wesley, 67-74.

10. Maes, P., and Kozierok, R., Learning Interface Agents, In *Proceedings of the AAAI'93 Conference.* MIT Press, Cambridge, Mass., 1988, 459-465

11. Maes, P., Agents that Reduce Work and Information Overload, *Commun. ACM* 37,7 (July 1994) 31-40

12. Mitchell, T., Caruana, R., Freitag, D., McDermott, J., Zabowski, D., Experience With a Learning Personal Assistant. *Commun. ACM* 37,7 (July 1994), 81-91

13. Norman, D., How Might People Interact with Agents, *Commun. ACM* 37,7 (July1994), 68-71

14. Ousterhout, J., *Tcl and the Tk Toolkit,* Addison-Wesley, Reading, Mass., 1994

15. Rich, C., Sidner, C., Adding a Collaborative Agent to Graphical User Interfaces, *ACM Symposium on User Interface Software Technology,* 1996. http://www.merl.com/pub/rich/uist96.ps.Z

16. Riecken, D., Intelligent Agents, *Commun. ACM* 37,7 (July 1994), 18-21