In Proceedings of the 8th European Conference on Cognitive Science Approaches to Process Control. September 24-26, 2001; Munich.

Associates with Etiquette: Meta-Communication to Make Human-Automation Interaction more Natural, Productive and Polite

Christopher A. Miller* and Harry B. Funk
Smart Information Flow Technologies
1272 Raymond Ave.
St. Paul, Minnesota, 55108 U.S.A.
{cmiller,hfunk}@SIFTech.com

Summary

Several different architectures for adaptive automation systems have now been demonstrated. Yet users remain ambivalent them—as our own work on the U.S. Air Force's Rotorcraft Pilot's Associate (RPA) reveals. Thus, we argue, research should be shifting away from how adaptive interfaces and automation can be made to work, and onto how they should *behave* in conjunction with human users. We find it useful to think about behavioral details and roles and responsibilities in terms of various 'etiquettes' for human-machine interactions. Results from RPA suggest that one aspect of establishing acceptable etiquette may be more and better inclusion of 'meta-communication' on the part of the system—the ability to report and accept feedback on the system's perceptions and intentions. We discuss these results in the broader context of etiquette recommendations for adaptive automation systems.

Introduction

The past several years have seen an explosion in the creation of "Adaptive" or "Intelligent" user interfaces which modify system performance, automation behavior and/or information presentation in response to some aspect of context, user, task or situation. Much of this research and growth, and much of the average user's familiarity with such systems, has been in the desktop software environment—with Microsoft™'s Office Assistant (the ubiquitous animated paperclip called "Clippit™" being the best known, if not always most loved, example. Works by Maybury [1998], Puerta [1996], and Bauer [2000] are examples of this trend. At the same time, "real world" examples of adaptive and intelligent interfaces—those which deal with a controlled system 'off the desktop' [Miller, 2000]—are moving closer to reality and fruition. Our own work on military attack/scout helicopters [Miller, Hannen, Guerlain, 1999], oil refineries, [Cochran, Miller and Bullemer, 1996] military command and control [Funk, et. al., 2000] as well as those of Bonner, et. al. [2000], Onken [Strohal and Onken, 1998], and Mitchell [1999], serve as examples.

As the number of such adaptive interaction efforts increases, we are collecting more data about what works and what doesn't. In our own experience, across multiple domains, it is clear that users want, even demand, to remain in charge of actions, yet they also want, even require, the benefits that intelligent, adaptive information and automation can provide them. Work on the Rotorcraft Pilot's Associate (RPA) Cockpit Information Manager [Miller, Hannen and Guerlain, 1999] proves instructive from several perspectives. For example, in developing RPA, we interviewed multiple pilots and designers to develop a consensus list of prioritized goals for a "good" intelligent cockpit configuration manager. Two of the top three items on the list were "Pilot remains in charge of task allocation" and "Pilot remains in charge of information presented." Thus, in spite of generally 'buying into' the need for and benefits to be provided from a sophisticated adaptive automation system, these pilots and designers remained ambivalent about how much control and authority they wanted to give it.

There are good reasons to design and use adaptive systems at the higher levels of automation. By definition [Sheridan, 1987; Parasuraman, Sheridan and Wickens, 2000], such systems share responsibility, authority and autonomy over many work behaviors with human operator(s) to accomplish their goals of reducing operator workload and information overload. While operators may wish to remain in charge, and it is critical that they do so, today's complex systems no longer permit them to be fully in control of all system operations—at least not in the same way as in earlier cockpits and workstations [cf. Miller, Pelican and Goldman, 2000; Perrow, 1999]. So, we are faced with a dilemma. Advanced and adaptive automation is necessary to allow humans to achieve the levels of performance required in today's world, yet human users frequently reject such automation when it makes them feel out of control. Worse yet may be when they fail to or cannot reject such automation—as the range of mode control errors and pilots' sense of failing to understand 'what it will do next' show all too clearly [Sarter, Woods, and Billings, 1997].

How can we build adaptive automation systems which leave the human in charge even when the human is less directly in control? The proposed solution, all too frequently, is to just make the automation 'better'—more aware of the user, more aware of the context, encompassing a broader range of behaviors, etc. The problem is that this puts the onus on the automation to know what is the right thing to do for the operator at any given time. While such a capability would certainly be nice, we believe it has proven (and will continue to prove) far too difficult to achieve with sufficient accuracy to be practical. We are not suggesting that task tracking, intent inferencing or cognitive modeling approaches [Charniak and Goldman, 1993; Hoshstrasser and Geddes, 1989; Coury, Santarelli & Mitchell, 2000] should be ignored, merely that they should be augmented. This augmentation should come through defining acceptable, desirable roles and relationships between human operators and advanced, adaptive automation systems—relationships that the automation can reliably and predictably deliver. We are finding it useful to think of such a package of defined roles and methods of relating as an 'etiquette' for human-machine interaction.

The remainder of this paper illustrates the benefits to be derived from designing adaptive automation around such an etiquette, using examples from our work on the Rotorcraft Pilot's Associate (RPA). We then go on to discuss the 'etiquette' of human-machine interactions and propose a tentative initial list of etiquette rules to stimulate future work.

Etiquette in the Rotorcraft Pilot's Associate

Description of the RPA

The US Air Force's Pilot's Associate programs were among the first efforts to implement large, adaptive interface and automation management systems [Banks and Lizza, 1991]. The particular type of automation targeted was called an 'associate' system because it was intended to provide many of the same functions and operate in the same relationship as a human associate in a single-seat fighter cockpit. 'Associates' are collections of intelligent aiding systems that, collectively, exhibit the behavior of a capable human [Riley, 1989; Miller and Riley, 1994]. They can (a) perform roughly the same breadth of activities as a human expert in the domain, (b) take initiative when necessary, but generally follow a human's lead, and (c) integrate over ongoing activities to exhibit robust, coordinated, appropriate behavior.

The US Army's Rotorcraft Pilot's Associate (RPA) program was a five year, $80 million research contract managed by the U.S. Army's Aviation Applied Technology Directorate at Ft. Eustis that built on the Pilot's Associate work, but extended it in many ways [Collucci, 1995].. The goal of RPA was to develop and demonstrate in flight an 'associate' system in a next-generation attack/scout helicopter. A critical sub-goal was to manage the information available in future helicopter operations so that human crews can attend to all and only relevant portions at a given time. RPA must accomplish this without increasing pilot workload or decreasing situation awareness. In practice, the RPA module designed to accomplish these goals was the Cockpit Information Manager (CIM) [Miller, Hannen and Guerlain, 1999]. CIM was designed to perform five major functions from the pilots' perspective:

1. *Page (or Format) Selection*—selection of a complete page or format to present on any of the aircraft's presentation devices. For example, the selection of a weapons page instead of a sensors page on the Right Multi-function Display, or the presentation of warbling tone at a specific 'location' via the 3D stereo sound system.
2. S*ymbol Selection/Declutter*—turning specific symbols on or off on a selected page (e.g., include/suppress intervisibility symbology on the Tactical Situation Display).
3. *Window Placement*—control of the location for pop-up windows which would overlay some other visual imagery on the Multi-Function Displays.
4. *Pan and Zoom* –control of centering and field of view of map and sensor displays.
5. *Task Allocation*—the assigning of tasks to various pre-defined, legal combinations of the two human pilots and automation.

Each behavior was adaptive and made use of an inferred task context to determine which information should be presented or which tasks should be allocated in what way. In addition to relying solely on inferred pilot tasks, however, we also implemented several mechanisms to allow the pilots to control and interact with the adaptive behaviors described above. Pilots could, during initial configuration, control the set of options that CIM was permitted to consider, and could apply preference weights to those options. Pilots could also, during the mission, individually turn each of the options on or off. Pilots could also command any display state they desired and the CIM would respond by avoiding modification to the commanded display for a period of time. In this sense, the pilot retained control over the behaviors that the CIM could, and was likely to, perform. CIM's behavior was not entirely predictable—after all, it could still choose between authorized options on the basis of what was appropriate in context—but the parameters of its relationship vis-a-vis the pilots were well defined.

A Crew Coordination Display

The types of pilot control over CIM functions mentioned above were not new to RPA. What was new was a serious look at the 'etiquette' which existed on the helicopter flight deck and an attempt to design the CIM to 'behave well' according to that etiquette.

Prior associate programs (especially the Pilot's Associate—cf. [Banks and Lizza, 1991]) were concerned about forcing the pilot to take on additional workload if s/he had to explicitly communicate and coordinate with the associate, and therefore relied more heavily on intent recognition alone. Direct pilot interactions with the associate (e.g., about what information the pilot wished to see or what actions he was in fact engaged in) were regarded as adding to that workload since they were, after all, interactions the pilot would not have had to perform if the associate were not present. While direct interactions might be warranted, they inevitably had to justify their added cost and were, generally, not favored.

By contrast, initial interactions with helicopter crews and reviews of domain training approaches revealed that, in the attack/scout helicopter domain, as much as a third of crew members time while in the cockpit is engaged in what might be called 'meta-communication' activities—discussion of plans and intentions, allocations and affirmation of responsibilities, maintenance of situation awareness, etc. In fact, helicopter crews were currently undergoing substantial training and review of Crew Resource Management procedures [e.g., Foushee and Helmreich, 1988], which were designed to strengthen crew coordination and team situation awareness. We believed that for the RPA CIM to fail to behave in accordance with this operational ‘etiquette’ – that is, to fail to be able to report on its activities, its perception of the activities of others, and to take instruction about the activities it should be engaged in—would guarantee that it fail to ‘play by the rules’ that human crew members expected.

Our response was to create a method for simple ‘meta-communication’ interactions between the pilots and the associate. Based on initial designs by Dr. Stephanie Guerlain, and ultimately implemented under the supervision of Matthew Hannen at Boeing [cf. Miller, Hannen and Guerlain, 1999], this “Crew Coordination and Task Awareness” display consisted of four small LED buttons located in the upper portion of each pilot’s main instrument panel. Each button was capable of displaying up to two eight-letter lines of text. The buttons were used to report, in textual form, (1) the associate’s current inference about the general, high-level mission context (e.g., that we’re currently engaged in an attack task rather than an evade task), (2) the associate’s inference about the highest priority current pilot task, (3) the task which the associate is engaged in currently which it believes has the highest priority, and (4) the highest priority, inferred copilot task. Pressing these buttons permit either pilot or co-pilot to override CIM’s current inferred tasks and assert new ones (from an automatically scrolled list of higher-level tasks from the overall task network) via a single push button input. Figure 1 shows the RPA cockpit simulation created for evaluation trials at the Boeing Company in Mesa, Arizona. The location of the Crew Coordination display is circled and an enlargement and interpretive sketch of the display is provided for clarity.

The goal of the Crew Coordination and Task Awareness display was to provide the crew with insight into ‘what the associate thought was going on’—as well as a direct ability to affect it. Although such interactions would actually add workload to pilots’ duties (over a perhaps unrealistic baseline where no interaction with the associate was needed or provided), we felt that the ‘etiquette’ of the flight deck demanded that the associate provide at least this level of meta-communication about its intents and its knowledge of the intent of others. For the associate not to have this capability might make it seem less trustworthy. We expected that the inclusion of a Crew Coordination display would facilitate user acceptance of the RPA by making it seem more like a ‘team player’ rather than a silent automaton, as well as allowing the crew to make task-based intent inputs which would improve the accuracy of RPA’s aiding overall. Since the inclusion of a direct method for viewing and interacting with the intent estimation and task network software was a new development in the RPA cockpit (over prior associate system work), we were especially interested in how pilots would regard it.

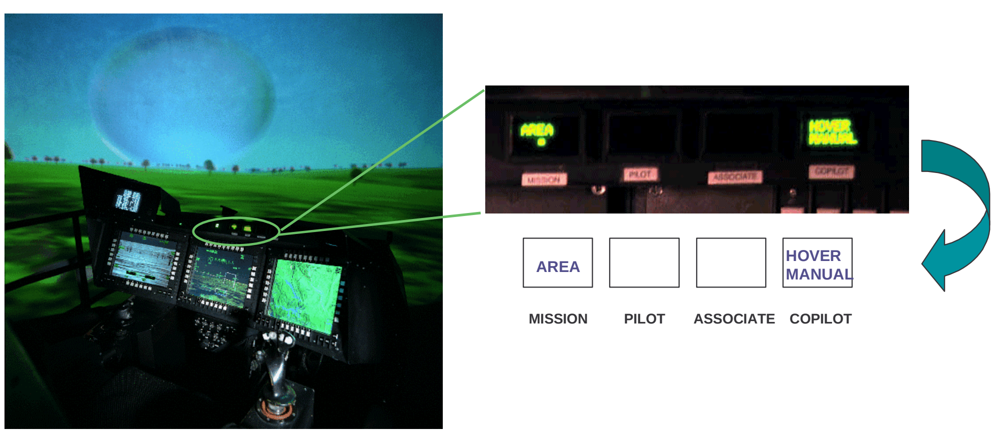


Figure 1. Location and appearance of the Crew Coordination and Task Awareness Display.

Evaluation Results—User Acceptance

Evaluation results designed to determine subjective user acceptance bear these hypotheses out. Extensive full mission simulations were performed to help evaluate the RPA CIM behaviors and implementation, as well as to assist in prioritizing issues for the flight demonstration. Four Crews (a total of eight pilots) trained and flew together, as they do in actual field operations. Crews were given realistic mission briefings and objectives and were permitted to make their own tactical decisions about how to achieve them. Tests were flown in Boeing's full mission simulator (see Figure 1) and included full fidelity RPA cockpits, dome visuals, an extensive range of passive and active threats, and human control of the aviation Tactical Operations Center, friendly artillery, and 1 to 3 wingmen. Realistic communications, including change of mission Fragmentary Operations Orders, were maintained between these players. Each pilot received an average of 10.8 hours of training in the simulator and 13.9 hours of classroom training over a two-week period.

Each crew flew 14 part-mission test scenarios, 7 with the full RPA Cognitive Decision Aiding System (CDAS), and 7 with an Advanced Mission Equipment Package (AMEP) alone. The AMEP was represented the capabilities of an advanced attack/scout helicopter platform, including an impressive array of automation and decision aids, but without the integrating support of an associate system. Each crew also flew four full-mission scenarios—two with the AMEP alone and two with the RPA CDAS in addition to the AMEP. Full-mission scenarios were designed to be highly realistic and crews were given free reign to pursue their commander's objectives via whatever methods they thought appropriate. Crews flew the two AMEP or CDAS full missions in sequence and then switched technology conditions and flew the remaining two missions with the other set of technologies. The sequence in which crew interacted with the different technology packages was counterbalanced to minimize training effects. The simulation test segments and missions were constructed to include numerous examples of the CIM page selection, window location, pan & zoom, and symbol selection behaviors in a variety of tactical mission contexts. (CIM's task allocation behaviors were not implemented in the simulation due to time and budget constraints.)

In order to obtain crew acceptance data, a questionnaire was administered to the pilot and copilot after each of the final AMEP and CDAS full-mission test trials. All of the questionnaire responses utilized complete verbal anchoring and a linear response scale with five equal intervals, in accordance with [Charlton, 1989]. The criteria value for satisfactory CIM behavior, was set at an average score of 3.5 or greater for each response.

The criterion was met for three of the four CIM behaviors. Figure 2 presents the average and range of pilots' ratings of the behaviors . In general, pilots found the CIM behaviors to be 'Of Use' or 'Of Considerable Use.' Figure 3 presents pilots' ratings of their perceptions of the frequency with which they had to override or correct CIM's actions. The average over the CIM behaviors fell between 'Seldom' and 'Now and Then' with symbol selection capabilities performing notably better.

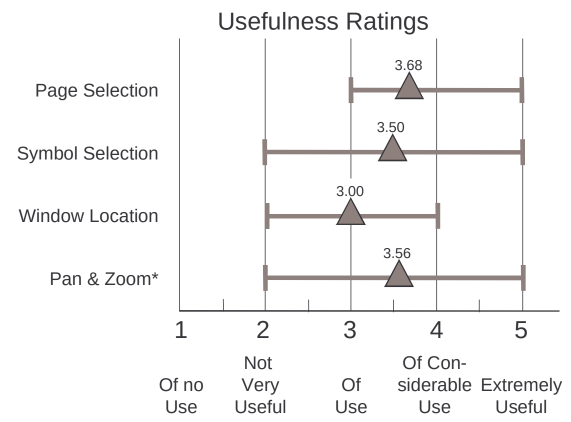
Figure 4 shows pilot ratings of CIM as a whole. CIM was seen as 'Frequently' providing the right information at the right time and was seen as almost always predictable in its behaviors. Finally, Table 1 compares pilot ratings of their effectiveness over four mission types with CDAS versus with the AMEP. On average, pilots found themselves to be more than half a point more ef-

Table 1. Perceived effectiveness in different mission tasks with CDAS and AMEP alone (where 3.0= 'Fair', 4.0='Good' and 5.0='Excellent'.)

Average Rating	AMEP	CDAS
Zone Reconnaissance	3.75	3.88
Area Reconnaissance	3.75	4.25
Deliberate Attack	4.13	4.75
Change to Attack	3.63	4.63

fective (10% of the scale length) with CDAS than without.

The RPA CDAS also produced overall benefits relative in one other critical area. Using TLX measures of subjective workload collected at the end of each trial, workload scores were consistently lower for CDAS conditions than for AMEP conditions (46 points versus 57 points). This difference was significant in an Analysis of Variance [F(1,6)=11.524, p<.05]. Furthermore, separate ANOVAs were conducted for each of the six TLX subscale ratings to determine CDAS' contributions to overall workload reduction. These results are presented in Table 2.



* Pan & Zoom questions were asked as pilot estimates of situation awareness with CDAS pan & zoom capabilities, using anchor terms "Terrible, Poor, Fair, Good, Excellent".

Figure 2. Pilots' average ratings and ranges for the usefulness of the four CIM behaviors.

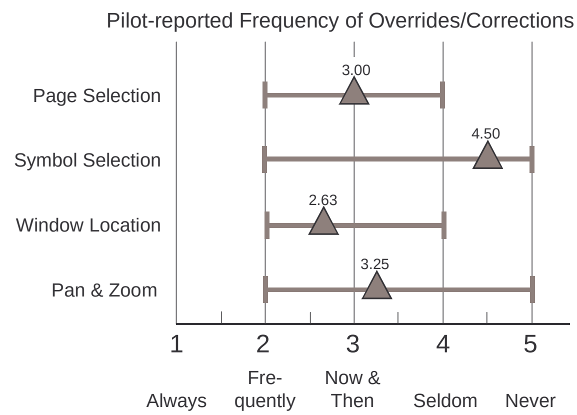


Figure 3. Pilots' average reported perceived frequency of overrides and corrections to the four CIM behaviors.

Reduced workload with the RPA CDAS is apparent in the mental demand, physical demand, temporal demand and effort subscales. There is also a marginal finding for the frustration subscale (p=.07). Means in all cases indicate that the RPA CDAS provides a benefit to the pilot. Examination of the perceived performance ratings, however, shows no effect of configuration. This may indicate that pilots use a different subjective criteria in rating their own performance, possibly judging it based on how well they felt they should have done in the given context, which would include the cockpit configuration.

There is something of a contradiction in the data above. Pilots said they 'Now and Then' or 'Frequently' overrode CIM's behaviors (cf. Figure 3), yet they found those behaviors 'Of Use' or 'Of Considerable Use' (cf. Figure 2), they thought their effectiveness was better with CDAS (cf. Table 1), and their TLX ratings confirm that CDAS offered significant perceived workload reductions. How can such levels of perceived usefulness be achieved along with such perceived error rates?

Table 2. Analysis of the TLX subjective workload subscale ratings.

TLX subscale	AMEP mean	CDAS mean	F-Value (df: 1,6)
Mental Demand	61.77	46.25	10.487*
Physical Demand	54.48	40.31	12.042*
Temporal Demand	62.08	45.73	14.061**
Perceived Performance	35.00	42.08	2.429
Effort	62.60	48.54	20.470**
Frustration	52.81	45.63	4.961

*p<.05 ** p<.01

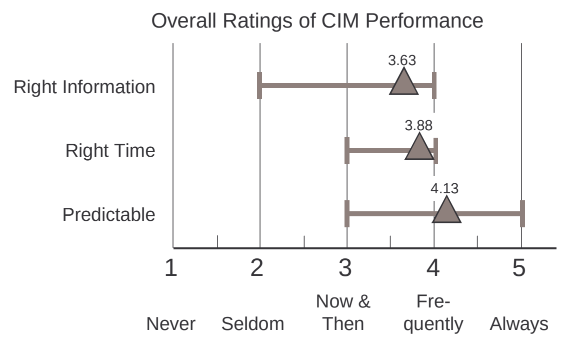


Figure 4. Pilots' average overall ratings of CIM performance.

Table 3. Perceived usefulness of the Crew Coordination and Task Awareness Display (where 4.0='Of Considerable Use' and 5.0='Extremely Useful'.

LED Button for:	Score
Mission Task	4.4
Pilot Task	4.3
Copilot Task	4.3
Associate Task	4.0

The answer may lie partly in the etiquette around which RPA was designed. As seen in Figure 4, CIM was seen as very predictable in its behaviors. It was also, generally, easy to override when it made a mistake. In addition, the inclusion of the Crew Coordination and Task Awareness display seems to have had a significant impact. Most pilots found this display 'Of Considerable Use' or 'Extremely Useful' (cf. Table 3). This provides some supporting evidence that the capability for the crew to interact directly with the associate's assumptions about active tasks was a capability that pilots welcomed—and that may have served to improve their overall impressions of CIM's capabilities and usefulness. Just like human colleagues, the RPA provides value when it is correct, and it facilitates interactions designed to manage, mitigate and correct its mistakes when it is not.

The Implications of Etiquette

Why Etiquette?

The inclusion of this novel interface type stemmed from taking seriously the 'associate' metaphor and using it to guide the behavior of sophisticated and semi-autonomous systems. 'Associates,' whether they be human or automated, must behave in ways appropriate to the established culture of the work environment and in ways that will further the goals of the team. We began with a close inspection of the 'etiquette' or culture used by team members in advanced attack/scout helicopter operations. Then, when attempting to design an 'associate system'—a highly capable, partly autonomous, dynamically adaptive aid which needed to interact with and, in a real sense, to be a part of the crew complement of this vehicle—we used our knowledge of that 'etiquette' to guide design decisions which seem to have paid off.

The success of this interface innovation has led us to think more seriously about the implications of the associate metaphor for adaptive automation in many domains, and about the role that the 'etiquette' of human-automation relationships does and should play. Etiquette seems to be a useful way of thinking about the formalizing the relationship between humans and automation systems. This thought process places the emphasis less on the construction of hardware, software or even knowledge architectures (though these must be in place) and more on the perceived behavior of the adaptive aiding system. What are the ways in which an automated associate *should* behave to support optimal human-system performance?

Etiquette rules are rarely created whole cloth by the Emily Posts or Miss Manners of the world. Instead, they attempt to observe good practices already existing in 'polite society' and then formulate them for others and/or infer from existing practices to propose etiquette for new situations. By proposing etiquette rules for adaptive automation and information management systems to follow, we should take a similar approach: observe good information exchange practices between humans and humans, or between humans and those systems that already exist, and attempt to both explicate good practices for others to follow and extend and generalize good practices to novel domains and situations.

Etiquette rules are not the same for all situations. While the mavens of etiquette may advocate a formalization of behavior for 'polite society', it is clear that there are different kinds of etiquette for different settings and domains. What may be appropriate in the boardroom would be strange in the parlor, and what might be genteel in a formal dinner would seem cold and contrived in a poker game. Furthermore, etiquette rules don't always have to be followed. More importantly for the types of domains we have been concerned with, attempts to make automated systems friendly and sociable [e.g., Reeves and Nass, 1996] may actually interfere with their ability to perform useful work in a team setting—just as human-human teams with a time critical job to do may forego many of the 'niceties' of social interaction. We maintain, however, that this type of goal-directed, team-coordinated behavior is simply another type of etiquette, and not an abandonment of all 'rules of good behavior'.

Finally, there may be times where the conscious and systematic violation of etiquette is highly useful. Nevertheless, consistent violation of rules appropriate to a domain relegates one to an undesirable position in society. We don't claim that every adaptive system should adhere to the same etiquette—just that most should try to find a good and useful notion of the 'rules of behavior' appropriate to the domain and their role in it, and then try to stick to them.

<u>A Tentative List of Etiquette Rules</u>

Given our experience in working on adaptive automation systems (especially, but not exclusively, the RPA work described above) and our familiarity with others in the literature, we have recently drafted a set of 12 'Etiquette Rules' for adaptive automation system behavior. This list attempts to be general; it is quite clear (given the arguments laid out above) that it should be adapted and extended to any specific domain of interaction. For example for rotorcraft pilots (and, probably, most human operators of high criticality systems), predictability in automation function is very important and should be sought after in interactions. To games players, for example, such a rule might be less important or even reversed. To some extent, this list has emphasized desirable behaviors over practicality of implementation though, as the Crew Coordination and Task Awareness display presented above illustrates, it is sometimes possible to fulfill an etiquette rule (i.e., number six below) with comparatively little interface sophistication. As discussed below, this list is evolving and changing. It is intended to provoke discussion and thought about behavioral standards more than to set those standards itself.

1. Make many, many correct interaction moves for every error made
2. Make it very, very easy to override and correct your errors
3. Know when you are wrong—the easiest way to do this is to let the human tell you—and then get out of the way.
4. Don't make the same mistake twice
5. Don't show off—Just because you can do something, doesn't mean you should.
6. Be able to talk explicitly about what you're doing and why—humans spend a lot of time in meta-communication activities facilitating coordination, especially in distributed work environments.
7. Be able to take instruction; not only will this help you adapt to the user's expectations, it may actually make you look smarter.
8. Make use of multiple modalities and information channels redundantly; understand the implications of your communications on *all* the levels on which it operates.

9. Don’t assume every user is the same—be sensitive and adapt to individual, cultural, social, contextual differences
10. Be aware of what the user knows—especially if s/he knows it because you recently conveyed it (i.e., don’t repeat yourself).
11. Try not to interrupt. There may be times when something you want to convey is important enough to warrant interruption, but this will usually not be the case. Err on the side of caution.
12. Be cute only to the extent that it furthers your interaction goals.

<u>The Very Idea . . .</u>

It is important to distinguish between the idea that there ought to be a list of etiquette rules for adaptive interface behavior and the specific list we’ve included above. That list is intended as the starting point for a discussion and series of inquiries whose endpoint might be a full understanding of proper adaptive interface behavior across a wide variety of domains and applications. We can, and should, argue about whether this set of etiquette rules is right or complete. We suspect that the discussion about what works and what doesn’t in different contexts will be informative.

Having drafted our list independently and on the basis of our own experience, we found it intriguing to see a very similar list drafted by Dr. Erik Horvitz [in Horvitz, 1999], one of the principal designers of the Microsoft™ ‘Paperclip’ Office Assistant. Horvitz sees his task as striving to create assistants “with the sensitivity of an intuitive, courteous butler.” That is one specific style of etiquette, it may not be right for all circumstances. In future work for our domains, we are intrigued by attempting to apply the lessons learned about intra-crew communication and resource allocation from the extensive work on Crew Resource Management [e.g., Foushee and Helmreich, 1988] to other goal-driven, high-criticality, ‘real world’ collaboration environments. In any event, the notion that there is an appropriate etiquette for various kinds of human-automation interaction is powerful. Now all that is needed is work designed to discover those ‘rules’ and to get our machines to play by them.

Acknowledgements

This work was supported by the U.S. Army AATD, contract number DAAJ02-93-C-0008. The authors would like to thank Stephanie Guerlain, Matt Hannen, Mark Ringer, Denny Rother, Erik Steinmetz, Ruben Yanas, Bernie McBryan, and Ray Higgins for their assistance in performing the work reported here.

References

Banks, S. & Lizza, C., (1991). “Pilot’s Associate; A cooperative knowledge-based system application.” *IEEE Expert*, June 1991, 18-29.

Bauer, M. Dengler, D., and Paul, G. (2000). Instructible Information Agents for Web Mining. In *Proceedings of the International Conference on Intelligent User Interfaces*. January 9-12; New Orleans, LA.

Bonner, M., Taylor, R. and Miller, C. (2000). Tasking Interface Manager: Affording pilot control of adaptive automation and aiding. In *Proceedings of the Annual Conference of the Ergonomics Society*, Grantham, UK, 4-6 April.

Charniak, E. and Goldman, R. (1993). A Bayesian approach to Plan Recognition. *Artificial Intelligence, 64*, 53-97.

Cochran, T., Miller, C. and Bullemer, P. (1996). “Abnormal Situation Management in Petrochemical Plants: Can a Pilot’s Associate Crack Crude?” In *Proceedings of the National Aerospace and Electronics Conference (NAECON*). Dayton, OH; May.

Charlton, S. G. (Ed.). (1989). *Operational Effectiveness Questionnaire Handbook*, Subjective Assessment Working Group, US Air Force Operational Test and Evaluation Center, Technical Paper 7.1, July.

Colucci, F. Rotorcraft pilot's associate update: The army's largest science and technology program. *Vertiflite*, (March/April 1995), 16-20.

Coury B., Santarelli, T. and Mitchell, T. (2000). Expert Modeling for Adaptive Automation and Intelligent Systems. In *Proceedings of Human Performance, Situation Assessment and Automation: User-Centered Design for the New Millennium*. Savannah, GA; October 15-19. 160-165.

Foushee, H.C. and Helmreich, R.L. (1988). Group Interaction and Flight Crew Performance. In Human Factors in Aviation. Academic Press, San Diego CA.

Funk, H., Miller, C., Johnson, C. and Richardson, J. (2000). Applying Intent-Sensitive Policy to Automated Resource Allocation: Command, Communication and Most Importantly, Control. In *Proceedings of the Conference on Human Interaction with Complex Systems*. Urbana-Champaign, Ill. May.

Horvitz, E. (1999). Principles of Mixed-Initiative User Interfaces. *Proceedings of CHI '99, ACM SIGCHI Conference on Human Factors in Computing Systems*, Pittsburgh, PA, May.

Hoshstrasser, B. and Geddes, N. (1989). OPAL operator intent inferencing for intelligent operator support systems. In V. Shalin and G. Boy, (Eds.) *Proceedings of the IJCAI-89 Workshop on Integrated human-Machine Intelligence in Aerospace Systems.* 53-70.

Maybury, M. and Wahlster, W. (1998). *Readings in Intelligent User Interfaces*. Morgan Kaufman.

Miller, C. (2000). Intelligent User Interfaces for Correspondence Domains: Moving IUIs 'Off the Desktop.' In *Proceedings of the International Conference on Intelligent User Interfaces*. January 9-12; New Orleans, LA.

Miller, C. and Riley, V. (1994). "Achieving the associate relationship; Lessons learned from 10 years of research and design." In *Proceedings of the 3rd International Conference on Human Computer Teamwork* . Cambridge, UK; September.

Miller, C., Pelican, M. and Goldman, R. (2000). "Tasking" Interfaces for Flexible Interaction with Automation: Keeping the Operator in Control. In *Proceedings of the Conference on Human Interaction with Complex Systems*. Urbana-Champaign, Ill. May.

Miller, C., Hannen, M. and Guerlain, S. (1999). The Rotorcraft Pilot's Associate Cockpit Information Manager: Acceptable Behavior from a New Crew Member. In *Proceedings of the American Helicopter Society's FORUM 55*, Montreal, Quebec, May 25-27.

Mitchell, C. (1999). Model-based design of human interaction with complex systems. In A.P. Sage and W. Rouse (Eds.) *Systems engineering and management handbook*. New York; John Wiley.

Parasuraman, R., Sheridan, T. and Wickens, C. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans, 30*(3). 286-296.

Perrow, C. (1999). *Normal Accidents*. Princeton University Press; Princeton, NJ.

Puerta, A. (1996). The MECANO Project: Comprehensive and Integrated Support for Model-Based Interface Development. In *Proceedings of CADUI96: Computer Aided Design of User Interfaces*. 1996. Numur, Belgium.

Reeves, B. and Nass, C. (1996). *The Media Equation: How people treat computers, television, and new media like real people and places*. Cambridge University Press; New York.

Riley, V. A general model of mixed-initiative human-machine systems. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Denver CO, October 1989) 124-128.

Sarter, N.B., Woods, D.D., and Billings, C.E. (1997). Automation Surprises. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (2nd edition) (pp. 1926-1943). New York, NY: Wiley.

Sheridan, T. (1987). Supervisory Control. In G. Salvendy (Ed.), *Handbook of Human Factors*. John Wiley & Sons, New York. 1244-1268.

Strohal, M. and Onken, R. (1998). Intent and error recognition as part of a knowledge-based cockpit assistant. In *Proceedings of SPIE, Vol. 3390: Applications and Science of Computational Intelligence*. April 13-17; Orlando, FL.