Etiquette in Human Computer Interactions: What Does it Mean for a Computer to be Polite? or Who Needs Polite Computers Anyway?

Caroline Hayes, Amit Pande
Department of Industrial Engineering
University of Minnesota
111 Church St S.E, Minneapolis, MN 55455
hayes@me.umn.edu, amit@me.umn.edu

Chris Miller
Smart Information Flow Technologies
2119 Oliver Avenue South Minneapolis
MN 55405
cmiller@sift.info

Abstract

We define *Human-computer interaction (HCI) etiquette rules* as a type of software design guideline aimed at facilitating smooth and effective interactions between humans and computers. They do so, by insuring that the software follows expected rules of interaction appropriate to the computer's role in a team. In particular, HCI etiquette rules from the cultural conventions of a particular group. In design of decision support tools to assist specific work groups such as a military intelligence analysts, a team of product designers, or the crew of a helicopter, it is particularly important to understand the sub-culture of that work group. In many domains, following the "etiquette" of the work groups can be key in getting human users to accept the software as part of the team, and to use its benefits. Software that does not follow rules of HCI etiquette, even if otherwise well designed may be rejected by users. For example, if the software interrupts too much during crisis periods, or fails to provide the type of interactions, such as justifications, or redirection of goals, that the user typically expects of his or her subordinates. The term human computer etiquette was first introduced by Miller and Funk. In this paper, our goals are to refine their definition of HCI etiquette, and define it as one (of many) types of HCI design rules, and to establish the necessity benefits of viewing HCI etiquette as an explicit category.

Introduction

In order to interact effectively with other humans, both humans *and* computers need to "follow the rules of the game." "The game" might be anything from a soccer game, a group design meeting, or an interaction between a commercial airline pilot and the support staff. Unfortunately, the rules are often unwritten, which makes it particularly difficult for computers to follow them. Humans often come to understand the rules of polite and acceptable interaction in a given context through interaction. Intelligent computer agents, however, need to have these rules spelled out for them by the software designer if they are to interact effectively with humans.

The "rules of the game" or rules of appropriate and fair social interactions is what we commonly refer as etiquette. We argue that etiquette is not just about pretty formalities that only Miss Manners takes seriously; etiquette is a powerful mechanism which has developed in *all* cultures specifically to facilitate smooth and efficient communications and interactions. If everyone understands and follows the same conventions, interactions can proceed with a minimum of friction. If no such rules exist, then interactions may be slowed to a standstill because individuals attempting to interact continually collide with each other or "step on each others toes" (either literally or cognitively). Thus, we argue that when the etiquette of a particular culture or work group is ignored, the effectiveness of interactions may be seriously compromised.

Many designers for computer associates and computer decision support tools often overlook "etiquette" issues in designing their systems to interact with humans. Furthermore, anecdotal evidence implies that in many cases they can get away with ignoring it; there are many examples of systems in which users will still use a system if poorly designed. However, there is also much evidence that in many other situations (particularly in high-stress, time critical decision making situations) that users will refuse to use a computer tool if it interrupts at an inappropriate time, or fails to accept re-direction. (In other words, if the computer tool fails to act as a respectful, biddable, and accountable assistant would.) Even in situations where the designer can "get away with" ignoring etiquette issues, we feel the effectiveness and usability of such systems can be greatly enhanced by incorporating HCI etiquette rules into the design.

Miller and Funk introduced the concept of etiquette in human computer interactions (which we will abbreviate as HCI etiquette) as the practice of behaving "in ways appropriate to the established culture of the work environment ... that will further the goals of the team (**?**)." Miller further describes human-computer *etiquette* as the "defined roles, acceptable behaviors and interaction moves of human and intelligent agent participants in a common setting (**?**)." They propose the definition of an HCI etiquette as one of many design issues that must be considered in creation of an associate system. *An associate system* is an intelligent computer tool that assists one or more humans to arrive at a decision, plan or design. Associate systems differ from automated problem solvers in that associate systems share the work of problem solving with the human, and tend to leave final decision making under the human user's control.

Our goals in the remainder of this paper are to discuss how the term *human computer etiquette* first arose, to refine this early definition of human computer etiquette, to put it in the context of other types of HCI design rules, and to characterize some of the properties of HCI etiquette rules.

Development of the Concept of HCI Etiquette

The motivations for introducing the concept of HCI etiquette arose during work on a project called Rotorcraft Pilot's Associate (RPA) (**?**). The goals of the project were to develop a computer associate to assist helicopter flight crews during flight. It was to do so by helping human crew members to identify relevant operational information in the cockpit without increasing their cognitive workload or decreasing their situational awareness.

Previous solutions were concerned with identifying ways to interact with users without requiring active input from them. The concern was that requiring active interaction with the associate program would increase the users' cognitive workload. These programs operated by attempting to recognize the current intent of the operator through observations of his or her interactions with the computer. By inferring the operator's current goals, the computer also inferred what information was appropriate to display (**?**). However, users are often frustrated if a system mis-infers their needs and presents the wrong information, particularly if they are unable to redirect the actions of the computer.

However, studies of the behavior of pilot crews show that as much as one third of crew member's time is spent in "meta-communications (**?**)." In other words, discussions of plans, intentions, responsibilities, etc. Furthermore, a study of pilots and cockpit designers showed that pilots have a strong desire to remain in control of both the tasks executed by a computer associate, and the information it presents (**?**). Thus, a model of an associate that requires (and accepts) no active input from the pilot but instead attempts to guess what is needed without being told, fits neither current crew practices, nor crew members' expressed preferences for interactions with an associate. The expected behavior for a subordinate (or computer associate) includes answering questions about what the subordinates current goals are, and receiving re-direction if the supervisor so desires. Doing so almost necessarily implies the need for communications between the pilot and associate on goals, plans, and responsibilities.

As we interpret Miller and Funk's paper, human computer etiquette is the concept they devised to describe the rules of effective interactions between humans and computer associates. They designed the Cockpit Information Manager (CIM) to support interactions, following what they perceived to be etiquette rules between pilots and associates. The CIM is a computer module and information display, included as a part of the RPA. The CIM's display enabled pilots to observe and (if necessary) reset the associates current inferences about the mission, pilot's, and co-pilot tasks (**?**).

Evaluations showed that despite the fact that pilots frequently made adjustments to the goals and information displayed by the CIM, their frustration levels and subjective workloads were significantly reduced when using the RPA with the CIM module, vs RPA alone (**?**). Although they do not explicitly state which etiquette rules they were following in design of the CIM the authors of this paper are inferring that they include rules such as:

- Make it very easy for the user to override and correct the associates errors.
- The associate should be able to explicitly tell the user what it is doing and why.
- Try not to interrupt the user.

Miller and Funk's Tentative List of Etiquette Rules

Miller and Funk additionally proposed an initial, more general list of HCI etiquette rules, which we paraphrase below:

1. Produce many correct results for every error made.
2. Make it very easy for the user to override and correct the associate's errors.
3. The associate should not make the same mistake twice.
4. The associate should be able to explicitly tell the user what it is doing and why.
5. An associate should be able to take instruction from the user.
6. An associate should communicate to the user through multiple modalies and information channels redundantly.
7. The associate system should try not to interrupt the user unless it is absolutely necessary.
8. Associate systems should be aware of what the user already knows, and what the associate system has already conveyed to him or her.
9. Designers of associate systems must be cautious in applying technology: do not automate a task simply because it is possible.
10. Designers of associate systems should not assume that all users are the same. The system should adapt to the individual where possible.

However on examining their etiquette rules, the questions arises "Are rules for appropriate HCI etiquette really different from rules for appropriate human computer interaction?" and "Do we really need a separate category of HCI to address human-computer etiquette?" We believe this ambiguity may arise because etiquette rules may have some overlap with more general rules for effective HCI. Furthermore, some of the specific rules listed above may be better classified as general HCI rules, rather than as etiquette rules.

Refining the Definition of Etiquette

We propose that rules for HCI etiquette should be viewed as a special sub-category of the general rules for effective HCI design. Etiquette rules differ from general rules for effective HCI design because etiquette focuses on the requirements necessary for convincing humans to accept an associate system as part of a problem solving team, and to allow it to continue participating:

Human computer etiquette defines the rules of behavior necessary for enabling computers to act as socially acceptable members of a larger team working together towards a common goal. Etiquette rules are those that specifically enable computer associates to gain initial acceptance as members of the team, and to maintain membership.

There are many types of system design rules aimed at facilitating effective human computer interactions. Each type arises from various types of information processing constraints. Figure 1 outlines a few of these constraints in the context of an information processing model, which includes a computer decision support tool. The computer tool is shown intervening between the human user, and the physical system that they are manipulating. Examples of constraints that impact the design of the tool's human computer interactions include: the human's information processing limitations (in terms of perception, memory, etc.), the physical work environment, and the social context of the users.

Some rules for effective HCI design focus on designing computer displays that respect the limitations of human perception, processing, memory, attention or reaction speed. For example, a design rule such as "use only one or a few shades of any given color in your display because humans can only distinguish a finite number of shades," is an example of a design rule aimed at working with the limitations of human color perception. Other rules focus on the constraints of the work environment, such as "Use auditory human computer interactions sparingly in a noisy work environment." Others yet, focus on the constraints of the physical systems manipulated, such as "Because the fuel tank can only hold 60 gallons, the simulated display of the fuel tank should also not allow the fuel tank capacity to exceed 60 gallons.

HCI etiquette rules focus, in contrast, on satisfying *social* interaction constraints and conventions developed within a national culture or specific work group (shown in grey in Figure 1). For example, we would consider a guideline such as "Associate systems should not interrupt humans" to be an example of an etiquette rule; a system that is constantly irritating users with information that is not relevant to the current goal will probably be turned off (i.e. not allowed to participate in the problem solving). The specific set of HCI etiquette rules adhered to is not only dependent on the conventions of the work context, but also on the roles that both the human and the computer play in that work group. Is the computer's role to be a slave, teacher, assistant or boss to the human? (The latter is usually not tolerated by humans). Is the human's role to be a manager, a colleague, a student, or simply someone to push the start button? For example, the etiquette rules appropriate for a computer tutor interacting with a domain novice, are likely to be very different from the etiquette rules that apply to a computer "gopher" gathering information for a general or company CEO.

Roles play a strong role in etiquette because they determine a "pecking order" or power precedence in work groups. Role may not only be determined solely by rank, but factors such as level of expertise, respect earned by the actor in the group, etc. An actor's rank in the pecking order, and the priority of the task on which they are working, largely determines the tolerance that the other group members have for that actor to interrupt or redirect the group focus. Thus, an important part of establishing a human-computer etiquette in any work domain, may be the definition of the roles and pecking order for all participants (both human and computer) and all tasks in the work group. This information can then be used to establish a set of HCI etiquette guidelines for when the computer should stay quiet and when it is appropriate to interrupt.

Lastly, unlike human to human etiquette rules, HCI etiquette rules apply only in one direction. Computers must be polite to humans, but humans do not necessarily need to be polite to computers.

In summary, computer etiquette is a necessary, but not sufficient condition for effective human-computer interaction. It is necessary that the human user accept the associate system as a part of the team in order for any human computer interactions. However, user acceptance does not guarantee that the interaction will be effective. In some cases, users will accept systems that actually decrease their performance.

The Nature of HCI Etiquette

Etiquette rules are about the established conventions governing communications within a work community. Thus, we view HCI Etiquette rules as a type of mental "affordances" established to facilitate communication between associate systems and human users.

Some of the issues that influence whether a human user is willing to accept a computer associate as a part of the team include:

- **Trust**. Can the user trust the associate to competently complete its assigned responsibilities? produce correct solutions most of the time? able to provide relevant and understandable explanations of goals and/or logic leading up to solutions.
- **Helpfulness**. To establish trust, the associate system must be able to provide explanations, when asked and those explanations must be at the appropriate *level of abstraction* w.r.t. the users domain knowledge, or lack there of. For experts this may mean a lack of unnecessary wordiness, facilitating efficient communications and work. For novices this may mean sufficiently detailed explanations, facilitating clear understanding and learning.
- **Coordination** of communications and other interactions. This issue is also related to trust. A user is more likely to trust a system if they know why it is doing what it is doing. An essential part of teamwork is making sure that team members are on the "same page." This rarely happens automatically. The associate system needs to provide a mechanism for allowing users to inspect its goals or other relevant aspects of its internal state.
- **Control** of the associate's goals and actions. This issue is also related to trust. Since no computer system or human is correct all of the time, a user is more likely to be willing to work with an associate if they can change critical aspects of the problem solving or solution. The associate

should allow the user to control or modify the facets of the problem solving process and solution that they wish to control.

- **Understanding of limitations.** Yet another issue related to trust is: does the user understand what the system can and cannot do? A user may fail to trust a system if he or she fails to understand what it is designed to do. There needs to be some communication or understanding on the part of the user as to what roles and tasks the associate can and should take on.
- **Floor control**. When multiple agents interact, there must be an established schema by which agents take "control of the floor" and by which interruptions are permitted. Commonly, one does not interrupt a discussion of a more critical issue with a less critical one. Persons of higher rank are given more leeway to introduce topics, or interrupt then those of lower rank. Associate systems are typically considered to be very low ranking (i.e. servants) and need to act accordingly.
- **Lack irritating behaviors**. A user will not use a system that irritates (by violating any of the etiquette rules above) or insults them too much. For example, an elderly person may attempt to defeat health monitoring software if they perceive it to assume them to be incompetent.
- **Convenience**.
- **Perceived value added**. A user will not use an associate system unless he or she perceives it as adding more value that the work required to use it. This may be a combination of the behaviors above, for example, Perceived value = (convenience + trust) / irritating behaviors.

Future issues and Conclusions

The concept of HCI etiquette is a very new one (although it has its roots in a much older literature pertaining to computers and social interactions.) Thus, there are many future issues to be explored. Under what circumstances, and in what type of domains is it most critical to consider HCI etiquette issues? Can we measure the benefits of following HCI etiquette rules? By what methods can a designer most effectively identify HCI etiquette rules for a new work group? To what extent can computers be made to learn etiquette rules while interacting in group problem solving (i.e. Can computers learn etiquette rules on the job)?

We believe HCI etiquette rules to be an important class of often over looked rules for effective HCI design, which deserve more study and attention. They are not sufficient by themselves to make a computer tool effective. However, we feel in many domains it is well worth the designer's effort to identify, articulate and implement HCI etiquette rules for a work group, so that they can reap the benefits of greater user acceptance, and interaction effectiveness.

References

Banks, S., and Lizza, C. 1991. Pilot's associate: A cooperative knowledge-based system application. Pages 18-29.

Foushee, H., and Helmreich, R. 1988. Group interaction and flight crew performance.

Miller, C. A., and Funk, H. B. 2001. Associates with etiquette: Meta-communication to make human-automation interaction more natural, productive and polite.

Miller, C.; M., H.; and Guerlain, S. 1999. The rotocraft pilot's associate cockpit information manager, acceptable behavior from a new crew member.

Miller, C. A. 2000. Rules of etiquette, or how a mannerly aui should comport itself to gain social acceptance and be perceived as gracious and well-behaved in polite society.

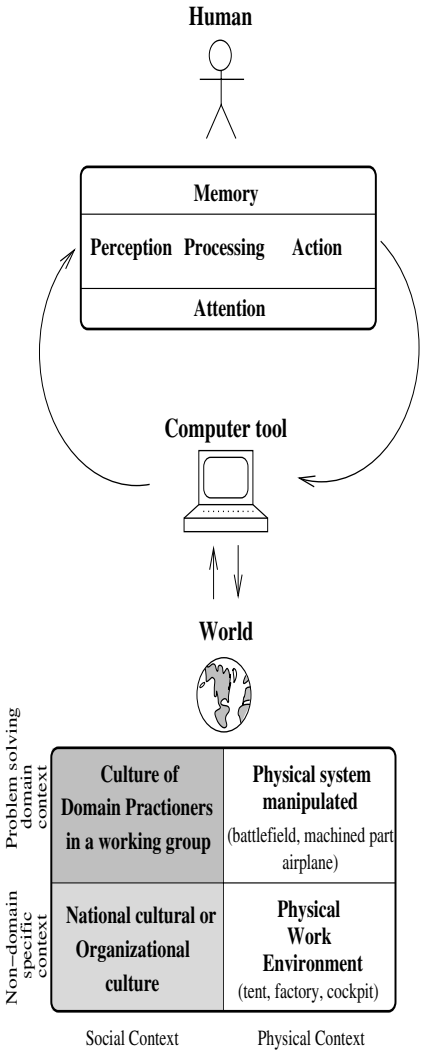

Figure 1: In information processing model, with computer tool.