



Interfaces That Give and Take Advice

Henry Lieberman

21.1 Introduction: Advance-Based Interfaces

Why is almost every kind of input by a human to a computer referred to as a “command”? This strikes up the image of the user as a military commander issuing orders, and the computer as an obedient soldier. But perhaps this is not the best kind of relationship that we can envision between a person and a computer to collaborate in problem solving.

The challenge for the new millenium will be to evolve the human computer relationship from a “command” structure to a more flexible and collaborative stance. I propose that a key idea in achieving this will be the development of computer interfaces based on the idea of *advice*.

Computer systems will act in an advisory role to give people suggestions, help and assistance. People will utilize advice as a means of affecting the computer’s operation without the rigidity of “commands.” In this paper, we will explore the notion of advice, give a few examples, and point to prospects for the future.

I believe that an important direction for the future is to explore how we could build human-computer interaction around the notion of *advice* as the primary means of communication. The computer can both give advice to the user and take advice from the user.

The idea of the computer relating to people as both advisor and advisee is becoming increasingly important as applications get more and more advanced. Sophisticated media interfaces are accruing more and more functionality, and soon the bottleneck will not be how to get our computers to do more for us but how to let them know what we would like them to do. Intelligent software will have the capability

476 CHAPTER 21 Interfaces That Give and Take Advice

to do more and more problem solving for us, but there is also the danger that the machine may take unwanted actions or we may distance ourselves too far from the problem-solving process. The best solution may sometimes be to set up the computer in an advisory capacity, leaving critical decisions and top-level control to the human.

Too much of today's human-computer interaction is couched in terms of *commands*. This is evidenced by the fact that the word "command" has almost come to mean any input whatsoever to a computer. The human "commands" the computer through icons and menu operations. Traditional computer interfaces tend to leave users feeling "commanded" by their rigid and inflexible nature, which accounts for much of the hostility the general public feels toward computers.

The best theories of management in business are moving away from viewing managers as autocratic "bosses" of passively obedient workers to a more cooperative model where managers act as advisors to help empower their workers. Similarly, we hope to abandon the military command-and-control view of the user-computer relationship and move toward a more cooperative model. The route to this is through the notion of advice.

Advice is softer. It needn't be as exact as commands. It need not specify everything the advisee must do but only address some aspect or part of the advisee's behavior. It can suggest action or modify action rather than specify action. It need not solve the whole problem—only part of it. It does not need to be given or received in a strict sequential order. It can critique past behavior or serve as guidelines for future behavior. In these qualities advice differs from what we ordinarily think of as "programming" computers.

Advice is ongoing. Although advice may be in the form of a conversation between the advisor and advisee, it assumes that both the advisor and advisee are active simultaneously. This is quite unlike traditional query-response computer interfaces that assume that either the human is working or the computer is working but not both at the same time. Advice assumes that both the advisor and the advisee will continue to operate even in the absence of the advice, and the goal of the advice is improvement of the advisee's behavior.

The necessity of advice in computer interfaces was recognized as early as 1959, when John McCarthy proposed as a goal a machine that could act as an *Advice Taker* (McCarthy 1959). He defined "common sense" as the knowledge that would enable a human or machine to meaningfully accept advice. McCarthy viewed advice-taking as a stepping-stone toward a fully autonomous artificial intelligence, but advice-taking interfaces have value of their own that is yet to be fully appreciated.

21.1.1 Agents and Advice

Intelligent software agents have recently become a popular movement for extending the human interface beyond traditional direct manipulation interfaces. Not all agents are advice-givers or advice-takers, but the role of advice as a means of communication between an agent and a user is a central component of agent competence.

In many agent projects, the user is encouraged to think of the agent as serving the role of a butler or secretary (Negroponte 1995). Someone in such a role can often provide services that are personalized to an individual's needs and desires, and they are often most appreciated when the agent can provide them without even being asked. When the agent is simply providing a service to the user, the issue of advice may not come into play at all.

But the user's relationship with the agent consists of more than having the agent do things for the user. First of all, if the agent is to act according to the user's desires and preferences, the agent must first *learn* what it is that the user wants, likes, and dislikes. The most natural way for the user to express desires and preferences to the agent is in the form of advice to the agent. As the agent performs services (or fails to in a situation where it really should), the user can *critique* the performance of the agent, acting as advice for future behavior. Finally the user may turn to the agent for advice in deciding what action to take which is one of the ways in which agents can be of the most helpful to the user.

Advice-giving and advice-taking agents can be a way of responding to critics who oppose the whole idea of agents. They argue that humans are abdicating responsibility to software agents and that there is a danger that the agent will autonomously take harmful actions. Advice gives the user an avenue of control without being burdensome and enables the user to benefit from the agent's work while retaining responsibility for decisions.

21.2 Examples of Advice in Interfaces

What would advice really look like in a computer interface?

I'll illustrate two examples from my own work, one in which the computer gives advice and another where it takes advice from the user. In both cases, the original goal of the project was not solely to explore the notion of advice, but as these projects developed, they convinced me of the importance of the advisory nature of the interaction.

21.2.1 Letizia: A Web Browser That Gives Advice

One way that an agent can advise the user is for the agent to try to accomplish the same task as it perceives the user to be doing. It can make its attempt while the user is idle or while working on the same or perhaps another task. When the agent feels it has accomplished something that might interest the user, it gives the benefit of its efforts to him or her as advice. It is up to the user to accept, reject, or ignore the advice.

Letizia is a user interface agent that assists a user browsing the World Wide Web (Lieberman 1997). It treats browsing as a cooperative venture between the human user and the agent. As the human is surfing [in the left window in the illustration], the

478 CHAPTER 21 Interfaces That Give and Take Advice

agent also surfs, and continuously shows its advice to the user in the windows on the right.

Letizia's advice has the following qualities.

- *It is personalized.* It watches the user's actions and compiles an interest profile, which it uses to filter its search.
- *It is ongoing.* Letizia operates continuously, making use of the user's idle time, and displays a continuous, "channel surfing" display of its recommendations. It reacts immediately to changes in the user's browsing behavior.
- *It is Noncoercive.* The user may, at any moment, choose to browse the pages that Letizia recommends or simply ignore it.

Right now, Letizia doesn't really accept explicit advice from the user, except in the form of observing the user's actions and making decisions based on them. However, explicit advice could be incorporated in a number of ways.

Users could give both feedback on the specific choices or criteria for determining those choices, all in real time as the program is running. Speech (see the next section) would also be a particularly useful modality for accepting advice, since the user is otherwise occupied by the browsing process. Users could supply commentary on both their own behavior and that of the agent and the relationship between them. Users could also explicitly advise and modify the profile that Letizia forms of them.

21.2.2 Mondrian: A Graphical Editor That Takes Advice

Mondrian (Lieberman 1995a) is a learning agent for an object-oriented graphical editor that learns new procedures demonstrated by the user by example. Mondrian takes advice from the user about how to learn new graphical procedures. Since the domain is graphical, often the advice comes in contextual forms.

Advice can be *demonstrative*. In the beginning of the demonstration, specific objects can be designated as being examples, and that action advises Mondrian that any further operations involving these examples may be generalized in the future.

Advice can be *graphical*. The user draws a diagram that shows Mondrian about part-whole relationships. In my experiments in a technical procedure domain, I allow the user to annotate selected frames of video by outlining images of parts and grouping them into part-whole relationships. This means that when Mondrian records an action involving that part, it is described in terms of its relationship in the object's structure.

Advice can be *verbal*. A speech recognition system is trained to recognize a set of visual relations, constraints, and predicates that modify the actions the user is taking or objects that the user is manipulating. When the user gives verbal advice, that changes how Mondrian is interpreting the user's actions (Stoehr and Lieberman 1995).

This gives the interaction a natural "show and tell" feel. In many speech applications, spoken commands simply replace typed or mouse commands. However, speech

is more appropriate as a medium for giving advice, since the hands and eyes are already occupied with interacting with the graphical interface.

Mondrian represents a class of interfaces where advice is given *by example*. The user uses concrete, visual examples to explain what is expected, together with actions that tell the system how to interpret the examples. Using examples is often easier for people than preparing advice in the abstract and less prone to error. An outstanding problem is achieving some sort of integration between the kind of advice-giving interfaces represented by Letizia and the kind of advice-taking interfaces represented by Mondrian.

21.3 Advice-Based Interfaces in AI and HCI

While advice-based interfaces are not yet widespread, many investigators are tackling conceptual problems that provide important capabilities that will enable such interfaces. Some of these come from the field of Artificial Intelligence (AI), where the basic representation and reasoning processes are studied, and some from the traditional field of Human-Computer Interaction (HCI), where the interaction issues such as feedback, learnability, and convenience are addressed. A true synthesis has not yet been fully achieved, but the beginnings and recent developments can be observed in the growing subfield of Intelligent User Interfaces (IUI), as evidenced the annual conferences of this subfield. Information on these conferences can be found at <http://www.iuiconf.org/>. The following sections discuss some of the important trends.

21.3.1 More Flexible Planning and Reasoning

Planning (determining a sequence of actions that is aimed at achieving a desired goal in a certain situation) has long been an important topic in AI. However, classical planning assumes that all the steps are worked out in advance before execution of the problem-solving steps starts, that all steps will have their desired effect, and that the world is relatively static between steps. Systems that will give and take advice often do not satisfy these assumptions, so that more flexible notions of planning and reasoning become more applicable to advice-based interfaces. Mixed-initiative planning allows some of the goals and actions to be specified by the machine and others by the user in an integrated fashion.

Collagen (Rich and Sidner 1998) is a good example of how such a mixed-initiative user interface can accept advice from the user. In their system, the user and an agent cooperate to plan a travel itinerary. Advice from the user enables dynamic recognition of plan patterns and incremental replanning on the part of the system. Either party can take the initiative [“Let’s work on <goal>”] or cede initiative [“What next?”].

480 CHAPTER 21 Interfaces That Give and Take Advice

21.3.2 Resource-Limited Reasoning

Some characteristics of human reasoning that are often not shared by formal systems are that humans get tired or frustrated, have limited attention, and have competing demands on their attention. These characteristics, far from being flaws, are actually important mechanisms for the control of problem-solving processes, as Minsky (1988) pointed out in his Society of Mind theory.

Many kinds of advice that a user would like to give to a system are about how to deal with these kinds of limitations on resources. We've already seen how Mondrian accepts advice from the user on what to pay attention to, thereby short-circuiting an otherwise potentially intractable computation about how to generalize an example.

Systems also need to consider the user's limitations when they are providing advice for that advice to be useful to the user. Systems that assume that the user has infinite attention, complete knowledge of the system, and infinite patience quickly become annoying. Horvitz (1999) describes a system that dynamically adjusts its advice according to a user model describing the user's attention in a situation-dependent manner.

21.3.3 Anytime Algorithms

A related recent development in theoretical AI is emphasis on so-called *anytime algorithms* (Zilberstein 1996). These algorithms for planning and problem solving, rather than insisting on running to completion, can be paused at any time [or at sufficiently frequent intervals] and can return the "best answer so far." Making algorithms used by software agents incremental in this fashion facilitates the dynamic adjustment of resources mentioned above. It also facilitates accepting advice, since the user can be assured of being able to interrupt the system before it gets too far "off-track."

21.3.4 Critics

In the Letizia example cited previously, advice from the system to the user takes the form of suggestions. Another role for the system's advice is to take the form of criticizing proposals from the user. This, of course, is a delicate manner, as users might react badly to what they judge as negative feedback from the system. However, users might appreciate well-presented reminders if they are overlooking important domain constraints. A classic example is Fischer's kitchen-design critic (Fischer and Nakakoji 1993), where the system informs the user if a given kitchen design violates the heuristic of minimizing the distance of the triangle between the kitchen, the sink, and the refrigerator.

21.3.5 Programming by Example

There are many kinds of advice lying along a spectrum from very explicit, almost procedural advice, to general hints and suggestions. At the explicit end, advice borders on

programming. As programming becomes more high level, it starts looking more like advice. Lieberman and Maulsby (1996) refer to this as the “spectrum of instructibility.” A middle ground that is gaining popularity is populated by systems, that, like Mondrian, take their advice in the form of demonstrations on specific examples, possibly along with a conversation between the human and the machine about how those examples are to be generalized. Such systems, referred to as Programming by Example or Programming by Demonstration systems, are a “sweet spot” in the inevitable tradeoff between convenience and control. A survey of the state of the art of such systems appears in Lieberman (2000).

21.3.6 Context-Sensitivity

An important aspect of good advice is that it is always sensitive to the context in which it is given, and making use of good advice depends crucially on the context in which it is received. Computer science, in general has had trouble with the idea of context because it goes against the grain of computer science’s constant search for abstraction. Computer science, in the tradition of mathematics, is always trying to make things more abstract, and the only way you can make rules or assertions more abstract is to take them out of context, ignoring details of particular situations. Now, though, in order to make intelligent agent software and embedded devices more useful to their users, we have to swing the pendulum back to taking more seriously the idea of context sensitivity. The problem for advice-giving and advice-taking interfaces is that identifying the relevant portions of the context and communicating them between the human and the machine puts too much burden on the user if it needs to be done explicitly. Thus, systems will have to figure out automatically what parts of the context (time, place, history of interaction, etc.) are relevant and automatically take them into account. This argument is fully developed in Lieberman and Selker (2000).

21.4 The Future of Advice-Oriented Interfaces

Letizia and Mondrian are two examples of the current state of the art in advice-giving and advice-taking interfaces. Current trends in interface development point toward the increasing need for such interfaces and some of the directions in which they might develop.

21.4.1 Internet Applications

The vastness and lack of organization of the Internet cries out for interfaces that advise users on how to look for, provide, and use information. Letizia is just one example of a wide class of such advisors. In the future, people will not be content

482 CHAPTER 21 Interfaces That Give and Take Advice

with simply browsing and consuming information as in today's "surfing." Net-based information will be integrated in with other software applications that process information, use it for problem solving, transcode it, and redistribute it. All these processes can be supported by advisory agents and benefit from the ability to accept advice.

Many people currently look to the Net for advice by searching for information "content" to advise their activities, either for work or play. People also use the net to make connections with people who may be able to serve as advisors or to offer advice to others. Advice-based interfaces could be developed to assist this process.

21.4.2 Physically Based Interfaces

A primary function of personal information devices such as PDAs, wearable computers, and computers embedded in household and office devices is that they will serve as reminders to people in their daily activities. In this role they are acting as advice givers, and care must be taken in designing these interfaces so that the user perceives them as helpful rather than annoying. Many of these devices will be so small that they will be unable to provide traditional display and mouse interfaces and have to accept advice from the user primarily by speech recognition and automatic observation and sensing of the user's activities. Understanding interfaces that give and take advice will be essential in integrating these devices into our daily lives.

21.4.3 Speech, Natural Language, and Gesture Interfaces

A large part of making users feel that their computers are acting intelligently is to make the communication between people and computers more like the interaction between people. This involves not only direct commands given via speech recognition but also indirect advice such as that accepted by Mondrian. Clues to the user's desires can be given by tone of voice and gestures or facial expressions recognized by a vision system. These secondary inputs act to advise the primary interaction specified by the surface commands. Theories of discourse, speech acts, and multi-agent communication can inform the advice-taking process so as to make use of this implicit advice in the human-computer dialog. This can streamline the communication process, requiring less explicit communication from the user, and make the computer seem more responsive and personalized.

21.4.4 Advice and the Design of Visual Communication

An important component of visual design in interactive interfaces, publishing on the Web, and other contexts is the relation between visual and textual information. Often the visual information serves as advice to the reader in how to interpret textual information, or vice versa. Problem-solving expertise in visual design is often communicated

through the use of concrete visual examples that serve as advice about how to visually represent information (Lieberman 1995b, 1996).

21.4.5 Advice as a Tool for Helping People Learn

Seymour Papert (1980) has long advocated “teaching the computer” through programming as a path for people, especially children, to become better problem solvers themselves. With more intelligent learning and programming environments, this can be extended to “advising the computer.” Advising has the advantage that it is less dependent on the precision of instruction and the exact of order of sequencing of events—precisely those aspects of learning programming that are most troublesome to beginners.

21.5 Conclusion

While there have been many specific examples of computer applications acting as advisors and advisees, the central role of advice in human-computer interaction has not yet been fully appreciated. Artificial intelligence may contribute commonsense knowledge and learning algorithms, but traditional AI has been more focused on the idea of a computer as a standalone problem solver. Human interface research offers more sensitivity to user needs, but it currently seems stuck on the direct-manipulation paradigm, which needs to be transformed into an advisor-advisee relationship. It is this relationship that we need to build the foundation of the digital future.

Take my word for it.

References

- Fischer, G., and Nakakoji, K. (1993). Embedding critics in design environments. *The Knowledge Engineering Review Journal, Special Issue on Expert Critiquing*, 8, 4, 285–307.
- Horvitz, E. (1996). Principles of mixed-initiative user interfaces. *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'99)*, 159–166. New York: ACM Press.
- Lieberman, H. (1995a). A demonstrational interface for recording technical procedures by annotation of videotaped examples, *International Journal of Human-Computer Studies*, 3, 383–417, 1995.

484 CHAPTER 21 Interfaces That Give and Take Advice

- Lieberman, H. (1995b). The visual language of experts in graphic design. *IEEE Symposium on Visual Languages*. Darmstadt, Germany.
- Lieberman, H. (1996). Intelligent graphics: A new paradigm. *Communications of the ACM*, 8.
- Lieberman, H., and Maulsby, D. (1996). Instructible agents: Software that just keeps getting better. *IBM Systems Journal*, 35, 3/4.
- Lieberman, H. (1997). Autonomous interface agents. *CHI 97*. Atlanta.
- Lieberman, H., and Selker, T. (2000). Out of context: Computers systems that adapt to, and learn from, context. *IBM Systems Journal*.
- Lieberman, H. (Ed.). (2000). *Your wish is my command: giving users the power to instruct their software*. San Francisco: Morgan Kaufmann.
- McCarthy, J. (1959). Programs with common sense. In V. Lifschitz *Formalization of Common Sense, Papers by John McCarthy*. Ablex.
- Minsky, M. (1988). *The society of mind*. Simon and Schuster.
- Negroponte, N. (1995). *Being digital*. A. Knopf.
- Papert, S. (1980). *Mindstorms*. Basic Books, Inc.
- Rich, C., and Sidner, C.L. (1998). COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8, 3/4, 315-350.
- Stoehr, E., and Lieberman, H. (1995). Hearing aid: Adding verbal hints to a learning interface. ACM Multimedia Conference. San Francisco.
- Zilberstein, S. (1996). Using anytime algorithms in intelligent systems. *AI Magazine*, 17, 3, 73-83.