When Should We Use Natural Features?

Mark C. Torrance *
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
torrance@ai.mit.edu

Abstract
Some features of computer programs can make the programs easier to use by enabling users to interact with them in the familiar ways they interact with other people. Including these features is hard, however, and doing a good job at it is even harder. We present some of these features, and compare the pros and cons of including them in a computer program. We conclude with criteria designers can use when deciding which of these features to include.

1 Introduction
People are very good at carrying on conversations. We successfully establish a shared context, form correct expectations about the beliefs and motivations of others, recover from misunderstandings and interruptions, and get things accomplished during conversations with other people. It would be nice to be able to improve the quality of human-computer interaction by tapping into this natural conversational ability. With effective enough software, perhaps programs can even reach a level where they deserve to be called collaborators.

2 Natural features
Some features of computer programs can make the programs easier to use by enabling users to interact with them in familiar ways. We call these natural features, because they provide the computer with interactive capabilities similar to the natural interactions between people.

2.1 Discourse Model
In a sophisticated interaction, people are able to negotiate several points at once by remembering issues that they have pushed aside and returning to them later. People also make offers and counter-offers, and eventually accept offers, in negotiating requests they make of each other. Typically, computers do not incorporate a very sophisticated model of discourse. There may be multiple projects being worked on at once, but computers typically either make everything available all the time (like a Macintosh), or impose a strict hierarchy (like a voice-mail system). Computer systems which employ more sophisticated models of discourse will be easier to use in certain applications. See, for example, the paper by Charles Rich [3] elsewhere in this volume for an example of an application domain and the possible use of discourse in its interface.

2.2 Inference
One of the most important things people bring to a conversation with each other is the power to make inferences. Much of the meat of a human conversation is left unsaid, because it is an obvious enough inference for both parties that it doesn't need to be articulated.

A computer system which is more effective at drawing inferences will be a more competent conversational partner in a complicated negotiation, or one in which a large amount of information needs to be exchanged.

2.3 Natural language
Natural language is a comfortable way for people to interact with each other. While it often lacks precision, it has the advantage that it can express many non-visual ideas quickly. Formal languages may offer more precision, but natural language is already familiar to a wide variety of potential users; this is perhaps its most significant advantage.

3 Reasons to include natural features
We here consider several reasons to include natural features in a computer system. Some of the reasons including natural features can be valuable include improved productivity or efficiency, effectiveness in an educational role, and entertainment value.

3.1 Productivity and efficiency
Natural features can improve the quality and efficiency of human-computer interaction by providing people with a familiar way to interact with the system. This benefit is realized most strongly when users will interact with the system only occasionally or when there are a large number of new users of a system, because in these cases the overhead of learning an unfamiliar user interface is significant. When people will use a system a lot, it may be that an unnatural user interface is just fine, as after the system is
3.2 Educational effectiveness

Existing programs designed for educational use have been criticized most strongly for their inflexible and repetitive nature. Teaching programs that have a greater sense of the progress of their students, as well as incorporating a richer dialogue and a less restrictive set of ways of learning, can already be imagined and may soon appear. These natural features have the potential to facilitate more effective learning by making the computer into a more flexible and less repetitive teacher.

3.3 Entertainment value

Natural features can enhance the quality of interactive computer entertainment by helping the author to create believable characters. Joe Bates’ work on the OZ interactive fiction system [1] relies in large part on natural features to design characters with whom the audience can empathize. Through the use of a simple discourse model, in some cases a natural language interface, and the modelling of animal emotional states, the audience identifies with the characters, views their problems as genuine and feels involved in their solution.

4 Problems with including natural features

There are several problems with including natural features in computer systems. It is hard to design the software to implement these features at all, and it is very hard to get them correct, and when they fail they can actually worsen the quality of the interaction over not including them at all. In addition, some systems are so simple already that it is easier to use them without any added features.

4.1 Hard to design

Human interaction is complex. Building computer programs that emulate even small aspects of this interaction has always been a great challenge. Some of these features, such as discourse, seem to be better understood than others, such as natural language, but even so there is still lots of room for research to even decide where to begin. It is important to keep the goals of the user interface design in mind when deciding whether a particular feature would actually improve to the quality of the interaction.

4.2 Hard to get correct

As hard as it is to include some competence on one or more of the natural features listed above, it is even harder to do a good job. Natural language has been known for quite a while to be a very challenging problem in general, and it remains quite hard even when the domain of discourse is restricted. Discourse theory may have more to say about how aspects of discourse can be incorporated into user interfaces (again, see Rich [3]), but this area of application is also new.

In any case, the interaction with the system will fail, or break down, to the extent that the feature was not implemented as completely as was needed.

4.3 Failures slow down the interaction

Breakdowns within the interaction, in which the user or the computer needs to reevaluate an assumption or fall back to a different way of understanding, are natural. This kind of breakdown occurs in human conversations as well. The presence of an effective discourse component that expects, recognizes and responds to such breakdowns will render them no more harmful than they are in human conversation. Winograd and Flores, in Understanding Computers and Cognition [2], discuss the issue of breakdowns in great detail.

More serious breakdowns occur when the user’s model of the program is violated. This can happen when the discourse component lacks a feature the user expects, when the natural language component exhibits a serious deficiency or misunderstanding, or if the system otherwise acts in ways the user doesn’t expect of a participant in a complicated conversation. In this case, the user may have to fall back to a mode of interaction more similar to conventional computer use. When this happens, the incompetent natural features of the program have done more harm than good, for they required the user to learn two interfaces instead of one. Furthermore, they may be in the way when the user tries to bypass them and just manipulate the program in a conventional way.

These factors should be taken into consideration by a designer who contemplates adding natural features to a system.

4.4 Simple systems

Some systems are simple enough to use as they are that natural features wouldn’t improve them at all, or at least wouldn’t improve them nearly enough to justify the research and development expense of implementing them. If it ain’t broke, don’t fix it.

5 Conclusion

In some cases, human-computer interaction can be improved by adding features to the computer system which make it more like a human. This improvement comes from the fact that people are familiar with ways of interacting with other people, and they have developed lots of context, cliches and metaphors for that interaction. A computer which uses the same metaphors in its interface should be easier to learn to use than conventional computers.

There are some problems with adding these features. It’s hard to add them at all, it’s hard to get them right, they may actually make it worse, and some programs don’t need them at all. But in cases where they can be helpful, the payoff may be huge.
References

