

Images of Ourselves: the Electronic Little People

Abstract:

There is a controversy over the appearance of user agents: should they be made accessible through anthropomorphism, or not? Those who say yes believe it makes agents' action more understandable, and that this predictability helps users direct them more effectively. Those who disagree believe that the "human-like helper" metaphor distorts rather than clarifies. Some are uncomfortable with the idea that we may become too dependent on agents, and devolve into incompetence.

One of the only certain predictions that can be made about computers is that they will change. Soon. There is quantitative change in implementation: physical size, price, capacity. There is the qualitative change of new methods of use: JCL, interaction, GUIs, databases, 4GLs, hypertext, multimedia. And there is cognitive change in the nature of their use: how people understand and use computers. This paper deals with that third area, in which our interaction with computers is changing. Researchers and designers are changing the way they think about computer interfaces, and as a part of this change process, the nature and direction of change is subject to debate. Some debates are replayed in new guises: one such is the disagreement about the uses of metaphor, which, as interactive agent-based interfaces are developed, becomes an argument about the nature of agents: are they tools, or anthropomorphical constructs?

Interface modes

There are two interface modes currently in wide use: *command mode* interfaces are conversational in nature, with the computer initiating action in response to a command issued by the user at the keyboard, or a series of commands collected into a script. DOS and UNIX are command-mode interfaces. A *direct manipulation* interface (examples are Windows and the Macintosh OS) involves the user as the explicit invoker of all activity. There is a growing consensus that command mode (third-person interaction) is well suited for situations where the job is laborious or repetitive, and where the system can be trusted to do the job properly. Direct control is needed if the job is critical, novel, or poorly defined. [Norman, 1988] Direct control interfaces afford a greater sense of mastery to the user, allowing interaction with the *work* to dominate interaction with the system.

The use of metaphor

Direct manipulation interfaces are often organized around a central metaphor. Metaphors are everywhere; they have been used since ancient times. The Greeks included metaphor in their list of the aids to rhetoric. With exaggeration, repetition, and the use of associated symbols, metaphor increases comprehension and retention. [Marcus, 1991] When encountering a new concept, we naturally look for structures and patterns that will help us learn and understand it. Analogy, simile, and metaphor are the basic building blocks humans use.

Metaphor is an invisible web of terms and associations that underlies the way we think and speak about a concept. How well the metaphor fits depends on the amount of structure, familiarity, applicability, representability, suitability to audience, and extensibility. A metaphor is only useful if it gives a realistic expectation about what will happen. [Erickson, 1990] Surely something this universally pervasive is good?

Not everyone thinks so. Critics note that metaphors applied to computer interfaces are seldom consistent. Metaphors can limit our thinking, and worse still, sometimes we don't recognize that this is happening. Why, for example, did column-based language syntax persist well beyond the use of 80-column punched cards? For that matter, why did the punched cards take their shape? (FYI, from the pattern cards controlling weave patterns in jacquard looms.) We no longer recognize the metaphor when its two parts—the tenor (analogy) and the vehicle (the implementation)—become separated. Sorting the cards (the original limited use) was like controlling a loom. Using the cards to program the computer was like using the cards as mechanically recognizable data. Programs input from electronic editors had to be like punched cards. So how is a computer program like a bedspread?

The danger in metaphor is that by invisibly shaping our perceptions it becomes a limit to thought. [Swigart, 1990] A metaphor with poor fit is especially annoying to some. They focus not on the areas where the fit is good, but where gaps are especially apparent, and these lapses create cognitive dissonance. They hate clumsy metaphors which are like, only different. They are unwilling to think of the difference as "magic".

A help becomes a hindrance

Interface metaphors provide a conceptual scheme, and must be carefully constructed, because, if misapplied, they lead only to confusion. But it is in the nature of things that a metaphor, which at first helps us grasp a novel concept, becomes limiting. It is only as we become comfortable in the new paradigm that we begin to see its limits. Sooner or later you run into the limits of the metaphor. [Laurel, 1991] So now the nature of this debate becomes clear: proponents of metaphor see its usefulness in establishing a new paradigm; detractors focus on the limitations imposed on exploring and expanding the paradigm.

New developments in interfaces: agents

All the arguments about metaphor are being replayed in a new context of agent-based interfaces. These more complex interfaces are a response to the same three types of changes mentioned in the introduction: increases in computer capabilities at a reasonable price; increasing complexity in the methods and data available; and growth beyond the limitations of current interface paradigms. Having the capacity to allow it, increasing complexity will necessitate changes in the user interface, as users can't personally direct all the actions that will be required. The growing number of untrained users intensifies this. Expected developments for user interfaces in the 90s include speech, agents, 3D display, video, hyper- and multi-media. These are approaching human interaction: speech, gestures, images, shared knowledge, common assumptions. [Marcus, 1991]

Agents are processes that act as guide, coach, amanuensis. As opposed to tools which we use, they are things we manage which use tools on our behalf. An agent-based interface could be called indirect management. Agents will become the *electronic labor force*, the "little people" of my title: like the shoemaker's elves, agents will perform their functions to prevent us being overburdened by complexity or volume of data. They may wade through Usenet data-streams to select only postings of interest, or run through a stuffed e-mail bin to categorize, file, and possibly reply on our behalf. Agents will form the basis of intelligent interfaces. It's happening now in research labs, there are some custom implementations now at work, and we can expect off-the-shelf commercial versions soon.

A new context for the metaphor debate

With this new development for interfaces, the metaphor debate takes on a new aspect: are intelligent interfaces agents—that is, mediators between user and computer—or are they tools with intelligently organized direct manipulation options?[Chin, 1991] Various representations are seen as "more honest" than the rest, mainly because that's how the designer's understanding is based. If we look at anthropomorphization as one kind of metaphor, it becomes easier to see that initial implementations will be based on the familiar, and that experience with the new paradigm will cause us to discard the initial construct. This process—adopting the metaphor *and* discarding it after discovering its limit—is natural, spontaneous, and unstoppable.

The essential nature of an agent is that it does things on behalf of the user. The user has two main concerns: how does the agent know what to do? and can it be trusted to function well? Competence derives from the way the agent is managed: who tells it what to do, and how. Trust is a cognitive issue: do the agent's actions adhere to an expected pattern? Competence issues can be resolved by judicious application of technology; trust is purely dependent on the user's mental construct.

Implications for the developer

There are clear implications for the developer of agent-based systems. The importance of clear-headed thinking cannot be over-stressed. Good design begins with an analysis of what functions will be enabled: what the user will DO. [Norman, 1988] The action can be plotted like a story: design the kinds of incidents that will occur, and in what order. Metaphors must be carefully chosen to provide a mental construct that will assist novice users while not limiting the competent. Agents should not be overly personified: sketch characters, not people. Select and represent only those traits appropriate to a particular set of actions or situations. Too much noise makes it harder to predict their action. [Laurel, 1991] It is wise to take into account cultural, professional, and personal culture: the skill set of the system developer will need to expand to include drama, psychology, story development, pacing, and characterization.

Three examples of agents in use today

Let's look at three examples of agents. They have varying degrees of explicit characterization, gain the knowledge of what they do in differing ways, and differ in the nature and expression of their feedback to the user.

Magnet, from No Hands Software, is advertised as the first intelligent agent for the Macintosh. This application is cast as a tool rather than an anthropomorphical entity. It uses a metaphor of attraction (hence the name.) A magnet operates on files, and at pre-defined times or events, pulls them to the folder location

where the magnet resides. Magnets are defined with file search/operation criteria, finder action, and trigger events or times. They can also be invoked manually. They provide the user with an activity log.

The OW Holmes system currently in use at a large Northwest law firm is a more robust expression of agent technology. Here there are a series of agents, referred to as the *electronic labor force*, each of which has been given a name and specific capabilities. They provide information, facilitate communication, and generate actions in a generalized workflow system. The elves are explicitly scripted by users using natural language: check the court docket for this matter and put the trial date on my calendar. Once created, elves perform their allotted tasks without supervision, notifying their human managers of anything requiring attention or intervention. In this system, anthropomorphization was deliberately chosen to introduce the legal and clerical staff to a paradigm of electronic helpers *under their direction*. These elves aren't magical; they're hard-working employees who are trained and ready for work. The semiotic shortcut of characterization allowed quick comprehension and predictability, making the users confident in using their agents.

At MIT's Media Lab, Pattie Maes and Robyn Kozierok have built a personal assistant to operate in a collaborative role. They have developed a generic architecture for learning interface agents [Maes, 1993] which include a "mail clerk" and a "calendar manager". These agents learn their jobs by observing the user, monitoring activities over a period of time, finding recurrent patterns and automating them. A caricature face in the corner of the screen provides feedback to the user. This explicit characterization is provided to allow quick comprehension of what the agent is up to: we humans observe and process facial expressions very well.

Summary

Quantitative, qualitative, and cognitive changes in the computer field are driving innovation. Interface agents are one response to these changes, and as with all new technologies, provide subject for argument. The debate over anthropomorphization of agents is a new expression of an old debate: the role of metaphor in interfaces. The nature of human perception and thought patterns leads us to approach new things in comparison to what we already know; metaphor is inevitable. It is important, though, to choose metaphor quite carefully, so that we do not unintentionally limit the usefulness by unnecessary elaboration or bounds.

Bibliography

Artificial Stupidity. The Economist v324 Aug 1 '92 p14.

Chin, D. Intelligent Interfaces as Agents in *Intelligent User Interfaces*, J Sullivan and S Tyler (eds.) ACM Press, NY NY, 1991.

Don, A. Anthropomorphism: From Eliza to Terminator 2, panel description, ACM proceedings of Chi'92, ACM Press.

Erickson, T. Working with Interface Metaphors in *The Art of Human-Computer Interface Design*, B Laurel (ed) Addison-Wesley, 1990.

Kay, A. User Interface: a Personal View in *The Art of Human-Computer Interface Design*, B Laurel (ed) Addison-Wesley, 1990.

Kelleher, J. Alan Kay: Prophet of Intimate Computing (interview) in Computerworld Special 25th Anniversary Edition: 25 People Who Changed the World, Computerworld v26 i1 Jun 92.

Laurel, B. Interface Agents: Metaphors with Character in *The Art of Human-Computer Interface Design*, B Laurel (ed) Addison-Wesley, 1990.

Laurel, B. *Computers as Theatre*, Addison-Wesley, 1991.

Levy, S. Let Your Agents Do the Walking, MacWorld v10 i5 May 93 pp37-42.

MacLean, A; Bellotti, V; Young, R; Moran, T. Reaching through Analogy: A Design Rationale Perspective on Roles of Analogy, in ACM proceedings of Chi'91, ACM Press.

Maes, P; Kozierok, R. Learning Interface Agents, to be published in the proceedings of the National Conference on Artificial Intelligence, July 1993.

Marcus, A; van Dam, A. User Interface Developments for the Nineties, in Computer, v24 i9, pp49-57, Sep 91. ISSN: 0018-9162.

Myers, B; Rosson, M. Survey on User Interface Programming, ACM Proceedings of Chi'92, ACM Press.

Nelson, T. The Right Way to Think About Software Design in *The Art of Human-Computer Interface Design*, B Laurel (ed) Addison-Wesley, 1990.

Norman, D. *The Psychology of Everyday Things*, Basic Books, 1988.

Santo, B. Call Your Agent. Electronic Engineering Times, Mar 8 93 pp46-47.

Swigart, R. A Writer's Desktop in *The Art of Human-Computer Interface Design*, B Laurel (ed) Addison-Wesley, 1990.