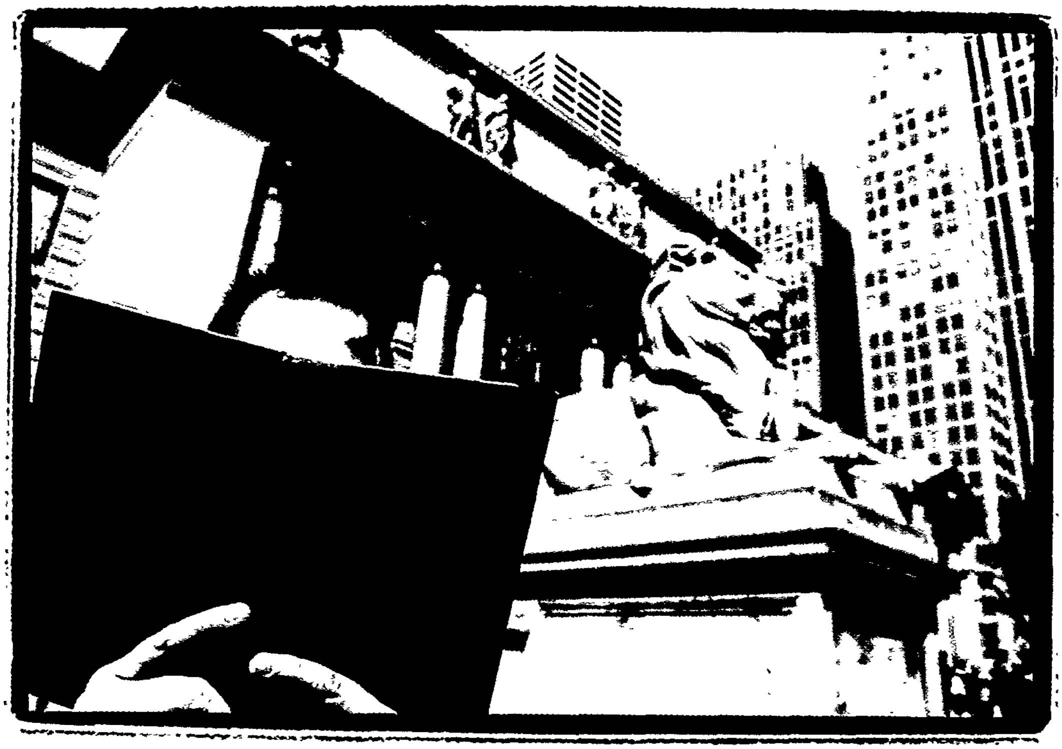
Designing a Language for Interactions *Terry Winograd*

Being a university teacher in the area of human-computer interaction can be both exciting and frustrating. There is a problem in trying to gather together appropriate materials in a field not yet mature enough to have produced a tradition of "classics."



Robert Vizzini

In an architecture course, for example, budding architects are introduced to the great genres of the past: the Greek temple, the Gothic cathedral, the modern or post-modern skyscraper. They learn what is important and possible by understanding how each of the different traditions dealt with the same problems in its own way. For after all, people and their needs remain much the same from millennium to millennium.

We aren't there yet in human-computer interaction. But we are starting to get there and the raw material is piling up. As pointed out by several authors in the first issue of interactions, we are in the process of moving beyond the interface genre that gave impetus to the field of HCI–the WIMP GUI established by the Xerox STAR and carried forth in Macintosh, Motif, Windows, and so on. Although this interface in all of its look-alike and work-alike forms still dominates the market, we are starting to see the appearance of "next generation" concepts. Once there is more than one interface style, it

Terry Winograd is Professor of Computer Science at Stanford University. He is a Founding Member and past President of Computer Professionals for Social Responsibility. Winograd is also a Consultant at Interval Research and Founder of Action Technologies, a developer of workflow software. At Stanford he directs the Project on People, Computers, and Design and is developing a program on Human-Computer Design. email: winograd@cs.stanford.edu

becomes easier to see it clearly as a specific genre rather than as the essence of what it means to be a human-computer interface.

Of course, it is myopic to think that this is brand new–other interfaces have been sitting under our noses all along, just not with the same degree of visibility within the academic and research community. The most prominent, of course, is video games. For someone looking at either the bottom line sales figures or at the breadth of psychological phenomena involved, it should seem puzzling that games and the techniques used to design and build them play such a tiny role in conferences, book and courses on human-computer interaction, when compared to spreadsheets, menu selection and scrollbars.

The HCI world came into its own by showing that it could be relevant to the development of successful "productivity software," but the market emphasis is shifting. Computers are moving from the phase where product emphasis is on productivity and business, into a consumer-centered phase where consumer choices are driven by feelings of engagement and appeal, both aesthetic and emotional.

Another new direction frequently pointed to by the media hubub is multimedia. We are beginning to see products that mix print, audio and visual media in non-trivial ways. Some are for entertainment, such as Voyager books early CD-ROM, *From Alice to Ocean*, and others for education as with *Making it Macintosh*, featured in the first issue of *interactions*. An additional twist is provided by the networked multimedia we find in browsers to the World Wide Web, such as Mosaic and Viola, which will make it possible for everyone to be a desktop multimedia producer and distributor–if they understand how to design it well.

Another indicator of change is the increasing use of "augmented reality" interfaces, in which the interface and the world are interwoven, as in Xerox Europarc's digital desk, or Myron Krueger's use of the body itself as an input device. These situate the "virtual realities" of the computer back into our world of work and play, and will in the long run be much more pervasive than the immersive goggle-and-gloves VR environments that have captivated more of the public imagination.

In each of these areas it is easy to bemoan the primitive nature of many of the current designs, the scarcity of well-thought-out standards or interaction styles, and the lack of careful studies and analyses. It is tempting to stick with teaching from the old books and articles, which carefully explain the detailed principles of windows and menus, the mechanisms of selection and command execution, and the mastery of a world of widgets. But the great majority of the students we are educating today will not be working in that world. They will be working on interaction designs that move away from the desktop, even away from the extended desktops of PDA's and wall-sized whiteboards.

Our challenge, then is to find ways to give them the underlying concepts and skills that will not be bound either to today's GUI or to the specifics of its successors. It is always necessary for the designer of software and its interactions to have a mastery of today's current widgets. But it is never sufficient. If we teach a compendium of techniques and stop with that, we have shortchanged our students. But what, then, do we teach as the timeless core of the discipline?

My own perspective is that we need to develop a language of software interaction–a way of framing problems and making distinctions that can orient the designer. Every developed profession has such a language. In some cases, such as medicine and law, much of the educational process is devoted to them. Even in early stages of development of a field, a shared language begins to emerge. A quick glance through the first issue of interactions shows a variety of terms that have taken on meaning for the readers of that magazine as tools for analyzing and doing interface design: "invisibility," "way-finding," "layering," "dialogue design," "metaphor," "narrative," "speech act," "ontology," and more.

Although all of these words have meanings in everyday language, they are used in a more specialized way when talking about interaction design. However, in being specialized they are not bound to specific interface styles or technologies–they point to universal aspects of designing any artifact with which people interact, whether it is a GUI or a VR theater experience. They also do not correspond to straightforward formalizations, as do terms such as "force," "mass," and "acceleration" in physics. When the skilled interaction designer uses a specialized language, he or she is not applying a formula but is using distinctions to help orient the user to certain aspects of what is needed and to open up a domain of possibilities for design.

If we want to develop this kind of competence in our students, we will not succeed by giving them definitions to read or by lecturing about meanings. Operational language is learned through what Donald Schön calls "reflection in action," engaging the student in the practice of design, and stopping from time to time to reflect, taking the raw material of the student's experience as the source of examples and opportunities to teach the terms of the trade. At the recent NSF workshop on new directions in teaching HCI, it was clear that HCI education is moving in this direction. In sessions on both the undergraduate and graduate curriculum, the focus was on courses that combined theory and practice in an integral way, using student projects as the touchstone for teaching.

This past year I have been working with David Liddle (one of the originators of the Xerox STAR and now head of Interval Research) to design an introductory course on Human-Computer Interaction Design. We wanted to find ways other than hands-on project work to bring students to an understanding of principles in practice. Project courses with adequate opportunity for reflection are highly labor-intensive as a teaching modality, and they are often limited by the scope of projects that are realistic for students to take on in a single course. We are attempting to apply a method that is often used in business and law schools, centered around case studies. Students read about and experience software examples, and the terms and concepts are learned in the discussion of concrete design experience. In many cases we have also been able to bring in people who were involved in the design of the products and engage them in the discussion as well. Examples range from the classics (such as Lotus 1-2-3) to flight simulators and World-Wide-Web interfaces.

Reflecting on the first prototype version of this course as it nears its end, it looks promising. We have found cases that cover a variety of interface styles and that can be used to show interaction ideas in practice. Discussion has been lively. There is an emerging body of concepts and distinctions that can be used to transcend the specifics of any interface and reveal the space of possibilities in which it represents one point. This "language of interactions" is still far from complete and far from consensus, but it is moving along with good speed. It will take more exploration, and more good writing and analysis before we have a solid body of material on which to depend. interactions magazine can play a great role in fostering the discussion and is already making strides in that direction. I look forward to being here for the fun.

References:

[1] Schön, Donald, *Educating the Reflective Practitioner*, The Jossey-Bass Higher Education Series, San Francisco, 1987.

[2] National Science Foundation Workshop on New Directions HCI Education and Research, Washington D.C., February 1994 (report forthcoming)