A METHOD FOR EVALUATING SOFTWARE COMMUNICABILITY

Clarisse S. de Souza **Raquel O. Prates** **Simone D. J. Barbosa**

Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente, 225
Rio de Janeiro, RJ – Brazil – 22453-900

{clarisse, raquel, sim}@inf.puc-rio.br

PUC-Rio Inf MCC11/99 Maio'99

ABSTRACT Computer tools are rational artifacts determined by the cultural background of designers and users. Communicating design assumptions, choices, and goals is a complex task, subject to constant misunderstandings that lead to a wide spectrum of breakdowns in human-computer interaction. This paper presents a method to help HCI practitioners evaluate the *communicability* of interactive systems, not only in terms of the effectiveness of their direct messages to users but also of the afforded conversational paths that should allow breakdowns to be circumvented. The whole proposal is tied to a semiotic background within which systems are like "discourse deputies" of their designers.

KEYWORDS Usability testing, semiotically-based HCI design, communicative breakdowns

A METHOD FOR EVALUATING SOFTWARE COMMUNICABILITY

Clarisse S. de Souza **Raquel O. Prates** **Simone D. J. Barbosa**

Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente, 225
Rio de Janeiro, RJ – Brazil – 22453-900

{clarisse, raquel, sim}@inf.puc-rio.br

ABSTRACT Computer tools are rational artifacts determined by the cultural background of designers and users. Communicating design assumptions, choices, and goals is a complex task, subject to constant misunderstandings that lead to a wide spectrum of breakdowns in human-computer interaction. This paper presents a method to help HCI practitioners evaluate the *communicability* of interactive systems, not only in terms of the effectiveness of their direct messages to users but also of the afforded conversational paths that should allow breakdowns to be circumvented. The whole proposal is tied to a semiotic background within which systems are like "discourse deputies" of their designers.

KEYWORDS Usability testing, semiotically-based HCI design, communicative breakdowns

INTRODUCTION

The ultimate goal of HCI design is to provide users with *usable* computer systems (Hartson 1998). An increasing number of methods, techniques, and frameworks emerge from research and practice, but none can fully guarantee that software produced using its rules or principles, models or tools, will achieve the highest scores of *usability*.

At the heart of the complex task faced by HCI designers is the fact that computer tools are cultural artifacts (Brown and Duguid 1992). Users must be able to interpret the various codes used by designers to convey the values and possibilities embedded in software. As Brown and Duguid remark, design "can, like all communications, attempt to cue relevant sets of codes and to dismiss others through its purposeful use of affordances" (op.cit. p.170).

Emphasizing the communicative nature of design, especially with respect to computer artifacts, is the keynote of semiotically based HCI research (e.g. Nadin'88, Andersen et al. '93, de Souza'93, Mullet & Sano'95). However, no methods for evaluating software *usability* from a semiotic perspective have yet been proposed. As a result, although many acknowledge the theoretical consistency and the explanatory power of work in the field of computer semiotics or semiotic engineering, for example, its practical benefits for professional HCI designers, and the corresponding additional affordances offered to users, have remained implicit and embedded in theoretical approaches.

In this paper, we make one step in the direction of bringing semiotically based research closer to the practice of HCI design and propose a method for evaluating the *communicability* of design rationale. This feature is important because, as a cultural artifact (through which designers communicate their own understanding of who the users are, what they need and expect, what they like and prefer, what they know and don't know), interactive software entails a conversation between users and a fixed representation of its designers' minds and ability to express and explain themselves.

The gist of the proposed method is to map a set of questions, interjections and statements with which users respond to interactive turns onto classes of known usability problems. It produces a picture of potential conversations about breakdowns that not only signals the occurrence of HCI problems of a given application, but can also be used to instruct the (re) design of this and other applications.

In section 2, we describe the empirical experiment we used to establish the relevant conversational categories of utterances for our method. In section 3, we explain how the method can be applied to analyze HCI design, and add our views on how it could also be used to support the design process itself. And finally, in section 4, we discuss our proposal with respect to current *usability* evaluation methods and approaches to HCI design.

A COMMUNICABILITY EVALUATION METHOD

In our method systems are viewed as a message (about *anticipated* users' needs and expectations) from designers to users. This message can send and receive other messages (with which users try to meet their *actual* needs and expectations). In this perspective, a system is a kind of "discourse deputy" for the designer in that it encompasses the set of all (and only) the conversational turns and themes she can predict at design time. In order to support the ongoing conversation especially when breakdowns occur, the designer must then afford questions, interjections, and statements that users can utter to express their minds.

Whereas most design and usability evaluation approaches concentrate on <u>users and qualities of solutions</u> afforded by designers (Nielsen 1993, Preece et al. 1994, Shneiderman 1998), we propose a shift to focus on the <u>conversation about solutions</u>. This shift should provide complementary insights about usability problems since one of the difficulties with current guideline-based methods and techniques is that guidelines often point at conflicting directions, and designers must introduce arbitrary choices that may give rise to considerable interactive problems (Baecker et al. 1995, Preece et al. 1994). A communicability evaluation method, however, grants the emergence of conflicts in design and supports users as they try to make sense of interaction with computer systems.

In order to enhance the *communicability* of design, all shifts and paths of conversation about the various topics included in the system's domain and about interaction itself must be carefully decided by designers. So, they should start by looking for expressions that translate users' doubts, frustrations, and confusion, and then plan for the appropriate responses to these.

Literature about explanatory dialog provides us with a set of questions users typically pose when trying to resolve problem situations (e.g. Moore 1995, Cawsey 1993). We may use them not only as a guide for the design of online HELP functions (Lewis and Norman 1986, Sellen and Nicol, 1990), but also as a conversational path to circumvent breakdowns. However, such typical questions are not likely to be equally relevant to the latter purpose. For instance, one typical question is what's this?. It expresses the kind of breakdown a user faces when he doesn't understand an interactive token afforded by the system. This breakdown is not at the same level of relevance as that associated to the question What if I do this?, which more often than not expresses an exploratory attitude of the user, and not necessarily a breakdown. In order to identify which are the most relevant ones for our purposes, we have conducted an experiment with 10 volunteers.

We have arbitrarily selected two HTML tag editors – Arachnophilia (Arachnophilia 1996-1998) and SpiderPad (SpiderPad 1997) – with which participants were requested to interact. They had varying levels of expertise in HTML and none had had any previous contact with either editor.

Participants were split up into two groups: one that worked first with Arachnophilia, and another that worked first with SpiderPad[1]. After 5 minutes of exploration with each editor, they were asked to: (1) create a nested list of items, with particular bullet types; (2) change the background color of an existing HTML page; (3a) create a 2x2 table with border and title (caption); and (3b) edit the table by merging the two cells in the first row. They were allowed a maximum of 4 minutes to perform each task.

All interactions were logged with Lotus® ScreenCam™ and videotaped. Testers took notes as they sat behind the participants and observed what they did. After an extensive examination of the records, the following set of utterances stood out as necessary and sufficient to account for all the observed interactive breakdowns. They have been treated as conversational categories[2], in that they can distinguish different sets of breakdown situations and user attitudes.

- **Where is? (What now?)**
  The user seems to be searching for a specific function. So, he sequentially (worse case) or thematically (better case) browses menus and/or toolbars for that function. This category includes a special case we have called What now?, applicable when a user does not know what to do and thus searches for the next step (as opposed to a function).

- **What's this? (Object or action?)**
  The user seems to be exploring the possibilities of interaction to gain more (or some)

[1] The order in which participants interacted with editors did not play a relevant role in the experiment.

understanding of what a specific function achieves. This category also includes cases in which users are confused about widgets being associated to objects instead of actions and vice-versa (Object or action?).

- **Oops! (Where am I?)**
  This category encompasses cases of unsuccessful actions, which the user immediately corrects (typically by pressing UNDO or by attempting to restore some previous state of interaction otherwise). Oops includes a special category Where am I? in which the user performs some action that is appropriate in another context, but not in the current one.
- **Why doesn't it? (What happened?)**
  This category is related to cases in which users expect some sort of outcome but do not achieve it. One subsequent scenario is that they then insist upon the same path, as if they are so sure that that function should do what they expected that they cannot accept the outcome. They carefully step through the path again and again to check that they are not doing something wrong. The other scenario (What happened?) is when they do not get a feedback from the system and are apparently unable to assign meaning to the function's outcome (halt for a moment).
- **Looks fine to me…**
  The user achieves some result he believes is the expected one. At times he misinterprets feedback from the editor(s) and does not realize that the result is not the expected one.
- **I can't do it.**
  The user is unable to achieve the proposed goal, either because he does not know how to or because he doesn't have enough time or will to do it.
- **I can do otherwise. (No, thanks)**
  The user defers some affordance provided by the editor's interface and finds another way around the problem. If this happens after a series of unsuccessful attempts, it is a case of I can do otherwise. If it happens after successful attempts but the user switches to other strategy, then it is a case of No thanks.

As can be seen from the above, some of the resulting categories are not questions likely to occur in explanatory dialogs but rather interjections (Oops!) or statements (I can do otherwise, No thanks, I can't do it, Looks fine to me). Furthermore, I can't do it and Looks fine to me are final statements that do not regard interaction *per se* but the completion of the task.

Our next step was to associate the seven categories to known classes of interaction and usability problems. Results are shown in Table I.

It should be noted that some navigation problems may be caused by meaning-assignment difficulties. For instance, some participants have repeatedly visited an option of the TOOLS menu named *LIST* in Arachnophilia (see Figure 1) because they believed it would lead them to a list creation dialog. However, Arachnophilia's designer had something else in mind (namely the display of lists of commands, macros, shortcuts and the like). Curiously enough, these participants overlooked the *LIST WIZARD* option in that same menu, and accessed it through another menu structure (as a submenu of COMMANDS).

	Navigation	Meaning Assignment	Declination of Affordance	Task

Why doesn't it? (What happened?)		*		
Where is? (What now?)	*			
What's this? (Object or action?)		*		
Oops! (Where am I?)	*	*		
Looks fine to me…				*
I can't do it.				*
I can do otherwise. (No, thanks)			*	

Table I: Mapping conversational categories onto interactive and usability problems

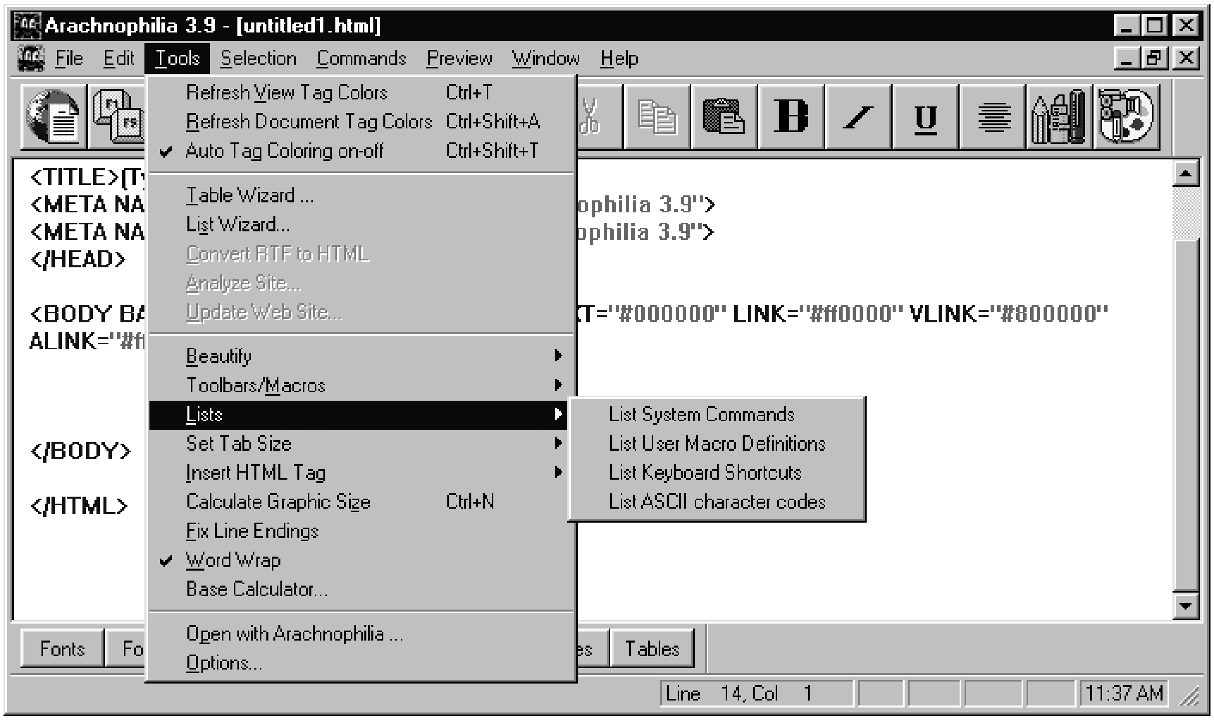

Figure 1: Arachnophilia's interface ambiguities about *lists*

Users may also decline perceived affordances because of inconvenient navigation structures (e.g. lack of shortcuts, deep nesting in menu structures, and the like). In SpiderPad, for instance, some users found that list items had to be inserted one by one in an existing list structure, by clicking on the <li> option of a command bar menu (see Figure 2). The cost/benefit of interaction was not perceived as beneficial by these users, who then began to type in the tag directly in the edited document. Also, occasionally users may totally miss affordances because they fail to assign the correct meanings to things (e.g. recognizing the meaning of icons in SpiderPad's vertical side bar in Figure 2).

Therefore, categories can always tell the more immediate problems encountered by the user. The deeper causes of such problems might, however, be derivable from the examination of threads of conversation (i.e. longer spans of dialog involving a sequence of categories).

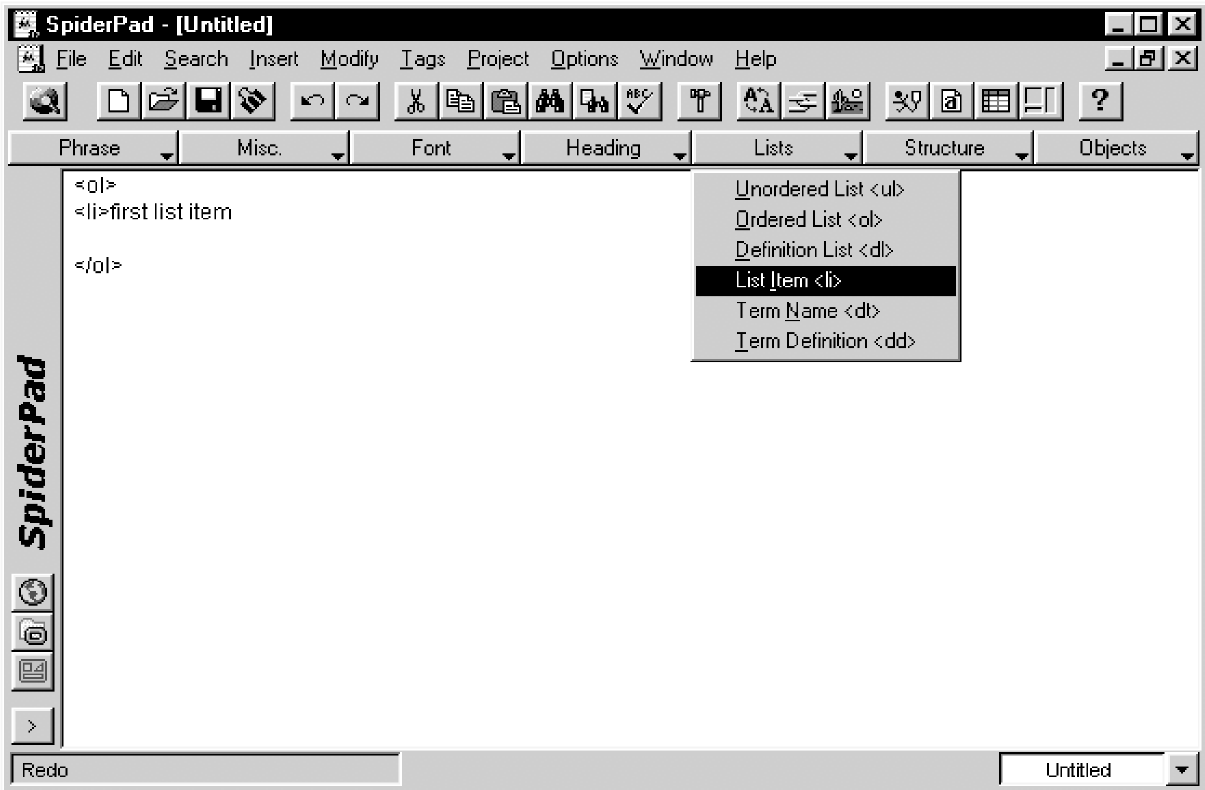

Figure 2: SpiderPad's affordance for the insertion of the <li> tag in HTML

	Arachnophilia		SpiderPad		Total
(a) Where is? What now?	65	48.87%	63	45.32%	128
(b) Oops! Where am I?	26	19.55%	23	16.55%	49
(c) I can't do it.	13	9.77%	11	7.91%	24
(d) I can do otherwise. No, thanks	13	9.77%	9	6.47%	22
(e) Looks fine to me…	10	7.52%	11	7.91%	21
(f) Why doesn't it? What happened?	2	1.5%	13	9.35%	15
(g) What's this? Object / action?	4	3.01%	9	6.47%	13
Totals	133	100%	139	100%	272

Table II: Comparing Arachnophilia and SpiderPad

As the last step in our experiment, we have turned to the kinds of insights we might have about Arachnophilia and SpiderPad by looking at the occurrences of such conversational categories throughout the tests. In Table II we can see a summary of each editor's performance. Integer numbers represent the totals and subtotals per editor of occurrences in each category. Percentages are calculated relative to the total of all occurrences for a given editor.

We can conclude from Table II that Arachnophilia and SpiderPad presented considerable problems of navigation to their users. Although both also presented difficulties in terms of assigning meanings to interface symbols, it is noteworthy that the subtotal of meaning-related categories in SpiderPad ((b)+(f)+(g)=45) is greater than in Arachnophilia ((b)+(f)+(g)=32).

If we group interaction-centered categories ((a),(b),(f),(g)), we see that Ararachnophilia performs slightly better than SpiderPad (97 *vs.* 108). On the other hand, by grouping usability-centered ((c),(d),(e)) categories, we see that SpiderPad performs slightly better than Arachnophilia (31 *vs.* 36). Interestingly, this observation is in line with the kind of discourse we find in each editor's help contents. Arachnophilia has a much more conversational and verbose help than SpiderPad. Arachnophilia's designer directly addresses the users and explicitly stands as the first person in discourse (we even know the designer's name), as evidenced by phrases like *"I can't know what your background is or how much you know about computers, so you may choose…"* (help content for topic <u>How to make your own page</u>). SpiderPad's help contents are totally impersonal texts (there is no mention to the designer's name), written in terse descriptive style. Moreover, in Arachnophilia help is organized in a tutorial way, whereas in SpiderPad it is organized in a functional way. This reinforces our interpretation that SpiderPad is metaphorically a toolbox for HTML, whereas Arachnophilia has something of it's designer's deputy in front of the user.

APPLYING THE METHOD

Our proposed method applies basically to single-user interfaces. Multi-user interfaces would probably require other utterances related to interacting with other users, such as <u>Who are you?</u>, <u>What are you doing?,</u> and <u>Where are you?</u>. The same is true of artificial intelligence applications, for which utterances related to the system's cognitive abilities are likely to occur (e.g. <u>Do you know this?</u>, <u>Can you learn this?</u>).

Thus, in single-user applications, the first step to applying the method is to ask users to perform some pre-defined task(s) using the application to be analyzed. The users' actions must be recorded using software that is able to capture mouse-pointer movements, screen events, etc. (e.g. Lotus® ScreenCam™). Testers should also take notes and videotape the experiment.

Next, testers interpret the records of the experiment and associate spans of interactions to the hypothetical user utterances, identifying potential conversation breakdowns. These associations can be perceived as the construction of users' hypothetical verbal protocols. Videotapes and notes on users' spontaneous utterances can be used to confirm or support hypotheses. The advantage of this strategy is that it can minimize the known problems of verbal protocol techniques (namely, that of straining the users with double activities – problem-solving and narrative (Preece et al. 1994)).

Having identified the breakdowns in interaction and the corresponding classes of utterances users are likely to have made, testers proceed to relate these to interaction and usability problems (navigation, meaning assignment, declination of affordances, achievement of tasks). The result is a map of critical interactive points in the context of the experiment, along with the perceived causes of breakdowns.

A designer can make further use of this method's components by allowing users to actually say the proposed utterances during interaction (e.g. by clicking on option of a pop-up dialog that is quickly accessed by the right button of the mouse), and providing the appropriate answers to the anticipated questions, interjections or statements. It would then be possible to estimate the difficulties faced by users and the quality of remedial actions designed to help them. This kind of *just-in-time* help would allow users to get back on track and keep up the productive conversation.

Traces of the proposed conversational turns can be found in popular applications such as MS Word 7.0, which affords a function named *WHAT'S THIS?* In its help menu (see Figure 3). This leads us to believe that our method can be easily incorporated by designers in their daily practice.

Figure 3: Option in Word 7.0 Help Menu

DISCUSSION

Some of the questions of our set of utterances are reinforced by the questions identified by Sellen and Nicol (1990) as being asked by users when they need help. These authors use these questions to guide their design of help, whereas we use them to evaluate the conversation breakdowns of an application, and thus the application. Nonetheless, in cases when the utterances are used to circumvent potential breakdowns and provide just-in-time help, they allow the users to articulate their vocabulary when in need of help. In this case, the users can say more than just help, they can also express what kind of help they need.

Our communication-based method serves both as a platform for inspecting design as well as for evaluating usability. Regarding inspection tasks, our categories are intuitive enough that they can be understood by non-experts in HCI. Thus, not only user interface professionals, but also software engineers and programmers may interact with an application having our categories at hand. Throughout interaction they may then tag utterances to critical points where they feel breakdowns are occurring. This technique is very low-cost and may provide designers with valuable insights about their work.

Regarding usability evaluation tasks, we have reported in this paper how to carry out tests based on communicative categories. Although tests have been applied to finished products, it is easy to imagine how the proposed utterances might be used in formative evaluation. From the very beginning of the design cycle, users could be offered these utterances (at a mouseclick, for instance) and resort to them whenever they felt appropriate. This could be used to tag and log conversational breakdown utterances, as well as to test the anticipated responses provided by designers.

Whether designers have made formative evaluation tests or only anticipated potential problems, they will be able to try and prevent breakdowns. However, even if they cannot be totally avoided (due to conflicting design choices, for instance), designers will still be able to circumvent breakdowns by allowing users to engage in conversation about them. This strategy has been mentioned earlier in user-centered approaches (Lewis and Norman 1986), but with an emphasis on supporting the design of explanation and help systems (i.e. on the system's discourse, and not the user's).

Since the proposed method is based on abstracted designer-user communication, it has the additional advantage that its core is independent of interface style and application domain. As mentioned in section 3 regarding multi-user and AI-based applications, novel interaction styles or fields of computation are likely to require only extensions to the proposed set of utterances.

In truth, in a semiotic perspective system designers project interactive TYPES, which users manipulate to produce interactive TOKENS representing solution states to their problems. We learn from Schön (Schön and Bennet 1996) that design results from a (metaphorical) *conversation with materials*. And, the production of interactive tokens is actually a design process in itself, during which users *converse with the available interactive types* and generate specific combinations and sequences of tokens which are expected to mean what the users mean. So, in this paper we step back from Schön's *metaphor* and essentially propose that users be afforded the *actual* expression of conversational turns with the designer's "discourse deputy" about the types of interactions that may occur.

In order to validate the communicability evaluation method proposed in this paper, a number of steps are already called for. First, we will carry out experiments with other users, applications, and designers. Second, we will analyze the insights gained from inspecting sequences of utterances and the conversation flow around breakdowns. Third, we intend to apply this method in formative evaluation and verify if its contributions for design are as beneficial as expected. Finally, we hope to extend this research to multi-user applications, for which we have already proposed a model that can support explanations about design rationale (Prates and de Souza 1998).

ACKNOWLEDGEMENT

The authors would like to thank CNPq for supporting them, with different types of grants, during the development of this research.

REFERENCES

Andersen P. B., Holmqvist, B. and Jensen, J. F. (1993). The Computer as Medium. Cambridge University Press.

Arachnophilia 3.9 (Build 5142). Paul Lutus. 1996-1998.

Baecker, R. M., Grudin, J., Buxton, W. A. S.,and Greenberg, S (1995). “Design and Evaluation”. Readings in Human-Computer Interaction: Toward the Year 2000. 73-91.

Brown, J.S. and Duguid, P. (1992). “Enacting Design for the Workplace”. In P. S. Adler and T. A. Winograd (Eds.) Usability-Turning Technologies into Tools. Oxford University Press. 164-197

Cawsey, A. (1993) Explanation and Interaction: The Computer Generation of Explanatory Dialogues. MIT Press.

de Souza, C. S. (1993). “The Semiotic Engineering of User Interface Languages”. *International Journal of Man-Machine Studies*, **39**, 753-773.

Hartson, H. R. (1998). “Human-computer interaction: Interdisciplinary roots and trends.” *The Journal of System and Software,* **43**, 103-118

Lewis, C. and Norman, D. (1986). “Design for Error”. In Norman, D. A. and Draper, S. W. (Eds.) User-Centered System Design. Lawrence Erlbaum. 411-432

Moore, J.D. (1995) Participating in Explanatory Dialogues. MIT Press.

Mullet, K. and Sano, D. (1995). Designing Visual Interfaces: Communication Oriented Techniques, Sunsoft Press.

Nadin, M. (1988). “Interface Design: A Semiotic Paradigm”. *Semiotica*, **69**(3/4), 269-302.

Nielsen, J (1993). Usability Engineering. Academic Press.

Prates, R. and de Souza, C. S. (1998). “Towards a semiotic environment for supporting the development of multi-user interfaces”. In Proceedings of Fourth International Workshop on Groupware (CRIWG’98). Brazil, 53-67.

Preece, J., Rogers, Y., Sharp. H., Benyon, D., Holland, S. and Carey, T. (1994). Human-Computer Interaction. Addison-Wesley.

Schön, D. and Bennett, J. (1996) “Reflective Conversation with Materials”. In T.A. Winograd (Ed.) Bringing Design to Software. ACM Press. 171-184.

Sellen, A. and Nicol, A. (1990). “Building User-entered On-line Help”. In B.Laurel (Ed.) The Art of Human-Computer Interface Design. Addison-Wesley.

Shneiderman, B. (1998) Designing the User Interface. Addison-Wesley.

SpiderPad Version 1.5.3. Six-Legged Software. 1996-1997.