Context, User Models and Interface Design

Sarah A. Douglas

CIS-TR-88-10
July 12, 1988

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
UNIVERSITY OF OREGON

Context, User Models and Interface Design[1]

Sarah A. Douglas

Department of Computer and Information Science
University of Oregon
Eugene, OR 97403
(503) 686-4408

1. INTRODUCTION

Most of the existing analytical descriptions of users characterize their performance as a function of the cognitive representation of the command sequences of the computer-based task (e.g. Anderson, Farrell, & Sauers, 1984; Card, Moran & Newell, 1983; Polson & Kieras, 1985; Norman, 1986). This is represented as goal-oriented schemata: procedures, plans, or production rules. Thus, the interface designer need only lay out the command sequences adequate to achieve a set of core tasks to make predictions about user behavior. These models are by and large restricted to descriptions of error-free, skilled (expert) performance or error-free learner subsets of expert knowledge.

Attempts to extend these analytical models to accommodate learner error soon find themselves coping with the problems of prior knowledge (c.f. Douglas & Moran, 1983; Riley, 1986). That is, elements of performance that are independent of the computer task representation. Additionally, the existing models make no attempt to represent the ongoing *interactive* nature of human behavior at the interface.

This problem of taking into account aspects of human performance at the interface which are independent of the task representation can be called the *context problem*. In the remainder of this paper I will attempt to delineate the nature of this problem by defining the notion of context, giving examples of context accommodation in interface design, and discussing the practical and theoretical problems that context creates for user models and interface design.

2. DEFINITIONS

There is a narrow, grammatical definition of context, that the meaning of an expression changes depending on the sequence of prior expressions. However, for purposes of completeness I will take a broader notion of context. The notion of context in language is an issue for the field known as pragmatics. Morris (1938) defined pragmatics as "the relations of signs to interpreters." Pragmatics treats text as a product of a dynamic process in which language is used as an instrument of communication in a particular situation by a speaker/writer to express meanings and achieve intentions with a hearer/reader. Knowing the context, or the particular

[1] Paper presented at the Workshop on User Models, Human Factors in Computing Systems and Graphics Interface Conference, April 5-9, 1987, Toronto, Canada.

situation of use, is required for complete understanding of the communication. The text alone is insufficient. For example, the sentence "I visit my children every weekend." communicates very different meanings given assumptions about the age and sex of the speaker. The stuff of contextual analysis is reference, presupposition, implicature and inference.

Lewis (1972) defines context as consisting of "a package of relevant factors, an index." The coordinates of the index are:

possible-world: states of affairs which might be, could be supposed to be or are

time: temporal aspects such as "today"

place: spatial aspects such as "here"

speaker: to account for pronouns such as "I"

hearer: to account for pronouns such as "you"

indicated object(s): deixtic expressions such as "this"

previous discourse: to account for references such as "the latter"

assignment: an infinite series of things (sets, sequences, etc.)

Rather than considering these coordinates as simple indexicals, one can broaden the definition to include a more general description of each of these aspects. For example, time and place can be extended to mean the spatio-temporal location of the discourse itself, not simply the lexical references to time and place contained within the discourse. Contexts are by definition recursive, since one context can "call" another.

Linguistics has only begun to explore the difficult issues of formalizing the notion of context. What is certain is that it is crucial to understanding language and communication in general. While user models frequently describe the interface in terms of linguistic descriptive levels of lexicon, syntax, and semantics, they rarely include pragmatics. Yet pragmatics is alive and well in the actual practice of interface design as the next section will demonstrate.

3. EXAMPLES

Existing interface designs make some attempt to accommodate the issue of context of interaction. In this section, I will give an illustrative selection of these features. Most of these concern temporal, interaction history and user specific aspects; only a few concern spatial. However, as bit-mapped graphic displays become more common, spatial context will become more important.

3.1. Multiple Processes

Operating systems, such UNIX, which allow multi-processing provide operating system commands to manage processes from within existing processes. These commands allow the creation, suspension, monitoring, and elimination of processes. Thus, in these systems the notion of process in equivalent to the notion of computing context.

3.2. Windows

Windowing systems on bit-mapped graphics display interfaces currently provide the user with a mechanism form segregating tasks or processes into visually and haptically separate interaction areas. The user can determine tasks of current attention focus by bringing that window into prominence: making it the top of the overlapped stack, expanding its size, etc. Designs for windowing provide mechanisms for visually highlighting or otherwise marking the currently active window(s) on either single-tasking or multi-tasking systems.

3.3. Interaction History

Some operating systems such as UNIX allow the user to create a list of the commands issued during a terminal session. This history file can then be used for reference during command creation. UNIX has special commands which have meanings such as "repeat the last command whatever it was", "repeat the command that had this substring in it", and "edit the last command's arguments as follows...". In addition, this history file can be used by the user as record of the session, thus providing a useful mnemonic device.

3.4. Undo

Undo or redo or restore are mechanisms that are commonly found in text editors or other command language interfaces. These restore the state of the system or file just prior to the last command issued or session initiation. It is not the case, however, that all states can be restored or undone since (obviously) some commands can issue non-retractable effects such as file deletion.

3.5. Saving State from Last Interaction

Some systems maintain the state of the system as a history file which is restored upon initiation of another user session. This state maintenance can include things such as preferences for formats in text editors, the state of the desktop, and the name of the last file edited.

3.6. Time-stamping Files

Most operating systems automatically retain temporal information on file creation and access. This can be used by the user to reconstruct episodes during which particular events occurred.

4. RELEVANCE OF CONTEXT FOR USER MODELING

Thus far, I have argued for the consideration of context as a legitimate component of any interaction, including human-computer. From the psychological standpoint it manifests itself as focus of attention and episodic reconstruction; from the computational standpoint it manifests itself as defaults of spatio-temporal references or recursive tasks; from the linguistic standpoint it manifests itself as language structures. Certainly, problems with context account for some component of error behavior (Norman, 1981). In order to integrate context into the user model however, we will have to examine such issues as how human beings detect context changes, how they landmark contexts, and how they recover contexts. Context shifts are inherent in any embedded help system, error handling, or multi-tasking.

Finally, there are significant practical and theoretical problems in considering context. From the practical point of view, the context problem is somewhat akin to the "frame problem" in artificial intelligence. That is, if you want to backtrack the system, you have to save the prior state information. Allowing users to save histories of interaction could require an enormous amount of storage. From the theoretical point of view, a context is fundamentally a situation of use. Recent research in collaboration may contribute to our understanding of context (Clark & Wilkes-Gibbes, 1986; Cohen, 1985; Suchman, 1986). However, these studies all point out that communication is filled with "errors". Thus, the notion of the error-free interface is a fantasy. Instead, designers should concentrate on mechanisms for detecting and repairing interaction failure. The recovery of prior context becomes crucial from this perspective. However, human beings frequently fail to detect context shifts in each others interaction. This underscores the fact that simply providing context mechanisms is no final solution to the interface problem of user error, since they themselves can be subject to misinterpretation and failure.

5. REFERENCES

Anderson, J.A., Farrell, R. and Sauers, R. (1984). Learning to Program in LISP. *Cognitive Science, 8,* 87-129.

Card, S., Moran, T.P. and Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, N.J.: Erlbaum.

Clark, J. and Wilkes-Gibbes, D. (1986). Referring as a Collaborative Process. *Cognition, 22,* 1-39.

Cohen, P.R. (1985). The Pragmatics of Referring and the Modality of Communication. *Computational Linguistics, 10,* 97-146.

Douglas, S.A. and Moran, T.P. (1983). Learning Text Editor Semantics by Analogy. *Human Factors in Computing Systems,* Proceedings of the 1983 SIGCHI Conference.

Lewis, D. (1972). General Semantics. In D. Davidson & G.H. Harman, *Semantics of Natural Language.* Dordrecht: Reidel.

Morris, C.W. (1938). Foundations of the Theory of Signs. Reprinted in Morris, C.W. (1971) *Writings on the General Theory of Signs.* The Hague: Mouton.

Norman, D.A. (1981). Categorization of action slips. *Psychological Review, 88,* 1, 1-15.

Norman, D.A. (1986). Cognitive Engineering. In D.A. Norman & S.W. Draper (Eds.), *User Centered System Design.* Hillsdale, NJ: Erlbaum.

Polson, P. and Kieras, D. (1985). A Quantitative Model of the Learning and Performance of Text Editing Knowledge. *Human Factors in Computing Systems,* Proceedings of the 1985 SIGCHI Conference.

Riley, M.S. (1986). User Understanding. In D.A. Norman & S.W. Draper (Eds.), *User Centered System Design.* Hillsdale, NJ: Erlbaum.

Suchman, L. (1987). *Plans and Situated Actions.* Cambridge University Press, *in press.*