Testing Predictions of Semiotic Engineering in HCI

Clarisse Sieckenius de Souza
Departamento de Informática — PUC-Rio
Rua Marquês de São Vicente 225
22453 - Rio de Janeiro, RJ — Brazil
email: clarisse@inf.puc-rio.br

Monografias em Ciência da Computação

Departamento de Informática — PUC-Rio

MCC 31/93

Abstract O presente artigo apresenta uma panorâmica do que é a Engenharia Semiótica e, apoiado em resultados obtidos com testes em escala reduzida, propõe um método para a testagem de previsões extraídas de princípios de projeto de interface motivados pela teoria semiótica. A Engenharia Semiótica é uma abordagem teórica que propusemos para a construção de linguagens de interface para usuários, dentro de uma perspectiva sobre a interação usuário-sistema (IUS) na qual os próprios sistemas constituem mensagens complexas enviadas por projetistas a usuários. A teoria semiótica que deu origem aos fundamentos desta abordagem é a Teoria da Produção Sígnica, de Umberto Eco. A meta de longo prazo da Engenharia Semiótica é contribuir para uma *teoria de projeto* na área de IUS. Por isto, os princípios fundamentais propostos anteriormente devem ser testados experimentalmente para que se avalie a teoria. Os métodos e resultados de testes preliminares aqui reportados sugerem procedimentos metodológicos para estudos experimentais em Engenharia Semiótica.

Keywords theoretic approaches to HCI design, semiotic theory, visual interface languages, test methods

Abstract This paper presents an overview of Semiotic Engineering and, with support of results achieved in a small-scale experiment, proposes a method for testing predictions derived from semiotically-motivated interface design principles. Semiotic Engineering is a theoretic approach we have proposed for the construction of user interface languages, within a perspective about HCI according to which systems are themselves complex messages sent from designers to users. The original semiotic theory from which our approach draws its fundamentals is Umberto Eco's Theory of Sign Production. The long-range goal of Semiotic Engineering is to contribute for a theory of design in HCI. Thus, the fundamental principles we have proposed must be experimentally tested as a means to evaluate the theory. Preliminary test methods and results reported here suggest methodological procedures for experimental studies in Semiotic Engineering.

Testing Predictions of Semiotic Engineering in HCI

Clarisse Sieckenius de Souza
Departamento de Informática — PUC-Rio
Rua Marquês de São Vicente 225
22453 - Rio de Janeiro, RJ — Brazil
email: clarisse@inf.puc-rio.br

Abstract This paper presents an overview of Semiotic Engineering and, with support of results achieved in a small-scale experiment, proposes a method for testing predictions derived from semiotically-motivated interface design principles. Semiotic Engineering is a theoretic approach we have proposed for the construction of user interface languages, within a perspective about HCI according to which systems are themselves complex messages sent from designers to users. The original semiotic theory from which our approach draws its fundamentals is Umberto Eco's Theory of Sign Production. The long-range goal of Semiotic Engineering is to contribute for a theory of design in HCI. Thus, the fundamental principles we have proposed must be experimentally tested as a means to evaluate the theory. Preliminary test methods and results reported here suggest methodological procedures for experimental studies in Semiotic Engineering.

Keywords theoretic approaches to HCI design, semiotic theory, visual interface languages, test methods

1. Introduction

The design of User Interface Languages (UIL's) involves a variety of decisions which are usually made based on talent and experience. Existing theoretically-oriented approaches, of either cognitive inclination [18] or semiotic inclination [01], have contributed to the goal of providing designers with a sound set of methods that ensure the construction of high-quality user interfaces. However, the challenge of building a sound theory to support HCI design, capable of explaining and predicting observable behavior, is still in order.

Cognitively-based approaches have provided a wealth of insights, ever since Card, Moran and Newell's model of information processing [04] appeared in the early 80's. Norman and Draper's User-Centered System Design [18] followed this psychological trend and presented a multidisciplinary approach converging to the *cognitive engineering* of computer artifacts. As suggested by its denomination, *cognitive engineering* [17] centers around the idea that HCI is fully governed by interpretation and evaluation activities, performed by users who have the challenge of translating goals into input events and judging systems reactions from perceived output events.

Communicative aspects in Norman and Draper's work have been explored in terms of *semantic and articulatory directness* [14] of the UIL — roughly a measure of the distance between the user's intentions and the meaning of UIL expressions and of the distance between the form of UIL expressions and their meanings, respectively. Such aspects have also been approached in terms of input/output inter-referentiality [10], emphasizing the fact that input and output languages, though possibly varying in appearance, must refer to a common set of interface objects, if they are to support efficient evaluation of the system's behavior in terms of the user's intended goals.

The *cognitive engineering* approach has set the basis for what we've called *semiotic engineering* [09], which is a semiotically-oriented pre-theory of UIL design. The backbone of *semiotic engineering* is Umberto Eco's Theory of Sign Production (TSP) [11], in which the author advocates the existence of four parameters that characterize the activity of producing messages to be transmitted in communication. Additionally, our pre-theory is committed to a broader communicative perspective in HCI, according to which computer systems are complex messages sent from designers to users [15, 16, 01], comparable to such messages as artwork, cultural practices, and the like.

This paper presents a summary description of *semiotic engineering* and reports the methods and results of preliminary tests meant to evaluate part of the predictions supported by our approach [09]. For these initial tests, a redesigned portion of the Macintosh desktop visual language has been contrasted with the original one [02], as in its system 7.1 version. The new design has followed *semiotic engineering* design principles which have apparently not been followed in the original design [08]. Our conclusion is that the test methods employed in this preliminary phase strongly suggest that they can be used in larger-scale tests, and that by the control of a number of variables the seemingly correct prediction of the theory is on its way of validation.

2. Semiotic engineering in HCI

The goal of semiotics is to build a general theory of signs — verbal and non-verbal — applicable to natural and artificial communication systems. As pointed out by Andersen [01], such goal is directly implied in the design of computer systems, although only recently has this connection been explicitly recognized by HCI researchers. From the wide variety of approaches semioticians have proposed, those that explicitly attempt to follow a more formal path have offered more attractive opportunities for computer science researchers than those that do not. This is Andersen's major motivation in adopting the Copenhagen School of Glossematics approach to the problem [13], whose aim was to set up a calculus or algebra to describe empirically observed natural language texts. However, Glossematics analytical bias common to many studies of natural language phenomena gears Andersen's Theory of Computer Semiotics towards methods and practices that concentrate on activities relative to the analysis of existing semiotic structures in the current users environment, prior to systems design.

In conjugation with Andersen's proposals, there has to be a complementary emphasis on the process of designing and realizing the ultimate designers message computer systems are supposed to transmit to users. In other words, analysis procedures must be accompanied by synthesis procedures. Such procedures should be able to indicate how to synthesize good computer messages, given the goals and constraints observed in analytical studies.

Semiotic engineering draws on Eco's Theory of Semiotics because this is precisely what it has set out to accomplish within the scope of all possible communication systems. As is the case of most theoretical work meant to account for mental processes involving creativity and design, Eco's theory is rich in insights, but lacking in formal rigor. Nevertheless, its alleged flaws may have more negative consequences when applied to natural communication systems than to artificial ones. One of the reasons for this is the fact that artificial codes are formal and embody clearly constraining abstraction principles known to message designers who select them as a means of expression. Another is that artificial codes are based on interpretive conventions supposedly known to its intended community of users (as senders and receivers), and this rules out many of the problems arising in natural language communication, where creativity of usage, polysemy of expressions, contextual ambiguity of utterances, and so many other factors, contribute to interpretive problems and rupture in communicative processes.

Eco's Theory of Semiotics is divided into two major sub theories — a Theory of Codes (TC), which is oriented towards the structural description of communication systems, and a Theory of Sign Production (TSP), which is oriented towards an account of the conditions under which communicators produce signs or messages. Computer languages are fundamentally connected to structural studies of language, and have long had explicit ties with linguistic research [05,06,07]. However, computer languages have been traditionally viewed as a sort of one-way communication code, insofar as programmers have been sending messages to compilers and interpreters. And in this case, the production of programming signs has been constrained by such factors as provability, reliability, orthogonality, portability, and the like [03].

HCI research has made programmers realize the need for a full two-way communication approach to interface design. Not only do users send messages to application programs (as top-level programmers), but they also receive and interpret messages coming from or *through* the application. As pointed out by Kammersgaard [15], some user interface research carried out under the inspiration of Artificial Intelligence has implicitly or explicitly assumed that messages received from users are sent by the application system. One of the critical issues with this design orientation is that users may be misled into believing that systems are cognitive peers of theirs, i.e. that they share their interpretations of the world. Winograd and Flores have extensively discussed the problems of this approach and their consequences for design [21].

An alternative view is one in which designers communicate with users *via* computer systems [15]. In a previous paper, we have discussed in detail the nature of computer systems in this light, and have proposed that they are actually metacommunication artifacts designed to be performing messages sent from programmers to users [09]. A critical design issue is then to understand the process of elaborating this complex message.

Eco's TSP proposes that there are four parameters which together account for the process of producing signs. They are related to: (a) the physical and mental effort a communicator has to perform when selecting the significant units that will be included in the message; (b) the optimization of the associative relation between form and content of significant units; (c) the segmentation of an expressive *continuum* in which verbal systems co-exist with non-verbal ones; and (d) the level of articulation of the communicative system in which the message is to be embedded.

The physical or mental labor involved in selecting expressive units ranges from the recognition that certain units are available as valid expressions for the intended meanings to the invention of novel units, given the inadequacy or nonexistence of available ones. The optimization of associations users will have to do between form and meaning involves, for example, choosing forms that immediately evoke intended meanings instead of those that do it indirectly (e.g. by means of analogies), or choosing an expressive system that is an established way of conveying the intended meanings. The segmentation of the expressive *continuum* is related to the distribution of content and form among a variety of communication systems, like the linguistic, the gestural, the pictorial, and many other ones among which — and very importantly — the computational one. And finally, the level of articulation refers to existence or not of combinatorial rules applicable to components within the chosen communication setting; for example, natural language is highly articulated, whereas painted pictures are practically non-articulated.

Of course, the above parameters interact with each other, and it is unlikely that they can all be set to the ideal values in the design of messages. Communicators intuitively or methodically balance out the influence of all parameters, given their message's goals and context of production. Although Eco avows that TSP isn't a finished theory, its current state can already impact the design of computer-human interaction in important ways.

The range of choices in HCI is considerably more restricted than in natural human interaction. To illustrate this, we can observe the fact that computer systems are highly articulated by necessity and that expressive resources are highly constrained by the technological state of the art in I/O devices. Moreover, by assuming that HCI is actually a mediated conversation between users and systems designers, the use of novel signs in the interface is qualified by the fact that such signs are only novel the first time users encounter them. From then on, no invention of signs is possible because computer systems are accomplished messages that do not exhibit signs that haven't been implicitly or explicitly included in them by programmers. Along this line, programmers can have a good grasp of how easy or how difficult it will be for users to associate expressions to their corresponding contents. Even if analogies are commonly used as a mechanism for such associations, their impact can be previously gauged in the light of how familiar the analogy is for the users population.

The *semiotic engineering* of UIL's is then governed by principles derived from the above TSP parameters. They aim at optimizing the users cognitive processes, which are well synthesized in Norman's *cognitive engineering* as those of Execution (relative to Input) and Evaluation (relative to Output).[17]. *Semiotic engineering* principles apply to both textual and non-textual interfaces, and state the following:

1. User Interface Language designers should produce signs they recognize as existing codified expressions of the intended contents.

2. User Interface Language designers should try to select expressions that are recognized as a token of an established (i.e. easily associated) type of expression system which accounts for the intended contents.

3. User Interface signs referring to domain objects and to computer-modeled solutions for existing problems should have forms directly or indirectly borrowed from the domain semiotic system, whereas those referring to I/O devices and operation system actions should be directly borrowed from a computer-specific subset of signs.

4. User Interface Language designers should always resort to expressions belonging to a recognizably codified (i.e. rule-based) system.

Such principles, just as Eco's TSP parameters, are expected to be the building blocks of a theoretical approach to UIL design. They map the territory for numerous kinds of investigations among which we can mention the sort of interactions they have with each other, the adequacy of message design against message interpretation, the adequacy of explanations and predictions they support for observed HCI phenomena, and their impact on the overall process of systems design.

A series of predictions and explanations such principles suggest have been reported in previous work [09]. In what follows we describe a pilot test designed to evaluate the correctness of one of our previous predictions.

3. Visual Grammars for *Copy* and *Move*

The Macintosh Desktop allows users to interact with the underlying file management system with direct manipulation [20] of visual objects on the screen. The copy and move file actions are performed by drag drop events involving icons related to files, disks and the desktop itself. Given its visual nature, we will call this communication code a Visual UIL, or V-UIL.

As stated in Section 2, all computer codes are articulated by necessity, which means that the V-UIL is rule governed and not an encapsulated association of expression and meaning. Users can perceive regularities between drag drop events (expressive level) and copy and move actions (content level). For example, as sketched in Figure 1, a file icon dragged from the desktop onto the hard disk icon causes the file to be moved from the desktop into the hard disk and vice-versa (note that there is nothing on the desktop before the second drag drop). If the icon is dragged onto the floppy disk icon, the file is copied into the floppy disk (the file icon remains on the desktop). The "vice-versa" part of the latter event, however, results in a curious situation experienced by most naïve users of the Macintosh system 7.1 interface. After the second drag drop of this pair, there will be two visual copies of the same icon (or file) on the desktop. In other words, the apparent symmetric drag drop action in both cases contradicts the expected symmetric result.

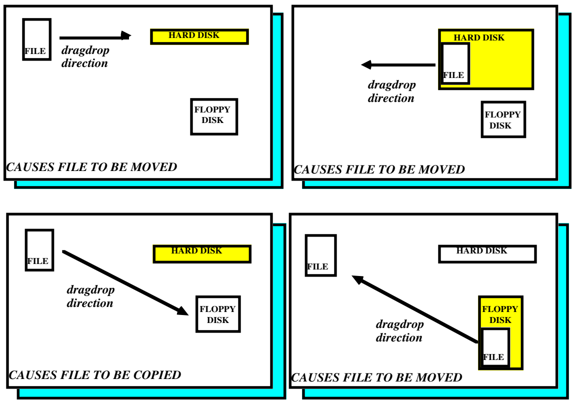

Figure 1: Drag drop Visual Effects in V-UIL

Of course, the intervening factor causing the unexpected asymmetric result of the drag drop is the active context in both cases. In the first pair of symmetric drag drops the active context was the same — the hard disk. In the second pair, though the active context was the hard disk in the desktop-to-floppy drag drop, it was the floppy disk in the symmetric floppy-to-desktop drag drop.

A detailed analysis of both tasks, using the GOMS paradigm [04] and Task-Action Grammars [18], has been carried out [08] and can be summarized by the following.

Firstly, drag drop interface events result in the effects shown in Table 1. Note that there is an ambiguity as to the effect of desktop-to-floppy and desktop-to-hard disk drag drops. This ambiguity is dispelled when users realize that the active context of the operation determines the kind of result to be expected. The expression of such a decisive factor in the V-UIL is the foreground color of the disk icons. When the hard disk is active, it shows a darker foreground color, whereas when it is inactive it shows a lighter hue. Having grasped this articulation, users correctly predict the result of all typical drag drop operations.

DRAG DROP FROM	DRAG DROP TO	RESULTING EFFECT
hard disk	floppy disk	Copy
floppy disk	hard disk	Copy
hard disk	desktop	Move
floppy disk	desktop	Move
desktop	hard disk	Copy or Move
desktop	floppy	Copy or Move

Table 1: Drag drop Effects on the Macintosh Desktop

Secondly, in *semiotic engineering* terms, we see that designers have chosen express part of the idea of movement (leaving or not leaving a copy of the moved object at the origin of its trajectory) in association with a spatial expressive system, which is absolutely appropriate, but part of it in a color code system. In terms of principle 2, above, the choice is partially optimal and partially sub-optimal. However, since such displacements are not the actual physical displacements users are familiar with outside the computer world, but underlying file management system operations of a very specific type, principle 3 has been possibly overlooked. Thus, the spatial code selected according to principle 2 motivates an analogy between two worlds, and reveals that the actual content of spatial expressions is not really the familiar one, but only a similar one. Consequently, we can dispute the idea that principle 2 is fully observed. Previous research has pointed at the potential dangers of the use of analogies in UIL's [12], and Eco's TSP parameters provide interesting insights into the reasons why analogies and metaphors are potential obstacles in communication [11].

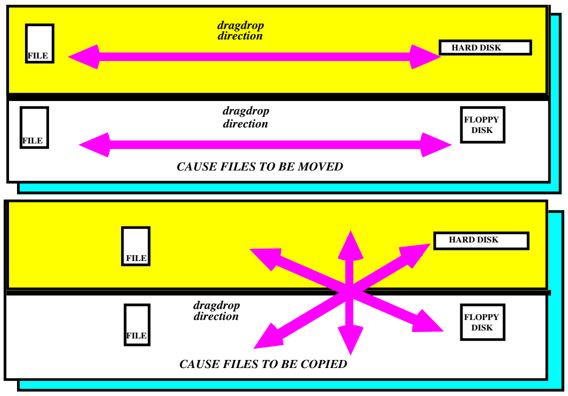

Figure 2: Drag drop Visual Effects in V-UIL*

Thirdly, what follows from the above is that there is an interesting tension between the choices between abiding or not to principle 2. We propose that this tension must be resolved by the concurrence of another principle, namely principle 4. What it says is that UIL, either visual or not, should exhibit clear codified articulation of expression and content. In this respect, as said before, in the original design the crucial disambiguating factor for proper anticipation of drag drop effects is expressed in a different code — a color code.

Of course this choice works, because there is a common higher-order visual code to which both colors and locations belong. However, at the fine-grained level of codification, there is a disparity of segmentation in the expressive *continuum*, in that contents related to the same global task (meaning) are separated into distinct signaling resources (form).

It can be seen that the four principles of *semiotic engineering* support an analysis of this V-UIL and provide the structure for explanations regarding observable ambiguities of the interface under appreciation. The second step in evaluating our approach's reach has been to make predictions regarding the design of another V-UIL — which we will refer to as V-UIL* — and carry out tests with novice users of both versions to see if predictions are right or wrong.
The kind of prediction we have chosen to examine is that relative to the effects of switching from one established expressive system (i.e. space) to another (i.e. color) in the articulation of unitary global contents (i.e. copy and move tasks). We have set out to design V-UIL* with the purpose of codifying the active context intervening in the interpretation of drag drop effects within the spatial code, alone. This choice has been judged superior to choosing the color code alone, due to the ease of association between visualized location of icons on the desktop and the actual physical location of files on the disks. For sake of control, we have abstracted principle 3 from our test and have assumed principle 2 overrides it.

V-UIL* then presents a new visualization of the Macintosh desktop, where the active context is associated to spatial coordinates. As sketched in Figure 2, each disk has its own desktop (as was the case in V-UIL), but desktops are visually expressed as two horizontal halves of the screen. The upper half is associated to the hard disk (the disk icon is shown on the upper right corner of the upper desktop), and the lower one with the floppy disk (the disk icon is shown in the upper right corner of the lower desktop).

4. Testing the Theory

In order to get a deeper perspective into this new kind of design before user testing, we have produced the same formal models of V-UIL* as we had used for V-UIL. A comparison between them [08] has shown that V-UIL* can be described by a smaller number of unconditional rules in the Task-Action Grammar than is the case of V-UIL. It has also shown that the relationship between GOMS conceptual objects and interface visual elements was made more direct by the emergence of the spatial notion of two hemispheres across which drag drops had consistently predictable effects. Table 2 summarizes the associations of drag drop events and their effects.

DRAG DROP FROM	DRAG DROP TO	EFFECT
upper hemisphere	upper hemisphere	Move
lower hemisphere	lower hemisphere	Move
upper hemisphere	lower hemisphere	Copy
lower hemisphere	upper hemisphere	Copy

Table 2: Drag drop Effects on the Redesigned Desktop

If we assume that systems are complex interactive messages sent from designers to users, the rules governing interaction are a legitimate part of the original message. *Semiotic engineering* supports predictions about the signs that will convey such rules. In this experiment, we focus solely on Copy and Move rules as expressed in V-UIL and V-UIL*. The prediction we want to test is that, according to principles 2 and 4, if copy and move articulatory rules in V-UIL's involve tokens of an established type of expression system, users interpretations of designers messages are systematically more adequate than if some tokens are borrowed from non-established types of expression systems.

An underlying assumption in our prediction is that the V-UIL expressive system which is more perceptively salient becomes the established one, whereas perceptively non-salient expressive systems do not necessarily do.

Ten non-Macintosh users, divided into groups A and B, have been selected as subjects for a preliminary experiment. Among them, 3 were casual computer users (A1#3, A2#3, B#1), 4 were non-HCI computer science researchers (A1#1 and 2, A2#1, B#2), and 3 were professional programmers (A2#2,B#2,B#3). Both groups interacted with a Visual BASIC® prototype of V-UIL and V-UIL* for a maximum of 5 minutes.

After interaction with each of the models, Group A was asked to fill out slots in 4 column tables (one for each model) with the following headings: (a) drag drop from; (b) drag drop to; (c) copy or move effect? ; and (d) always or under certain conditions?. Slots (a) and (b) were already filled out (covering all possible drag drop trajectories), and slots (c) and (d) were to be filled out by subjects. Group B was not asked to fill out the tables — the only task was to say what were the underlying rules to copy and move files in each of the 2 models.

Group A consisted of 6 subjects, randomly divided into 2 subgroups: Group A1 who interacted with the original model first, and group A2 who interacted with the new model first. Group B consisted of 4 subjects, all of whom interacted with the new model first. Results shown in Figures 3 and 4 show the following:

❑ Successful interpretations of model 2 were consistently more frequent than those of model 1 in both groups.

❑ The only subject who correctly interpreted model 1, also provided a correct interpretation of model 2

❑ Group A2 subjects consistently achieved higher scores in the total number of correct slots than Group A1 subjects.

❑ Complete failures in Group A occurred only with subjects who examined model 1 before model 2.

❑ All subjects in Group A2 achieved the highest score possible in the Move column slots of V-UIL*.

❑ Subjects in Group A2 had exactly the same scores as each other in all column slots of V-UIL. (Actually, they made exactly the same mistakes as each other.)

❑ All subjects in Group A1 achieved the lowest score possible in the Move column slots of V-UIL.

❑ In Group A2 there were more cases of homogeneous interpretations of copy/move effects (3 in 4) than in Group A1 (1 in 4).

5. Conclusions

Such results have suggested some variables to be controlled in full-scale tests for the above prediction. Firstly, the **users background** with computers seems to have played a role in the only complete failure of the V-UIL* (subject A1#3 is a casual computer user, not acquainted with the Macintosh desktop model, who got the absolute opposite interpretation of visual representations of copy and move actions). Moreover, subject B#4 is a professional support programmer, who is acquainted with a variety of direct manipulation interfaces for graphics applications.

Secondly, the **order of presentation** may have significantly influenced successful interpretations of either models. There may be another variable related to this one, namely the **influence of a previous satisfactory model** of visual language commands for copy and move actions.

Subjects Id		V-UIL Model			V-UIL* Model		
Sub-group	Subj. #	Copy Slots OK	Move Slots OK	Total Slots OK	Copy Slots OK	Move Slots OK	Total Slots OK
A1	1	2/4	0/4	2/8	7/8	1/4	8/12
	2	0/4	0/4	0/8	2/8	4/4	6/12
	3	2/4	0/4	2/8	0/8	0/4	0/12
A2	1	2/4	2/4	4/8	2/8	4/4	6/12
	2	2/4	2/4	4/8	8/8	4/4	12/12
	3	2/4	2/4	4/8	8/8	4/4	12/12

Figure 3: Group A Results

Subject #	V-UIL Model	V-UIL* Model
1	Failed	Failed
2	Failed	Failed
3	Failed	Succeeded
4	Succeeded	Succeeded

Figure 4: Group B Results

Thirdly, the **semantic directness** [14] of move commands as compared to copy commands in either V-UIL. In group A, there have been more cases of highest scores in the Move columns than in the Copy columns, and the only cases of the latter have also been cases of the former.

In conclusion, we suggest that the method used to carry out the pilot tests is adequate for a full-scale testing of the prediction made *semiotic engineering* principles, that if copy and move articulatory rules in V-UIL's involve tokens of an established type of expression system, users interpretations of designers messages are systematically more adequate than if some tokens are borrowed from non-established types of expression systems. Moreover, we strongly suspect, based on the above indications, that the prediction is correct, and possibly so is another one concerning principle 3 (see Section 3), that the codification of copy actions in both V-UIL's is sub-optimal because of intervening analogies with real world displacements of physical objects.
Further steps in adjusting and validating the theory as far as the above prediction is concerned will include tests with other types of commands, other types of expressive systems (including textual ones) and other types of grammatical rules within each system.

Acknowledgments

The author would like to thank the participants in the reported experiments, without whom this paper wouldn't have been written.

6. References

[01] Andersen, P.B. (1990) *A Theory of Computer Semiotics*. Cambridge. Cambridge University Press.

[02] Apple Computer, Inc. (1987) *Human Interface Guidelines: The Apple Desktop Interface*. Reading, Ma. Addison-Wesley.

[03] Appleby, D. (1990) *Programming Languages*. New York. MacGraw-Hill.

[04] Card, S.K.; Moran, T.P.; Newell, A. (1983) *The Psychology of Human-Computer Interaction* . Hillsdale, NJ. Lawrence Erlbaum Associates.

[05] *Syntactic Structures*. The Hague. Mouton.

[06] Chomsky, N. (1963) Formal Properties of Grammars. Luce, Bush and Galanter (eds.) *Handbook of Mathematical Psychology*. Vol. II. New York,NY. Wiley. pp. 323-418

[07] Chomsky, N. (1965) *Aspects of the Theory of Syntax*. The Hague. Mouton.

[08] de Souza, C.S. (1993a) Regularidade e Generalização em Interfaces Gráficas. *Proceedings of SIBGRAPI'93*. Recife, Pe. October 18-22, 1993.

[09] de Souza, C.S. (1993b) The Semiotic Engineering of User Interface Languages. *International Journal of Man-Machine Studies* (39), in print. 1993

[10] Draper, S.W. (1986) Display Managers as the Basis for User-Machine Communication. Norman and Draper (eds.) *User Centered System Design* . pp. 339-352

[11] Eco, U. (1976) *A Theory of Semiotics* . Bloomington, IN. Indiana University Press. (First Midland Book Edition, 1979)

[12] Halasz, F. and T.P. Moran (1982) Analogy Considered Harmful. *Proceedings of the Conference on Human Factors in Computing Systems*, March 15-17, 1982, Gaithersbury, Md.

[13] Hjelmslev, L. (1963) *Prolegomena to a Theory of Language*. Menasha,WI. University of Winsconsin Press.

[14] Hutchins, E.L.; Hollan, J.D.; and Norman, D.A. (1986) Direct Manipulation Interfaces. Norman and Draper (eds.) *User Centered System Design* . pp. 87-124

[15] Kammersgaard, J. (19988) Four different perspectives on human-computer interaction. *International Journal of Man-Machine Studies* (1988) **28**, pp. 343-362.

[16] Nadin, M. (1988) Interface design: A semiotic paradigm. *Semiotica* 69 — 3/**4** (1988), pp.269-302.

[17] Norman, D.A. (1986) Cognitive Engineering. Norman and Draper (eds.) *User Centered System Design* . pp.31-61

[18] Norman, D.A. and S.W. Draper (1986) *User Centered System Design*. Hillsdale, NJ. Lawrence Erlbaum Associates.

[19] Payne,S.J. e Green,T.R.G. (1986) Task-Action Grammars: a Model of the Mental Representation of Task Languages. **Human-Computer Interaction** 1986, volume 2, pp. 93-133.

[20] Schneiderman, B. (1983) Direct Manipulation: a Step beyond Programming Languages. *IEEE Computer* 16(8), 57-69

[21] Winograd, T and F. Flores (1986) *Understanding Computers and Cognition: New Foundations for Design* . Norwood, NJ. Ablex.