On Classes and Costs of Multimodal Interfaces

Clarisse Sieckenius de Souza[1]
Raquel Oliveira Prates
Flávio Miguel Varejão[2]

Departamento de Informática — PUC-Rio
Rua Marquês de São Vicente 225
22453-900 Rio de Janeiro, RJ — Brazil
email: {clarisse,raquel,fvarejao}@inf.puc-rio.br

Abstract

Multimodal Interfaces have considerably enhanced computer-human interaction by allowing designers and users to communicate via different codes, and often different channels. Nevertheless, multimodality does not involve a homogeneous set of design options. Rather, there are quite different situations for which we propose a taxonomy of alternatives based on each mode's expressive power relative to the application's conceptual model. According to it, optional and mandatory cases of multimodality can be identified and associated to specific cognitive and computational costs. We report the results of a small-scale exploratory experiment carried out to gauge the incremental costs of maintaining high-quality multimodes for layout applications. And, we conclude that there are certain semiotic conditions that maximize the chances of success in multimodal interface design, a point which is further supported by research results achieved in the field of Cognitive Science.

Keywords: Multimodal Interfaces, Computer Semiotics, Language Expressiveness, Human Cognition, Implementation Costs

1. Introduction

Multimodal Interfaces have extended the interactive capacities of computer systems by allowing people to resort to a variety of communicative modes (e.g. natural and visual languages) and media (e.g. audio and video) in search of achieving an efficient and effective usage of software tools. However, if all modes of communication in a multimodal interface have the same expressive power, multimodality hardly helps users gain a broader perspective about computer systems. Therefore, each mode should have intrinsic semiotic distinctions relative to others, so that the main purpose of multimodality is achieved.

Nevertheless, when communicative modes are semiotically different, it is not possible for them to share exactly the same semantic model (e.g. the system's conceptual model). If they did, either more expressive modes would be unable to realize their full communicative

[1] This author is supported by CNPq research grant number 303232/89-7.
[2] This author is an Assistant Professor at UFES - the Federal University of Espírito Santo, Brazil.

potential, or less expressive modes would be unable to compute all the meanings included in their own semantic model.

Thus, multimodality must be adopted conditionally, and the criteria to decide on its adoption must be not only of a cognitive, but also of a semiotic nature. Whatever the choice of modes, a price will be paid for a wider range of communication between user and system. This price can be gauged by the nature and dimensions of extensions or changes imposed to a system's minimal conceptual model, which is usually the semantic model of unimodal interfaces.

In order to demonstrate the points above, we propose a taxonomy of the communicative distribution of modes in Multimodal Interfaces, providing some criteria for maximizing the efficiency and effectiveness of multimodality. We illustrate the modeling costs of multimodality through a pilot experiment with a layout application having two interface modes textual (natural language output) and visual (drag drop icons and spatial configurations). Small-scale user testing has shown that in this sort of application multimodality is desirable, but that the relatively simple spatial model that is sufficient to represent layout for visual communication must be considerably extended to support the semantics of useful high-quality natural language texts about static spatial configurations or subsequent stages of spatial reasoning and decision-making in typical tasks.

Further theoretic investigations should show the extent to which classes of interface modes are semiotically supplementary or complementary to each other across certain kinds of applications or tasks. As an example, Stenning and Oberlander's theoretical framework for the analysis of reasoning based on graphical and linguistic representation systems [Stenning and Oberlander'95] has supplied a convenient embedding for some exploratory investigations reported here. On a more empirical basis, future large-scale user testing should reveal the actual parameters of interpretability and expressiveness of alternative interface codes in situated computer-human interaction.

2. Semiotic Conditions of Multimodality

The ultimate goal of user interface design is to allow users to understand fully a system's computing potential, so that they may creatively wield this software tool and solve novel classes of problems [Adler & Winograd'92]. Thus, the codes an interface designer chooses in order to convey his or her own conception of what people (could or should) wish to do with a given computer application achieve more than one target. They allow users to interact with the application and execute a number of actions; to learn which computations can be done with their input (and possibly how they are done); and they trigger the users imagination as to which other tasks could be performed, partially or totally, by analogy with the tasks the interface directly refers to. In sum, computer interfaces are themselves a message that reveals its range of meanings through other messages it is able to generate and interpret. They are metacommunications artifacts [de Souza'93].

As in natural human communication, it is almost always the case that the conceptual model of a complex object cannot be expressed in its

full extent by means of one and only communicative code. Even natural language, which is the most expressive coded system we are able to use, falls short of allowing people to understand some kinds of messages. In such cases, speakers usually resort to auxiliary codes like pictures, gestures and a variety of higher-order linguistic constructs, in order to help their listeners capture the nuances of what they have in mind.

Multimodal Interfaces have expanded the boundaries of computer-human interaction by allowing designers to use a number of communicative codes to compound their global message to users. However, a wider range of choices brings about greater complexity for design. Since computer resources are considerably limited in terms of knowledge representation, manipulation and retrieval, designers must choose a set of modes that optimally combine with each other in the mosaic of perspectives they offer about the system's potential. Communicative optimality is taken here in a very loose sense: It stands for the quality of codes being simultaneously easy to understand and use, perspicuous in referring to all meaningful objects of a given semantic model, and flexible in accounting for different levels of abstractions about conceptual objects.

One of the key criteria for choosing more than one mode of interaction is that they differ from each other in at least one of the three dimensions above: ease of use, perspicuity, and flexibility. If two modes are equally easy, perspicuous and flexible, they are probably presented at the interface for matters of taste or culture only. Because users don't gain or lose understanding about the system by choosing one or the other, multimodality in this case is not really achieving an expansion of the communicative boundaries of computer-human interaction. Interesting cases of multimodality appear when modes vary along one or more of the three dimensions, and thus offer users a number of insights about the software tool. In Figure 1, we sketch 4 alternative situations that depict the sort of distribution in which two modes (and, by extension, any number of modes) may be found relative to each other.

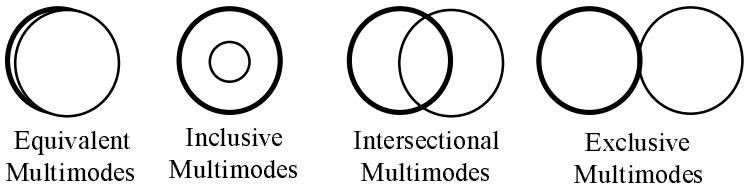


Figure 1: A Taxonomic Framework for Multimodality

Equivalent multimodes are those virtually interchangeable among each other. It should be noted that real equivalence is in theory rare, since different codes potentially evoke different interpretants - i.e. different second-order signs generated in the process of interpreting a first-order sign [Peirce'31]. As an example, although the expressions "x is equivalent to y" and "x⇔y" could be taken to be virtually equivalent to each other in terms of ease of use, perspicuity and flexibility, their occurrence in writing can set the reader on different interpretive frames of mind. And this difference is precisely the subtle reason why some authors choose to describe their objects in Logic Notation instead of its direct Natural Language translation.

Inclusive Multimodes are those in which some mode encompasses some other. The encompassed mode is fully intersected by the encompassing mode, and thus it does not extend the boundaries of communication. Nevertheless, its existence may be justified because of its very ability to refer only to a subset of the objects or qualities referable by the encompassing code. One of the benefits of such distribution is the ability of the encompassed mode to set the focus of users attention on only a clearly circumscribed range of meanings for certain kinds of tasks.

Intersectional multimodes and exclusive multimodes are evidently a case of mandatory multimodality for effective and efficient communication between user and system. If only one mode existed in either case, it would be impossible for users to have access to parts of the system's conceptual model. Therefore, full usability could never be achieved. Exclusive multimodes seem to be just a more drastic case of intersectional multimodes, but the issue is rather qualitative than quantitative. With intersectional multimodes, unlike with exclusive multimodes, symbols in one mode can refer to semantic entities referable in the other mode. Thus, not only can users be led more smoothly from one communicative domain to the other, but also the intersection between codes shows how different codes distinctively approach the same contents. And this is of crucial importance for users to guess the expressive power of each code.

It follows from the above that multimodality ranges from the optional to the mandatory, along a continuum of expressive modes. In all cases, the semantics of the different modes is defined by the system's conceptual model, which ideally characterizes all and only the meanings that the user must master to learn, use, and modify the original power of software tools. If it had not been so, it would be possible to have modes whose referred meanings fell outside the scope of the computer application, or to accept that a given interface did not warrant a system's full usability.

Nevertheless, a system's "conceptual model" is an abstraction that programmers must implement in a way or another at the time the actual software machine is created. The implementation step is a translation of the model into a very specific linguistic environment which will react to input data in a principled way. Expressions formulated in the various interface modes will be (recursively) translated into expressions that belong to the implementation environment and this is where the cost/benefit analysis of multimodality plays a major role. Theories of formal languages prove the correspondence between the expressive power of computable languages and the complexity of the abstract symbol-processing machine that is able to compute their expressions [Chomsky'59, Aho and Ullmann'72]. Thus, the more expressive the mode at the interface level, the more expensive the implementation of its processor. This simple rationale might work as a criterion to rule out optional multimodality in a number of cases. For instance, in case of inclusive multimodes, computer complexity will certainly be weighted against focal control if the encompassed mode is itself an extensive context-sensitive language. By the same token, equivalent multimodes are likely to be implemented only within the boundaries of computationally inexpensive modes of interaction, like regular languages or structurally simple context-free languages.

Recent work by Stenning and Oberlander has established an insightful correspondence between Chomsky's hierarchy of formal languages and what they call Abstract Representation Systems - ARS [Stenning and Oberlander'95]. They have outlined a cognitive theory of graphical and linguistic reasoning and argued that graphical representation limits abstraction and aids processibility. They have proposed a hierarchy of three types of ARS. Minimal Abstract Representation Systems (MARS) allow for exactly one semantic model to be built out of each expression in the representation system. Limited Abstract Representation Systems (LARS) allow for more than one semantic model to correspond to their expressions, because they can leave some features of the semantic model unmarked, for example. And finally Unlimited Abstract Representation Systems (UARS) may not only have more than one semantic model that corresponds to their expressions by virtue of a certain set of interpretive keys, but they may also support a change in such interpretive keys (which leads to novel correspondences between expressions and semantic model configurations). The authors argue that what distinguishes graphical and linguistic representation systems (such as Euler's Circles and Natural Language, for instance, in representing syllogisms) is the specificity of the former, in contrast with the virtual unlimited abstraction capacity of the latter.

Their point is of particular interest to our discussion here because they conclude that limiting abstraction is a most relevant cognitive strategy to gain comprehension of complex models. Thus, they provide us with a theoretical background to examine aspects of a cost/benefit analysis for real multimodal interfaces. In fact, their example of reasoning in linguistic and graphical mode is a case of inclusive multimodes. Euler's Circles are hierarchically less powerful than natural language, and they can be used, as pointed out above, to set the focus of a reasoning process onto a clearly circumscribed subset of abstractions. We will report in the following a small-scale experiment with a prototype layout system whose interface included visual (a top view of a furnished office room) and textual (short descriptive and narrative texts in Portuguese) modes. Our example is also an instance of inclusive multimodes, and we will show a stepwise refinement of the interface code that helps us gauge implementation costs.

3. A Multimodal Interface for a Layout System

Experimental research [Kirsh and Maglio'94] has shown that in games requiring spatial reasoning, like TETRIS, it is not the case that all of the players actions are meant to directly decrease the distance between the game's current state and the final target state. Some actions are performed just in order to allow the player to have different perspectives about the shapes on the screen. Thus, the problem of fitting shapes into existing slots may be restated in terms of new and potentially simpler views of the object. Such actions are called "epistemic" actions, and their purpose is to facilitate the formulation of the (spatial) problem, as suggested in Figure 2. In (A), the rotation of the shape is directly interpreted as a means to make it fit into the slot of the platform (local target). In (B), however, no rotation is actually needed to make the falling shape fit into the slot. Nevertheless, the player may still successively rotate the shape to examine all the opportunities of fitting, and choose the best one within a restricted time frame.

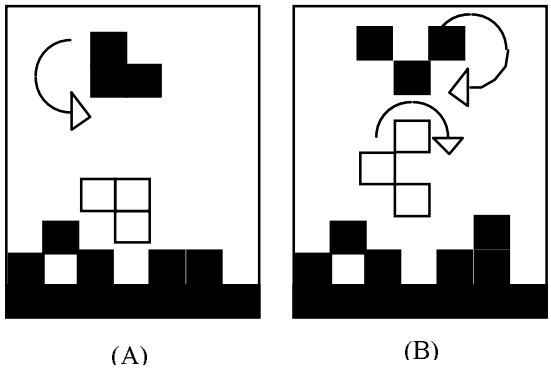


Figure 2: Example of an epistemic action [Kirsh and Maglio'94]

Although layout systems are not games, and users are not competing against time in their efforts to make objects fit into a given frame, spatial reasoning is still the main cognitive task performed. The flexibility to perform epistemic actions in layout tasks naturally ranks direct manipulation interfaces as the best design choice for computer-human interaction.

However, if the application is also meant to allow users to retrieve the steps of the whole layout process for purposes of documentation or redesign, for instance, strict direct manipulation seems to impose unnatural expressive constraints upon the interface. Suppose that in order to convey the various stages in the configuration process, the interface could offer users not more than a playback of the layout sequence — an animation of the successive steps.

Whereas the reproduction of static individual layout stages, final or other, may meet users cognitive needs in a variety of situations (e.g. to show that a certain piece of furniture has been tentatively located in a position where it is not currently found), the same strategy does not seem to be just as adequate in the case of the dynamic aspects of the task (e.g. to show that one given piece of furniture has been successively moved to all corners of a room before being located next to the door). A playback of a sequence of individual layout states showing that a certain object has changed positions but has eventually returned to its original location provides the necessary elements for users to **infer** that the layout designer has considered a number of hypotheses before backtracking to his/her original solution. However, the same layout sequence, if expressed in narrative natural language (NL) text, would be likely to include words like **RETURN** which directly refer to an abstraction of the visual pattern faithfully reproduced in the animation.

In sum, the expressive resources of visual representations compared to textual NL representations seem to fit differently tasks involving static or dynamic computations in spatial reasoning. Combined with Stenning and Oberlander's claims about the role of specificity in spatial and linguistic reasoning, this point supports the choice of inclusive multimodes. The visual system in the direct manipulation mode is actually a MARS (i.e. each visualization corresponds to just one semantic model of the objects in space), and the linguistic system is potentially a UARS, if full natural language interpretation and generation were really possible. For sake of the current discussion, we will deprive the linguistic system of its

metalinguistic power (i.e. the ability of deriving new semantic models from changes performed in the system's key assertions) and downgrade it into a LARS (i.e. a representation system that can generate more than one corresponding semantic model by means of abstractions about the value of certain semantic parameters). This capacity is especially interesting to distinguish epistemic actions from final actions in solving spatial configuration problems. Narrative NL texts can be generated with such abstractions that only states which actually approached the final solution are mentioned, all others being loosely marked by adverbial forms like "eventually" or "finally", for instance.

In order to illustrate the issues above, we have designed a very simple program to help users make the layout of a typical office in our university. Offices are shared by two people, and furnished with desks, computers, bookshelves, and small cabinets. The program should prevent bookshelves from being placed against a door or a window, and we have built a simple spatial model to account for that. The model has then been translated into a VISUAL and a TEXTUAL language. The textual grammar was designed to support the generation of narrative texts. Descriptive texts could be generated by a subset of it.

We have then designed a number of spatial memory tests, performed by two groups of subjects. Each group went through a different sequence but not a different set of tests. Since both modes were generated from the same underlying model, we wanted to have indications about the following:

(I) Can the same model support equally good textual and visual interface modes in spatial configuration problems?

(II) Should there be different models to support the alternative interface modes?

These two questions portray the issues we have presented in section 2. As has been seen, there is considerable theoretical motivation to say that the answer to (I) is "no" and the answer to (II) is "yes". But, our purpose in this section is to evaluate the extent of both answers, so that some measure of the actual costs of multimodality can be obtained. Moreover, the class of multimodality we are dealing with (inclusive multimodes) is really optional (though arguably desirable), since in theory natural language could precisely describe layout states and processes (even if at the expense of great cognitive complexity). Certainly, if we had cases of intersectional or exclusive multimodes, it would not be really a matter of choosing multimodality, but of using it as the only reasonable design alternative for the interface.

In order to build a model of a typical office, we have started by structuring the room space to be furnished in 9 sequentially numbered square regions, as shown in Figure 3. The office door is located in region 7, and the window extends along two adjacent regions 6 and 9. Another distinctive region number 5 stands for the geometric center of the room, and all other regions are adjacent to walls only.

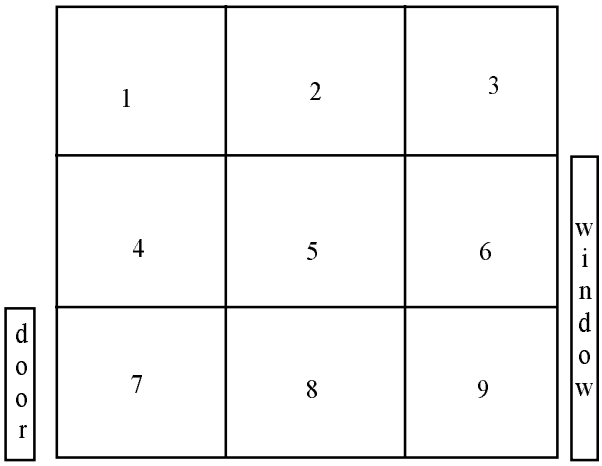


Figure 3: Spatial structuring of the room

This structuring, regardless of any claims about its being the best fit for the problem, is a convenient one, since all the constraints about object locations can be easily stated in terms of such regions. For example, desks can be placed anywhere except in region 7; bookshelves can be placed only against walls (i.e. not in regions 5 and 6); and so on. Moreover, the attributes of each region allow for inferences as to the location of furniture against perpendicular walls (e.g. the only available wall for a bookshelf in region 9 is that perpendicular to the window). Objects have been characterized by their physical constraints relevant for the task. Thus, computers have been constrained to be placed only on top of desks, cabinets and bookshelves have never been allowed to be put against the same wall, nor have been cabinets and desks. Bookshelves, however, have been allowed to be placed above desks against the wall.

As shown in Figure 4, the direct manipulation interface represented space schematically in 2D, with a top view of the office and the furniture. Objects have been visually abstracted in terms of quadrilateral shapes equivalent to their top surface. Therefore, computers have been represented as squares whose surface is approximately 1/4 of the desk surface (which is also a square, for sake of distinction from cabinets and bookshelves). Cabinets and bookshelves, on their turn, have been represented by the same rectangular surface and distinguished by spectrally distant colors (dark green and bright yellow).

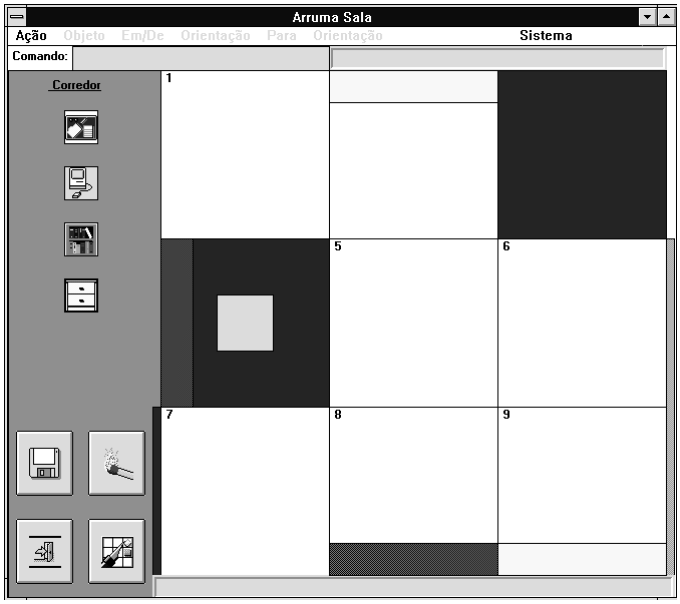


Figure 4: Screen Dump of the Layout interface

So, users could furnish the office space by drag drop operations performed on the objects available on the screen. None of the subjects has reported any difficulty to understand what the program did, what the visual objects stood for, and how they were manipulated. During the tests, the representation of layout steps was registered in the underlying MARS, which captured the sequential index of the action, the operation performed, the object involved, and the direction of the action. For instance, if a user placed a desk in region 1 and a computer on top of it, the underlying representation would be roughly the following:

STEP 1	MOVE	DESK-1	from(OUTSIDE) to(REGION1)
STEP 2	MOVE	COMPUTER-1	from(OUTSIDE) to(REGION1)

Note that the recorded history does not include the fact that the computer is on top of the desk. This piece of knowledge is used to evaluate a user's attempt to place a computer on top of a cabinet, for example. Because this action cannot succeed at the interface level, it is never achieved as a valid step in the configuration process. Therefore, such knowledge is absent from the task history. Previous work on the influence of alternative interface modes upon the users problem-solving strategies [Svendsen'91] has shown that mechanisms such as this blocking of "nonsense" configurations in direct manipulation interfaces actually encourage more immediate (i.e. less pondered) trial-and-error behavior, compared to linguistic interfaces. In the latter, users apparently "think more" before they command actions, and thus they make less mistakes.

Also, there is an inferential tradeoff in the history representation if redundancies are to be eliminated. With only 2 statements, we can either choose to say that both the desk and the computer are in region 1, or say that the desk is in region 1 and that the computer is on top of it. The cost of the latter, compared to the former, is the maintenance of inter-object references in the layout process history. As it can be noticed, the adopted region-based strategy

works with a fixed-frame spatial reference MARS which provides for a very simple and effective history retrieval function.

Another noteworthy aspect of the implemented matrix-based model above is that it is clearly optimized for computer processing of space knowledge; not for human. For example, the gain in representation and understanding if the model could directly express such relations as "in front of", "across", "next to", and so on, would be very large. But, strictly speaking, despite the greater sophistication and interpretability for humans, the actual gain in "effectiveness" because of the enriched model could be null — i.e. the set of layout mistakes a user can do in the poorly-expressive model is the same as in the highly-expressive model. Moreover, it could be argued that there are losses in "efficiency" brought forth by the enriched model, since with more entities and relations to compute upon, execution takes more time. These considerations often play a leading role in the selection of implementation strategies for semantic models, and keeping them in mind is important for the rest of our analysis.

So, if users want to playback all their steps in solving a layout configuration problem, it suffices to input the backlog file to the system, and the subsequent frame animation will show all the users actions. This feature certainly supports the design rationale documentation process, but it also imposes some cognitive loads on users. To mention the most obvious one, this playback strategy conceals some abstract concepts a natural language report of the process could directly reveal. For instance, the use of the word RETURN to signal that a certain piece of furniture is being moved back to a place where it had been before has no equivalent in the frame animation. The same is true for such other concepts and words as EXCHANGE (or SWAP) and CENTER. This observation is a natural counterpart of Svendsen's point about users more rational behavior in linguistic interfaces. Because the language is more powerful to express such global abstract patterns, as well as lower-level components, the user's perspective about the problem is automatically expanded to account for time periods that are longer than one frame (e.g. RETURN, after "n" frames), or space ranges that refer to more than one region (e.g. the CENTER of the room).

We have started our experiment by trying to use exactly the same backlog file (an implementation of MARS) as input to the animation sequence and to a text generation system. The text generator used a Brazilian Portuguese grammar and dictionary in which notions such as RETURN, SWAP or EXCHANGE and CENTER, could appear in a narrative text about the problem-solving history. However, the larger grain in both time and space dimensions implied abstractions of typical LARS. For example, the path a certain object followed before "returning" to its original space could be left undetermined in narrative text. This was not true of the "playback" mode — a typical MARS. So, minor extensions to MARS have been made.

Whereas in the visual interface mode, the animated representation of subsequent state sequences has been successfully rendered, in the narrative text a number of problems have emerged. Two of these have been selected to demonstrate the costs of having multimodal interfaces: the spatial reference problem and the higher-order abstract move pattern problem.

The spatial reference problem amounts to the low-quality text provided by a fixed-frame reference system which was computationally attractive to reproduce the configuration history visually. This system only supported text as that in (a), whereas a much more desirable rendition of it would clearly be that shown in (b).

(a) "Pedro colocou a primeira mesa na regiªo 1 e a segunda mesa na regiªo 3. Em seguida, ele posicionou o primeiro armÆrio na regiªo 2." [Automatic Generation which translates into: "Peter placed the first desk in region 1 and the second desk in region 3. Next, he put the first cabinet in region 2."]

(b) "Pedro colocou a primeira mesa na regiªo 1 e a segunda mesa em frente a ela, contra a parede oposta. Em seguida, ele posicionou o primeiro armÆrio entre as duas mesas." [Manual Generation which translates into: "Peter placed the first desk in region 1 and the second desk in front of it, across the room. Next, he put the first cabinet between the two desks."]

The major difference between (a) and (b) is the object-oriented dynamic nature of the reference system used in (b) for the realization of adequate spatial expressions. For (b) to be automatically generated, the original fixed-frame system must be replaced by or translated into an object-oriented, hierarchical and dynamic system in which the possibility of inter-object spatial references may override the office space fixed references.

The higher-order abstract move pattern problem amounts to the low-quality of text provided by a flat representation of state sequences (as in (c) below) as compared to articulated representations of different state change patterns (as in (d) below). Such notions as RETURN (or GO BACK), for example, are the result of computations performed upon the whole history (or larger portions) of the problem-solving steps, in which the existence of two statements saying that OBJECT X MOVED to(LOCATION Y) actually MEAN that the object was returned to a previous position. We have in fact added this feature to the text generation system and have been able to evaluate the considerable computational cost it adds to a simple readout of the subsequent historical steps: a dedicated knowledge base for the textual interface mode.

(c) "Pedro moveu a segunda mesa da regiªo 9 para a regiªo 3.[...] A seguir, ele moveu a segunda mesa da regiªo 3 para a regiªo 9." [Automatic Generation which translates into: "Peter moved the second desk from region 9 to region 3. [...] Then, he moved the second desk from region 3 to region 9."]

(d) "Pedro moveu a segunda mesa da regiªo 9 para a regiªo 3.[...] A seguir, ele retornou a segunda mesa para a regiªo 9." [Automatic Generation which translates into: "Peter moved the second desk from region 9 to region 3. [...] Then, he returned the second desk to region 9."]

We can conclude that, no matter the quality or sophistication of the historical knowledge used in our experiments, "ideal" renditions of visual and textual reports of a user's problem-solving steps cannot be derived from the same knowledge representation language. On the one hand, if the visual interface is considered per se, very simple

coding can be used to reproduce the exact animated sequence equivalent to the original performance. However, text directly derived from it, without additional spatial inferences, has very poor quality. On the other hand, if the textual interface is considered per se, certain levels of articulation in the knowledge representation language (e.g. dynamic object-oriented referencing and higher-order abstract move patterning) will have no perceptible effect upon the animated sequence of problem-solving steps. This would characterize a certain waste of expressiveness from the point of view of the visual language.

Stating the problem in cognitive terms, this amounts to the fact that in our case automatically-generated NL text is based on LARS, and that playback animation of successive layout snapshots is based on MARS. A good example of this is when the user SWAPS two objects. Because swapping is achieved in more than one step (i.e. the user selects the first object, places it somewhere else, brings the second object to the vacant position formerly held by the first object, and then finally places the first object in the second object's former position), a playback may wrongly induce users to think that the actual position held temporarily by one of the objects ("somewhere else", in our natural language phrasing) is of some importance. This misconception can only be avoided by such abstractions as those concealed under the word "swap". Additionally, since what Kirsh and Maglio have called "epistemic" actions cannot be distinguished from final actions in MARS, although they could be signalled by certain abstractions in LARS, the latter must be actually used when to focus users on the relevant aspects of a task is a desirable or necessary feature.

In order to gauge the effects of a suboptimal fixed-frame referencing system in a multimodal environment, we have carried out tests exploring static and dynamic reconstitution of layout configuration states. By comparing results obtained by subjects with visual and textual input, we tried to have **indications** about how inadequate underlying knowledge representation systems affect interface quality. We looked for answers to two basic questions:

> (III) Is any of the current output modes sufficient to account for both a layout description of configuration states and a report of the layout process and history?
>
> (IV) Is any of the output modes best fit for one retrieval task (state/process information) than the other?

Questions (III) and (IV) should also indicate the class of multimodality that should be designed for the application. As it has been designed for the test, the application supported inclusive multimodes. However, if more abstractions (and more expressiveness) were added to both modes (graphic and textual), so that, for instance, both modes turned into legitimate LARS, intersectional multimodes could be found to be the best design option for multimodal layout interfaces.

4. Indications of Small-Scale Experiments

We have started our experiment by designing a layout process representation that would give evidences of relevant abstractions for contrasting the expressiveness of MARS-based and LARS-based codes. So, we first configured an arbitrary initial layout state, and to this we made special subsequent changes until we eventually reached a final state. The whole process was recorded in a backlog file, in the format illustrated in section 3. From this file we automatically generated a narrative text, in Brazilian Portuguese [Prates and dos Santos'94] and an animated playback sequence. A descriptive text in the same language was also generated to represent the final layout picture in the last frame of the animation.

Two groups of 4 computer-literate subjects have made a set of 6 tests each: (a) memorizing and reproducing a layout state described in NL text; (b) memorizing and reproducing the exact steps of a layout process reported in NL text; (c) reproducing a layout process described in NL text permanently accessible for reading; (d) reproducing and transforming a layout state into another, both represented on a screen permanently accessible for viewing; (e) memorizing and reproducing a layout state represented on a screen; (f) memorizing and reproducing the exact steps of a layout process represented by an animated sequence of screens.

Since our main interest was to assess how difficult it was for subjects to memorize textual and visual representations of states and processes, tests (c) and (d) were dummy tests. They were introduced just as a buffer between the other two sets of tests, because the configurations in textual and visual representations were exactly the same, and one of the interesting observations would be to check how many subjects realized they were doing the same thing again, but in another code.

Groups performed the tests in different order. Group TV performed first the textual memorization tests and then the visual ones, whereas group VT performed the tests in reversed order. Subjects were introduced to the interface, and were left free to experiment for a while until they thought they knew how to operate with the mouse. Each memorization test was carried out by first allowing subjects to read or playback the material at their own pace; then, the material was taken away from them and the experiment began. Because dummy tests did not require memorization, the material for these was not taken away from subjects.

The whole experiment was not meant to be used to demonstrate or prove a point, but rather to raise a point. In other words, we did not want to use out data and observations to propose a cost function to calculate implementation overheads or to make claims for and against multimodality. What we aimed at was to use the experiment as an exploratory net, in which we expected to catch insights and indications about important issues in multimodal interface design. Thus, records of the subjects performance included all steps in each test, the time it took to achieve it, and observations made by testers during the experiment. Analysis of results revealed that time information did not actually contribute to identify relevant performance distinctions among subjects and groups. However, the number and type of mistakes, along with some observed patterns of

behavior have provided valuable resources for a cost/benefit analysis.

The visual snapshots and sequences will not be shown here, but they were all variations on what is shown in Figure 4, above. The computer-generated descriptive text was a Portuguese translation of the following:

> *Peter rearranged his office. When he was finished, furniture was placed like this:*
>
> *In region 2 there was a cabinet. In region 3, a desk. In region 4 were a desk, a computer and a bookshelf. And finally, there was a bookshelf in region 8 and a cabinet in region 9.*

And the computer-generated narrative text was a Portuguese translation of the following:

> *Peter decided to rearrange his office. It was a hard task to decide on the final location of each piece of furniture. Many attempts were made.*
>
> *Initially, Peter placed the first desk and the first computer in region 1. He then placed the second desk in region 9, the first bookshelf in region 2 and the second bookshelf in region 8. Then he placed the first cabinet in region 4.*
>
> *After that, Peter moved the second desk from region 9 to region 3. Then, he returned the second desk to region 9. Afterwards, he took out the first bookshelf from region 2.*
>
> *At last, Peter finished the task.*
>
> *Yes, at last!*

Ordinal references to pieces of furniture (e.g. the first desk) were made because there was a fixed set of only two pieces of each kind that could possibly be used in the layout.

Five classes of mistakes have been identified. (i) Users were able to remember an object, but they were mistaken about its position. (ii) Users were able to remember a position, but they placed the wrong object in it. (iii) Users introduced spurious objects in spurious positions. (iv) Users omitted an object from the layout. (v) Users were able to recall a correct step in the process, but they performed it in the wrong order.

Class (v) evidently occurred only in process memorization tests and is somewhat orthogonal to the others. Nevertheless, it is of paramount importance in memorizing processes. Mistakes were not equally serious. For instance, the omission of an object is more serious than mistakes about its position. We have considered that in layout tasks, optimal memorization included retrieval of a triple —— {object, position, sequential order of action}. A decayed performance would imply cumulative losses of entities and relations present in the ideal model. Whenever object and position were properly retrieved from memory, the final model staticly reproduced the ideal. But, a mistake in the order of steps is relevant for process memorization,

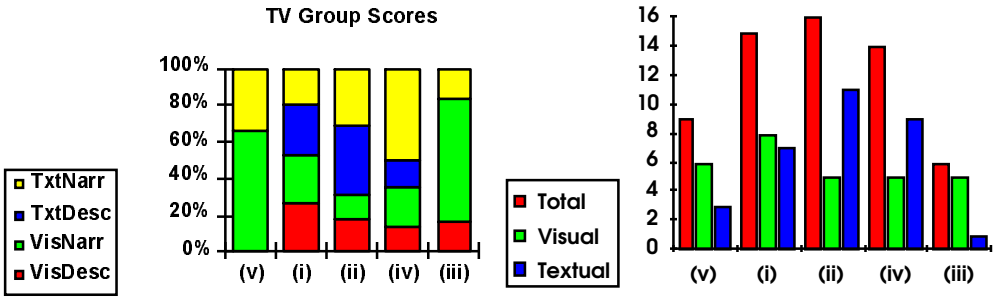
although not for final configuration. Within the remaining set of mistakes, losses were increasingly serious if they involved missing an object s position, recalling that a position was taken but missing the correct object for it, and forgetting an object (and therefore its position). A serious mistake was to introduce objects which did not belong to the test model. The following tables and charts show various contrasts of performance between the two groups. Tables range mistakes in ascending order.

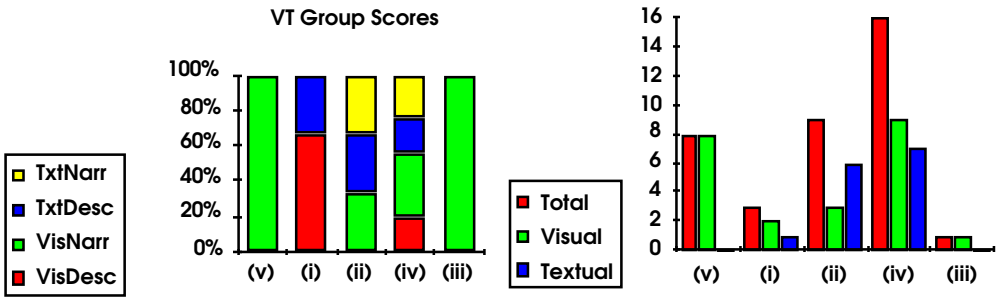
TV Group

	Test	+Act/ −Seq	+Obj/ −Pos	−Obj/ +Pos	Omit	−Obj/ −Pos	# F	# S
Sub-Total	Txt Desc	0	4	6	2	0	12	16
Sub-Total	Txt Narr	3	3	5	7	1	19	17
Sub-Total	Vis Desc	0	4	3	2	1	10	19
Sub-Total	Vis Narr	6	4	2	3	4	19	19
Total		9	15	16	14	6	60	71

VT Group

	Test	+Act/ −Seq	+Obj/ −Pos	−Obj/ +Pos	Omit	−Obj/ −Pos	# F	# T
Sub Total	Vis Desc	0	2	0	3	0	5	23
Sub Total	Vis Narr	8	0	3	6	1	18	19
Sub Total	Txt Desc	0	1	3	3	0	7	21
Sub Total	Txt Narr	0	0	3	4	0	7	29
Total		8	3	9	16	1	37	92

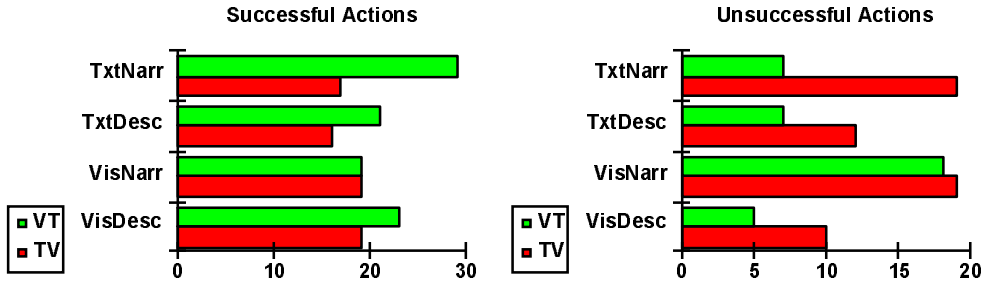


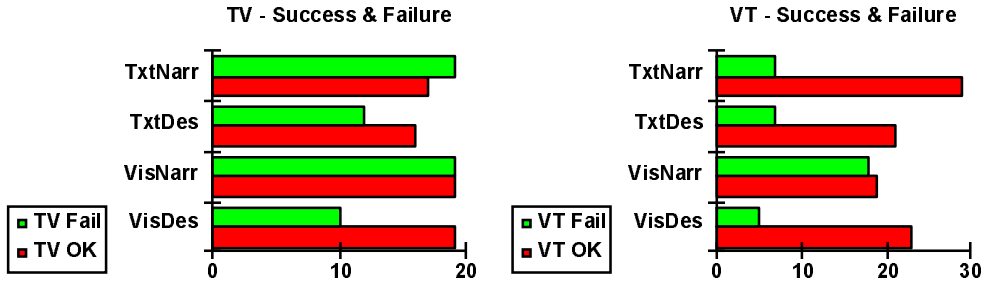


Another interesting contrast is that between the number of successful and unsuccessful actions in each group's global performance. This is shown in the table below.

TV and VT Group's Global Performance Scores

Group	VisDesc OK	VisNarr OK	TxtDesc OK	TxtNarr OK
TV	19	19	16	17
VT	23	19	21	29
Group	VisDesc Fail	VisNarr Fail	TxtDesc OK	TxtNarr Fail
TV	10	19	12	19
VT	5	18	7	7





All the tables and charts above have been analyzed in terms of large-grain distinctions. Scores differing from each other in the range of ± 3 (absolute score values for each type of mistake) have not been interpreted as indications. Nevertheless, interesting facts emerged.

The TV Group showed a homogenous pattern of success across the four tests. However, the group's pattern of failure showed a pronounced difficulty in Narrative Tests, in both Textual and Visual Modes.

Moreover, the best Success/Failure ratio in this group was found in the Visual Description Test; the second best was in the Textual Description.

The VT Group, however, showed a more heterogeneous behavior. Best results were found in the Textual Narrative Test (29 successful retrievals), against the lowest score in the Visual Narrative Test (19 retrievals). It is quite interesting to see that in this group, the best and worst performances were in the Narrative Tests, which are indeed much more difficult than the Descriptive ones, and that the best were obtained with the Textual Mode and the worst with the Visual Mode. Also, this group had much better success/fail ratios than the TV Group. With the exception of the Visual Narrative Test, where both groups had 50% of success in the test, the VT Group had a pronouncedly superior performance than the TV Group.

So far, we have two important indications:

Using a Visual Mode to represent layout process history causes cognitive difficulties for users; and

Reasoning made from a visual representation of the layout model seems to benefit subsequent reasoning of the same kind made from a textual representation, but benefits from textual representations to reasoning about visual representations do not show evidence of the same contribution.

An observation of the distribution of mistakes among classes (i) through (v) in both groups suggests that the relatively large number of mistakes of type (ii) (placing one object instead of another in a given position of the model) may have been strongly influenced by the fact that cabinets and bookshelves had the same shape, and were distinguished only by color attributes.

The TV Group had more problems in the Visual Narrative Test than in its Textual counterpart (actually, twice as many occurrences). This group also had much more difficulties to assign the right objects to the model's occupied positions in the Textual Mode than in the Visual Mode, in both tests (again, practically twice as many occurrences). Type (iv) mistakes in this group were pronouncedly more frequent in the Textual Narrative Test, than in the other three tests. There have also been relatively many occurrences of spurious introductions of objects in the Visual Narrative Test, especially when compared to single isolated occurrences of (iii) in the other three tests.

The VT Group, however, had almost no problems with the Textual Narrative Test. No mistakes in the sequence of actions were noticed, no objects belonging to the system's model were placed in the wrong position, and the only few occurrences of mistakes (7, against 19 in the other group) were 4 omissions and 3 wrong objects in the model's marked positions. Such performance presented a very strong contrast with the group's performance in the Visual Narrative Test (18 failures, against 7) with pronounced frequencies of mistakes in the order of actions and omissions.

From these other observations, two more indications emerged:

The fixed-frame model from which positions were realized in NL text apparently caused unexpected difficulties in the processing of narrative texts; and

The order of events is easier to retrieve from textual representations than from visual representation, and thus favors Textual Modes instead of Visual Modes for process retrieval information.

Thus, questions (III) and (IV) at the end of section 3 have already an answer. The Textual Mode is clearly superior to the Visual one for the retrieval of layout process steps. However, neither modes are sufficient to account for the cognitive needs of users in layout documentation and memorization tasks. Moreover, questions (I) and (II) may also have found an answer, although maybe not as good as the previous ones. The same underlying model does not seem to fit equally well the needs of high-quality multimodes in our experiment, since automatically generated texts are not good in Portuguese. Thus, at least some improvement would be needed to meet the quality needs of the interface and this could only increase the overhead from the apparently sufficient spatial model for the Visual Mode.

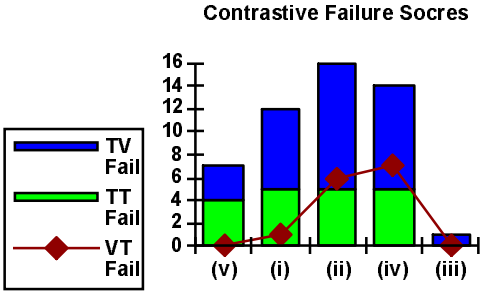
We have asked ten other subjects, with the same background as those who participated in the experiment, to freely write Brazilian Portuguese texts describing and narrating the configurations used in the tests. In absolutely all texts the reference system used was not fixed: As expected, all subjects used a dynamic inter-object reference system to describe and narrate how objects were placed in the office room. Other interesting features of such texts were a more extensive use of the door and window as landmarks in the space to be furnished and a sophisticated perspective of the observers, who often viewed themselves inside the office and reported what they could see from where they were. Additionally, as proposed by existing research about the indeterminacy problem in spatial reasoning from NL inputs [Ioerger'94], subjects proposed descriptions and narratives that did not translate into one and only semantic model — a MARS. All texts were at least evidences of LARS, if not UARS (since some subjects clearly changed referring strategies within the same span of text).

One last experiment about the implementation costs of good-quality multimodes was then made, in which we manually generated NL texts with one improved feature — a dynamic object-oriented reference system for object locations, instead of the fixed frame used in the first experiment. Thus, the narrative and descriptive texts included expressions like: "Peter placed the first desk in region 1 and a computer on top of it", or "he placed a bookshelf in region 2 and another against the opposing wall, across the room". In other words, we made an extensive use of anaphoric expressions, but all within a LARS model. Some solutions, however, were still not present in our text, which made use of the underlying grid and referred to positions in terms of the "regions".

We selected another group of four subjects, not present in any of the previous experiments, and asked them to make only two tests: Textual Description and Textual Narrative. Here are the results they had.

TT Group

	Test	+Act/ -Seq	+Obj/ -Pos	-Obj/ +Pos	Omit	-Obj/ -Pos
Sub Total	Txt Desc	0	1	4	3	0
Sub Total	Txt Narr	4	4	1	2	0
Total		4	5	5	5	0



As can be seen, this new Group, which we have coined the TT Group, has been virtually as successful as the TV Group regarding mistakes of type (v), and has been systematically more successful than this group in all other cases. Also, the TT Group has equaled the VT Group performance with types (ii), (iv) and (iii), even without having benefited from previous memory tests from visual representations.

The indications from TT Group scores is that the improvement of text quality actually impacted performance in a positive way. Nevertheless, further improvements could still be introduced in the textual input, so that subjects had even less cognitive loads than they did in this last test. The important conclusion is that such improvements really cost a high price for the implemented semantic models. The NL Mode requires a sophisticated reasoner and anaphoric-reference generator that is not essential for the semantic model of the Visual Mode, as seen from VT Group scores. So, we drew interesting conclusions from the experiments which are presented and commented in the following section.

5. Concluding Remarks

This paper has addressed two major interrelated issues faced by Multimodal Interface Design — the class of multimodality available to computer-human interaction and the implementation costs incurred by multimodal design options. Our intent is not to give the final word about which modes should or should not be selected in such interfaces, but rather to point at some evidences provided by both theoretically-based analysis and empirical observation of incremental design implementations.

We have started our study by identifying four different classes of multimodality. The conceptual framework for our taxonomy has been a set-theoretic approach to the semantic coverage of alternative codes, like textual and visual for instance. The classes ranged from

optional cases of multimodal design to mandatory cases. Optional multimodality is one in which multiple modes at the user interface level are virtually equivalent to each other. They do not complement each other's shortcomings in expressing the finesses of a semantic model and are, therefore, offered for sake of satisfying users taste or ability criteria. However, we have remarked that in theory it is quite rare to have two modes that are exactly equivalent, since each one triggers a set of interpretants and associations that is very seldom the same as the set triggered by the other.

Interesting cases of multimodality arose from different set relations, like inclusion, exclusion, and intersection. Inclusive multimodality is that in which one of the modes is able to say all that the other says, and more. The justification for having more than one mode in such situation is that the less expressive mode helps designers predict the focus and spectrum of alternatives a user has when using that mode. We have provided connections with recent research in Cognitive Science, and shown that this case is actually related to limiting abstractions for efficient processibility [Stenning and Oberlander'95]. In it, authors have proposed a gradient of abstract representation systems parallel to Chomsky's hierarchy of formal languages [Chomsky'59] and proposed that specificity helps users reason when the level of the content's abstractions is high. So, we have applied their model to our framework and recast the issues about multimodality costs in terms of building one of Stenning and Oberlander's representation systems: MARS, LARS or UARS.

We have proposed that Inclusive Multimodes typically involved MARS included in LARS, since UARS are computationally complex and involve metalinguistic modeling. It should be noticed that Intelligent Multimedia Interfaces [Maybury'93] could involve the three concurrent models in the semantic component of the interface. Previous research has pointed out at important knowledge representation issues for multimedia presentations [Arens, Hovy and Vossers'93], which are related to the ones discussed here.

An immediate observation from a theoretic point of view, where abstract representation systems were taken as the basis for the semantic component of a code, is that Inclusive Multimodes where MARS are encompassed by LARS always impose an important implementation overhead to design. The full expressiveness of the LARS-based code is necessarily warranted by a more costly model, in computational terms, than a MARS. In other words, whenever interface languages are found in different points of Chomsky's Hierarchy, the languages interpreter must be designed for the most expressive one. Alternatively, each language may have its own interpreter, but then one must translate into the other.

We have provided examples from an experiment we have done, trying to use the same basic MARS model as the semantic source for a Visual and a Textual Mode in a Multimodal Interface for a Layout Application. Results have strongly suggested that this sort of application clearly benefits from multimodality, and that the Textual Mode is better to represent evolutionary spatial models (like layout process and history) than the Visual Mode. Combined with each other, Visual and Textual Modes can lead to high-standard performance in memorization tests about layout configurations. These results confirm points previously raised by empirical researchers [Krause'93; Neilson and

Lee'94]. In particular, the level of precision and specificity of NL descriptions of spatial relations has been remarked to be very low [Neilsen and Lee'94; Ioerger'94]. But, we have shown that this is so because of higher-order abstractions on larger grain patterns of observation, whose expression in visual layout interfaces is unwieldy. By the same token, the full specification of objects precise locations in space via NL text imposes certain burdens of information processing that, in its turn, are also unwieldy for users.

The case of Inclusive Textual-Visual Multimodes has also provided examples of how the LARS-based (or the more expressive) code can cooperate to give users only the relevant information [Grice'75]. We have observed during experimental tests that users interacting with visual representations did not immediately realize (if the ever realized at all) that some movements of objects across space were only meant to enable a SWAPPING, for instance. They often interpreted the intermediary position an object occupied during a SWAPPING operation as an intentional positioning of the layout agent — which is not adequate. Because the visual code cannot distinguish intentional moves from pivotal ones, users may easily be led off the track of relevance. The same is not true of NL renditions of the model, since such words as 'swap' directly access a higher-level abstraction pattern on movements and guides users attention to the point being made. The same mechanism in NL can distinguish what Kirsh and Maglio called "epistemic" actions from top-level goal-driven actions.

Our taxonomy of multimodality also provided us with two other instances of multicode relations. One is Intersectional Multimodes, which we defined as a case in which two or more modes intersect with, but do not include, each other in semantic terms. This is a very powerful case of multimodality because the intersected area of the various modes can demonstrate to users the expressive distinctions between modes. In other words, if Mode A and Mode B can both express the same content C, and they do it in different ways, whenever users choose one or the other to manipulate and compute meanings they gain understanding about the potential of each mode, and make inferences about their respective ranges of expressiveness. Thus, this type of multimodality is particularly welcome to maximize users understanding about the communicative codes used by system's designers to convey their global usability message to users.

A very appealing application of Intersectional Multimodality in Layout Systems is the use of NL text and 3D direct manipulation. As mentioned above, precise information about an object's position in 3D rendered in natural language is so difficult that processible narrative text of layout history can hardly be written (e.g. "He placed the desk 4 inches apart from the cabinet that faced the bookshelf and was 4 inches away from the wall behind it... "). Thus, precise positioning could be easily expressed in 3D direct manipulation and recovered in listings.

In the case of Exclusive Multimodes, alternative modes do not have any semantic intersection with one another. This is clearly difficult for users to manage, since by switching from one mode to the next, they are actually changing from one world to the next. Cross-references between modes and worlds are impossible at the interface

level, and thus the cognitive load for users is considerable. Unlike Inclusive and Intersectional Multimodes, Exclusive Multimodes are costly for both machine and human, thus characterizing a last-resource design choice for computer-human interaction.

Our study has been guided by theoretic threads borrowed form Linguistics, Semiotics, Theory of Computing, Software Engineering and CHI. Of course the experiment was small, and we have been careful to avoid overstatements about the issues involved based solely on empirical data. This paper reports our first step in applying a wider semiotic perspective to Multimodal Interface Design. A previous attempt at applying it to CHI in general resulted in what we've called the Semiotic Engineering of User Interface Languages [de Souza'93]. There, we treated the problem in a One-Mode environment and draw a number of conclusions about how such languages should be designed. The interesting improvement of this paper compared to its predecessor is that in Multimodal environments, the perceived interactions among modes may considerably increase the chances of users actually getting the whole of the designers message. Thus, multimodality may serve a novel semiotic goal teaching users about computer language expressiveness and limitations, which is ever more difficult with just one communicative mode available.

6. References

[Adler and Winograd'92] Adler, P.S. and Winograd, T.A. (1992). *Usability: Turning Technology into Tools*. New York: Oxford University Press.

[Aho and Ullman'72] Aho, A.V. and Ullman, J.D. (1972). *The theory of parsing, translation, and compiling*. Englewoods Cliffs, NJ: Prentice-Hall.

[Arens, Hovy and Vossers] Arens, Y., Hovy, E.H. and Vossers, M. (1993). On the Knowledge Underlying Multimedia Presentations. In M.T. Maybury, Ed. *Intelligent Multimedia Interfaces*. AAAI Press, Menlo Park, pp. 328-354.

[Chomsky'59] Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control*, **2**, pp. 137-167.

[de Souza'93] de Souza, C.S. (1993). The semiotic engineering of user interface languages. *International Journal of Man-Machine Studies*, **39**, pp. 753-773.

[Grice'75] Grice, H.P. (1975). Logic and conversation. In P. Cole and J.L. Morgan, Eds. *Syntax and semantics: 3. Speech acts.* New York: Academic Press.

[Ioerger'94] Ioerger, T.R. (1994). The Manipulation of Images to Handle Indeterminacy in Spatial Reasoning. *Cognitive Science*, **18**, pp. 551-593.

[Kirsh and Maglio'94] Kirsch, D. and Maglio, P. (1994). On Distinguishing Epistemic from Pragmatic Action. *Cognitive Science*, **18**, pp. 513-549.

[Krause'93] Krause, J. (1993). A Multilayered Empirical Approach to Multimodality: Towards Mixed Solutions of Natural Language and Graphical Interfaces. In M.T. Maybury, Ed. *Intelligent Multimedia Interfaces*. AAAI Press, Menlo Park, pp. 328-354.

[Maybury'93] Maybury, M. (Ed.). (1993). *Intelligent Multimedia Interfaces*. AAAI Press, Menlo Park, pp. 328-354.

[Neilsen and Lee'94] Neilsen, I. and Lee, J. (1994). Conversations with graphics: implications for the design of natural language/graphics interfaces. *International Journal of Human-Computer Interaction*, **40**, pp. 509-541.

[Peirce'31] Peirce, C.S. (1931-1958). *Collected Papers*. Cambridge, MA: Harvard University Press.

[Prates and dos Santos'94] Prates, R.O. and dos Santos, A.L.S.C. (1994). Program developed for Final Project for Interactive Computational Linguistic Course. Departamento de Informática, PUC-Rio.

[Stenning and Oberlander'95] Stenning, K. and Oberlander, J. (1995). A Cognitive Theory of Graphical and Linguistic Reasoning: Logic and Implementation. *Cognitive Science*, **19**, pp. 97-140.

[Svendsen'91] Svendsen, G.B. (1991). The influence of interface style on problem solving. *International Journal of Man-Machine Studies,* **35**, 379-397.