

SRI International

Technote 514 • November, 1991

The Role of Natural Language in a Multimodal Interface*

Prepared by:

**Philip R. Cohen
Senior Computer Scientist**

**Computer Dialogue Laboratory
Artificial Intelligence Center
Computing and Engineering Sciences Division**

Approved for Unlimited Distribution

* This paper is an invited address to the '91 FRIEND21 International Symposium on Next Generation Human Interface, Tokyo, Japan, 1991

The Role of Natural Language in a Multimodal Interface*

Philip R. Cohen
Senior Computer Scientist
Computer Dialogue Laboratory
Artificial Intelligence Center
SRI International

1 Abstract

Although graphics and direct manipulation are effective interface technologies for some classes of problems, they are limited in many ways. In particular, they provide little support for identifying objects not on the screen, for specifying temporal relations, for identifying and operating on large sets and subsets of entities, and for using the context of interaction. On the other hand, these are precisely strengths of natural language. This paper presents an interface that blends natural language processing and direct manipulation technologies, using each for their characteristic advantages. Specifically, the paper shows how to use natural language to describe objects and temporal relations, and how to use direct manipulation for overcoming hard natural language problems involving the establishment and use of context and pronominal reference. This work has been implemented in SRI's Shoptalk system, a prototype information and decision-support system for manufacturing.

2 Introduction

Stimulated in part by the accelerating developments in multimedia computing, researchers have been striving to develop multimodal interfaces. In this paper, I want to suggest that we should be concerned not just with building interfaces that make available two or more communication modalities, but with developing *integrated* interfaces, in which the modalities forge a productive synthesis. To build such interfaces in a more illuminating, and, it is hoped, more efficient, manner than just trial and error requires some guiding principles. This paper advocates one such principle, namely to *use the strengths of one modality to overcome weaknesses of another* and demonstrates an interface that conforms to it.

Before beginning, it is worth distinguishing the term “media” from “modality.” Although there is no consistent usage in the literature, the term “medium” is used here to focus on the production, storage, and transmission by the machine of signals, such as those recorded from video, imagery, sound, and handwriting. The term “modality” is used to concentrate on the syntactic, semantic, and pragmatic properties of the signal — in other words, on how the signal functions to communicate. Whereas the technology needed to incorporate new communication media need not involve any analysis of the signals, a discussion of human-computer communication modalities presupposes some medium, and inherently involves, at some level, the machine's determination of and response to the *content* of the message.

This paper is concerned only with two human-computer communication/interaction modalities — direct manipulation and natural language. To keep the discussion simple, the term “natural language” will be used here without regard for the transmission medium (e.g., keyboard, speech, handwriting) or the various modalities incorporating it, even though research has shown that spoken interaction and keyboard interaction differ in many ways [1, 3, 13, 16]. Still, those differences are not germane to the general points raised here about how to formulate and adhere to modality integration principles.

Furthermore, I do not attempt a precise distinction between graphical user interfaces (GUIs) that include pointing and menu selection, and direct manipulation interfaces (DMIs). DMIs rely on the graphical techniques

*This paper is a keynote address delivered to the '91 Friend21 International Symposium on Next Generation Human Interfaces, Tokyo, Japan, November, 1991.

supported by GUIs, but merely using a GUI does not guarantee one has built a DMI. I will try to be specific where possible, but not too precise, in that characterizations given here of DMIs may be true in virtue of general properties inherited from GUIs.

3 Modality Integration

3.1 Strengths and Weaknesses of Direct Manipulation

Many writers have identified numerous virtues of direct manipulation interfaces (e.g., [9, 18]). Among those virtues, it is argued that

- If well-designed and based on familiar metaphors that allow direct engagement with a semantic object, direct manipulation interfaces are intuitive.
- If well-designed, such interfaces can have a consistent “look and feel,” enabling users of one program to learn to another program quickly.
- The typical “select—act” style of GUIs is a natural form of interaction.¹
- The use of menus makes apparent the available options, thereby curtailing user errors in formulating commands and in specifying their arguments.

It is no exaggeration to say that GUIs and DMIs have been so successful that no serious computer company would attempt to sell a machine without them.

However, such interfaces do not suffice for all needs. One clear weakness is the paucity of means available for identifying entities. Merely allowing users to select currently displayed entities provides them little support for identifying objects not on the screen, for specifying temporal relations, for identifying and operating on large sets and subsets of entities, and for using the context of interaction. What is missing is a way for users to *describe* entities, by which I mean to use an expression in a language (natural or artificial) to *denote* or *pick out* an object, set, time period, and so forth.² At a minimum, a description language should include some way to find entities having a given set of properties, to say how many are of interest, to say which properties are definitely not of interest, and to supply temporal constraints on those properties. Moreover, a useful feature of a description language is the ability to reuse the results of previous descriptions. Some of these capabilities are found in formal query languages, and all (and more) are found in natural languages.

Finally, because of the emphasis given to rapid, graphical response to actions [18], the time of action in DMIs is tied closely to the time of action invocation. Although some systems can delay actions to specific future times, GUIs offer little support to users who want to execute actions at an unknown, but describable, future time. Without a means for describing interesting times, the users’ options in dealing with delayed actions are severely limited.

3.2 Strengths and Weaknesses of Natural Language

English, or any other natural language, provides a set of finely honed descriptive tools such as the use of noun phrases for identifying objects, verb phrases for identifying events, and tense and aspect for describing time periods. By the very nature of sentences, these capabilities are deployed simultaneously, as sentences must be about something, and most often describe events situated in time.

Coupled with this ability to describe entities, natural languages offer the ability to avoid extensive redescription through the use of pronouns and other anaphoric expressions. Such expressions are usually intended to denote the same entities as earlier ones, and the listener/reader is intended to infer the connection. Thus, the use of anaphora provides an “economical” benefit to the speaker, at the expense of the listener’s having to draw inferences.

However, still other costs are involved when natural language is incorporated into an interface. Pure natural language systems suffer from an opacity of linguistic and conceptual coverage, in that the user knows the system

¹We conjecture that this is so partly because human spoken dialogues often take an analogous form — having separate speech acts to identify the intended referents, and subsequent speech acts to request actions to be performed upon them [3, 13].

²Of course, the elimination of descriptions was a conscious design decision.

cannot interpret every utterance, but does not know precisely what it can interpret. Often, multiple attempts must be made to pose a query or command that the system can interpret correctly. Thus, such systems are error-prone and, some claim [17], lead to frustration and disillusionment. Moreover, many natural language sentences are ambiguous, and parsers are adept at finding more ambiguities than people do. Hence, a natural language system often engages in some form of clarification or confirmation subdialogue to ascertain if its interpretation is in fact the intended one.

Still another disadvantage is that reference resolution algorithms do not always supply the correct answer, in part because systems have underdeveloped knowledge bases, and in part because the system has little access to the discourse situation the user finds himself in, even if the system's prior utterances and graphical presentations have created that discourse situation. To complicate matters, systems currently have difficulty following the shifts in context inherent in dialogue. These contextual and world knowledge limitations undermine the search for (co)referents of anaphoric expressions, and provide another reason that natural language systems are usually designed to confirm their interpretations.

In summary, these two modalities have complementary advantages and disadvantages, which are summarized in Figure 1.³

	Direct Manipulation	Natural Language
Strengths	<ol style="list-style-type: none"> 1. Intuitive 2. Consistent Look and Feel 3. Options Apparent 4. Fail Safe 5. Feedback 6. Point, Act 7. "Direct Engagement" with semantic object 8. Acting in "here and now" 	<ol style="list-style-type: none"> 1. Intuitive 2. Description, including: <ol style="list-style-type: none"> a. Quantification b. Negation c. Temporal Information 3. Context 4. Anaphora (e.g., pronouns) 5. Delayed action possible
Weaknesses	<ol style="list-style-type: none"> 1. Description, including: <ol style="list-style-type: none"> a. Quantification b. Negation c. Temporal Information 2. Anaphora 3. Operations on large sets of objects 4. Delayed actions difficult 	<ol style="list-style-type: none"> 1. Coverage is opaque 2. "Overkill" for short or frequent queries 3. Difficulty of establishing and navigating context 4. Anaphora is problematic 5. Error prone 6. Ambiguous

Figure 1: Direct manipulation and natural language are complementary interface technologies

The remainder of this paper illustrates how to build an interface that compensates for the aforementioned weaknesses of one interface technology via the strengths of the other. The discussion involves examples drawn from the Shoptalk system at SRI International.

4 Background: Shoptalk

Shoptalk is a prototype manufacturing information and decision-support system developed at SRI International to help factory personnel perform tasks such as quality assurance monitoring, work-in-progress tracking, and production scheduling. The system allows users to query databases on the current state and recent history of the factory with a combination of English and graphical interaction techniques, and to examine alternative factory scenarios by running a discrete event simulator. Shoptalk features an integrated interface that permits intermixing

³To facilitate reference to the table entries, an entity will be encoded by its axes and numbers. For example, "S-DM-3" denotes "Options apparent" as a Strength of Direct Manipulation; "W-NL-2" denotes "Overkill ..." as a Weakness of Natural Language.

Move-lots	
Which lots:	each Dram lot
To which station:	the manual inspection station with the smallest queue
Whenever:	it has been masked here <point to Mask-station-1>

Figure 2: Completed Move-lots form

natural-language queries and descriptions with mouse pointing, menu selection, and graphical output. The current version of the system demonstrates the application of the technology to semiconductor and printed-circuit board manufacturing, but the basic system is equally applicable to a wide variety of domains.

5 A Principled Merger of Interface Technologies

The user of Shoptalk is expected to be performing analysis tasks and/or operations planning. In this section, the system's interface tools that support these tasks are described in relation to the entries of Table 1. A number of these techniques were first described in [4].

5.1 The Invocation of Commands

Shoptalk allows users to execute actions at the present or at some unknown (but user-specified) future time. Following the typical GUI methodology, actions are made available to the user through a menu of commands. However, mediating between the selection of an action and its execution is a "form" into which the user supplies arguments. The user may point at objects and "deposit" them into slots in the form, and/or he may type (or speak) natural language expressions into those slots.⁴ Thus, the user can describe an arbitrary number of objects of the right type, and apply the action to them. The interface designer can specify the types of objects expected in the "prompt" in a field, thereby conveying to the user an aspect of the system's conceptual coverage (W-NL-1), and at the same time, allowing for semantic type-checking to be done.

The "when" field of an action allows a user to enter an invocation condition, such that if the system can prove the condition holds, it takes the action. The condition can be as simple as an integer, denoting a time point, or as complex as a natural language sentence. Through menu selection and toggling, the user can alter the temporal period during which the system will apply the invocation condition. Thus, not only can an action be invoked on an arbitrary number of arguments, it can be invoked when a user-specified condition is true.

For example, considering a semiconductor factory, Figure 2 shows how one can move a collection of semiconductor lots (groups of wafers) of a certain type whenever a certain condition arises by (1) invoking Move-lots from a menu, (2) typing (or speaking) "each Dram lot" into the Which lots slot, (3) typing "the manual inspection station with the smallest queue" in the To which station slot and (4) typing "it has been masked here <point to Mask-station-1>" in the Whenever field. Shoptalk parses each of these phrases, and assembles a semantic representation for the conditional action. When it can prove that the condition specified in the When or Whenever field is true, it invokes the action. In the present case, lots of the right type would be rerouted to a different station once they have been masked at a given masking machine. The user does not need to know which lots they are, where they will go, nor when the moves will take place. This is an example of a "standing order" that would be extremely difficult to express to a system that uses only a graphical user interface unless someone has anticipated just this form of rerouting operation and provided precisely the right set of menus.

This one capability offers advantages over both the usual GUI style of command invocation, and over command languages. Specifically, it overcomes the descriptive weakness of direct manipulation (W-DM-1a,b,c), and the difficulty of formulating delayed actions (W-DM-4). At the same time the use of menu selection to invoke actions addresses the conceptual coverage weakness of natural language interaction (W-NL-1) in that the command options available to the user are still limited, and the types of arguments expected can be displayed.

⁴See [4] for a discussion of our integration of pointing and language. Discussions of other approaches can be found in [12, 19].

In addition to these advantages, the multimodal form technique helps natural language processing by limiting the ambiguity of command invocation (W-NL-6) in at least two ways. First, in some domains, there might be numerous similar commands distinguished, for example, by the types of argument. Thus, in the semiconductor manufacturing world, one might move a wafer, move a piece of equipment, move a worker, and so forth. Given multiple meanings, a parser that is analyzing the verb “move,” would likely entertain a number of word-sense hypotheses at least until the verb’s arguments have been interpreted. By selecting the relevant sense from a menu, the parsing process can be made simpler.

A second ambiguity-related advantage of the use of forms augmented with natural language is the opportunity to specify certain prepositional phrase attachments by filling in slots. Prepositional phrases are syntactically “every way ambiguous” and the number of attachments forms a Catalan series [2].⁵ When the case roles marked by various prepositions can be filled in directly, the number of possible parses to be considered can be substantially reduced. For example, consider “Put the block in the box on the table in the corner by the door.” This sentence has four preposition/noun combinations, and would have 14 parses (= Cat_4). However, for a form with the following argument structure:

Put

Object = “the block in the box”

Destination = “on the table in the corner by the door,”

the number of parses can be characterized as $Cat_1 \times Cat_3 = 1 \times 5$. With more prepositional phrases the savings may become substantial. In summary, the structure of the form reduces the complexity of the natural language processor’s task by making explicit the intended word sense of the action, and by reducing the combinatorics inherent in determining the attachment of the prepositional phrases.

5.2 Anaphora, Follow-up Questions, and Context

Shoptalk is designed to support problem solving through question answering. But, no one wants to ask just one question. During problem solving, answers to questions lead the questioner to think of still other required information, and this leads him or her to ask follow-up questions. Shoptalk presents answers to questions in their own window, thereby graphically limiting context. To ask a follow-up question to a given question, a user must ask the follow-up in the latter’s answer window.

A characteristic of such follow-up questions is the use of anaphora. To date, the determination of the referents for anaphoric noun phrases has been extremely difficult (W-NL-4). Shoptalk provides a facility that alleviates some of the difficulty. For present purposes, anaphora will be treated as a unitary phenomenon, although it is well-known that pronouns and definite noun phrases behave differently (e.g., [5, 7]). We have integrated a number of anaphoric reference techniques, including Hobbs’s method for resolving intrasentential anaphora [8], a method for converting the problem of anaphora into one of pointing, and the provision of a technique for using and manipulating focus spaces [5] via windows.

Regarding the second technique, Webber [20] has argued that a system that engages in a dialogue should compute what is available for subsequent reference after each utterance or sentence. Shoptalk takes this suggestion one step further by explicitly displaying, as “buttons” or icons, what *it* takes to be available for reference in a subsequent follow-up question. By selecting a “focus icon” or pushing a “focus button,” the user reduces the space of referents [4]. Through this technique of augmenting anaphoric reference with pointing, we are attempting to deploy the fail-safe nature of direct manipulation (S-DM-4) to overcome an error-prone aspect of natural language processing.

Of course, most natural language database query systems now offer some limited form of anaphoric reference, and those methods use, at least implicitly, some mechanism to limit the context of search for potential referents. However, the context mechanisms developed do not provide a full tree structure, as various researchers have argued is needed [6, 14], but rather a bounded linear structure in which users can make anaphoric reference to entities brought into focus by some small number of prior questions and/or answers. A full tree structure is not

⁵The Catalan series is defined to be

$$Cat_n = \binom{2n}{n} - \binom{2n}{n-1}$$

maintained in part because the semantics of discourse markers (such as “Ok, now”), whose use enables speakers and listeners to navigate the implicit discourse structure tree, is still unclear (W-NL-3). We have developed a simple technique allowing the user to avoid these hard problems through the use of the explicit depiction and manipulation of context.

Rather than leave the discourse structure implicit, and force users to guess the context in which their utterances are being interpreted, Shoptalk displays the discourse structure as a tree of queries, and allows the user to view and manipulate the discourse graphically (see Figure 3).

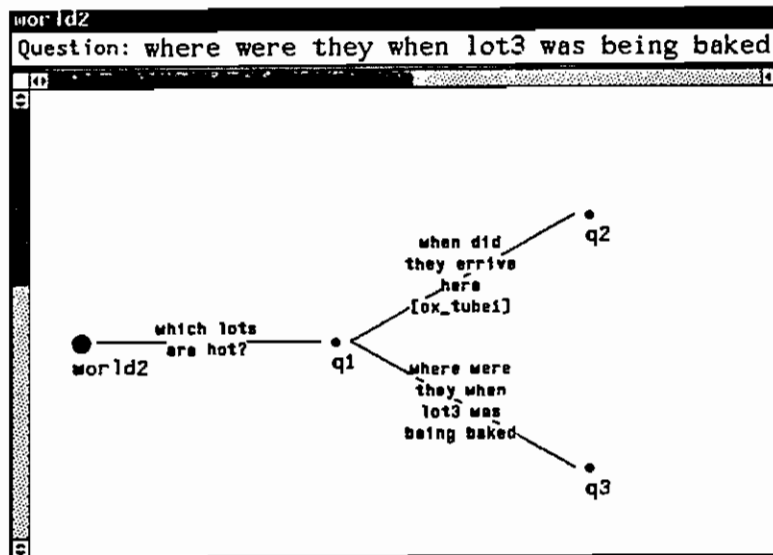


Figure 3: A context tree

A return to a prior context is made simply by selecting a node in the tree.⁶ By doing this, two problems from the user’s perspective are being treated. First, the current discourse context is made apparent, allowing the user to decide explicitly whether or not the question in mind should be a follow-up to a prior one. Once a follow-up is asked in a given context, the tree is extended. Second, by displaying the discourse as a tree, the user can follow up on any query in the discourse, not just those recently asked. As a result of this technique, users can graphically navigate a discourse tree, save promising lines of inquiry for reuse, and perform other context manipulation actions.

5.3 Describing vs. Directly Manipulating Time

One might conjecture that pure direct manipulation would be a powerful interface modality for dealing with time, especially since very attractive graphical renditions of multiple time lines have recently been developed [10]. However, I argue below that the development and manipulation of time lines would benefit from a merger with a linguistic approach.

Because time is a central feature of manufacturing systems, Shoptalk provides a powerful simulation environment that enables a user to analyze the future effects of newly posed operating scenarios. However, in addition to learning about the final state of a simulation, a user may have a bona fide interest in learning about intermediate states. Thus, a simulation system should provide the capability for viewing the state of the system under study at any prior time [15]. Two methods were implemented for rewinding the simulation to a prior time (or times). First, Shoptalk allowed the user to scan backwards by dragging a “slider” or typing in a precise time (see figure 4). Although attractive, this method of directly manipulating time has a number of disadvantages (W-DM-1c). First, the user may simply not know what time is of interest. Armed only with a time slider, the user would have to devise a search strategy. A linear search would be particularly poor, while one that merely stepped through the

⁶ A similar technique is proposed in [11].


Time: [65] 0  100
When: a lot was being baked

Figure 4: Time slider, coupled with time expression

events again, as in a slow-motion replay, would be only marginally better. Second, if the scale of the slider is too large, each pixel may represent too wide a time interval, thereby physically preventing the user from even selecting the desired time through direct manipulation. Finally, sliders allow one to select only a single time point. But if the user is interested in *all* the times when a certain condition arose, he would have to exhaustively reapply the temporal search strategy to find the relevant times. Unfortunately, there are simply too many time points from which to pick.

To overcome these limitations, Shoptalk offers a means for the user to describe the time(s) of interest (S-NL-1c) (see Figure 4). In response to a natural language expression typed or spoken into the *When* slot of the slider, the system composes a menu of *all* time points or intervals satisfying the expressed condition, and resets the slider to the first period found. The distinction between time point and interval is made on the basis of the semantics of the verb, as well as its tense and aspect. If as above, the user asks to rewind the simulation to the times when “a lot was being baked,” the menu will be composed of intervals. If instead, the input were “a lot arrived at the oven,” then a menu of time points would be constructed. To sum up, directly manipulating time has its attractions, but also its drawbacks. As with the other examples given here, when combined with natural language for describing complex event relationships and time periods, the result is clearly more useful.

6 Conclusion

This paper has shown how to build interface components that integrate both direct manipulation and natural language processing, by following the principle of using the strengths of each modality to compensate for the weaknesses of the other. The resulting integrated interface provides more power than either technology could in isolation. Future research should attempt to refine and evaluate this principle and the resulting interface techniques empirically. In addition, we should strive both to identify other principles of interface integration, as well as to apply them to assist in integrating the wide array of emerging modalities.

7 Acknowledgments

Contributions by Mary Dalrymple, David Kashtan, Doug Moran, Sharon Oviatt, and Fernando Pereira have been invaluable to the development of the Shoptalk system.

References

- [1] A. Chapanis, R. N. Parrish, R. B. Ochsman, and G. D. Weeks. Studies in interactive communication: II. The effects of four communication modes on the linguistic performance of teams during cooperative problem solving. *Human Factors*, 19(2):101–125, April 1977.
- [2] K. Church and R. Patil. Coping with syntactic ambiguity or how to put the block in the box on the table. *American Journal of Computational Linguistics*, 8(3-4):139–149, 1982.
- [3] P. R. Cohen. The pragmatics of referring and the modality of communication. *Computational Linguistics*, 10(2):97–146, April-June 1984.
- [4] P. R. Cohen, M. Dalrymple, D. B. Moran, F. C. N. Pereira, J. W. Sullivan, R. A. Gargan, J. L. Schlossberg, and S. W. Tyler. Synergistic use of direct manipulation and natural language. In *Human Factors in Computing Systems: CHI'89 Conference Proceedings*, Austin, Texas, April 1989.

- [5] B. J. Grosz. The representation and use of focus in dialogue understanding. Technical Report 151, Artificial Intelligence Center, SRI International, Menlo Park, California, July 1977.
- [6] B. J. Grosz. Focusing and description in natural language dialogues. In A. K. Joshi, B. Webber, and I. Sag, editors, *Elements of Discourse Understanding*. Cambridge University Press, 1981.
- [7] B. J. Grosz, A. K. Joshi, and S. Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 44–50, Cambridge, Mass, 1983.
- [8] J. R. Hobbs. Resolving pronoun reference. *Lingua*, 44, 1978. Reprinted in *Readings in Natural Language Processing*, Grosz, B. J., Sparck Jones, K., and Webber, B. L. eds., Morgan Kaufman Publishers, Inc., Los Altos, California, 1986.
- [9] E. L. Hutchins, J. D. Hollan, and D. A. Norman. Direct manipulation interfaces. In D. A. Norman and S. W. Draper, editors, *User Centered System Design*, pages 87–124. Lawrence Erlbaum Publisher, Hillsdale, New Jersey, 1986.
- [10] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In S. P. Robertson, G. M. Olson, and J. S. Olson, editors, *Human Factors in Computing Systems: CHI'91 Conference Proceedings*, pages 173–179, New Orleans, Louisiana, May 1991. SIGCHI, ACM Press.
- [11] J. D. Moore and W. R. Swartout. Pointing: A way toward explanation dialogue. In *Proceedings of the Eight National Conference on Artificial Intelligence*, pages 457–464, Cambridge, Massachusetts, July 1990. American Association for Artificial Intelligence, AAAI Press/MIT Press.
- [12] J. G. Neal and S. C. Shapiro. Intelligent multi-media interface technology. In J. W. Sullivan and S. W. Tyler, editors, *Intelligent User Interfaces*, chapter 3, pages 45–68. ACM Press Frontier Series, Addison Wesley Publishing Co., New York, New York, 1991.
- [13] S. L. Oviatt and P. R. Cohen. The contributing influence of speech and interaction on human discourse patterns. In J. W. Sullivan and S. W. Tyler, editors, *Intelligent User Interfaces*, chapter 3, pages 69–83. ACM Press Frontier Series, Addison-Wesley Publishing Co., New York, New York, 1991.
- [14] R. Reichman. *Plain-speaking: A theory and grammar of spontaneous discourse*. PhD thesis, Department of Computer Science, Harvard University, Cambridge, Massachusetts, 1981.
- [15] J. Rothenberg. Knowledge-based simulation at the RAND Corporation. In P. A. Fishwick and R. B. Modjeski, editors, *Knowledge-based Simulation*, Advances in Simulation 4, pages 133–161. Springer-Verlag, New York, 1991.
- [16] A. D. Rubin. A theoretical taxonomy of the differences between oral and written language. In *Theoretical Issues in Reading Comprehension*. Lawrence Erlbaum Assocs., Hillsdale, N. J., 1980.
- [17] B. Shneiderman. Natural vs. precise concise languages for human operation of computers: Research issues and experimental approaches. In *Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics*, pages 139–141, Philadelphia, Pennsylvania, June 1980.
- [18] Ben Shneiderman. Direct manipulation: A step beyond programming languages. *IEEE Computer*, 16(8):57–69, 1983.
- [19] W. Wahlster. User and discourse models for multimodal communication. In J. W. Sullivan and S. W. Tyler, editors, *Intelligent User Interfaces*, chapter 3, pages 45–68. ACM Press Frontier Series, Addison Wesley Publishing Co., New York, New York, 1991.
- [20] B. L. Webber. So what can we talk about now? In M. Brady and R. Berwick, editors, *Computational Models of Discourse*. MIT Press, Cambridge, Massachusetts, 1983. Reprinted in *Readings in Natural Language Processing*, Grosz, B. J., Sparck Jones, K., and Webber, B. L. eds., Morgan Kaufman Publishers, Inc., Los Altos, California, 1986.

