Combining Discourse Strategies to Generate Descriptions to Users Along a Naive/Expert Spectrum

Cecile L. Paris

Department of Computer Science
Columbia University
New York, N.Y. 10027
Ph: (212) 280 8125
CECILE@CS.COLUMBIA.EDU

ABSTRACT:

A question answering system that provides access to a large amount of data will be most useful if it can tailor its answer to each user. In particular, a user's level of knowledge about the domain of discourse is an important factor in this tailoring. In previous work we determined that a user's level of domain knowledge affects the kind of information provided in an answer to a user's question as opposed to just the amount of information, as was previously proposed. We also explained how two distinct discourse strategies could be used to generate texts aimed at naive and expert users. Users are not necessarily truly expert or fully naive however, but can be anywhere along a knowledge spectrum whose extremes are naive and expert In this work, we show how our generation system, TAILOR, can use information about a user's level of expertise to combine several discourse strategies in a single text, choosing the most appropriate at each point in the generation process, in order to generate texts for users anywhere along the knowledge spectrum. TAILOR'S ability to combine discourse strategies based on a user model allows for the generation of a wider variety of texts and the most appropriate one for the user.

I Introduction

A question answering system that provides access to a large amount of data will be most useful if it can tailor its answer to each user. In particular, a user's level of knowledge about the domain of discourse is an important factor in this tailoring. The answer should not provide information that is obvious to the user. However, if the answer assumes knowledge that the user docs not have, it may be very hard for the user to understand the answer. In previous work [Paris 85], we determined that a user's level of domain knowledge affects the kind of information provided in an answer* to a user's question as opposed to just the amount of information, as was previously proposed [Wallis and Shortliffe 82]. We also explained how two distinct discourse strategies could be used to generate texts aimed at naive and expert users [Paris 85; Paris 87]. Users are not necessarily truly expert or fully naive however, but can fall anywhere along a knowledge spectrum whose extremes are naive and expert In this work, we show how a generation system can use the information contained in a user model to combine several discourse strategies in a single text, choosing the most appropriate at each point in the generation process, in order to generate texts for users anywhere along this knowledge spectrum

In past work on text generation, discourse strategies have been used to guide a generation system in deciding what to say [Weiner 80; Mann 84; McKeown 85; Kulrich 85; McCoy 86]. If a generation system had a choice of several strategies for composing a text, it would choose one based on the question type and use that strategy to generate the entire text Our system's ability to combine discourse strategies based on a user model allows for the generation of a wider variety of texts and the most appropriate one for the user.

II The problem: describing complex physical objects

In our work, we are mainly concerned with describing complex physical objects. Our choice of domain has been motivated by RESEARCHER, a program being developed at Columbia University that reads, remembers and generalizes from patent abstracts written in English [Lebowitz 83; Lebowitz 85]. The abstracts describe complex physical objects in terms of spatial and functional information. Being able to provide descriptions of the physical objects it has learned about is an important component of RESEARCHER. We assume that a good description is one that provides meaningful information about an object and that allows the user to build a mental functional model of the object

We have built a program called TAILOR to implement and test our discourse strategies. TAILOR generates descriptions tailored to the user's level of expertise, using the two distinct strategies we found in naturally occurring texts. The discourse strategies guide TAILOR to choose the appropriate information from the knowledge base, and the user model dictates which strategy to use at each point in the generation process. TAILOR, which can be thought of as the generation component of RESEARCHER, uses the knowledge base built by RESEARCHER, as depicted in Figure 1.

In the domain of complex physical objects, the amount of information contained in the knowledge base is very large and the information can be very detailed. The knowledge base contains several different kinds of information: spatial, functional and attributive (properties attached to objects). TAILOR must choose among facts representing different kinds of information about an object and facts at different levels of detail in the knowledge base, making the decision process a complicated one. TAILOR cannot simply state all the facts contained in the knowledge base about a particular object Rather, it should pick facts that are appropriate and construct a coherent text from these facts [McKeown 82; McKeown 85]. A user model representing what the user presumably knows about the domain can help guide the

*We have done our analysis with respect to requests for descriptions.

system in choosing facts that the user understands and does not already know (and cannot easily infer), thereby improving the resulting answer.

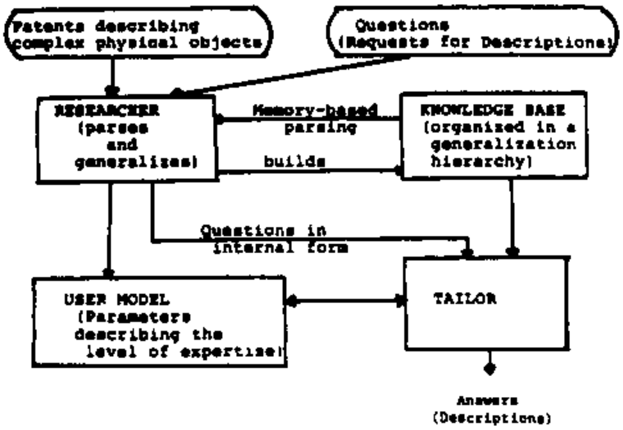

Figure 1: RESEARCHER and the TAILOR System

strategy is most appropriate when a user is an expert in the domain of the knowledge base, since s/he will be able to infer how the parts of an object work together in order to perform the object function. It is then not necessary to include the process information associated with the object

• *The Process Trace,* which follows the links between events in the underlying knowledge base in order to trace through the process information that explains the function of the object. A precise formulation of this strategy has been developed in [Paris and McKeown 87]. It is summarized in Figure 3. The process trace strategy is most appropriate when the user is a novice with respect to the domain of the

Examples of descriptions generated with the two strategies given the corresponding user models are shown in Figures 4 and 5. In both figures, the description given is that of a microphone. In Figure 4, after identifying the microphone as a member of a superordinate class, its parts and their properties are given. In Figure 5, the text follows the causal links associated in the knowledge base in order to explain how the microphone works.

Description of the microphone

User Model: *expert*

Based on the user model, the system decides on the process strategy.

The microphone changes soundwaves into current A person speaking into the microphone causes the soundwaves to hit the diaphragm of the microphone. The diaphragm is aluminium, and disc-shaped. The soundwaves hitting the diaphragm causes the diaphragm to vibrate. The soundwave_intensity increasing causes the diaphragm to spring forward. The diaphragm springing forward causes the granules to compress. The granules compressing causes the resistance of the granules to reduce. The resistance reducing causes the current to increase. The soundwave_intensity reducing causes the diaphragm to spring backward. The diaphragm springing backward causes the granules to decompress. The granules decompressing causes the resistance to increase. The resistance increasing causes the current to reduce. The diaphragm vibrating causes the current to vary. The current varies like the intensity varies.

Figure 5: Example of a description

Since the knowledge base contains many different objects, most users will be neither fully naive nor truly expert with respect to the knowledge base. "Naive" and "expert" are the extremes of a knowledge spectrum, and most users will be at intermediate points in the spectrum. Such users may have *local expertise* about some objects in the knowledge base and not others. Hence it is not sufficient for the user model to indicate whether a user is naive or expert The user model must distinguish more precisely the user's level of domain knowledge about separate sub-areas of the knowledge base and individual objects. Finally, the user model must also contain information about whether the user is knowledgeable about the basic concepts that may be involved in a process description. For example, in our domain, such concepts include electricity and magnetism.

If a user has local expertise about some parts of an object and not others, a complete process trace (including a process trace for every part) is unnecessary. On the other hand, providing a description based only on the constituency schema may not be totally understandable. For such users, we use both strategies to produce a description, using the constituency schema for the known parts and the process trace for the new ones.

As an example, suppose we are providing a description of an elevator to a user who knows how a motor functions, but not how an elevator works. In describing the elevator to this user it is necessary to first describe how the parts of the elevator are related, using the process trace strategy. If a long description is desired, the individual parts must be described in turn. It should not be necessary to fully explain what the motor does, as the user already knows about it The constituency schema strategy can thus be used to describe the motor. For the other parts, however, the process strategy is still appropriate in order to explain their mechanism.

Our analysis of naturally occurring texts confirmed that several strategies can be used to generate a text. We found example of texts that combine the constituency schema and the process trace to generate descriptions aimed at users with intermediate levels of domain knowledge. Consider the text shown in Figure 6, taken from [Chemical 78]. The description starts with the constituency schema strategy but ends with a process trace: the "ir [Infra red] spectrometer" is first described in terms of its parts; each part is then described in turn (depth-attributive); finally, the authors revert to a process trace to describe the "thermocouple detector," assuming it is unknown to the reader. To fully understand this text, the reader must already know (or be able to infer information) about the ir spectrometer's purpose, the "ir radiation," the "monochromator" and the purpose of a spectrometer.

(1) The ir spectrometer consists of three essential features: a source of ir radiation, a monochromator and a detector. (2) The primary sources or ir radiation are the Globar and Nernst glower. (3) The Globar is a silicon carbide rod heated to 1200 degrees C. (4) The Nernst glower is a rod containing a mixture of yttrium, zirconium, and erbium oxides that is heated electrically to 15000 degrees C. (5) Earlier ir spectrometers contained prism monochromators but today gratings are used almost exclusively. (6) Most detectors in modern spectrometers operate on the thermocouple principle. (7) Two dissimilar metal wires are connected to form a junction. (8) Incident radiation causes a temperature rise at the junction and the difference in the temperature between head and tail causes a flow of current in the wires which is proportional to the intensity of the radiation.

1. Constituency (parts)
2-4. Depth attributive for the ir radiation (properties)
5,6. Depth attributive for the monochromators
7. Depth attributive for the thermocouple spectrometer
8. Process trace for the thermocouple principle.

Figure 6: Text from the *Encyclopedia of Chemical Technology*

We can now present the decision algorithm employed by TAILOR to decide which strategy must be used to begin generating a description. If the user is not an expert but has local expertise about the object to be described, the constituency schema can be used initially. If the user knows about nearly all of the functional parts of the objects, the constituency schema can still be used, provided that the process trace is used when details about the unknown parts need to be given. The decision algorithm is described in Figure 7.

To be able to combine the two strategies to generate a description, we must specify the points at which it is possible to switch strategy. Any time an object is introduced and needs to be described, the system can provide either structural information (using the constituency schema) or functional information (using the process trace). This gives the following decision points in the strategies:

Is the user an *expert* user?
Yes : Use constituency schema
No : Is the user a *naive* user?
  Yes: Use process trace
  No: Is the object in the user model (i.e., does the user have local expertise about this object?)
    Yes: Use constituency schema
    No: Look at all the functional parts of the objects. If the user has local expertise about all but one of these pans, use the constituency schema;
    Else use process trace.

Figure 7: The decision algorithm

- Within the constituency schema:
  - after identifying an object as a member of some generic class, we can provide a process trace for the prototypical object of this class.
  - after mentioning the parts of an object, the constituency schema dictates to provide structural information for each subpart. Instead, we could provide a functional description of one or more of the parts. This has been done in the *Encyclopedia of Chemical Technology* text, for example.
- Within the process trace:
  - when a part is introduced while traversing the causal links, the process strategy dictates to include attributes of this part (to describe it). Here, we could choose to describe the part more fully using the constituency schema instead.
  - When the subparts have to be described, the constituency schema can be used to provide structural information about them instead of including functional information.

Figure 8 summarizes the two strategies in their simplest form; the decision points are denoted by *"switch."*

Constituency Schema	Process Trace
- Give the superordinate *Switch?*	- Next causal link
- List the parts *Switch?*	- Attributes of a part *Switch?*
- or information about the parts	- Substeps
- Properties	- Back to next causal link

Figure 8: The two strategies and their decision points

While we could generate all the possible combinations allowed by the decision points, some combinations are better than others. We have accounted for these preferences using heuristics that decide *when* to mix the strategies. These heuristics limit the texts we can produce by eliminating some combinations that produce inappropriate texts given the users' knowledge about the domain. The heuristics examine the user model to see what items are known to the user (or what items have a superordinate known to the user), and decide whether it is appropriate to continue with the same strategy or to switch. The decision test is essentially the same as the one used to choose the initial strategy.

Figure 9 shows an example of a text generated by TAILOR by combining the two strategies. Based on the user model that indicates local expertise about *loudspeaker,* TAILOR chooses the constituency schema. It first identifies the *telephone* by providing its purpose and then introduces its parts. Structural information is then provided about each of the parts, except for the *transmitter,* because the transmitter plays an important role in the function of a telephone and the user model shows no local expertise about it. Thus TAILOR chooses to provide process information for the transmitter, switching momentarily to the process trace strategy.

Description of the telephone

User Model: Local Expertise: Loudspeaker

The telephone has two main functional part: the transmitter (an instance of a microphone) and the receiver (an instance of the loudspeaker). Because the user knows one of the two parts of the telephone, the system decides on the constituency schema strategy at first. However, before providing structural information about each subparts, the system consults the user model and decides to switch strategy to describe the transmitter since the user has no local expertise about it.

The telephone changes soundwaves into soundwaves. The telephone has a various-shaped housing, a transmitter that changes soundwaves into current, a curly-shaped cord, a line, a receiver to change current into soundwaves, and a dialing jnechanism. The transmitter is a microphone. A person speaking into the microphone causes the soundwaves to hit the diaphragm of the microphone. The soundwaves hitting the diaphragm causes the diaphragm to vibrate. The diaphragm vibrating causes the current to vary. The current varies like the intensity varies. The receiver is a loudspeaker with a small aluminium diaphragm. The housing contains the transmitter. The housing contains the. receiver. The housing is connected to the dialing jnechanism by the cord. The line connects the dialing_mechanism to the wall.

Figure 9: A description generated by mixing the two strategies

IV TAILOR system and the implementation of the strategies

A top-level diagram of TAILOR is shown in Figure 10. Input to TAILOR includes: (1) a request for the description of an object, and (2) the level of expertise of the user, which is either naive/expert for users at the ends of the knowledge spectrum, or, for users with intermediate levels of expertise, a set of parameters that describe the user's level of knowledge about the domain, including:

- whether the user knows (or does not know) the basic underlying concepts that are important in the domain of the knowledge base.
- the specific objects the user knows in the domain.

The *textual component* of the system decides what to include in the description. It examines the *knowledge base* and chooses appropriate facts, guided by the user model and the discourse strategies. The output of the textual component is a conceptual representation of the content of the description. This representation is passed through an *interface* which makes lexical choices for the various concepts included in the description. The interface utilizes the focus of a proposition and the past discourse to guide its decision process. Finally, a *surface generator* constructs

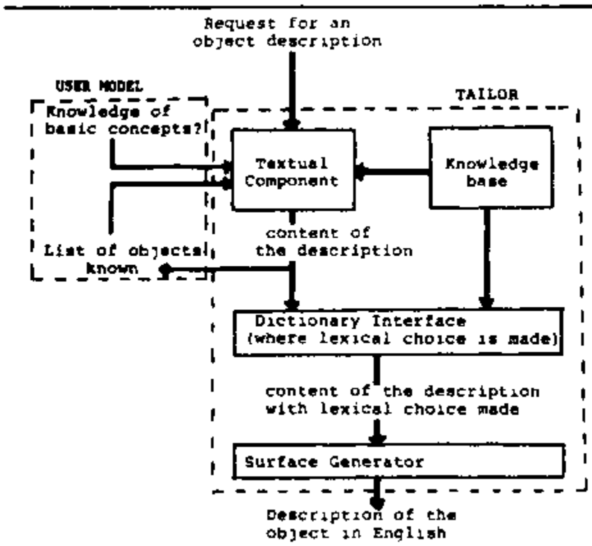

Figure 10: The TAILOR System

English sentences. The surface generator is based on the generator used by [McKeown 85] in the TEXT system. This generator unifies the input with a functional grammar [Kay 79] to produce English sentences. We have extended and improved the performance of this program, and augmented the functional grammar it uses [Paris and Kwee 85; McKeown and Paris 87].

Strategy Implementation

The constituency schema and the process trace strategies are implemented using an augmented transition network (ATN) [Woods 73]. The arcs joining the various nodes in the network dictate what information is to be retrieved from the knowledge base, under what conditions (the arcs contain a test), and which node to go next.

In the constituency schema, shown in Figure 11, the arcs correspond to the predicates of the schema. (See [Paris 85; Paris and McKeown 87] for details about traversing the ATN, and [McKeown 85] for details of a similar system.) These predicates, which are enumerated below, define the type of information to be retrieved from the database.

- The *identification* predicate represents the more general concept of which the present object is an instance.
- The *constituency* predicate gives the components of an entity, if there are any.
- The *attributive* predicate provides different attributes of a part (such as its shape or material).
- The *cause-effect* predicate that provides some causal relations between entities or relations.

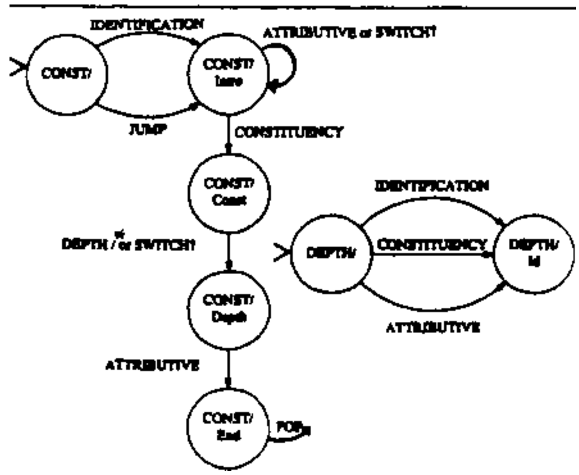

Figure 11: The Constituency Schema

For the process trace, shown in Figure 12, the arcs are not predicates *per se,* but dictate how to trace the knowledge base to form an answer, mainly by following the causal links in the knowledge base. The arcs are:

- *Next-main-link:* this arc dictates to follow the next link on the *main path.* The main path is the sequence of events that is performed in order for an object to achieve its function.
- *Side-link?: A side-link* is a link that is not part of the main path, but comes off an event on the main path. This arc tests to see whether there is a side link caused by an event at this point. The decision to mention the side link is based on the importance of that link.***
- *Attributive:* This arc is similar to the attributive predicate in the constituency schema. If information about a part just introduced is available in the knowledge base, this arc will be taken.
- *Substeps?:* If an event consists of several substeps, the substeps are traced first.**** To traverse the substeps, the subroutine *substep,* shown in Figure 12, is called for each substep. This subroutine is very similar to the main graph, but does not allow for a further decomposition of events.

By representing the two strategies in the same format, we immediately obtain the control structures necessary to switch strategies, since it is possible to jump from a node in one part of the network to a node in a different part. The decision points are marked as special tests *(Switch?)* on the arcs joining nodes, as shown in Figures 11 and 12.

***In a complex knowledge base, there can be different kinds of links between processes, including causal links, temporal links or analogical links. Some links are more important to mention than others. Since only the most important links are mentioned, links have to be given importance weights. (See [Paris and McKeown 87] for details.)

****Note that we can also choose to not follow the substeps in order to generate a shorter description. This factor is incorporated into the test of the arc.

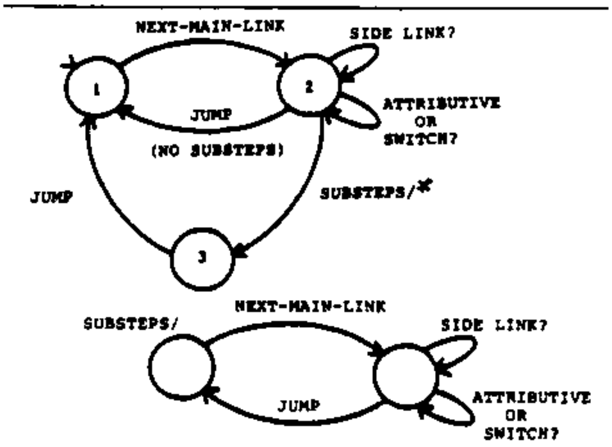

Figure 12: The Process Trace

TAILOR uses the decision process described earlier to decide on which strategy to employ. This test is invoked before beginning generation and at any point in the schema when it is possible to switch strategies. The process of stepping through the ATN and switching strategy is shown in Figure 13. (The corresponding generated text was shown in Figure 9.) Note that to avoid generating very long texts, substeps are omitted when the process trace is chosen to describe a subpart

For each object of the knowledge base, TAILOR can currently generate descriptions aimed at expert and naive users, and texts that combine the two strategies for users who are along the knowledge spectrum and only know a few objects.

V Further work

We are investigating extensions of the tests used in the decision process. Such extensions might include checking on the length of the discourse and the number of parts to avoid generating overwhelmingly long texts, or taking the generalization hierarchy into consideration. Furthermore, we are extending our work to examine the recursive use of both strategies. Currently, TAILOR does not call the constituency schema recursively more than once (i.e., the constituency schema can be called recursively for each subpart of the top level object being described, but not for the components of these subparts). Similarly, the process is traced only one level down, that is only the top-level relations are broken into substeps. This kind of recursion however could occur as deeply as the knowledge base allows for. We think that such a level of detail is appropriate only if the user asks for a longer description. We are planning on implementing a mechanism that would allow the users to request such a description. The problem of depth on the process trace also needs to be studied further. We expect the user's domain knowledge, in particular knowledge about basic concepts, to play a role in determining the depth needed: at some point, one cannot subdivide the processes any further if the user does not know basic underlying concepts as the process explanation will cease to be understandable. For example,

```
User Modal: Local Expertise: Loudspeaker

  Stepping through the Constituency Schema to describe
  a TELEPHONE.   Switching to the Prooess Traoe to
  desoribe the TRANSMITTER.

Applying the predicates to TELEPHONE:

 Identification predicate:
   DEVICE used-for: change soundwaves into soundwaves

 Constituency predicate:
   DIALING_MECHANI8M
   TRANSMITTER used-for: change soundwaves into current
   LINE
   CORD
   RECEIVER used-for: change current into soundwaves
   HOUSING properties: oolor: various, shape: various

 Need to switch to process traoe for the TRANSMITTER
 Introduction:
 Identification predicate: MICROPHONE

 Causal link (1) :
       <M-CAUSES) relates the two relations:
                  [ONE] P-8PEAKS-INT0 [TRANSMITTER]
                  [SOUNDWAVES] P-HIT8 [DIAPHRAGM]

 Causal link (2) :
       (M-CAUSES) relates the two relations:
                  [SOUNDWAVES] P-HIT8 [DIAPHRAGM]
                  [DIAPHRAGM] P-VIBRATE8

 [8ubsteps omitted]

 Causal link (3) :
       (M-CAUSES) relates the two relations:
                  [DIAPHRAGM] P-VIBRATES
                  [CURRENT] P-VARIES

 Side Link (4) :
       (M-EQUIVALENT-TO) relates the two relations:
                  [CURRENT] P-VARIES
                  [SOUNDWAVE-INTENSITY] P-VARIES

Returning to the constituency schema:

Applying the predicates to RECEIVER:
 Identification: RECEIVER is a LOUDSPEAKER
                 difference: small aluminium diaphragm

Applying the predicates to HOUSING:
 Attributive: HOUSING r-oontains TRANSMITTER
              HOUSING r-oontains RECEIVER
              HOUSING r-oonnected-to DIALINC_MECHANI8M
                                   by CORD

Applying the predicates to LINE:
 Attributive: DIALING_MEGHANISM r-oonnected-to WALL
                                               by LINE
```

Figure 13: Switching strategy

explaining how a disc drive works in terms of current and magnetism will not be informative if the user does not understand these terms. The process explanation, in such a case, should be kept at a higher level. The depth of the knowledge base itself may also affect the depth of the description, as parts or processes may be considered too minor to be mentioned beyond a certain threshold.

VI Conclusions

In this paper we have shown how a user model can determine which of several discourse strategies to use in generating a description of physical objects. We have also shown how to mix several strategies to generate a single text, using information about a user s local expertise about the domain of discourse to decide when to switch strategy. Finally we presented TAILOR, a program that generates descriptions tailored to users along the knowledge spectrum going from naive to expert. TAILOR uses two distinct discourse strategies, and, based on the user's level of expertise, decides which one to use and when to switch strategy during the generation process.

Acknowledgments

Many thanks go to Kathy McKeown and Michael LebowiU for helping in both the research and the writing of this paper, and to Kwee TjocLiong for his initial work on the functional grammar.

This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-84-C-0165, and the National Science Foundation grant ISI-84-51438.

References

[Chemical 78] Encyclopedia of Chemical Techology. 1978 Kirk and Othmer, Eds; John Wiley & Sons, New York, NY 1978.

[Kay 79] Kay, Martin. Functional Grammar. In *Proceedings of the 5 th meeting of the Berkeley Linguistics Society.* Berkeley Linguistics Society, 1979.

[Kukich 85] Kukich, K. Explanation Structures in XSEL. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, Chicago, Illinois, 1985.

[Lebowitz 83] Lebowitz, M. RESEARCHER: An Overview. In *Proceedings of the Third National Conference on Artificial Intelligence.* American Association of Artificial Intelligence, Washington, DC, 1983.

[Lebowitz 85] Lebowitz, M. RESEARCHER: An experimental intelligent information system. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence,* pages 858 - 862. Los Angeles, 1985.

[Mann 84] Mann,W.C. Discourse Structures for Text Generation. In *Proceedings ofCOUNG '84.* Association for Computational Linguistics, Stanford, Ca., July, 1984.

[McCoy 86] McCoy, K.F. The ROMPER System: Responding to Object-Related Misconceptions Using Perspective. In *Proceedings of the 24th Annual Meeting of the ACL.* Association of Computational Linguistics, New York City, New York, June, 1986.

[McKeown 82] McKeown, K.R. *Generating Natural Language Text in Response to Questions About Database Structure.* PhD thesis, University of Pennsylvania, May, 1982. Also a Technical report, No MS-CIS-82-05, University of Pennsylvania, 1982.

[McKeown 85] McKeown, K.R. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, Cambridge, England, 1985.

[McKeown and Paris 87] McKeown, K. R. and Paris, C. L. Functional Unification Grammar Revisited. In *Proceedings of the 25th Annual Meeting of the ACL.* Association of Computational Linguistics, Palo Alto, California, 1987.

[Paris 85] Paris, C. L. Description Strategies for Naive and Expert Users. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics.* Chicago, 1985.

[Paris 87] Paris, C. L. Tailoring Object Descriptions to the User's Level of Expertise. *To Appear in the Computational Linguistics Journal,* Special Issue on User Modelling, 1987. Paper presented at the International Workshop on User Modelling, August 1986, Maria Laach, West Germany.

[Paris and Kwee 85] Paris, C. L. and Kwee TL. *Guide to the Unification Process and its Implementation; Progress Report on Extending the Grammar.* Technical Report, Computer Science Department, Columbia University, New York, NY 10027, 1985.

[Paris and McKeown 87] Paris, C. L. and McKeown, K. R. Discourse Strategies for Describing Complex Physical Objects. In G. Kempcn (editor), *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics.* Kluwer Academic Publishers, Boston/Dordrecht, 1987. Paper presented at the Third International Workshop on Natural Language Generation, August 1986, Nijmegen, The Netherlands.

[Wallis and Shortliffe 82] Wallis, J.W. and Shortliffe, E.H. *Explanatory Power for Medical Expert Systems: Studies in the Representation of Causal Relationships for Clinical Consultation.* Technical Report STAN-CS-82-923, Stanford University, 1982. Heuristics programming Project Department of Medecine and Computer Science.

[Weincr 80] Weiner, J. BLAH, a System that Explains its Reasoning. *Artificial Intelligence Journal,* 15:19 - 48, 1980.

[Woods 73] Woods, W. An Experimental Parsing System for Transition Network Grammars. In Rustin, R. *{editor). Natural Language Processing.* Algorithmics Press, New York, 1973.