

TAILORING OBJECT DESCRIPTIONS TO A USER'S LEVEL OF EXPERTISE

Cécile L. Paris

Department of Computer Science
Columbia University
New York, NY 10027

A question answering program providing access to a large amount of data will be most useful if it can tailor its answers to each individual user. In particular, a user's level of knowledge about the domain of discourse is an important factor in this tailoring if the answer provided is to be both informative and understandable to the user. In this research, we address the issue of how the user's domain knowledge can affect an answer. By studying texts, we found that the user's level of domain knowledge affected the *kind* of information provided and not just the *amount* of information, as was previously assumed. Depending on the user's assumed domain knowledge, a description can be either parts-oriented or process-oriented. Thus the user's level of expertise in a domain can guide a system in choosing the appropriate facts from the knowledge base to include in an answer. We propose two distinct descriptive strategies that can be used in a question answering program, and show how they can be mixed to include the appropriate information from the knowledge base, given the user's domain knowledge. We have implemented these strategies in TAILOR, a computer system that generates descriptions of devices. TAILOR uses one of the two discourse strategies identified in texts to construct a description for either a novice or an expert. It can merge the strategies automatically to produce a wide range of different descriptions to users who fall between the extremes of novice or expert, without requiring an a priori set of user stereotypes.

1 INTRODUCTION

The tailoring of answers according to a person's domain knowledge frequently occurs in human communication. For example, an explanation aimed at a child of how a car engine works will be different than one aimed at an adult, and an explanation adequate for a music student is probably too superficial for one in mechanical engineering. We have found further evidence of this phenomenon in naturally occurring texts (*Collier* 1962, *Britannica* 1984, *Britannica-Junior* 1963, *New Book of Knowledge* 1967, *Encyclopedia of Science* 1982, *Chevrolet* 1978, *Weissler* 1973), where the information presented to readers varies with their assumed level of general knowledge.

Likewise, a question answering program that provides access to a large amount of data to many different users could be more effective if it could customize its answers to each user, retrieving from its knowledge base the facts that are most appropriate and useful for a given user. Much research to date has focused on tailoring an answer depending on a user's goals, but

customizing an answer depending on what the user knows about the domain of the question has been overlooked. This is an important factor in tailoring an answer if the answer provided is to be both informative and understandable to the user. The answer should not provide information that is obvious to the user (Grice 1975). However, if the answer assumes knowledge that the user does not have, it may be very hard (if not impossible) for the user to understand the answer (Wilson and Anderson 1986). In this paper we show the feasibility of incorporating the user's domain knowledge, or *level of expertise*, into a generation system and address the issue of how this factor might affect an answer.

Through an analysis of texts, we found that two distinct discourse strategies were used in describing texts. We postulate that the writers' choice of strategy might be based on the assumed domain knowledge of the expected readers. If so, then the reader's level of knowledge about the domain affects the *kind* of information provided as opposed to just the *amount* of information. In previous generation systems, the user's

Copyright 1988 by the Association for Computational Linguistics. Permission to copy without fee all or part of this material is granted provided that the copies are not made for direct commercial advantage and the CL reference and this copyright notice are included on the first page. To copy otherwise, or to republish, requires a fee and/or specific permission.

0362-613X/88/0100●-●\$03.00

domain knowledge affected only the amount of detail provided in the text (Wallis and Shortliffe 1982).

In this work, we show how the two discourse strategies found in texts can be used to provide answers to users whose domain knowledge falls anywhere along a knowledge spectrum, from naive to expert. We have implemented them in TAILOR, a computer system that generates descriptions of physical objects.

TAILOR uses one of the two discourse strategies identified in texts to construct a description for either a novice or an expert. It can merge the strategies automatically in a systematic way to produce a wide variety of different descriptions for users who fall between the extremes of novice and expert. This means TAILOR is able to generate descriptions to a whole range of users, rather than just for an a priori set of user stereotypes.

1.1 PREVIOUS WORK ON USER MODELING IN QUESTION ANSWERING PROGRAMS

In studying the factors involved in tailoring the content of an answer to a user, research to date has focused mainly on the problems of inferring and using user goals, plans, and beliefs (Appelt 1982, 1985; Carberry 1983, and this issue; McKeown et al. 1985), recognizing and dealing with misconceptions (Kaplan 1982; McCoy 1983; McCoy 1986, and this issue; Quilici et al., this issue), and superposing various stereotypes (Rich 1979). The issue addressed here differs from these because we are not concerned with the users' goals in asking the question, nor with correcting their view of the domain, but rather with providing an answer that is optimally informative (without being overwhelming) given how much the user knows about the domain. We are not interested in building a user model using stereotypes (as was Rich), but in determining an answer based on a user model involving user types. As in McCoy (1986, and this issue), we are more concerned about using information from the user model to generate an answer than building the model itself.

While the need for a model of the user's domain knowledge in question answering systems has been noted by various researchers (Lehnert 1977; McKeown 1985), few programs have actually had one. The HAMS system (Hoeppner et al. 1984) has a model of the user's knowledge, but this knowledge is mainly used for anaphora resolution and production. In our work, we are more interested in studying how a user's knowledge affects the content of an answer as opposed to its phrasing. Wallis and Shortliffe (1982), who have used the naive/expert distinction in providing an answer (or explanation), did so mainly by giving more or less detail, without addressing the issue of whether the level of detail was the only important factor to vary. The issue we confront in this work is identifying the role played by a user's level of knowledge in determining the content of an answer. The UNIX Consultant (UC) (Chin 1986) uses a user's knowledge level about the UNIX system to provide help to its users. UC, however, uses stereotypes for both the user and the knowledge base

(set of UNIX commands). Stereotypes for the knowledge base include "simple", "mundane", and "complex". UC matches the user type against the command type to decide on the answer. In this work, we are looking at a different kind of domain, the domain of complex physical objects, in which this categorization of the knowledge base is not possible. Furthermore, we would like to be able to tailor answers to users whose domain knowledge level falls anywhere along a knowledge spectrum without necessarily having to classify users in several different stereotypes.

1.2 THE DOMAIN

In our work, we are mainly concerned with describing complex devices such as telescopes, telephones, and disk drives to users with different levels of expertise. Our choice of domain has been motivated by RESEARCHER, a program being developed at Columbia University. RESEARCHER reads, remembers, and generalizes from patent abstracts written in English (Lebowitz 1983, 1985, 1986). The resulting knowledge base is organized in a generalization hierarchy. The abstracts describe complex physical objects in which spatial and functional relations are important. In this domain, the amount of information contained in the knowledge base is very large and the information can be very detailed. Moreover, the knowledge base contains several different kinds of information: spatial, functional, and attributive (properties attached to objects). A program can choose from among facts representing different kinds of information about an object, and facts at different levels of detail in the knowledge base, rendering the decision process a complicated one.

A request for the description of an object cannot be translated into a simple database query and thus cannot be answered by a straightforward retrieval from the knowledge base. This type of question has been termed high level questions. (Tennant 1978, McKeown 1985). There are no clear constraints on what information should be included in the answer. Since the amount of information contained in the knowledge base is very large and the information very detailed, a program cannot just state all the facts contained in the knowledge base about the object as there will typically be too many. Rather, it needs to select a subset of facts to present to the user. As the answer will be composed of several facts, a generation program needs to organize these facts in order to construct from them a coherent text (McKeown 1985). When a generation system can choose among many facts, a user model representing what the user presumably knows about the domain can guide the system in choosing information that the user understands and does not already know (and cannot easily infer), thereby improving the resulting answer.

Descriptions are important because they can be used to answer other types of high level questions. For example, to compare two objects, it may be necessary to describe each of them. Furthermore, with a knowl-

edge base of physical objects, users are likely to ask such questions. In order to focus on how the level of expertise affects a description, we have not considered how the goal of the questioner could affect the description. It is clear that in a sophisticated question answering program the user's goal should also play an important part. An answer for a user whose goal is to buy an object should include different kinds of information than an answer for a user who wants to repair this object. Detecting and using the user's goal to provide an appropriate response has been the focus of extensive research (Appelt 1985, Carberry 1983, McKeown et al. 1985). In this work, being more concerned with the role played by the user's domain knowledge, we simply assume that users want to find some information about an object. A description should provide meaningful information about an object and allow the user to build a mental functional model of the object. Therefore, we assume that the goal of a description is to help the user construct a mental functional model of the object under consideration.

2 IDENTIFYING WHAT NEEDS TO BE IN THE USER MODEL

Our goal is to provide a characterization of the role of the user's domain knowledge in generating descriptions that is computationally usable by a generation system. Even though we will not be addressing the problem of how to determine how much the user knows about the domain, we still have to ask what kinds of knowledge a user can possess about a domain that can affect generation and that can be explicitly represented in a user model. Instances of these kinds of knowledge will be the information contained in our user model. Having identified what needs to be in the user model, we will take the user model as given, and study how a system can use the information about a user's domain knowledge contained in the user model to tailor the answer.

Analysis of natural language texts suggests the existence of at least two kinds of domain knowledge that affect the type of descriptions that can be provided in our domain:

- *knowledge about specific items in the knowledge base.* We define "knowing" about an object to mean knowing about the existence of the object, its purpose and how this purpose is achieved (that is, how the various subparts of the object work together to achieve it). Knowing about an object thus means understanding the functionality of the object and the mechanical processes associated with it.
- *knowledge about various basic underlying concepts.* In a domain of complex physical objects, such concepts might include *electricity* and *voltages*.

We define an *expert* user as one whose knowledge about the domain includes functionality of objects and mechanical processes. An expert user knows all the underlying

basic concepts and the majority of the generalized objects contained in the knowledge base for a particular domain. Given an object that is new but similar to a known one, such a user has enough domain knowledge to infer how the parts of this new object work together to perform a function. A *naive* user is one who does not know about specific objects in the knowledge base, and does not necessarily understand the underlying basic concepts.

A user is not necessarily naive or expert, however. For example, a user may know about several objects in the knowledge base. In this work, instead of rating the user as having some intermediate level of domain knowledge, we use explicit parameters to indicate the user's knowledge. These parameters are a list of items in the knowledge base which are known to the user and information about whether the user understands the underlying basic concepts. So, for example, a user model in our system may contain a parameter indicating that the user only has local expertise with respect to disk drives. We retain the terminology naive and expert only for users at the two ends of the knowledge spectrum. Thus our emphasis is on studying how object descriptions can be varied when these parameters vary. We will not try to categorize a user's domain knowledge, or attempt to determine what levels exist between the two extremes.

3 TWO DESCRIPTIONS STRATEGIES FOUND IN TEXTS: CONSTITUENCY SCHEMA AND PROCESS TRACE

3.1 THE TEXTUAL ANALYSIS

To develop effective strategies for tailoring a description to a particular level of expertise, we began by studying descriptions in a variety of texts: adult encyclopedias (*Britannica* 1984, *Collier* 1962) and junior encyclopedias (*Britannica-Junior* 1963, *New Book of Knowledge* 1967, *Encyclopedia of Science* 1982), manuals (Chevrolet 1978, Weissler 1973), and high school textbooks. This range of texts was chosen because it provided a good source of descriptions¹ and because these texts seem to address audiences at the two ends of the knowledge spectrum: naive and expert. Texts from adult encyclopedias are directed at an audience much more knowledgeable in general than the audience addressed by high school textbooks and junior encyclopedias. Likewise, the Chevrolet manual is aimed at knowledgeable users (i.e., professional mechanics), while the other manual claims to be directed towards novices. We studied descriptions of devices, taking the description of the same object in all sources whenever possible. The descriptions we have studied are generally several paragraphs in length.

Besides providing us with examples of descriptions, encyclopedias have the added advantage (for our study) that people read them for a variety of reasons. Yet, they all obtain the same texts (and therefore the same information). An encyclopedia is thus providing its readers with information about an object without taking the reader's goals into account.

We analyzed the different texts using methods devel-

{Identification (description of an object in terms of its superordinate)}²
 Attributive* (associating properties with an entity)/
 Cause-effect*
 Constituency (description of subparts or subtypes)
 {Depth-identification / Depth-attributive
 {Particular Illustration / Evidence}
 {Comparison ; Analogy} }⁺
 {Attributive / Explanation/Analogy}

Figure 1. The Constituency Schema.

oped by other researchers (Hobbs 1978a, 1980; Mann 1984, McKeown 1985), decomposing paragraphs in terms of their primitive rhetorical structure in an attempt to find consistent structures in the texts. We found that the texts fell into two groups: most of the descriptions in the adult encyclopedia entries and in the car manual for mechanics were organized around object subparts and their properties, while the descriptions in the junior entries and in the car manual for novices traced process information.

3.2 TEXTS ORGANIZED AROUND SUBPARTS

The texts from the adult encyclopedias and the Chevrolet manual can be characterized in terms of the discourse strategy the *constituency schema*, one of the textual structures posited in McKeown (1985). In her work on natural language generation, McKeown studied the problems of what to say when there are many facts to choose from and how to organize a text coherently. To this end, she examined texts and transcripts in order to determine whether there were standard patterns of discourse structure used in naturally occurring texts. Where patterns of discourse structure could be identified for various discourse goals, these patterns could be used to guide a generation system in choosing and organizing facts to construct a text.

McKeown analyzed the texts by classifying each sentence as one of a set of rhetorical predicates. Rhetorical predicates characterize the structural purpose of sentences and have been discussed by a variety of linguists (Shepherd 1926, Grimes 1975, Hobbs 1978b). Some examples are *constituency* (description of subparts or subtypes), *attributive* (providing detail about an entity or event), and *analogy*. McKeown found that in the texts she studied, some combinations of predicates were more likely to occur than others, and that for each discourse situation (such as providing a definition), some combination was the most frequent. McKeown encoded these standard combinations as schemas that are associated with a particular discourse situation. One of these schemas is the constituency schema, which is used to describe an object (or concept) in terms of its subparts and their properties. The constituency schema is shown in Figure 1.

The descriptions from the adult encyclopedias and the car manual for mechanics followed the pattern

The following text illustrates the decomposition of a description using the **Constituency Schema**.

- 1) *The hand-sets introduced in 1947* 2) *consist of a receiver and a transmitter in a single housing available in black or colored plastic.*
- 3) *The transmitter diaphragm is clamped rigidly at its edges* 4) *to improve the high frequency response.* 5) *The diaphragm is coupled to a doubly resonant system* 6) *-a cavity and an air chamber-* 7) *which broadens the response.* 8) *The carbon chamber contains carbon granules,* 9) *the contact resistance of which is varied by the diaphragm's vibration.*
- 10) *The receiver includes a ring-shaped magnet system around a coil and a ring shaped armature of anadium Permendur.* 11) *Current in the coil makes the armature vibrate in the air gap.* 12) *An attached phenolic-impregnated fabric diaphragm, shaped like a dome,* 13) *vibrates and sets the air in the canal of the ear in motion.*

Using rhetorical predicates, we can classify the sentences of the above description in the following way:

- | | |
|---|---|
| 1. Attributive | |
| 2. Constituency | |
| Depth-attributive for the transmitter (Description of the transmitter) | Depth-attributive for the receiver (Description of the receiver) |
| 3. Depth-Attributive | 10. Depth-Attributive |
| 4. Cause-effect | 11. Cause-effect |
| 5. Depth-Attributive | 12. Attributive |
| 6. Depth-identification | 13. Cause-effect |
| 7. Cause-effect | |
| 8. Depth-Attributive | |
| 9. Cause-effect | |

Figure 2. Constituency Schema Example.

shown in this schema: the descriptions are organized around the parts of the object. An example of such a description is shown in Figure 2³. This entry, describing the telephone, is taken from *Collier* (1962). In the first paragraph, the parts (constituents) of the telephone are given. Then, each main part is described in turn: first the transmitter, then the receiver. In the descriptions from this set of texts, the parts are also described with their subparts and their properties (depth-attributive).

3.3 TEXTS FROM JUNIOR ENCYCLOPEDIAS AND FROM THE CAR MANUAL FOR NOVICES

The texts from junior encyclopedias, high school textbooks, and the car manual for novices are organized in a significantly different manner. No known schema or other organizing structure consistently accounted for the descriptions in the junior encyclopedia texts. In looking for other types of organizing strategies, we discovered that the main strategy used in these descriptions is to trace through the process that allows the object to perform its function, that is, to mainly describe

I. 1) When one speaks into the transmitter of a modern telephone, these sound waves strike against an aluminium disk or diaphragm and cause it to vibrate back and forth in just the same way the molecules of air are vibrating.

II. 2) The center of this diaphragm is connected with the carbon button originally invented by Thomas A. Edison. 3) This is a little brass box filled with granules of carbon composed of especially selected and treated coal. 4) The front and back of the button are insulated.

III. 5) The talking current is passed through this box so that the electricity must find its way from granule to granule inside the box. 6) When the diaphragm moves inward under the pressure from the sound waves the carbon grains are pushed together and the electricity finds an easier path. 7) Thus a strong current flows through the line. 8) When a thin portion of the sound wave comes along, the diaphragm springs back, allowing the carbon particles to be more loosely packed, and consequently less current can find its way through. 9) So a varying or undulating current is sent over the line whose vibrations exactly correspond to the vibrations caused by the speaker's voice. 10) This current then flows through the line to the coils of an electromagnet in the receiver.

IV. 11) Very near to the poles of this magnet is a thin iron disc.

V. 12) When the current becomes stronger it pulls the disc toward it. 13) As a weaker current flows through the magnet, it is not strong enough to attract the disc and it springs back. 14) Thus the diaphragm in the receiver is made to vibrate in and out. . .

Figure 3. Description from a junior entry.

processes associated with the operation of the object. We characterize these descriptions as *process descriptions*. An example of such a description, from *Britannica-Junior* (1963), is presented in Figure 3⁴.

We see that the organizing principle of this text is the mechanical process description. The process description gets interrupted when descriptive information can be included concerning a subpart that was just mentioned as part of the process description. (Such information is shown indented in the example). Furthermore, in this text, not only is the description made mainly through a process trace, but this process trace is given in great detail and substeps are explained if there are any.

The information contained in this group of descriptions corresponds to the causal links that connect the various processes contained in the knowledge base. To generate such a description, it is then necessary to follow these links, giving rise to a process trace that

(For each object, given a chain of causal links)

(1) Follow the next causal link

(2) {Mention an important side link}

(3) {Give attributive information about a part just introduced}

(4) {Follow the substeps if there are any. (These substeps can be omitted for brevity.)}

(5) Go back to (1).

(This process can be repeated for each subpart of the object.)

Figure 4. The Process Trace.

describes how the object functions. The algorithm used for the process trace, summarized in Figure 4, is as follows: given the chain of causal links that constitutes the functional information of the object, the first link is taken. If there is an important side effect at this point, it is mentioned. If a new part was introduced when the causal link was mentioned, attributive information about the part may be included. If the step just explained can be subdivided into substeps, the trace may continue at the substeps level. Finally, the next causal link is taken and the algorithm repeats. (This strategy is described in detail in Paris and McKeown (1987) and Paris (1987).)

Substeps happen, for example, in the following case: “. . . causes the diaphragm to vibrate”. This step can be divided into the two substeps: “the diaphragm moves inward” and “the diaphragm moves outward”. Substeps can also arise when a complex object is made of several other complex parts. The strategy first describes how the parts work together to achieve the object's function. If a long description is desired, it is possible to step through each of the parts, describing how it achieves its own function. This is similar to the schema recursion for the constituency schema mentioned by McKeown (1985).

3.4 TAILORING DESCRIPTIONS

Given that the two types of descriptions occur in texts, a system that generates device descriptions should also be able to provide the two kinds of descriptions. We have thus implemented both strategies in our generation system.

We use the constituency schema when a user has expertise about the domain of discourse, giving rise to a parts-oriented description. Recall that we assumed that the goal of a description is to allow the user to form a mental model of the functionality of the object. Research in psychology indicates that expert users have more knowledge not only about individual components, but also about the causal models involved and the interconnections among parts (Lancaster and Kolodner 1987; Chi et al. 1981). Expert users, then, are likely to have functional knowledge about the domain and to know how parts might interact with each other. As they

can use this knowledge when reading the description⁵, they should be able to pull all the parts provided in the description together in order to "understand" the description as a whole. That is, they should be able to figure out how the parts fit together to form an object capable of performing a function. Since the reader is able to construct a mental model, it is unnecessary to include the process information in the description. Actually, assuming that the reader will be able to infer the processes involved, providing such useless information would contradict the principles of cooperative behavior (Grice 1975).

On the other hand, a user who does not have enough knowledge to infer the processes linking the parts would be unlikely to understand a mostly structural description of an object and to be able to construct a functional mental model of the object from such a description. For this sort of naive user, if the description is to be informative and understandable, it must describe how the parts perform the function of the object. The description must therefore include process information. Previous research in reading comprehension strengthens our belief that a user who does not have knowledge about the functions of the various parts will not be able to make sense of a description centered around parts. Wilson and Anderson (1986) demonstrate the importance of prior knowledge in comprehending new texts, and show how readers can fail to understand a text mainly because the text contains implicit knowledge that the readers do not have. We thus propose to use the process trace when providing a description to a naive user.

To summarize, we suggest that the user's domain knowledge affects the content of a description with respect to the kind of information included, and not just to the level of detail, and postulate that the choice of strategy might be based on the assumed level of expertise of reader/user. Namely, the process trace can be used when the expected readers are relatively naive about the domain of discourse, while the constituency schema can be used when the expected readers have expertise about the domain. We will show how a user's level of expertise can be incorporated in a generation system and how it can guide the system in choosing a discourse strategy.

4 MIXING THE STRATEGIES

The two strategies presented account for the main differences found between the adult and junior encyclopedia entries and we proposed to use them to describe objects to naive or expert users. Users are not necessarily either naive or expert in a domain however. They may have *local expertise*, knowing about some objects in the domain and not others (Paris 1984). Such users would not be considered naive users, but, as there are many objects they do not know in the domain, they would not be considered expert users either. The user

models for such users would indicate for which objects of the knowledge base they have local expertise. We believe that, to describe objects to users with intermediate levels of expertise, a combination of the two strategies presented for naive and expert users is appropriate. Based on the user model, a generation program can decide which strategy to use for which object.

As an example, suppose we are providing a description of an elevator to a user who knows the function of a motor and how this function is achieved, but not how the elevator itself works. In describing the elevator to this user it is necessary to first describe how the parts of the elevator work together, using the process trace strategy, since the user model would indicate local expertise about the "motor" only. This local expertise is too narrow to allow the user to understand how all the parts of the elevator work together to perform their required function. A process explanation is thus necessary. In describing the individual parts 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 mechanisms.

Such combinations were actually also found in naturally occurring texts. Figure 5 presents an example of a text that uses a combination of the constituency schema and the process trace to generate a description aimed at users with intermediate levels of domain knowledge. This text is taken from the *Encyclopedia of Chemical Technology* (Chemical 1978). 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" and the "monochromator". In Figure 5, the text corresponding to the process trace is shown in italics.

4.1 DECISION POINTS WITHIN THE STRATEGIES

Since we want to be able to combine the two strategies to generate a description aimed at users with intermediate levels of expertise, we must specify under which conditions one strategy would be preferable to the other and how to switch from one strategy to the other. The decision of which strategy to use at any point is based on information about the user's domain knowledge contained in the user model. Notice that it would be hard without a thorough psychological study to specify the exact conditions necessary to choose one strategy over the other and to switch from one strategy to the other. However, we have identified some heuristics that determine when to mix the strategies. (Should data from psychological experiments later become available, we

(1) The IR spectrometer consists of three essential features: a source of IR radiation, a monochromator and a detector. (2) The primary sources of 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 1500 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.*

Text Decomposition

1. Constituency
- 2-4. Depth identification for the IR radiation
- 5,6. Depth identification for the monochromators
7. Depth identification for the thermocouple spectrometer
8. Process trace for the thermocouple principle.

Figure 5. Text from the Encyclopedia of Chemical Technology.

could use the results as our heuristics to generate appropriate texts given the user's domain knowledge.) To decide on the strategy to use, the program looks in the user model to check whether the user is an "expert" or has a local expertise about the object to be described or about its superordinate in the generalization hierarchy. In either case, the constituency schema is used; otherwise, the process trace is chosen. This process is repeated when the program has to decide on which strategy to employ to describe the subparts.

If the user knows about most of the parts that play an important role in the mechanical process of the object⁶, it is possible to describe the object with the constituency schema instead of explicitly describing the process information that connects these parts. Note however that, in this case, providing a process trace for the top-level description, thus indicating how the parts work together, might still provide an adequate description. Currently, "most" is set to be at least half of the functionally important parts⁷.

To mix the strategies, we must specify when it might be possible to switch. Whenever an object is introduced and needs to be described, the system must decide whether to provide chiefly structural information (with the constituency schema) or functional information (with the process trace). This gives us some clear decision points in the strategies:

- Within the constituency schema:

Constituency Schema (with decision points)
Identification (introduction of the superordinate)
If there is no local expertise for the superordinate do a Process Trace (for the superordinate) before proceeding.
Constituency (description of the subparts)
For each part, do:
*If there is local expertise on this part (or its superordinate), do **Depth-identification***
Else do a Process Trace (for the part)
Attributive

Figure 6. The Constituency Schema strategy and its decision points.

- after the *identification* predicate: once the superordinate of an object has been introduced, we could provide a process trace for this superordinate.
- After the *constituency* predicate: after mentioning the parts of an object, the constituency schema dictates to fill the depth identification predicate 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 presented in Figure 5, 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 also choose to describe the part more fully with the constituency schema.
 - 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 6 and 7 summarizes the two strategies in their simplest form with the decision points.

Process Trace (with decision points)
Next causal link
Properties of a part mentioned during the process trace
If a fuller description of the part is desired, do Constituency Schema (for the part)
Substeps
Back to next causal link
 Repeat for each of the subparts:
If there is local expertise on this part (or its superordinate), do Constituency Schema
Else do a Process Trace

Figure 7. The Process Trace strategy and its decision points.

The decision to switch strategy is based on the user model: the program looks into the user model and checks the local expertise of the user. If there is no local expertise about the object, the program decides to follow the process trace; otherwise, the constituency schema is used. The user model is also examined to decide whether substeps should be traced: if the substeps involve any basic underlying concepts and the user model indicates that the user knows these concepts, the substeps are traced. Otherwise, the program does not trace through the substeps, as they might confuse the user. In that case, after the description has been generated, the user is asked whether he or she would like to see the substeps, although they might involve unknown concepts.

Figure 8 shows an example of a text generated by TAILOR by combining the two strategies. More examples can be found in Paris (1987.) 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, or about the microphone, its superordinate in the generalization hierarchy. Thus TAILOR chooses to provide process information for the transmitter, switching momentarily to the process trace strategy. The process trace is shown underlined in the figure.

5 TAILOR

5.1 OVERVIEW OF THE SYSTEM

We have implemented the discourse strategies presented above in TAILOR, a program that generates descriptions tailored to a user's level of expertise. The discourse strategies guide the program to choose the appropriate information from the knowledge base, under the constraint of the user model. TAILOR generates descriptions aimed at users anywhere along the knowledge spectrum. TAILOR uses the knowledge base built by RESEARCHER, as depicted in Figure 9, and looks at the information contained in the user model to decide on the strategy to employ. After generating a description, TAILOR updates the user model based on the objects that were included in the description provided to the user. At this point, no parsing of the questions is done (the input consists of a request for a description). In the ideal system, RESEARCHER would parse the question using the same parser as that used in reading patent abstracts, produce the request, and hand it to TAILOR. The question (along with other factors) could also be used to determine to the level of expertise. The user model includes the parameters that TAILOR uses to constrain its decision process during generation. The

Description of the telephone, given a user model

User Model: Local Expertise: Loudspeaker (a pointer to the concept of a loudspeaker in the knowledge base); Basic concepts: electricity

The telephone has two main functional parts: the transmitter (an instance of a microphone) and the receiver (an instance of a 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 subpart, the system consults the user model and decides to switch strategy to describe the transmitter since the user has no local expertise about it.

TAILOR output:

The telephone is a device that transmits soundwaves. The telephone has a housing that has various shapes and various colors, a transmitter that changes soundwaves into current, a curly-shaped cord, a line, a receiver to change current into soundwaves and a dialing mechanism. The transmitter is a microphone with a small diaphragm. 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 vibration of the diaphragm 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 and it contains the receiver. The housing is connected to the dialing mechanism by the cord. The line connects the dialing mechanism to the wall.

Figure 8. A description generated by TAILOR combining the two strategies.

user model is not determined by the program but is given as input.

RESEARCHER's knowledge base contains detailed descriptions of complex devices, including both structural and functional information about the objects. We use a frame-based knowledge representation (Wasserman and Lebowitz 1983, Wasserman 1985) in which the basic frames represent objects. The knowledge base contains about 120 object frames and 150 frames of other types. Objects are organized in a generalization hierarchy. In addition to the generalization links, or *instance-of* links, there exist two additional kinds of links joining entities: *part-of* links, which indicate that an entity is a part in a larger structure, and *relations*, which convey information about spatial or functional relationships. Functional relations corresponds to the various events (or processes) that occur. Finally, there are links between relations, that is links between events. These links include cause-effect relations, temporal relations (such as "X happens at the same time as

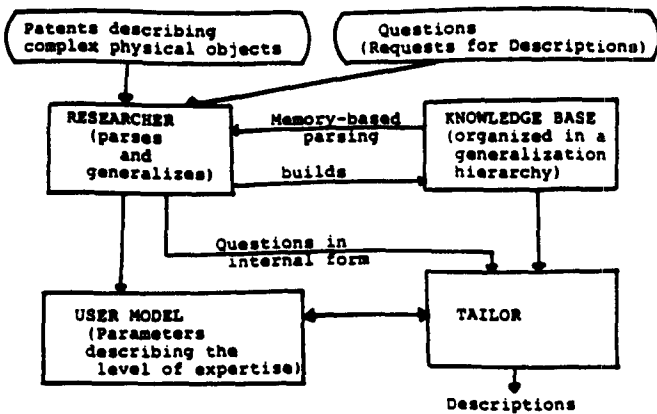


Figure 9. RESEARCHER and the TAILOR System.

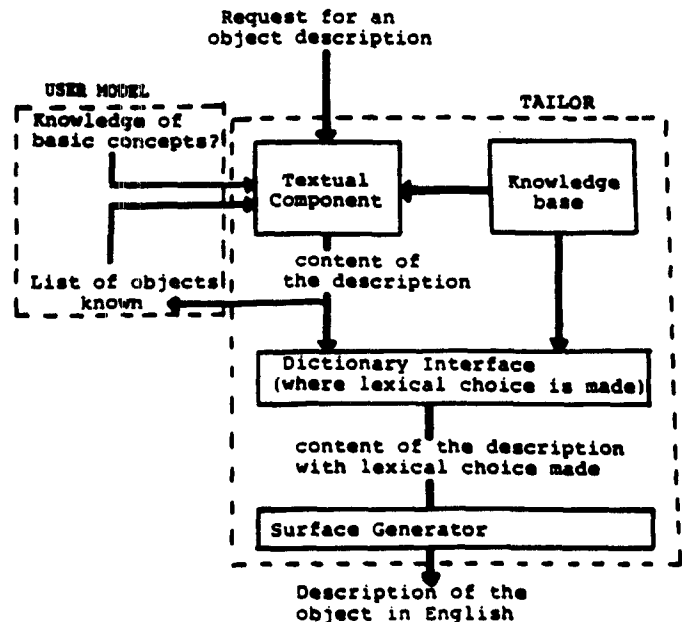


Figure 10. The TAILOR System.

Y''), and analogical relations (such as "X corresponds to Y").

A top-level diagram of TAILOR is shown in Figure 10. Input to TAILOR includes: 1. a request for the definition of an object, and 2. the level of expertise of the user, that is, either one of the two stereotypes naive or expert, or, for users with intermediate levels of expertise, the set of parameters that describe the user's level of knowledge about the domain, including:

- A list of basic underlying concepts that are important in the domain of the knowledge base and that the user understands.
- The specific objects the user knows in the domain, that is the user's local expertise (a list of pointers into the knowledge base).

The textual component of the system decides what to include in the description. It looks into the knowledge base and, based on the user model and the discourse strategies, chooses appropriate facts. 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 uses the focus of a proposition and the past discourse to guide its decision process. (However, as our emphasis in this work is on the *content* of a description as opposed to its *phrasing*, we have not studied in depth the complexity and subtleties of lexical choice.) Finally, a surface generator constructs English sentences. The surface generator is based on the one used by McKeown (1985) in the TEXT system. This generator unifies the input with a functional grammar (Kay 1979) to produce English sentences. We have extended and improved the performance of this program, and augmented the functional grammar it uses (Paris and Kwee 1985; McKeown and Paris 1987).

5.2 IMPLEMENTATION OF THE STRATEGIES

The constituency schema and the process trace strategies are implemented using an augmented transition network (ATN) (Woods 1973). The arcs joining the various nodes in the network specify what information is to be retrieved from the knowledge base, under what conditions (the arcs contain a test), and which node to go to next. Figures 11 and 13 present the nets used for the strategies.

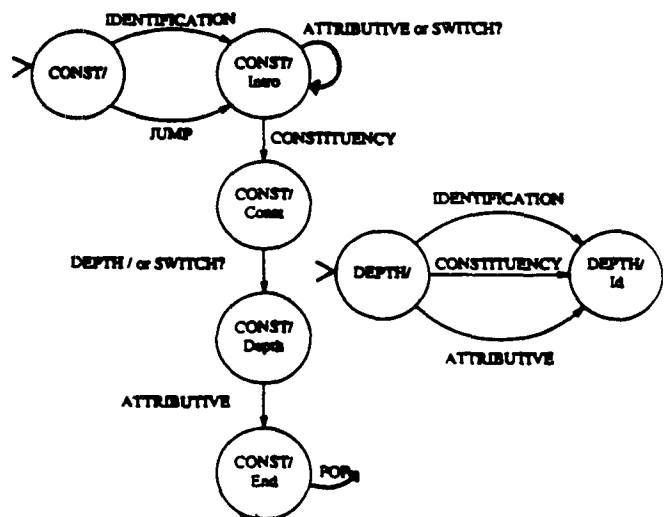


Figure 11. The Constituency Schema.

In the constituency schema, shown in Figure 11, the arcs correspond to the predicates from the schema. (See McKeown (1985) for details of a similar system.) These

predicates define the type of information to be taken from the data base. They are:

- *identification*, presenting the more general concept of which the present object is an instance;
- *constituency*, giving the components of an entity, if there are any;
- *attributive*, providing different attributes of an object (such as its shape or material); and
- *cause-effect*, providing some causal relations between entities or relations.

The process of filling the ATN for the constituency schema strategy to describe a microphone is shown in Figure 12. The constituency schema would be chosen by TAILOR if the user model exhibits local expertise about the microphone (or if the user is classified as an expert). In this example, the identification predicate is first applied to the microphone. The identification predicate provides the superordinate of the object together with the function of the object. The constituency predicate is then applied, providing the subparts of the microphone, together with their properties or purposes. Finally, the depth-identification predicate is applied to each subpart, the doubly-resonant system and the dia-

;Stepping through the Constituency Schema to describe ; a MICROPHONE. (English noun-phrases are used here for clarity sake.)

Applying the predicates to MICROPHONE:

Identification
 predicate: DEVICE; (used-for:
 change soundwaves into current)

Constituency
 predicate: DIAPHRAGM
 (shape disc, material aluminium),
 DOUBLY-RESONANT-SYSTEM
 (used-for: broaden response)

Depth-constituency for DOUBLY-RESONANT-SYSTEM:

CARBON-CHAMBER, AIR-CHAMBER

Depth-attributive for DIAPHRAGM:

(edges clamped)

TAILOR output:

The microphone is a device that changes soundwaves into current. It has a disc-shaped aluminium diaphragm and a doubly-resonant system to broaden the response. The system has a carbon chamber and an air chamber. The diaphragm is clamped at its edges.

Figure 12. Stepping through the Constituency Schema.

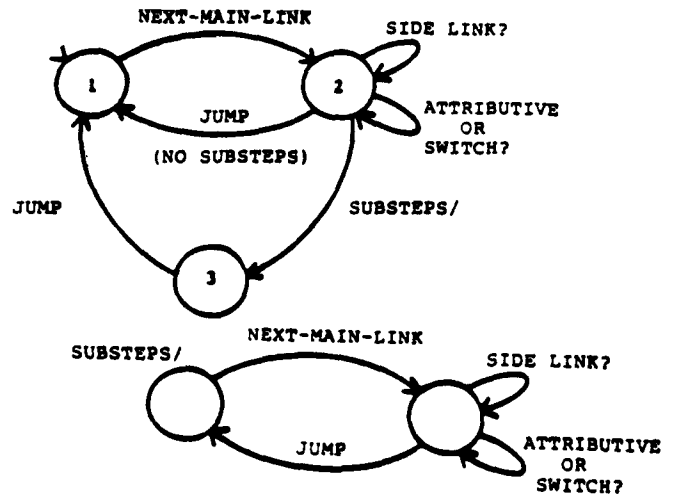


Figure 13. The Process Trace.

phragm. The English output shown in the figure (and in the following figures) is the actual output from TAILOR.

The network for the process trace is shown in Figure 13. An example of following the process trace strategy is presented in Figure 14. In this network, the arcs dictate how to trace the knowledge base to form an answer, but they are not linguistic predicates as in the network corresponding to the constituency schema. They mainly dictate how to follow the causal links in the knowledge base. (Details about the process trace can be found in Paris and McKeown (1987) and Paris (1984)). 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 that 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.⁸
- *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* is called for each substep. This subroutine is very similar to the main graph, but does not allow for a further decomposition of events.

In Figure 14, the link "the diaphragm vibrates" can be divided into substeps, namely "the diaphragm goes forward" and "the diaphragm goes backward".

;The program traces the process information for a naive user

Causal link(1): {M-CAUSES} relates the two relations:

[ONE] P-SPEAKS-INTO [MICROPHONE]
[SOUNDWAVES] P-HITS [DIAPHRAGM]

Attributive Information about DIAPHRAGM: [material: aluminium]
[shape: disc]

Causal link(2): {M-CAUSES} relates the two relations:

[SOUNDWAVES] P-HITS [DIAPHRAGM]
[DIAPHRAGM] P-VIBRATES

;Substeps: "Soundwaves increasing" and "soundwaves decreasing"

Substep1 {M-CAUSES} relates the two relations:

[SOUNDWAVES] P-INCREASES [DIAPHRAGM]
[DIAPHRAGM] P-SPRING [Direction: forward]

{M-CAUSES} relates the two relations:

[DIAPHRAGM] P-SPRING [Direction: forward]
[GRANULES] P-COMPRESS

{M-CAUSES} relates the two relations:

[GRANULES] P-COMPRESSES
[RESISTANCE] P-DECREASES

{M-CAUSES} relates the two relations:

[RESISTANCE] P-DECREASES
[CURRENT] P-INCREASE

Substep2 {M-CAUSES} relates the two relations:

[SOUNDWAVES] P-DECREASES [DIAPHRAGM]
[DIAPHRAGM] P-SPRING [Direction: backward]

...

; [The remainder of the trace is omitted here for brevity.]

TAILOR output:

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. When the intensity of the soundwaves increases, the diaphragm springs forward. This causes the granules of the button to be compressed. The compression of the granules causes the resistance of the granules to decrease. This causes the current to increase. Then, when the intensity decreases, the diaphragm springs backward. This causes the granules to be decompressed. The decompression of the granules causes the resistance to increase. This causes the current to decrease. The vibration of the diaphragm causes the current to vary. The current varies like the intensity varies.

Figure 14. Tracing the process information for a naive user.

The process trace can be continued at the substep level. Then, once the substeps have been traced through, the trace returns to the top-level description. (Only one substep is fully shown in Figure 14 for brevity. See (Paris 1985) for details.) We can also choose to not follow the substeps in order to generate a shorter description. This factor is incorporated into the arc test.

By representing the two strategies in this formalism, we immediately obtain the control structure 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 thus marked as special tests on the arcs joining nodes.

As a general test for deciding which strategy to use to describe a part, TAILOR looks into the user model to check if a superordinate of the part (or the part itself) is known to the user. If the part is known, the constituency schema is used. Otherwise, the process trace is chosen. This test is invoked before beginning generation and at any point in the schema where it is possible to switch strategies. The test also checks on the length of the discourse planned so far and the number of parts to avoid generating overwhelmingly long texts. The process of stepping through the ATN and switching strategy is shown in Figure 15. (The corresponding generated text was shown in Figure 8.) Note that to avoid generating very long texts, substeps are omitted when the process trace is chosen to describe a subpart. After the

User Model: Local Expertise: Loudspeaker
 ; Stepping through the Constituency Schema to describe a TELEPHONE.
 ; Switching to the Process Trace to describe the TRANSMITTER.

Applying the predicates to TELEPHONE:
 Identification predicate: DEVICE; (used-for:
 change soundwaves into soundwaves)
 Constituency predicate: DIALING MECHANISM
 TRANSMITTER
 (used-for: change soundwaves into current)
 LINE
 CORD
 RECEIVER
 (used-for: change current into soundwaves)
 HOUSING
 (properties: color: various, shape: various)

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

Causal link (1): {M-CAUSES}
 relates the two relations:
 [ONE] P-SPEAKS-INTO [TRANSMITTER]
 [SOUNDWAVES] P-HITS [DIAPHRAGM]

Causal link (2): {M-CAUSES}
 relates the two relations:
 [SOUNDWAVES] P-HITS [DIAPHRAGM]
 [DIAPHRAGM] P-VIBRATES

[Substeps 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-contains TRANSMITTER
 HOUSING r-contains RECEIVER
 HOUSING r-connected-to DIALING MECHANISM
 by CORD

Applying the predicates to LINE:
 Attributive: DIALING MECHANISM r-connected-to WALL
 by LINE

Figure 15. Switching strategy.

initial description is generated, a mechanism allows the user to request the information that was omitted for brevity sake (see Paris 1987).

For each object of the knowledge base, TAILOR can 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. The ability to combine strategies allows TAILOR to generate a greater variety of texts than would otherwise be possible. For example, starting with the constituency schema as the initial strategy for describing the pulse-telephone, which has one superordinate

and three functionally important parts, TAILOR can generate eight different descriptions depending on various user models. This ability also allows TAILOR to generate a description tailored to any individual user model.

6 FURTHER WORK AND RELATED ISSUES

We are extending our work by examining the recursive use of both strategies. Currently, TAILOR can call the strategies recursively for each subpart of the top level object being described, but not for the components of these subparts. Similarly, the process is only traced one level down, that is only the top-level relations are broken into substeps. This kind of recursion 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 plan to implement a mechanism that would allow the user to request such a description, either initially or after a description has already been provided by the system and the user wants additional information. As we already mentioned, we think that the user's domain knowledge, in particular knowledge of basic concepts, plays a role in determining the depth needed. 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.

6.1 DETERMINING THE LEVEL OF EXPERTISE

In this work, we have not addressed the issue of determining the level of expertise of the user. This is obviously an important question that needs to be studied. We believe that it is possible to infer the user's level of expertise. Some relevant factors are:

- *The user type.* Some classes of users may be likely to be naive while others may be likely to be expert. Identifying a user as part of these classes can give insight into how much that user might know, and provide a starting point. This is similar to using stereotypes to model the user (Rich 1979; Chin 1986).
- *The question type.* The kind of question asked ("What is a tape recorder?" as opposed to "Does this disk drive have three bearings?") can give further information about how much the user knows.
- *The depth in the knowledge base* (both in the components tree and the generalization hierarchy) of the object of the question ("Describe a disc drive" as opposed to "Describe the track assembly in a disk drive")(Paris 1984).
- *The previous discourse.* If a user asks the same question twice, it is indicative that the answer was not understood, perhaps because it assumed knowledge the user did not have.

It would be necessary to study how the user's domain knowledge can be inferred from these factors and how these factors affect each other. This seems to be a hard problem.

Finally, our user model at this point is coarse grained, in that it contains a list of objects and concepts that the user knows. A more detailed model might include exactly which facts the user knows about objects, and specifically which basic concepts are understood. We have shown that a system benefits from a user model that indicates a user's knowledge about the domain, even when the model is not a detailed one. While we feel that a detailed user model would be much harder to obtain, it would be interesting to see whether such a model would allow a system to provide more appropriate answers.

6.2 INCORPORATING THE USER'S GOAL

It is clear that the goal of the user also influences the content of an answer, or of a description. It would be interesting to examine how the goal and the level of expertise combined can help a program in choosing appropriate facts from the knowledge base. In this research, we found that the level of expertise affects the kind of information to include in a description. This is also clearly true for a user's goal: the information provided in the description should help the user in pursuing his/her goal. Thus, depending on a user's goal, the information included would vary. The interaction of these two factors can become quite complex. However, until both factors are fully understood individually, we feel that it will be very hard to determine exactly how they interact.

6.3 FEASIBILITY AND EXTENSIBILITY OF THIS APPROACH

In this work, we make the assumption that a system that tailors its answer to its users will be most useful. This is true only provided that this tailoring does not hinder the system's performance or increase its complexity significantly. We argue that tailoring a description to a user's level of expertise using the method described above will not add to the cost of generation and yet might provide better answers. Whether a generation system tailors its answers to users or not, it needs to employ a discourse strategy to guide its decision process lest the resulting text be incoherent. The two discourse strategies used by TAILOR are of comparable complexities, and there is not much cost added to combine them.

While this work was done only with respect to generating descriptions of complex devices, we think this approach will be useful in any information seeking environment to which users with different background and knowledge levels have access. Such environment could be a large knowledge base of facts (of the kind of an encyclopedia), or a help system. Providing different information in an answer might also be done in explaining the behavior of an expert system that is used both as a teaching tool and as a problem solving engine.

7 CONCLUSIONS

In this paper we have demonstrated that the user's domain knowledge can be used as a factor in tailoring an answer. In particular, we have shown how the description of a complex physical object might be tailored to a user's level of expertise. We presented different kinds of knowledge users can have, explaining how a system can take them into consideration in order to generate a description. From our studies of texts, we have found two distinct discourse strategies that are used in describing complex devices. We postulated that the level of expertise of the user affects the kind of information given as opposed to just the amount of detail provided. Even though we conducted this study in the domain of complex physical objects, we believe this result can extend to other domains. We thus propose that a user model containing information about the user's domain knowledge can be used in a question answering system to guide the decision process. We presented the two distinct descriptive strategies that can be used in a question answering program and showed how they can be mixed to include the appropriate information from the knowledge base, based on the information contained in user model.

Finally, we presented TAILOR, a program that generates descriptions tailored to users with various levels of expertise. TAILOR employs one of the two discourse strategies described to generate a text for a novice or an expert. TAILOR is also able to automatically mix the strategies to provide device descriptions tailored to users whose domain knowledge fall anywhere along the knowledge spectrum. By representing explicitly the user's domain knowledge in terms of parameters, TAILOR does not require an a priori set of stereotypes but can provide wide variety of descriptions for a whole range of users.

ACKNOWLEDGMENTS

Many thanks go to Kathleen McKeown and Michael Lebowitz for their help in both the research and the writing of this paper, and to Tjoeliong Kwee for his work on the functional grammar. We also thank the anonymous reviewer for his/her useful comments.

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

- Anderson, R.C.; Spiro, R.J.; and Anderson, M.C. 1977 *Schemata as scaffolding for the representation of information in connected discourse*. Technical Report 24, Center for the Study of Reading, Urbana, IL.
- Appelt, D.E. 1982 Planning Natural Language Utterances. In *Proceedings of the Second National Conference on Artificial Intelligence*, Pittsburgh, PA: 59-62.
- Appelt, D.E. 1985 *Planning English Sentences*. Cambridge University Press, Cambridge, England.
- The New Encyclopedia Britannica* 1984 Encyclopedia Britannica Incorporation.
- Britannica Junior Encyclopedia* 1963 Encyclopedia Britannica Incorporation, William Benton Publisher.
- Carberry, S. 1983 Tracking User Goals in an Information-Seeking Environment. In *Proceedings of the American Association of Artificial Intelligence*.
- Carberry, S. (this issue) Plan Recognition and User Modeling. *Computational Linguistics*. Computation on User Modeling, 1988.
- Kirk and Othmer, (eds.) 1978 *Encyclopedia of Chemical Technology*. John Wiley & Sons, New York, NY.
- Chevrolet Service Manual 1978 General Motors Corporation.
- Chi, M.T.H.; Glaser, R.; and Rees, E. 1981 Expertise in Problem Solving. *Advances in the Psychology of Human Intelligence*. Lawrence Erlbaum, Hillsdale, NJ.
- Chin, D.N. 1986 User Modeling in UC, the UNIX Consultant. In *Proceedings of Computer Human Interactions*.
- William Halsey (ed.) 1962 *Collier's Encyclopedia* The Crowell-Collier Publishing Company.
- The New Encyclopedia of Science* 1982 Raintree Publishers.
- Grice, H.P. 1975 Logic and Conversation. In Cole P. and Morgan J.L. (eds.), *Syntax and Semantics*. Academic Press, New York, NY.
- Grimes, J.E. 1975 *The Thread of Discourse*. Mouton, The Hague, Paris, France.
- Hobbs, J. 1978 *Why is a Discourse Coherent?* Technical Report 176, SRI International, Menlo Park, CA.
- Hobbs, J. 1978 *Coherence and Coreference*. Technical Note 168, SRI International, Menlo Park, CA.
- Hobbs, J. and Evans, D. 1980 Conversation as Planned Behavior. *Cognitive Science*, 4(4): 349-377.
- Hoepfner, W.; Morik, K.; and Marburger, H. 1984 *Talking it Over: The Natural Dialog System HAM-ANS*. Technical Report ANS-26, Research Unit for Information Science and Artificial Intelligence, University of Hamburg, W. Germany.
- Kaplan, S.J. 1982 Cooperative Responses from a Portable Natural Language Query System. *Artificial Intelligence* 2(19).
- Kay, Martin 1979 Functional Grammar. In *Proceedings of the 5th meeting of the Berkeley Linguistics Society*, Berkeley, CA.
- Lancaster, J.S., and Kolodner, J.L. 1987 *Problem Solving in a Natural Task as a Function of Experience*. Technical Report, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA.
- Lebowitz, M. 1983 RESEARCHER: An Overview. In *Proceedings of the Third National Conference on Artificial Intelligence*. American Association of Artificial Intelligence, Washington, DC.
- Lebowitz, M. 1985 RESEARCHER: An experimental intelligent information system. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA: 858-862.
- Lebowitz, M. 1986 An experiment in intelligent information systems: RESEARCHER. *Intelligent Library and Information Systems*, Ellis Horwood, London, England.
- Lehnert, W. 1977 A Conceptual Theory of Question Answering. In *Proceedings of the International Joint Conferences of Artificial Intelligence*, Cambridge, MA.
- Mann, W.C. 1984 *Discourse Structure for Text Generation*. Technical Report ISI/RR-84-127, Information Sciences Institute, Marina del Rey, CA.
- McCoy, K.F. 1983 Correcting Misconceptions: What to say When the User is Mistaken. In *Proceedings of the Conference on Human Factors in Computing Systems*. Human Factors, Boston, MA.
- McCoy, K.F. 1986 The ROMPER System: Responding to Object-Related Misconceptions Using Perspective. In *Proceedings of the 24th Annual Meeting of the Association of Computational Linguistics*, New York, NY.
- McCoy, K.F. (this issue) Reasoning on a Highlighted User Model to Respond to Misconceptions. *Computational Linguistics* 1988.
- McKeown, K.R. 1985 *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge, England.

- McKeown, K.R.; Wish, M.; and Matthews, K. 1985 Tailoring Explanations for the User. In *Proceedings of the Ninth International Joint Conferences on Artificial Intelligence*, Los Angeles, CA.
- McKeown, K.R. and Paris, C.L. 1987 Functional Unification Grammar Revisited. In *Proceedings of the 25th Annual Meeting of the Association of Computational Linguistics*, Palo Alto, CA.
- The New Book of Knowledge-The Children's Encyclopedia* 1967 Grolier Incorporation, New York, NY.
- Paris, C.L. 1984 Determining the Level of Expertise. In *Proceedings of the First Annual Workshop on Theoretical Issues in Conceptual Information Processing*. Atlanta, GA.
- Paris, C.L. 1985 Description Strategies for Naive and Expert Users. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*. Chicago, IL.
- Paris, C.L. and Kwee, T.L. 1985 *Guide to the Unification Process and its Implementation; Progress Report on Extending the Grammar*. Technical Report, Computer Science Department, Columbia University, New York, NY.
- Paris, C.L. and McKeown, K.R. 1986 Discourse Strategies for Describing Complex Physical Objects. Paper presented at the Third International Workshop on Natural Language Generation, Nijmegen, The Netherlands. In G. Kemper (ed.), *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*. Kluwer Academic Publishers, Boston, MA/Dordrecht, Holland.
- Paris, C.L. 1987 The Use of Explicit User Models in Text Generation: Tailoring to a User's Level of Expertise. Ph.D. thesis. Columbia University, Department of Computer Science, New York.
- Quilici, A.; Dyer, M.; and Flowers M. (this issue) Providing Explanatory Responses to Plan-Oriented Misconceptions. *Computational Linguistics* 1988.
- Rich, E.A. 1979 User Modeling Via Stereotypes. *Cognitive Science* 3: 329-354.
- Shepherd, H.R. 1926 *The Fine Art of Writing*. Macmillan Co., New York, NY.
- Tennant, H. 1978 *The Evaluation of Natural Language Question Answerers*. Technical Report, University of Illinois at Urbana/Champaign Ph.D. proposal, Department of Computer Science, Advanced Automation Group, Coordinated Science Laboratory.
- Wallis, J.W. and Shortliffe, E.H. 1982 *Explanatory Power for Medical Expert Systems: Studies in the Representation of Causal Relationships for Clinical Consultation*. Technical Report STAN-CS-82-923, Heuristics programming Project, Department of Medicine and Computer Science, Stanford University, Stanford, CA.
- Wasserman, K. 1985 *Unifying Representation and Generalization: Understanding Hierarchically Structured Objects*. Department of Computer Science, Ph.D. thesis, Columbia University, New York, NY.
- Wasserman, K. and Lebowitz, M. 1983 Representing Complex Physical Objects. *Cognition and Brain Theory* 6(3): 333-352.
- Weissler, A. and Weissler, P. 1973 *A woman's guide to fixing the car*. Walker and Company, New York, NY.
- Wilson, P.T. and Anderson, R.C. 1986 What They Don't Know Will Hurt Them: The Role of Prior Knowledge in Comprehension. *Reading Comprehension: From Research to Practice*. Erlbaum, Hillsdale, NJ.
- Woods, W. 1973 An Experimental Parsing System for Transition Network Grammars. In Rustin, R. (ed.), *Natural Language Processing*. Algorithmics Press, New York, NY.

NOTES

1. We hope that by choosing several sources, stylistic differences on our results are minimized. We studied about 15 examples from each encyclopedia and textbook and a few from the manuals.
2. We are using McKeown's notation: "{}" indicate optionality, "/" alternatives, "+" that the item may appear 1 or more times, and "*" that the item may appear 0 or more times. Finally, ";" is used to indicate that the propositions could not be clearly classified as corresponding to one predicate. We changed McKeown's schema slightly, by adding the identification predicate as an option for the first predicate of the schema.
3. The original entry was in one paragraph only. We divided it into three paragraphs for clarity. More details about this analysis are given in Paris (1985).
4. The original entry contained two paragraphs. The second one has been divided for clarity.
5. Research in reading comprehension indicates that readers indeed use their previous knowledge in order to understand new texts (Anderson et al. 1977; Wilson and Anderson 1986).
6. Some parts do not play an important part on the mechanical process associated with the object. For example, the housing of the telephone does not have an important role in the functionality of the telephone. Such a part would not be involved in one of the causal links contained in the knowledge base.
7. We do not claim that this is the optimal value for the threshold indicating at which point the constituency schema should be used. This threshold cannot be set with certitude without further experimentation. Also note that, while these heuristics allow the system to generate reasonable descriptions given the user's domain knowledge, there is no clear "best" descriptions.
8. We have mentioned that there are different kinds of links between events. The types of links are ranked in order of importance, and the most important ones are mentioned. There are actually different types of side-links. These are not indicated in the figure for simplicity. The reader is referred to Paris (1987) for details.