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

Figure 3. Description of a telephone from an adult encyclopedia

and the receiver in sentences 9 to 13. Providing information about the subparts of an object is termed giving depth-attributive information. In this text, *depth-attributive* information includes the use of both the constituency and attributive predicates.

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 these descriptions. Moreover, identifying the rhetorical predicates of the sentences did not provide insights to a useful organizing structure. The structure resulting from using rhetorical predicates is one that contains mainly cause-effect relations. Although this structure is a simple schema, it is not useful for generation as it does not provide enough constraints in deciding what to include in a text. In looking for other types of organizing strategies, we discovered that the main strategy used in these descriptions is to trace through

	Text	Predicates
1)	When one speaks into the transmitter of a modern telephone, these sound waves strike against an aluminum disk 2) or diaphragm	Cause-effect renaming
3)	and cause it to vibrate back and forth	Cause-effect
4)	in just the same way the molecules of air are vibrating.	Cause-effect
5)	The center of this diaphragm is connected with the carbon button	Attributive
6)	originally invented by Thomas A. Edison.	Elaboration
7)	This is a little brass box filled with granules of carbon	Attributive
8)	composed of especially selected and treated coal.	Attributive
9)	The front and back of the button are insulated.	Attributive
10)	The talking current is passed through this box so that the electricity must find its way from granule to granule inside the box.	Cause-effect
11)	When the diaphragm moves inward under the pressure from the sound waves the carbon grains are pushed together	Cause-effect
12)	and the electricity finds an easier path.	Cause-effect
13)	Thus a strong current flows through the line.	Cause-effect
14)	When a thin portion of the sound waves comes along, the diaphragm springs back,	Cause-effect
15)	allowing the carbon particles to be more loosely packed, and	Attributive
16)	consequently less current can find its way through.	Cause-effect
17)	So a varying or undulating current is sent over the line whose vibrations exactly correspond to the vibrations caused by the speaker's voice.	Cause-effect
18)	This current then flows through the line to the coils of an electromagnet in the receiver.	Cause-effect
19)	Very near to the poles of this magnet is a thin iron disc.	Attributive
20)	When the current becomes stronger it pulls the disc toward it.	Cause-effect
21)	As a weaker current flows through the magnet, it is not strong enough to attract the disk and it springs back.	Cause-effect
22)	Thus the diaphragm in the receiver is made to vibrate in and out	Cause-effect

Figure 4. Description of a telephone from a junior encyclopedia

the process that allows the object to perform its function, that is, to mainly describe processes associated with the operation of the object. We characterize these descriptions as *process descriptions*. An example of such a description, from [BRIT63], is presented in Figure 4.

In this description, the author starts by explaining what happens when someone speaks into a telephone (sentence 1), and why a diaphragm located inside the

transmitter vibrates (3). While introducing the diaphragm as part of a causal link, the author also includes properties about the diaphragm: it is "aluminum" and "disc-shaped." Now that the diaphragm has been introduced as part of the process, it is described more fully in sentences (5) through (9) with various properties and structural relations. The description then reverts to tracing the process information, explaining how the soundwaves are transformed into varying current and describing how the telephone achieves its function of transmitting soundwaves.

As the vibration process in (3) can be decomposed into two substeps (i.e., "the diaphragm moves inward" and "the diaphragm springs back"), the author traces each one in turn (sentences (11) through (13) for the first substep, and (14) through (16) for the second). Sentence (17) summarizes these substeps, also indicating that the process trace will now continue at the top level, not the substep level. Sentence (18) continues the process trace by describing where current flows next. In the remaining text, the author explains how the varying current is changed back into soundwaves in the receiver of the telephone.

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. 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 describes how the object functions. The algorithm used for the process trace, summarized in Figure 5, 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

---

[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 5. The process trace

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. (The knowledge base contains various kinds of links between events, each kind having a different importance factor. TAILOR has an algorithm for automatically finding the main path for a given object. See [PARI 87a] for details.)

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 [MCKE85b].

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.

As a result of the difference in organizing structures, the type of information that is included in the descriptions is also different: in the texts from the adult encyclopedias and the manual for experts, the information included is mainly structural, while in the texts from the junior encyclopedias and the manual for novice the information included is mainly functional (or process oriented). One of tl main differences between the two audiences at which the two groups of texts we aimed is their assumed level of domain knowledge. We postulate that the writers choice of strategy might be based on the assumed domain knowledge of the expected readers.[5] If so, the reader's level of knowledge about the domain affects the *kind* of information provided as opposed to just the *amount* of information, as was previously assumed [WALL82, SLEE82b]. This is significant as it adds a dimension (the different kinds of information available) along which a system can tailor its answers to users having different levels of domain knowledge.

In TAILOR, the constituency schema is used 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 [LANC87, CHI81]. 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

[5] No psychological experiments have been conducted to confirm this hypothesis. It is based on observations about what happens in natural texts.

when reading the description,[6] 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 was 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 [GRIC75].

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 [WILS86] 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 the 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 now 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 [PARI84]. 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

[6] Research in reading comprehension indicates that readers indeed use their previous knowledge in order to understand new texts [ANDR77, WILS86].

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 6 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 [MARH87]. 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, th "IR radiation" and the "monochromator". In Figure 6, the text corresponding t the process trace is in italics.

---

(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 6. Naturally occuring text combining the two strategies

4.1. Generating a Description Based on the User Model

Before starting to generate a text, TAILOR must decide whether to use the constituency schema or the process trace for the overall framework of the text. While constructing the description, TAILOR can switch to the other strategy, thus combining the two strategies in one text. This is done for users with intermediate levels of expertise. There is therefore a need to specify 1) how to choose an initial strategy and 2) how to combine the strategies. In TAILOR, the decision on which strategy to use at any point is based on the user model.

4.1.1. Choosing a Strategy for the Overall Structure of the Description

To choose the initial strategy, TAILOR looks into the knowledge base to see whether it is possible to describe the object using either strategy. In particular, it checks whether there is process information associated with the object. If the object has no such information in its frame, the constituency schema is chosen by default. This will happen in two cases: first, when the object's function cannot be broken down into a sequence of events (for example, a "cover" has no mechanism associated with it), and second, when the object's function and the way it is performed is identical to that of its superordinate in the generalization hierarchy (for example, the function of the *pulse telephone* is the same as that of a *telephone*). Process information in this case is stored in the frame of the superordinate. The pulse telephone frame contains the part that differs from the telephone, namely the pulse dialer. Although the constituency schema will be chosen for the initial strategy of the description of a pulse telephone, it will be possible to switch to the process trace to explain the function of its superordinate. Similarly, it will be possible to switch to the process trace to explain the function of the "new" (or different) part, i.e., the pulse dialer.

If there was process information (or a mechanism) associated with the object, it is possible to describe the object with either strategy. TAILOR examines the user model to check the user's local expertise. If the user model does not indicate any local expertise about either the object to be described or its superordinate in the generalization hierarchy, the process trace is chosen. Otherwise, the constituency schema is chosen. (If local expertise is indicated about the superordinate of the object, the user will most likely be able to infer the processes for the object given his or her knowledge about the superordinate; therefore, the constituency schema is still adequate for the description.)

The constituency schema is also chosen when the user model indicates local expertise about all or most of the functionally important subparts of the object to be described.

A distinction is made between parts that play a significant role in the mechanical process of a device and other parts, because knowledge about the former will help the user to comprehend the mechanism of the device. For example, the *housing* of a telephone does not play an important role in allowing the telephone to transmit soundwaves. Knowing about the *housing* or about a *cover*, the housing superordinate in the generalization hierarchy, does not help a user understand how the telephone works. Understanding how a microphone and a loudspeaker work,

on the other hand, is useful since the transmitter of a telephone is essentially a microphone, and the receiver a loudspeaker.

If the user already knows about most of the functionally important subparts of a device, he or she is likely to be able to infer how the parts fit together functionally, so it is not necessary to include a process description for the object. The overall description of a text in that case will be based on the constituency schema. "Most" is represented in TAILOR as a parameter that can be set to any value for experimental purposes. Its particular value in the current implementation is half the number of functionally important parts.[7] If the user does not have local expertise about most of the subparts of the device, the process trace is chosen, and a process explanation of how the subparts function together is provided.

To summarize, the constituency schema is chosen as the overall structure of the text in the following cases:

1. The object has no mechanism associated with its frame in the knowledge base.
2. The user model indicates local expertise about either the object or its superordinate in the generalization hierarchy.
3. The user model indicates that the user knows about most of the functionally important parts of the object.

The process trace is chosen as the initial strategy in all other cases. The basic decision algorithm is shown in Figure 7.

---

1) Is there a mechanism associated with the object to be described?

No: Use the constituency schema
Yes: 2) Is the object to be described (or its superordinate) in the user model?
(i. e., does the user have local expertise about this object or its superordinate?)

Yes: Use the constituency schema
No: 3) Collect all the functional parts of the object.
If the user has local expertise about most of these parts,
use the constituency schema
Otherwise use the process trace.

Figure 7. The decision algorithm for the initial strategy

Whether the constituency schema or the process trace is chosen as the overall strategy for the description, a mixture of structural and process information can still be provided depending on the user model. For example, if the constituency schema

[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 are no clear "best" descriptions.

is chosen initially, the process trace can be used for the unknown parts, providing the user with a process explanation for these parts. Likewise, if the process trace is taken at first, structural information about the parts already known to the user is still included. "Most" only determines the initial strategy (i.e., the overall structure of the text). It does not determine whether a mixture of structural and functional information will be included.

4.1.2. Decision Points Within the Strategies

To generate a description aimed at users with intermediate levels of expertise, the two strategies must be combined. We must specify *when* to switch from one strategy to the other and *under which conditions* to do so. 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 could use the results as our heuristics to generate appropriate texts given the user's domain knowledge.)

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 or functional information. Thus, each time an object needs to be described, a strategy must be chosen and a strategy switch might occur.

This gives us some clear decision points in the strategies:

- Within the constituency schema:
  - After the *identification* predicate: once the parent of an object in the generalization hierarchy has been introduced, a process trace can be provided for the superordinate.
  - After the *constituency* predicate: after mentioning the parts of an object, the constituency schema dictates filling 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 6, 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.

Figures 8 and 9 summarize the two strategies in their simplest form with the decision points.

Like the initial decision for a strategy, the decision to *switch* strategy is mainly based on the user model. If the initial strategy is the constituency schema, TAILOR

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 about this part (or its superordinate),
do **Depth-identification**
Otherwise do a Process Trace (for the part)
Attributive

Figure 8. The constituency schema strategy and its 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 about this part (or its superordinate),
do Constituency Schema
Otherwise do a Process Trace

Figure 9. The process trace strategy and its decision points

considers switching to the process trace for the superordinate and for the subparts of the device being described. The test for switching is largely the same as the initial test. TAILOR first checks whether the object under consideration (either the superordinate or each of the subparts) has some process information in its frame, or in its superordinate's frame. If so, TAILOR examines the user model to see whether the object or its superordinate is included in the user's local expertise. Switching is considered only if it is not included.

Two more conditions are tested before switching occurs. If the process explanation of the object includes *basic concepts* of which the user lacks knowledge, the process explanation would involve steps the user is not likely to comprehend. As a result the process explanation will not be understandable. For example, if the process explanation of a part involves the concepts of electricity and magnetism, it will not be informative if the user does not understand them. In that case, switching does not occur.

TAILOR keeps a record that switching was not allowed because of lack of knowledge about basic concepts, and, after the initial description is given, the user is given the option to ask for the process description for the part, even though it contains concepts he or she might not understand.

The last condition that needs to be satisfied before the switch can occur is a test on the length of the text constructed so far and, in case of subparts, the number of parts that need to be described using the process trace. This is done to avoid generating very long texts. A variable, which can be arbitrarily set to any value, sets a threshold beyond which the process trace will not be included at this point.[8] TAILOR remembers that the process explanations for some parts were not included because of length, and, after the text corresponding to the initial strategy is produced, TAILOR asks the user whether it should provide these explanations. When all the conditions are satisfied, TAILOR decides to switch to the process trace; otherwise, it continues with the constituency schema.

If the process trace is chosen for the overall structure of the text, a test similar to the one described above is performed when an object has been introduced as part of the process explanation. In this case, however, switching is considered only if the user model indicates local expertise about the part just introduced. As above, a test on the length of the text generated so far and the number of parts to be described using the constituency schema is also done to determine whether the switch should occur when the part is mentioned as part of the process or, at the user's request, after the initial text is produced. Again, when all the conditions are satisfied, TAILOR switches to the constituency schema for the part.

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.

Figures 10 and 11 show examples of texts generated by TAILOR by combining the two strategies. In Figure 10, based on the user model that indicates local expertise about *loudspeaker*, TAILOR chooses the constituency schema. The *telephone* is first identified with its purpose, and its subparts are introduced. Then, the constituency schema directs TAILOR to provide attributive information about each of the subparts. At this point, it is also possible to switch to the process trace for some subparts. TAILOR examines each subpart to decide whether there is a need to switch strategy for any of them. Two of the subparts have a mechanism associated with them (or with their parent in the generalization hierarchy): the transmitter and the receiver. It is thus possible to describe either of these parts with the process trace. Because the user model indicates local expertise about the *loudspeaker*, the receiver's superordinate, TAILOR decides to use attributive information and not a process explanation to describe this part. As no local expertise about the transmitter or the microphone is indicated in the user model, TAILOR chooses to provide process information for the transmitter, switching momentarily to the process trace. After the process explanation for the transmitter is completed,

[8] Note that this test can also be used to test schema recursion while going through the constituency schema, when the user model indicates local expertise about a part and switching is not allowed.

Telephone

User Model: Local Expertise — *Loudspeaker*
Basic Concepts — *electricity*

The telephone has two main functional parts: the transmitter (an instance of a microphone) and a 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 process of the transmitter since the user has no local expertise about it.

TAILOR output:

The telephone is a device that transmits soundwaves.	Identification
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.	Constituency
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.	Switch to process trace for the transmitter
The receiver is a loudspeaker with a small aluminum 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.	Attributive information for the other subparts

Figure 10. Combining the strategies: using the constituency schema as the overall structure of the text and switching to the process trace for one part

TAILOR returns to the constituency schema, and attributive information is provided about each of the other subparts. The process trace is shown in italics in the figure.

Figure 11 provides the description of a radio station. In this example, the process trace is chosen for the initial description of the object, and switching to the constituency schema occurs for one of the parts. The user model indicates local expertise about radio transmitters. The process trace is chosen for the overall structure of the text, because the user does not know most of the parts of the radio station. The description starts by introducing the object with its

Radio Station

User Model: Local expertise — *radio transmitter*
Basic Concepts? — *electricity*

A radio station has a few functionally important parts, only one of which is known to the user. The process trace is thus chosen as the overall structure of the text. TAILOR switches to the constituency schema for the radio transmitter as the user model indicates local expertise about it.

TAILOR output:

A radio-station is a device that broadcasts a signal when a human speaks into a microphone. Because the human speaks into the microphone, a current is produced. Then, the current flows through the wire into the radio-transmitter.	Process Trace
The radio-transmitter has an oscillator that produces a varying current when a battery produces another current, an amplifier that produces a strong varying audio-frequency signal from a varying audio-frequency signal, a mixer, and an amplifier to produce a strong radio-frequency signal from a radio-frequency signal. An amplifier has a vacuum-tube to produce a strong current from a power source and a metal grid. The oscillator has a high-resistance resistor, a battery, a capacitor, a low-resistance resistor and a transistor.	Switch to constituency schema for the radio transmitter
The current flowing through the wire into the radio-transmitter causes the strong varying radio-frequency signal to be produced. Then, the strong varying radio-frequency signal flows into the antenna of the radio-station. The antenna is long and made of metal. Because the strong radio-frequency signal flows into the antenna, the strong radio-frequency signal is broadcasted.	Continue the process trace

Figure 11. Starting with the process trace and switching to the constituency schema for one part

purpose and following the main sequence of events that occur when the object performs its function. When the radio transmitter is introduced as part of the process explanation, TAILOR switches to the constituency schema to describe it, as the user model shows local expertise about radio transmitters. After the radio transmitter is described with its subparts, the process explanation is resumed.

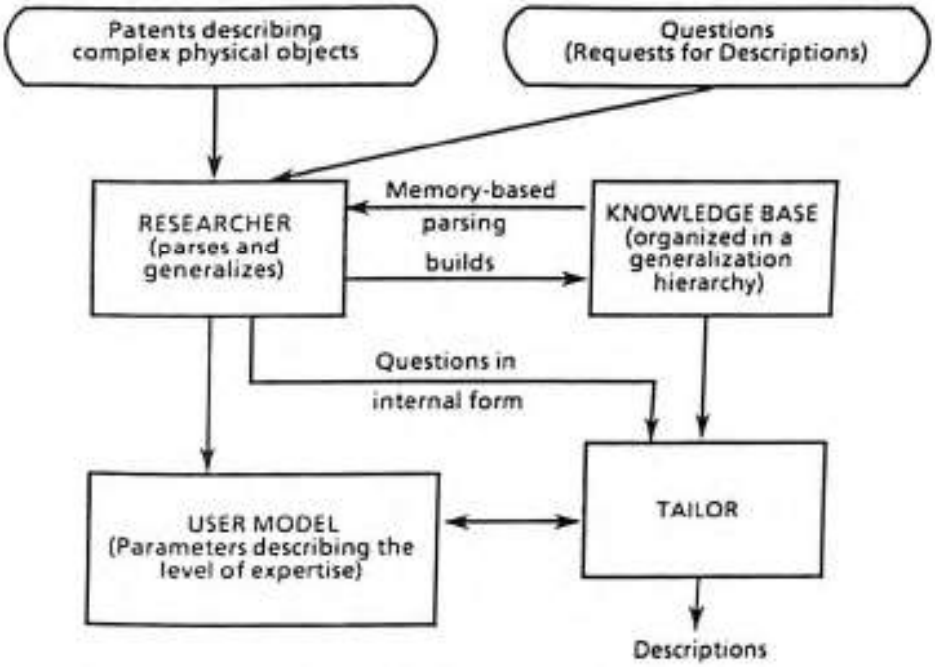


Figure 12. RESEARCHER and the TAILOR system

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 12, and looks at the information contained in the user model to decide on the strategy to employ. At this point, TAILOR is not really an interactive system and there is little feedback from discourse. In TAILOR, the user issues a request for a description and the description is generated. The object that was just described is added into the user model, as TAILOR assumes the user now has local expertise about it. TAILOR does not keep a record of all the questions asked by the user and all the answers provided. 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 the level of expertise. The user model includes the parameters that TAILOR uses to constrain its decision process during generation. The user model is not determined by the program but 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 [WASS83, WASS85] in which the basic frames represent objects. These structures 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 correspond 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 Y") and *analogical* relations (such as "X corresponds to Y").

A top-level diagram of TAILOR is shown in Figure 13. Input to TAILOR includes: (1) a request for the definition of an object, and (2) the level of expertise of the user, that is, the set of parameters that describe the user's level of knowledge about the domain, including:

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

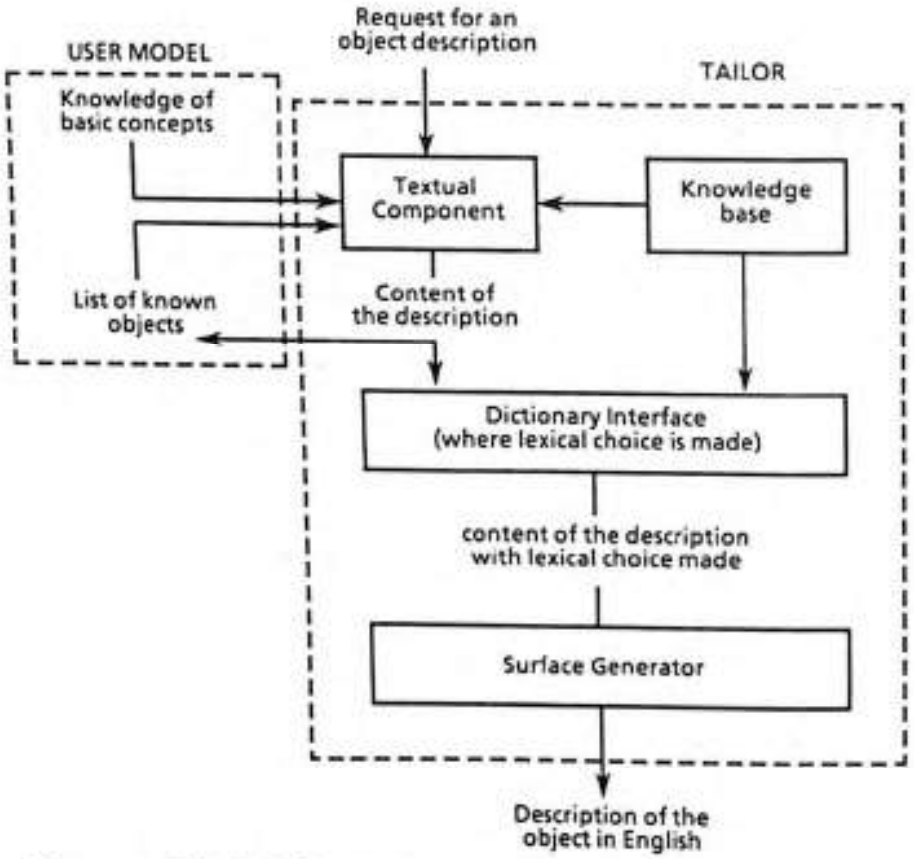


Figure 13. The TAILOR system

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 syntactic and 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 [MCKE85b] in the TEXT system. This generator unifies the input with a functional grammar [KAYM79] to produce English sentences. We have extended and improved the performance of this program, and augmented the functional grammar it uses [PARI85b, MCKE87].

5.2. Implementation of the Strategies

The constituency schema and the process trace strategies are implemented using an augmented transition network (ATN) [WOOD73]. 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 14 and 16 present the nets used for the strategies. (In these figures, the schemas do not include the links allowing the program to switch from one strategy to the other.)

In the constituency schema, shown in Figure 14, the arcs correspond to the predicates from the schema. These predicates define the type of information to be taken from the database.[9] 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).
- **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 15. 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 *diaphragm*. The English output shown in the figure (and in the other figures) is the actual output from TAILOR.

[9] More details can be found in [PARI87a].

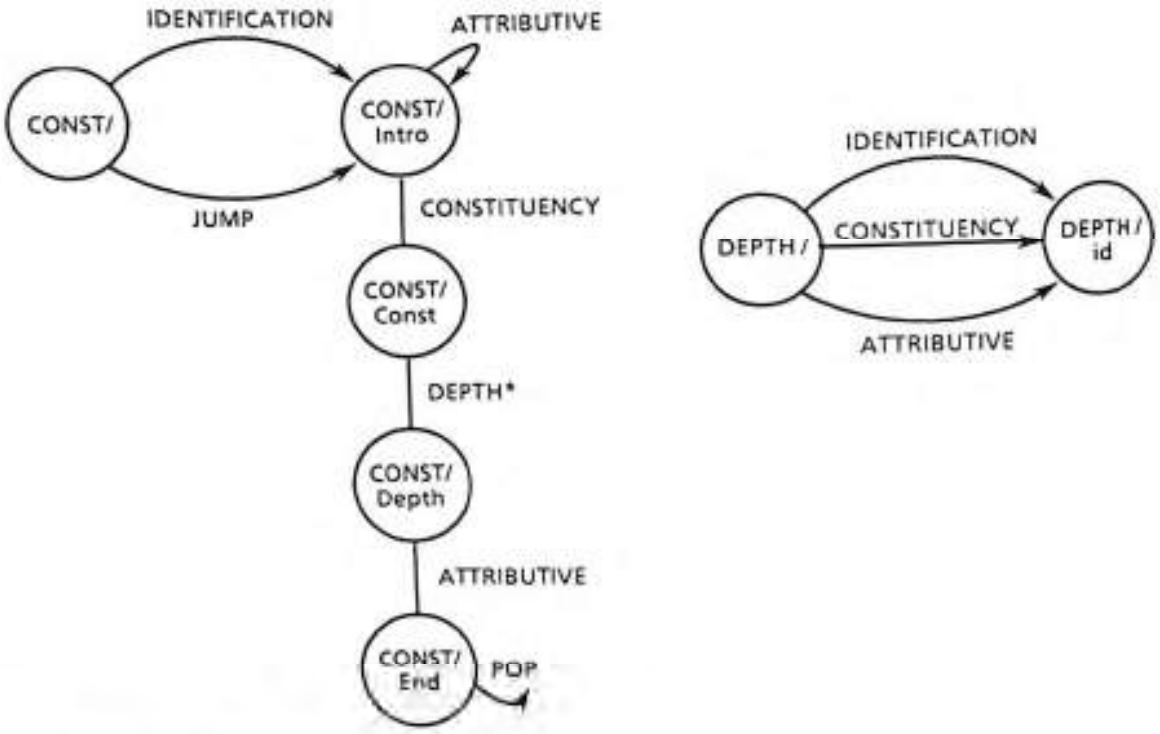


Figure 14. The constituency schema

The network for the process trace is shown in Figure 16.[10] An example of following the process trace strategy is presented in Figure 17. 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 [PARI87a, PARI87b]). 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. (TAILOR finds the main path and sidelinks automatically.)
- **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.[11]
- **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.

[10] The network shown here is actually slightly simplified for clarity. The full network can be found in [PARI87a].

[11] 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.

```
; Stepping through the Constituency Schema to describe
; a MICROPHONE

Applying the predicates to MICROPHONE:

  Identification predicate: DEVICE;
                              (used-for: change soundwaves into current)
  Constituency predicate:  DIAPHRAGM
                              (shape disc, material aluminum),
                           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 changes soundwaves into current. It has a disc-shaped aluminum 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 15. Stepping through the constituency schema

- **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 17, the link "the diaphragm vibrates" can be divided into substeps, namely "the diaphragm goes forward" and "the diaphragm goes backward". 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 17 for brevity. See [PARI85a] 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 network. The decision points are thus marked as special tests on the arcs joining the nodes.

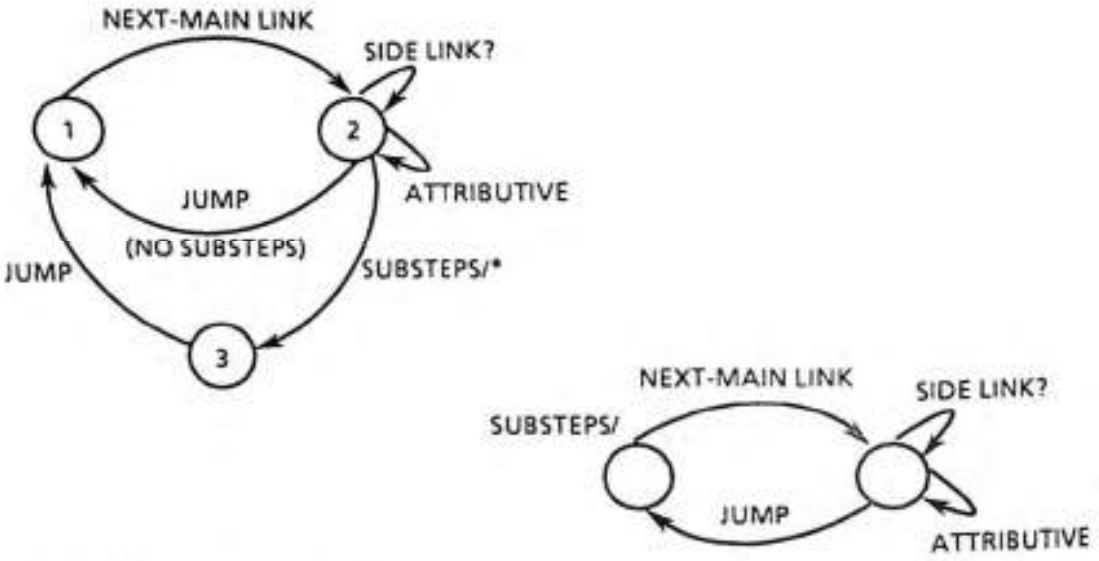


Figure 16. The process trace

As a general test for deciding which strategy to use to describe a part, we look 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 process of stepping through the ATN and switching strategy is shown in Figure 18. (The corresponding generated text was shown in Figure 10.) Note that to avoid generating very long texts, substeps are omitted when the process trace is chosen to describe a subpart.[12]

6. Further Work and Related Issues

6.1. Recursive Use of the Strategies

TAILOR can also use the two strategies recursively, although this was not one of our main concerns. TAILOR can currently call the constituency schema 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 have implemented a mechanism that allows the user to request such a description, either initially or after

[12] After generating the description, TAILOR asks the user whether it should now provide a process explanation at the substeps level.

; 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: aluminum]
[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 soudwaves to hit the diaphragm of the microphone. The diaphragm is aluminum and disc-shaped. The soundwaves hitting the diaphragm cause the diaphragm to vibrate. The soundwave_intensity increasing causes the diaphragm to spring forward. Because the diaphragm springs forward the granules compress. This causes the resistance of the granules to decrease. Because the resistance decreases the current increases. The soundwave_intensity decreasing causes the diaphragm to spring backward. This backward causes the granules to decompress. Because the granules decompress the resistance increases. The resistance increasing causes the current to decrease. The vibration of the diaphragm causes the current to vary. The current varies like the intensity varies.

Figure 17. Tracing the process information for a naive user

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 aluminum 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 18. Switching strategy

a description has already been provided by the system and the user wants additional information. We could extend this work to have TAILOR decide whether a deeper recursion is necessary for a particular user. 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: 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, explaining how a disk drive works in terms of current and magnetism will not be informative if the user does not understand these concepts. 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.

6.2. 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 [RICH79b, CHIN86].
- **The question type.** The kind of question asked ("What is a tape recoder?" 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") [PARI84].
- **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.
- **Inference rules.** Inferences rules relating objects in the knowledge base can be used to extend an existing user model as in [SLEE85]. For example, an inference rule might indicate that expertise about microphones also implies expertise about loudspeakers, as their mechanisms are similar. Knowing that the user is familiar with some items of the knowledge base might thus allow a system to infer expertise about other items [KASS87a, CHIN*].
- **Asking the user questions.** It is possible to ask the user a few questions about himself to get a starting point for the user model. This is the approach taken in both GRUNDY [RICH79b] and UMFE [SLEE85] for example.

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 difficult problem.

Finally, our user model at this point is coarse grained, in that it contains a list of objects that the user knows and whether he or she understands the basic underlying concepts. 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.

6.3. 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 or 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.

6.4. 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 only true 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 a control structure that will guide the decision process both in determining facts to take from the knowledge base and in organizing them, lest the resulting text be incoherent. A discourse strategy is one way to guide its decision process that has been used successfully in previous generation systems. The two discourse strategies used by TAILOR are of comparable complexities to those used in other systems and, in fact, one is identical. Other systems employing discourse strategies must decide which strategy to use to produce a text. The decision is usually based on the question type and the structure and content of the knowledge base. In TAILOR, the user model plays a role, but this does not add any cost to the decision process. Furthermore, to combine the strategies does not add cost either, especially because of the formalism used to represent the strategies. By representing both strategies with an ATN, the control structure necessary to combine the strategies is readily available. It is not more costly to jump from one node in one network to another node in the same network than to go from one network to another.

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 an environment could be a large knowledge base of facts (such as an encyclopedia), a help system, an expert system which needs to communicate with specialists, students and knowledge engineers, or simply with different types of users, such as the expert system for educational diagnosis proposed by [COHE*], and tutoring systems. Providing different information in an answer might also be done in explaining the behavior of an expert system which is used both as a teaching tool and as a problem solving engine.

Independently of whether the texts provided here are the most appropriate for a given user, we have added a new dimension along which a system can tailor its answers to users having different levels of domain knowledge, and TAILOR's ability to combine strategies is important, because it allows TAILOR to generate a greater variety of texts than would otherwise be possible. For example, for the pulse telephone which has a superordinate and three functionally important parts, TAILOR can generate eight different descriptions, even though the constituency schema is always chosen as the initial strategy (as the pulse telephone has no mechanism in its frame).[13] The ability to combine strategies allows a generation system to have distinct strategies clearly defined without losing flexibility. Discourse strategies help a system to construct a text with a given structure, and it is desirable to have such strategies to constrain the generation process. Given the strategies and no mechanism for combining them, a system is limited in the kinds of texts it can generate by the number of strategies. If combining strategies can be achieved, the variety of texts that can be generated is greatly increased.

By explicitly representing the user's knowledge about the domain, we have gained the ability to tailor texts to a whole range of user, without having a predefined set of user types.

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 devices, we believe this result extends to other domains, that is, that varying the kind of information to include in an answer is important in other domains as well, in order to tailor the

[13] Note that this number takes into account the fact that the heuristics chosen in TAILOR limit the number of combinations allowed. Furthermore, knowledge about basic concepts plays no role for this particular object. If it did, there would be yet more combinations.

answers to users with different levels of domain knowledge.[14] We thus proposed 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 described 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 the 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 falls 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 a wide variety of descriptions for a whole range of users.

Acknowledgements. 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 and Jean Marc Lagoutte for their work on the functional grammar. We also thank Robin Cohen for her useful comments.

8. References

[ANDR77], [APPE82a], [APPE85], [BAKE57], [BRIT63], [BRIT84], [CARB83], [CARB88], [CARB*], [CHEV78], [CHI81], [CHIN86], [COHE*], [COLL62], [DAVS84], [ENCY82], [GRIC75], [GRIM75], [HOBB78a], [HOBB78b], [HOBB80], [HOEP84b], [KAPS82], [KAY79], [LANC87], [LEBO83], [LEBO85], [LEBO86], [LEHN77], [MANN84], [MARH87], [MCCO83], [MCCO86], [MCCO88], [MCCO*], [MCKE85b], [MCKE85c], [MCKE87], [NEW67], [PARI84], [PARI85a], [PARI85b], [PARI87a], [PARI87b], [QUIL88], [QUIL*], [RICH79b], [RICH*], [SHEP26], [SLEE82b], [SLEE85], [TENN78], [VERW62], [WALL82], [WASS83], [WASS85], [WEIS73], [WILS86], [WOOD73]

[14] Note that, for other domains, one would need to investigate the kinds of information that exist and which are appropriate for the different user type.

9 Highlighting a User Model to Respond to Misconceptions [1]

Kathleen F. McCoy

Abstract[2]

Responses to misconceptions given by human conversational partners often contain information refuting the possible reasoning which may have led to the misconceptions. This work describes generating such responses by reasoning on a highlighted model of the user to identify possible sources of the error. Domain-independent structural characterizations of a highlighted user model which suggest the source of several types of misconceptions are identified. Each is associated with a response strategy. The system can respond to a particular misconception by consulting the highlighted user model for one of the identified structural configurations and responding with the associated strategy. Transcripts of misconceptions corrected by human conversational partners show, however, that misconception responses must reflect the user's perspective on the objects involved. This context-dependent aspec of correcting misconceptions is captured by a new notion of object perspectiv which serves to highlight certain aspects of the user model due to the precedin discourse.

1. Introduction

When people interact with a database or expert system, it is reasonable to expect that they might reveal a misconception about an object modeled by the system. Since a human conversational partner would correct such a misconception if it was important to the current goals of the conversation, our database and expert systems should also be equipped with this ability.

A study of transcripts of humans responding to misconceptions revealed that a response to a misconception important to the current discourse goals of the participants can be viewed as consisting of three parts: (1) a denial of the incorrect information, (2) a statement of the correct information, and (3) justification for the

[1] This paper is a revised and extended version of [MCCO88]

[2] Some of this work was done while the author was at the University of Pennsylvania and was partially supported by the ARO grant DAA20-84-K-0061 and the NFS grant MCS81-07290.

denial and correction given. Further, the justification often seemed to refute support that might have led to the misconception.

In order for a computer system to mimic this behavior, a way of generating appropriate justifications must be found. The justification hinges on the support accorded the misconception. If we could (domain independently) characterize both the various kinds of support someone might have for a particular kind of misconception and an appropriate justification refuting that support, then the behavior could be mimicked. For instance, suppose the user gives an object a property it does not have (e.g., "What is the interest on this stock"). One way a person might come to such a misconception is by either confusing the object being discussed (e.g., stock) with a similar object that does have the property (e.g., a bond) or by making a bad analogy from such an object. A highlighted user model configuration that indicates such support would be one containing an important object which has the property in question and is similar to the object involved in the misconception. A reasonable response to use when such a user model configuration is present is one that points out the error so it does not continue (e.g., "did you mean bond?"). If we can characterize all support/justification pairs in this kind of domain independent way, a computer system could appropriately respond to entire classes of misconceptions.

The task of characterizing the possible supports for a misconception in a domain independent fashion, however, seems insurmountable since the support someone might have for a misconception is so varied. Further transcript analysis shows, however, that for a given type of misconception (e.g., a misclassification), only a few kinds of support were corrected for in the justification part of the response. In fact, one could abstract out a small number of correction strategies used by the human conversational partners, based on the kind of justification given for the denial and correction. Therefore, since the number of support/justification pairs is small, it is feasible to characterize correctable support for a misconception and associate each with a correction strategy whose justification refutes the support. A generation system that has access to a highlighted model of the user's beliefs about the domain could respond to a misconception by looking in this model for one of the characterized supports and respond with the associated strategy.

Note that in order for this method of correcting misconceptions to mimic what is found in human conversation, the user model must be highlighted by previous discourse. The kind of response given by a human conversation partner is not only dependent on the beliefs about the person being corrected, but also on the context in which the misconception occurred. For instance, we could imagine the following dialog where the user exhibits the misconception that T-bills have a penalty. A reasonable response is shown.

U. I am interested in investing in some securities to use as savings instruments. I want something short-term and I don't have a lot of money to invest so the instrument must have small denominations. I am a bit concerned about the penalties for early withdrawal. What is the penalty on a T-bill?

R. Treasury Bills do not have a penalty. Were you thinking of a Money Market Certificate?

This response might be prompted by R thinking that U came to the misconception by confusing T-Bills with a Money Market Certificate.

On the other hand, it is reasonable that the response might be different given a different preceding dialog. For example:

U. I am interested in investing in some securities. Safety is very important to me, so I would probably like to get something from the government. I am a bit concerned about the penalties for early withdrawal. What is the penalty on a T-bill?

R. Treasury Bills do not have a penalty. Were you thinking of a Treasury Bond?

This response may have been prompted by R thinking that a confusion between T-Bills and T-Bonds was the source of the misconception.

These responses could be generated by the methods described if the user model was affected by previous discourse and if this highlighting was taken into account when accessing the user model. Currently, when a user model is accessed, all user knowledge has equal importance. When people engage in a conversation, however, certain aspects of their domain model become more important than others. This importance is more than just a highlighting of those things that have been explicitly mentioned. Rather, certain things that are somehow related to those things explicitly mentioned in previous discourse are also highlighted. In fact, a way of looking at the domain is usually established.

A new notion of object perspective will be introduced that will enable this highlighting effect of previous discourse to be incorporated into the user model. Thus when the user model is accessed, certain things in it will be highlighted while other things will be suppressed. I will show how this variable highlighting of the user model can explain why the response to a particular misconception by a particular user may vary.

2. Knowledge Available

The work being done here is in the context of a natural language interface to a database or expert system. It is an attempt to define a module of an interface which could generate a cooperative natural sounding response to a misconception – the kind of response that would be generated by a helpful human conversational partner. In this work, a *misconception* is defined to be some discrepancy between system beliefs and user beliefs (as exhibited through the conversation). Upon encountering such a misconception, the job of the misconception corrector module is to attempt to bring the user's knowledge into line with the system's knowledge.

The scope of this work is limited by several assumptions about the kind of knowledge available. First, since we are concentrating on *object-related* misconceptions, we assume the system's model of the world contains an object taxonomy with attribute/value pairs attached to the objects.

Second, we assume that we have a user model available. Thus we will not be concerned with inferring a model of the user. Inferring the user model is the subject of several other papers in this volume (see [RICH*, CHIN*, FINI*] for example). We further assume that the model of the user is in the same form as the system

model. That is, it contains an object taxonomy with attribute/values attached. The content of the user model, however, may be quite different from the content of the system's model of the world. Thus, the way objects fit into the object taxonomy and which attributes are associated with which objects may differ between the two models. I assume that the user model starts out as a model of a canonical user and is updated as the conversation progresses.

Third, we assume the system has available to it certain pieces of contextual and discourse information which serve to highlight the user model. This highlighting (explored below) is gained from a new notion of object perspective and from a record of items and attributes which have been explicitly focused on in the discourse.

Given the kind of information assumed in the system's and user's models of the world, there are two kinds of misconceptions that may occur: misclassifications (the user may place an object in the object taxonomy where it does not belong) and misattributions (the user may give an object an attribute/value it does not have). We are interested in generating a response to these kinds of misconceptions in situations in which the goals of the conversation warrant that a "full" misconception response be given. For each of these kinds of misconceptions, a small number of response strategies[3] were found in the transcript study that could be supported by the assumed knowledge bases. These strategies were abstracted into response schemas. In Section 4, examples of the response strategies found for misclassifications will be examined. For each strategy an abstract specification of the content will be given. Next we will look at what beliefs about the user might have prompted the use of each strategy, and characterize these beliefs in terms of the structure of the highlighted user model. A misconception can be responded to by looking in the highlighted user model for one of the pre-identified structural characterizations and instantiating the associated schema. Section 5 examines response strategies for misattributions. The sections following that will concentrate on the highlighting of the user model due to object perspective. Object perspective will be defined and it will be shown how the highlighting gained from object perspective affects responding to misconceptions. In particular, perspective's influence on the assessment of object similarity (crucial in the identified structural configurations) will be discussed.

The methods described here have been implemented in the ROMPER system (Responding to Object-related Misconceptions using PERspective). The system takes as input a specification of the information that is inconsistent with the system's model of the world, the current perspective (described below), and a record of past focus. It produces a formal specification of the response. This response specification is passed into the Mumble system [MCDO80] which, using a grammar and dictionary written by Robin Karlin [KARL85], produces an actual English response.

The implemented system works in the financial securities domain. In order to show the generality of this approach, two different domains will be used in this paper. The motivation for the system's method of choosing a response strategy will use examples from the domain containing whales and fish. In the last sections of the paper (those pertaining to the new notion of object perspective), the examples

[3] The dialogs contained in this paper were not taken directly from the transcripts. They are used to illustrate the kind of responses found.

will be taken from the domain of financial securities since the ideas can be better motivated with a more complex domain.

3. Related Work on Correcting Misconceptions

The method for responding to misconceptions described in this paper should be contrasted with the way that misconceptions are handled by AI systems today. For the most part misconceptions have been left to the intelligent computer aided instruction (ICAI) systems which basically use an *a priori* listing of misconception-response pairs (see [BROW78, SLEE82b, STEV79, STEV80, WOOL83, WOOL84a], etc.). Responding to a misconception is reduced to matching the user's input against the *a priori* listing and outputting the associated response should a match be successful. The major problem with these systems is that it is impossible to list *every* misconception and, due to the system's inability to reason about the misconception itself, the system is completely at a loss when faced with a misconception absent from the *a priori* listing.

The work of Sleeman [SLEE82b] on inferring defective algebra rules (mal-rules) is an attempt to avoid the hand-encoded *a priori* listing of misconceptions. Sleeman proposes on-line inference of mal-rules based on the answer the student has given to a particular problem. The problem is worked backward from the answer using correct rules as long as possible. A mal-rule is then inferred which "closes the gap" between the problem gotten by working backward from the student's response and the actual problem given. Although Sleeman's work is a major improvement over the *a priori* listing approach, it still has several problems. First, there is no measure of how *reasonable* or *likely* a particular mal-rule is. In addition, once a mal-rule has been inferred, no indication is given concerning how the misconception should be corrected.

[KAPS79and MAYS80a] differ from the previous work mentioned in that it, like the work described here, is concerned with correcting an entire *class* of misconceptions. Thus a system based on Kaplan and Mays' methods can handle a potentially infinite number of individual misconceptions. They were concerned with correcting *presumption failures* occurring in the context of a natural language interface to a database system. They worked on detecting and correcting such misconceptions based on domain independent linguistic cues from the user and an enhanced model of the domain. For instance, the query "Which faculty take courses?" indicates a presumption failure. A truthful response of "none" to this query would confirm the user's erroneous belief that faculty can take courses. Mays suggests correcting a query like the one above by (1) denying that the "take" relation can hold between faculty and courses, and (2) describing all correct alternatives which can be reached by abstracting on each of the objects and on the relation involved. This method would produce the following kind of response:

R. I don't believe that faculty can take courses. Faculty teach courses. Students take courses.

Although responses such as this would probably be helpful to the user, they have the potential (given a complicated domain) for being overly verbose and containing information that the user does not care about. This paper can be seen as an extension of Kaplan and Mays' work in that it offers a more sophisticated way of correcting a misconception once it has been detected.[4]

[QUIL*] discusses an approach to correcting misconceptions that is very similar to the one advocated in this paper. In Quilici's case, however, the class of misconceptions is different. While this work is concerned with correcting misconceptions about objects, Quilici concentrates on correcting misconceptions about actions. His user model contains information about plan knowledge and he has characterized support for several types of misconceptions in terms of this knowledge. As is the case with this work, each kind of support has an associated response strategy. When a misconception is detected, the system analyzes its user model and system knowledge to determine the probable support and responds accordingly.

4. Misclassifications

Three major strategies for correcting misclassifications that could be supported by the assumed knowledge bases were found in the transcripts. The first is exemplified by the following dialog:

U. I thought whales were fish.
R. No, they are mammals. You may have thought they were fish because they are fin-bearing and live in the water. However, they are mammals since, (while fish have gills) whales breathe through lungs and feed their young with milk.

Let us first examine the content of this response. It can be seen as consisting of three parts. In the first sentence we have the denial of the incorrect information and the statement of the corresponding correct information. The remaining sentences comprise the justification for the denial and correction given. Notice that it refutes a kind of support that may have led to the misconception. It is easiest to see this by examining an abstraction of the response and looking at the content of the justification:

MISCONCEPTION = X *is-a* Y
RESPONSE =
1. X *is-NOT-a* Y
2. X *is-a Type-of* (X)
3. X is like Y because both share *attributes-of* (X) ∩ *attributes-of* (Y)
4. BUT X has *attributes-of* (X) − *attributes-of* (Y)
5. WHILE Y has *attributes-of* (Y) − *attributes-of* (X)

The justification is contained in 3-5 of the rule. Notice that it can be seen as a concede/override pair. First, information is conceded that actually supports the

[4] Correction was not the major thrust of Kaplan or Mays' work.

misconception. This consists of attributes that whales and fish have in common. These attributes are a potential source of the misconception since an object that shares many attributes with a superordinate is likely to belong to that superordinate class. By conceding the intersecting attributes, R is acknowledging the apparent support for the misconception lest U think that that part of his/her domain model is wrong.

The response goes on to override the conceded information by offering attributes that differentiate whales and fish. These are attributes that whales have but fish cannot, and attributes that fish have that whales do not.

By analyzing the justification in this way, not only do we get an abstract representation that can be used by a generation system, but we get some insight into when it would be reasonable to use such a response. It makes sense to use a response that first concedes a similarity between the misclassified object and the object that was wrongly given as its superordinate (henceforth called the *posited superordinate*) and then overrides that conceded information, when such a similarity is a possible source of the misconception. This might be the case when *the highlighted user model shows that the user believes that such a similarity exists.* Thus a generation system may generate its response by instantiating the above rule if the user has misclassified the object X as a Y and an analysis of the highlighted user model shows that the user believes X is similar to Y. Because of this reason for using this response strategy, it has been termed the "like super" response strategy.

A second kind of strategy is exemplified by the following dialog. Imagine a recent conversation centered on sharks being large fish that people are typically afraid of. Following that, the conversation turns to classifying whales. The user says:

U. I thought whales were fish too.

R. No, they are mammals. You may have thought they were fish since they are like the fish, sharks, in that both are large aquatic creatures and both scare people. However, whales are mammals since, (while fish have gills) whales breathe through lungs and feed their young with milk.

Again, this response can be seen as consisting of the three parts with the denial and correction being contained in the first sentence. As in the previous example, the remaining sentences comprise the justification in the form of a concede/override pair. The conceded information, while dependent on attributes, differs from that given above. The abstract content rule can perhaps best illustrate this difference:

MISCONCEPTION = X *is-a* Y

RESPONSE =

1. X *is-NOT-a* Y
2. X *is-a Type-of* (X)
3. ∃ Z (Z *is-a* Y ∧ X is like Z
   because both have *attributes-of* (X) ∩ *attributes-of* (Z))
4. HOWEVER X *is-a Type-of* (X)
   because X has *attributes-of* (*Type-of* (X)) − *attributes-of* (Y)

The major difference between this response and the like-super response is that instead of conceding a similarity between the misclassified object and the posited superordinate itself, a similarity between the misclassified object and a *descendant* of the posited superordinate is conceded. This response is reasonable if the user arrived at the misconception by reasoning: "I don't know how to classify whales, but I do know how to classify sharks. Sharks are fish and whales are very similar to sharks. Therefore it is reasonable to believe that they are classified the same." Evidence for this kind of reasoning might be a similarity found in the highlighted user model between the misclassified object and some descendant of the posited superordinate. Finding such a similarity would justify the use of this "like-some-super" strategy.

A final strategy found for misclassification is exemplified by the following:

U. I thought whales were fish.
R. No, they are mammals.

This is the degenerate "no support" strategy which contains no justification for the correction. At first glance the use of this strategy seems rather puzzling. Since I have assumed that the misconception is important, one would expect a full response to be given. However, no justification was included in the response.

This response can be explained in the following way. The model for correcting misconceptions says that there are a small number of ways in which a correction of a particular kind of misconception can be justified. Each kind of justification refutes a kind of faulty reasoning that may have led to the misconception. Thus, in order to respond to a misconception, the highlighted user model is consulted in order to find evidence for one of the pre-identified kinds of faulty reasoning.

This last response is an example of a response in a situation where no evidence was found for any of the pre-identified kinds of faulty reasoning. If R could not figure out what led to the misconception, then R could not include justification to refute the support. The most that R could do in this situation was deny the incorrect information and state the corresponding correct information. For this reason the response has been termed the "no-support" response and it can be used when the highlighted user model contains none of the evidence that would key off any of the other correction strategies.

5. Misattributions

Misattributions are the second kind of misconceptions that may arise given the kind of knowledge assumed. A misattribution occurs when the user gives an object an attribute/value that does not exist in the system's domain model. In the case of misattributions, the justification took the form of offering another statement that would change what the user said in a small way in order to make it a correct statement. This is illustrated in the following dialog which is an example of the first kind of correction strategy found. Imagine that we are discussing sea animals and have previously discussed sharks and their attributes. We move to whales:

U. I thought whales had gills.
R. No, they breathe through lungs. Were you thinking of sharks?

The response might go on to discuss the similarities and differences between whales and sharks.

Notice what was done in this response. We have the denial and the correction as before. The justification offers another object that is similar to the object involved in the misconception, but for which the misattribution is true. This content is captured in the following rule:

MISCONCEPTION = X has *attribute* Y with *value* V
RESPONSE =
1. X does not have *attribute* Y with *value* V
2. X has *attribute* Y with *value* W
3. ∃ Z (Z has *attribute* Y with *value* V) did you mean Z?

This response would be effective if the user's misconception results from either confusing the object being discussed with some other object that has the misattributed property, or through an incorrect analogy with such an object. Mention of the object in the case of confusion prevents continuation of the confusion. In the case of an incorrect analogy, mention of the other object alerts the user to the inappropriateness of the analogy and also prevents the continued use of the analogy.

It makes sense to use the "wrong object" strategy when a possible source of the misconception is either an object confusion or the use of a bad analogy. Evidence for this would be the existence of some object from the highlighted user model that has the misattributed property and is similar to the misconception object.

A second kind of correction strategy used for misattributions is shown here:

U. I thought whales had gills.
R. No, whales breathe through lungs. Did you mean you thought they had fins?

This is very similar to the last response except, instead of offering another object, the strategy offers a similar attribute that makes the user's statement correct. The content rule for this response follows:

MISCONCEPTION = X has *attribute* Y with *value* V
RESPONSE =
1. X does not have *attribute* Y with *value* V
2. X has *attribute* Y with *value* W
3. ∃ Q ∃ U(X has *attribute* Q with *value* U ∧ similar(Y,Q)
   did you mean X has *attribute* Q with *value* U?)

This strategy, termed the "wrong attribute" strategy, is used when there is a similarity of attributes within the highlighted user model.

The no-support strategy also occurs in the case of misattributions.

6. A Rule for Choosing a Strategy

Given the association of highlighted user model configurations with response strategies given in the previous sections, we can come up with a method for deciding

how to respond based on the type of misconception along with an analysis of the highlighted user model. Basically the highlighted user model analysis looks for one of the pre-identified configurations and, if one is found, suggests instantiating the associated strategy. The following rule captures the way this is implemented in the ROMPER system.

IF misconception = "X *is-a* Y"
  THEN
    IF *similar*(X,Y)
      THEN instantiate like-super schema
    ELSEIF ∃ Z (Z *is-a* Y) ∧ *similar*(X,Z)
      THEN instantiate like-some-super schema
    ELSE instantiate no-support schema
ELSEIF misconception = "X has *attribute* Y with *value* V"
  THEN
    IF ∃ Z ((Z has *attribute* Y with *value* V) ∧ *similar*(X,Z))
      THEN instantiate wrong-object schema
    ELSEIF ∃ Q ∃ U ((X has *attribute* Q with *value* U) ∧ *similar*(Y,Q))
      THEN instantiate wrong-attribute schema
    ELSE instantiate no-support schema

This rule is a simplification since it goes through the configurations in a pre-determined order and chooses the first strategy with a matching configuration. Thus it places preferences on some strategies over others. While this preference is well motivated in the choice of a strategy containing justification over a no-support strategy (see [MCCO87] for details) it is a simplification for choosing between two strategies containing justification. An arbitrary preference has been made for the like-super strategy over the like-some-super strategy and for the wrong-object strategy over the wrong-attribute strategy. In reality a more complicated model which is perhaps based on the discourse situation is needed.

For instance, suppose we have just been discussing sharks and you misclassify whales as fish. It probably makes more sense for me to check the configurations that might involve sharks (or any other object recently focused on) rather than those that involve things foreign to the conversation so far. In this situation if the highlighted user model indicates a similarity with whales and sharks, the like-some-super strategy should probably be used even though the highlighted user model might also indicate a similarity between whales and fish. Precisely how this should be done is an area of future research.

7. Highlighting and Object Similarity

In the previous section a rule was introduced that shows how the ROMPER system determines which schema to use for a particular misconception. This rule calls for

analyzing a highlighted user model. In this section we show how the highlighting of the user model is taken into account in the above rule. Notice that the user model analysis in each case described above hinged on an assessment of object similarity. It is in this assessment that the highlighting on the user model is taken into account in the ROMPER system. ROMPER's similarity metric allows several contextual components to be taken into account. One of these components is the highlighting of the user model gained from object perspective.

To date, most AI systems do not assess object similarity in a way that is context-dependent. Several systems that do assess object similarity (e.g., [RUME73, MCKE82, CARB84, WEIN84]) use a metric based on distance in the generalization hierarchy. Basically, two objects that have a common immediate superordinate (i.e., are siblings in the hierarchy) are seen as very similar, while objects whose lowest common ancestor is several levels up in the hierarchy are seen as quite different.

One problem with using a metric based on distance in the generalization hierarchy has to do with the way in which this distance should be defined. The metric yields ambiguous results when objects can be classified in more than one way. With such a rich classification, there may be several lowest common ancestors of the objects being compared. A decision must be made about which of these lowest common ancestors should be considered since the similarity assessment of the objects might vary widely as a result. For instance, a treasury bond and a corporate bond may be assessed as being very similar since they have a common immediate superordinate of bonds. On the other hand, both of these objects can be classified along other dimensions. If we look at a treasury bond as being a treasury issue which is a type of US Government security, it seems quite different from a corporate bond which is a corporate security.

A second major problem with the distance similarity metric is that it is *context invariant*; contextual information has no way of affecting the assessments. However, in order for our model of correcting misconceptions to work, it is crucial that the similarity metric be sensitive to context. In addition, human judgements of object similarity [TVER77] have been found to shift both when the set of objects under discussion are altered (e.g., a violin and an electric guitar may be judged quite similar when in a group with a clarinet and an oboe, and may be judged quite different when the other members of the group are a cello and an electric bass), and when the salience of attributes is altered (e.g., in a group containing a red triangle, a blue triangle, and a red square, the red triangle might be judged similar to the blue triangle when attribute shape is stressed, but may be judged similar to the red square when attribute color is stressed).

One metric that avoids these problems was introduced by Tversky ([TVER77]). Tversky's metric, rather than relying on distance in some space, is based on the common and disjoint features of the objects involved. The metric, termed a contrast model, allows context to be taken into account in several places.

Suppose we have two objects a and b where A is the set of properties associated with object a and B is the set of properties associated with object b. Tversky's measure can be expressed as:

$$s(a,b) = \theta f(A \cup B) - \alpha f(A - B) - \beta f(B - A)$$

for some θ, α, and $\beta \geq 0$.

In the above equation θ, α, and β are parameters which represent the importance of each component of the equation. The function f maps over the features and yields a salience rating for each. In essence, the contrast model states that the similarity of two objects is some function of their common features minus some function of their disjoint features. The importance of each particular feature involved (determined by the function f) and the importance of each component of the equation (determined by θ, α, and β) may change with context.

Although Tversky discusses in general terms how these functions might be set, he gives no concrete methods for doing so. For instance, to set θ, α, and β he turns to the relative prominence of objects a and b in the discourse. The more prominent an object is, the more its attributes should have an impact on the similarity rating. Thus, finding the relative prominence of objects a and b in the discourse would help set these values. If a is relatively more important, then functions θ and α should be greater than β, resulting in the attributes of the more prominent object having a greater influence over the similarity assessment. While I would conjecture that information about the focus of the discourse [GROS81, SIDN83a, GROS83] might give an indication of an object's prominence and would therefore be useful in setting the values of θ, α, and β, in this work I have assumed a value of 1 for the θ, α, and β and have concentrated on setting the f function.

We turn, then, to the problem of finding a value for the f function: the measure of salience for each property of the objects involved. Other work, such as [CARN70b, WEIN84], has hand-encoded salience values for attributes of individual objects directly into the knowledge base, permanently setting the f function. This approach is not sufficient for setting the f function for Tversky's metric since it is crucial that the f function be able to change with context. In order to make this happen, the salience values computed by f must change with context.

To see this, consider our ability to explicitly mention an attribute to increase its salience. In the example earlier with the red and blue triangles and the red square, if the request for a similarity judgement had been preceded by "Look at the pretty colors of these objects", the red triangle and red square would have probably been judged to be most similar. Thus we see that explicit mention of an attribute in a discourse is one way that the f function might be affected by previous discourse.

Explicit mention of an attribute is only one way in which the salience of the attributes may change dynamically. Another aspect of dynamic salience comes from the point of view, or perspective, applied to the domain. A building can be referred to as being an architectural work, for example, or as being someone's home. The two different views of the building cause different sets of attributes to become salient. Notice that this set of attributes is in addition to the attributes that have been explicitly mentioned in the discourse. From an architectural work point of view, attributes like the architect's name, date of building, and particular architectural features become salient. On the other hand, from the home point of view, attributes like the kitchen size, number of bedrooms, and living space become important. If we can find a way of modeling how these "precompiled" sets of attributes become highlighted in a discourse, we will have a principled method for setting the f function needed for Tversky's similarity metric. The next section discusses how this highlighting can be modeled by a computer system.

8. Object Perspective

The notion of point of view or object perspective has been noted by other researchers in artificial intelligence. Its ability to explain the changing attribute salience has been attributed to a limited inheritance mechanism (see e.g. [GROS77b, BOBR77, TOU82]). According to the limited inheritance theory, an object viewed from a particular perspective is seen as having one particular superordinate, although in fact it may have many. The object inherits properties only from the superordinate in perspective. Therefore different perspectives on the same object cause different properties to be inherited (and therefore highlighted).

While the limited inheritance theory of object perspective is intuitively appealing, in practice it is problematic (see [MCCO85a] for details), and is unable to handle some additional effects which should be handled intuitively by object perspective. Some of these effects will be discussed here.

The first has to do with the availability of object attributes. The proposed limited inheritance mechanism makes attributes inherited from superordinates other than the one in perspective *unavailable*. This seems a bit too strong. When we discuss a building as an architectural work, I may comment on the number of bedrooms the building has. While you may think my comment irrelevant to the current conversation, you would still be able to understand it and even evaluate its truth or falsity. In the limited inheritance account of object perspective, however, this would not be possible. You would not even be able to tell that the number of bedrooms is an attribute the building might have.

A second problem with the limited inheritance account of object perspective concerns its close ties with the generalization hierarchy. The account says that an object viewed through a particular perspective inherits the attributes only from the superordinate in perspective. What attributes does that superordinate have? The superordinate itself may have multiple classifications and thus potentially multiple perspectives. Thus, in order to figure out what attributes a particular concept should inherit, we must figure out not only what perspective it is being viewed from, but also what perspective the perspective superordinate is being viewed from, and so on. But this seems to be much more work than is necessary!

A third problem with the limited inheritance account of object perspective has to do with the scope of a perspective; a perspective may be taken on an individual object. However, during the course of a conversation it is usually the case that more than one object will be discussed. When this happens, usually the same kinds of things are discussed about the objects. In essence, a particular highlighting of attributes (or point of view) seems to be in force during the conversation. Yet this highlighting is applied to different objects – some of which may not even have the same superordinates. What seems to be happening is that the conversational partners are viewing an entire group of objects from the same perspective. This cannot be accounted for by the previous definition of object perspective unless each of the objects under discussion can be said to have the same (immediate) superordinate.

A final effect which is not accounted for by the limited inheritance mechanism, yet seems to hinge on the view being taken of the domain, has to do with the heightened importance of some objects during a discourse. Like the importance of

attributes, the relative importance of some objects in the discourse cannot solely be accounted for by explicit mention. Some objects are more likely to be mentioned and discussed in a discourse than others. For instance, when discussing a building as an architectural work, I might reasonably mention the library down the street that was designed by the same architect. On the other hand, I will probably not mention my apartment. Along the same lines, in discussing that building as a home, my apartment is a likely candidate for mention in the conversation. Thus, this effect seems to be tied to the notion of object perspective, yet is not addressed by the limited inheritance account.

In light of these problems, a new definition of object perspective is needed. We need the new definition to be able to account for the dynamic highlighting of "precompiled" groups of attributes. At the same time, we need it to avoid the problems of the limited inheritance account. We propose that the following account be used:

1. Instead of tying perspective into the generalization hierarchy of objects as has been done in the past, the new notion of perspective is independent of that hierarchy. "Perspectives" which can be taken on the objects in the domain will be defined and will be located in a structure which is orthogonal to the generalization hierarchy.
2. A number of perspectives are available for any domain of discourse and any given domain object may be viewed from any one of several perspectives for that domain.
3. Each perspective comprises a set of attributes with associated salience values. It is these salience values that dictate which attributes are highlighted and which are suppressed.
4. One such perspective is designated *active* at any particular point in the discourse.[5]

This notion of object perspective works as follows. An object or group of objects is still said to be viewed through a perspective. In particular, any object which is accessed by the system is viewed through the current *active* perspective. However, instead of dictating which attributes an object inherits, the active perspective affects the salience values of the attributes that an object possesses (either directly or inherited through the generalization hierarchy). The active perspective essentially acts as a filter on an object's attributes. By raising the salience of the attributes, it highlights those attributes which have a high salience rating in the active perspective. By lowering the salience of the attributes, it suppresses those attributes which are either given a low salience value or do not appear in the active perspective.

Let us look at how this new notion of object perspective handles the problems with the old account. The first two problems are eliminated rather easily by the definition itself. In this new account, the perspective does not affect what attributes

[5] Saying that exactly one perspective is active is actually a simplification. It may be the case that a number of perspectives can be active. In this case the resulting "active perspective" will be some function of the individual active perspectives. The exact nature of this function is an open research question.

an object inherits; it simply affects the salience values on those attributes. As a result, we do not have the problem, that of some attributes being unavailable. Instead, their salience rating makes them unimportant in the current context.

The second problem, that of potentially having to assess perspectives on many objects in order to evaluate the perspective on the object under discussion, is handled by decoupling the perspectives and the generalization hierarchy. In the new definition, "perspectives" are defined broadly enough so as to be applicable to any domain object (see below for an example). As a result, one avoids the problem of having to evaluate the perspective of a superordinate in order to determine how that superordinate's attributes should be highlighted.

The third problem with the previous notion of object perspective was that it was not able to handle the feeling of a perspective being evident for a whole group of objects. This new notion of object perspective handles this by forcing it to be the case. As long as the active perspective does not change, every object that is accessed by the system is viewed through the same perspective.

Finally, this notion of object perspective is able to account for the feeling of heightened object importance within a perspective. An object is important within a perspective if it has many of the attributes to which the perspective gives a high salience rating. It can be thought of as the whole becoming highlighted because the parts are highlighted. In other words:

$$important(X) = \sum_{X \text{ has } P} salience(P)$$

In the example given earlier, the library was highlighted (and thus worth mentioning) in the architectural works perspective because it contained many attributes highlighted by that perspective. On the other hand, my apartment was unimportant in that perspective because it didn't contain any attributes made salient in the architectural works perspective.

9. Using Perspective to Set *f*

We return to the problem of setting the *f* function in Tversky's metric. *f* maps over the attributes of the objects and returns a salience rating for each. So far we have identified three sources of salience values:

1. static salience encoded directly into the generalization hierarchy.
2. dynamic salience gained by an attribute being explicitly mentioned in a discourse.
3. perspective salience gained by viewing the domain objects through the active perspective.

I propose that the *f* function for Tversky's metric be set by taking into account all three of the above components of salience. This would yield an *f* function that was context-dependent and would help the similarity metric exhibit many desirable properties.

10. Modeling a Domain with Perspectives

In some natural language systems, a model of a particular domain includes the usual object taxonomy containing all of the objects in the domain and all of the attributes associated with those objects. In some systems (e.g., [CARN70b, WEIN84]), the attributes of the objects in the domain model are given static salience values to reflect the attributes' importance. The norm in building knowledge bases is to omit even the static salience values and to assume equal importance for all attributes an object possesses. We will show how a typical knowledge base can be augmented with perspective information and how this information is taken into account when accessing the knowledge base.

Figure 1 shows a small segment of a typical domain model. The domain is that of financial securities. Three of the objects from this domain, Money Market Certificates, Treasury Bills, and Treasury Bonds, are shown with the attributes they possess.

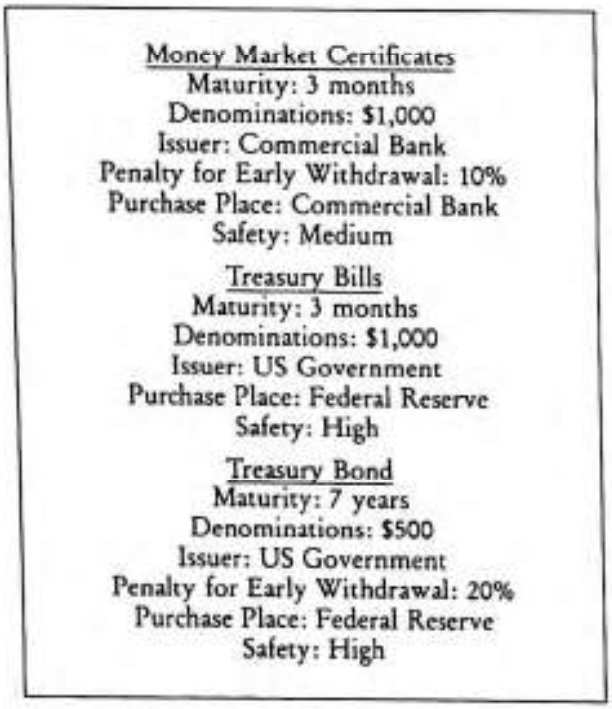


Figure 1. Objects in "flat" domain model

In systems as they are defined today, a group of objects defined in this way and arranged in a generalization hierarchy would constitute the domain model. If a system were to access any one of the objects in the domain it would be given the objects with attributes as they are listed in the figure.

In order to get a dynamic highlighting of the domain model, we must build, in addition to the object taxonomy, a separate structure containing the perspectives that can be taken on the domain objects. This means that we must think about the

different points of view that can be taken on the objects in the domain and compile sets of attributes from our model which capture the important domain concepts in that point of view. There will be attributes in each perspective that do not occur with all of the objects in the domain. At the same time, there will be attributes of individual objects that do not appear in a particular perspective. The perspective simply captures the attributes that are important in a particular point of view. The perspectives found in Figure 2 might be reasonable for the domain where we are assuming salience values from low salience of 0 to high salience of 1).

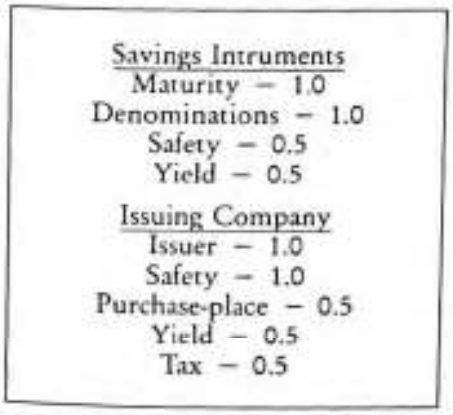


Figure 2. Sample perspectives

The perspective of Savings Instruments highlights maturity and denominations, and somewhat highlights safety and yield. This indicates that when people are discussing securities as savings instruments, they are most interested in how long their money will be tied up and in what denominations they can save their money. The perspective of Issuing Company, on the other hand, highlights different attributes. When securities are discussed from this perspective, things like the name of the company and the stability of an investment in the company become important. Other attributes of the securities are ignored (recall that attributes not mentioned in the perspective get assigned a low salience rating).

Notice how the objects look when accessed, depending on which of the two different perspectives are active. For instance, through the savings instruments perspective the objects' attributes take on the salience shown in Figure 3.

Attributes are not shown in the figure that have 0 salience. Recall that when no static salience is given, the attributes of the objects derive their salience directly from the active perspective. Attributes in the perspective that the objects do not have (e.g., yield) are ignored. Attributes of the objects not occurring in the perspective are given the lowest salience rating. The very same objects look different when viewed through the other perspective. This is shown in Figure 4.

The salience values derived from perspective can be used for many tasks. Section 12 will show how they can be used in conjunction with Tversky's similarity metric to generate context sensitive responses to misconceptions.

Another task that perspective is useful for is in generating more concise responses to users. For instance, the schemas for correcting misconceptions intro-

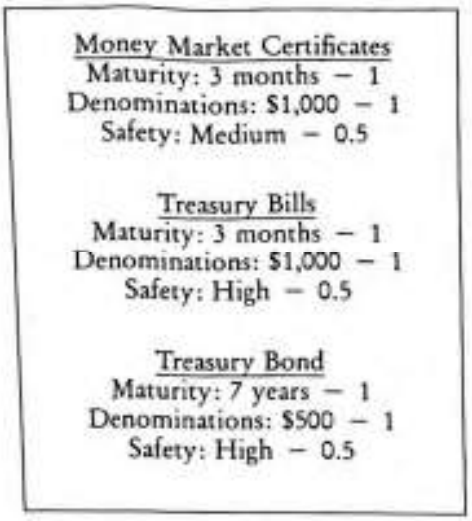


Figure 3. Objects throught savings instruments perspective



Figure 4. Objects throught issuing company perspective

duced earlier call for the inclusion of sets of attributes in many of the responses. For example, the like-super schema used for misclassifications calls for listing the attributes the misconception object and the posited superordinate have in common. In many cases this set of attributes may be quite large. In addition, the user is not likely to be interested in the entire set. Perspective is used in ROMPER to limit the listed attributes to those that are likely to be most interesting to the user by listing those attributes that are given high salience in the active perspective.

Cohen and Jones use a notion similar to object perspective to limit the information included in their responses [COHE*]. The difference between the two notions is that while the attributes given high salience in a perspective can be described as what things are currently of interest, their notion is more closely tied to what the system believes the user knows. They limit the information included in a response to that which the system believes the user does not know. The idea is that the unknown information is most helpful and interesting to the user. Cohen and Jones' work is in the context of an expert system where the system knowledge is partitioned into several classes. Different types of users are assumed to know about different classes of knowledge. When the system is asked to explain its reasoning, the "perspective" chosen is one that includes information from the classes of knowledge that the user is believed not to know. Thus their notion can be seen as a crude highlighting mechanism where the knowledge classes unknown to the user are highlighted. In the case of Cohen and Jones, however, this "highlighting" is achieved in response schemas instead of in the knowledge base access as is the case with the perspectives described here.

11. Choosing the Active Perspective

In the implemented ROMPER system, the perspective being taken on the domain is input to the system. In order for the notion of object perspective to be truly beneficial, however, there must be a mechanism for choosing the active perspective based on previous discourse. While this topic is still very much open to investigation, some preliminary research has revealed several factors that might influence the choice of active perspective.

Perhaps one of the most influential pieces of information useful in choosing a perspective is the user's current goal. The task of determining and tracking the user's goals throughout a discourse is discussed in this volume in [CARB*]. In [MCKE85c] the user's goal completely determines which perspective is active. In their work, each perspective which can be taken on the domain objects is indexed by potential goals. Thus, once the system has determined what the user's goal probably is, it has also determined what perspective the user has probably taken on the domain objects.

While the user's goal is a good source of information to use to determine the probable perspective, other factors may also influence this choice. These include the attributes and objects mentioned so far in the dialog. The mentioned attributes are obviously thought to be important and one would therefore expect them to be given a fairly high salience rating in the active perspective. Thus, the choice of active perspective can be narrowed down to those in which the mentioned attributes appear with high salience.

Using the mentioned attributes to determine the active perspective is similar to what is done in [MORI*] in determining *isotopy*. The notion of isotopy is similar to that of object perspective. It is a kind of coherence found in a discourse established by semantic ties or threads found from one exchange to the next. The semantic ties from isotopy are based on the attribute/values mentioned in the dialog. For instance, in Morik's system, a user booking a hotel room might first ask about the view the room has and then about whether or not the hotel has a night club. While these two attributes might seem unrelated, they do have the common feature (isotopy) of determining how quiet/loud the room is. Thus the system, using this information, might offer additional information which also addresses the quietness/loudness of the room. The notion of isotopy is similar to perspective in that the attributes which provide the coherence in isotopy would likely be found in the same perspective as defined in this paper. For instance, one could imagine a "loudness" perspective in the hotel domain which contained all of the attributes which contributed to the loudness of a hotel room. A helpful system might offer attributes given high salience in this perspective to a user who has shown interest in the loudness of the room by referring to several of the attributes given high salience in that perspective.

Another source of information which can be used in determining the active perspective is the objects mentioned so far in the dialog. One would expect that the active perspective would deem the mentioned objects important. Therefore the system might look for perspectives that give high salience ratings to many of the attributes associated with objects that have been mentioned in the discourse.

In this section I have identified several factors which influence the choice of active perspective. This choice, however, is a question which remains as an open research topic. Still unanswered are questions such as: when does the active perspective change? How long is a perspective active? Is there a relationship between a discourse unit [GROS85] and perspective? Is there any structure to the space of perspectives that would put constraints on moving from one active perspective to another? These questions must be taken up in future research on perspective.

12. Perspective's Influence on Responses

In this section an example is given which indicates how the choice of perspective influences the correction of a misconception. Recall that in correcting a misattribution one of the correction schemas used by ROMPER called for a similar object to be offered as a possible object of confusion. A study of transcripts reveals, however, that this schema may be instantiated in different ways depending on the context. Consider once again the following dialogs that were first seen in the introduction:

U. I am interested in investing in some securities to use as savings instruments. I want something short-term and I don't have a lot of money to invest so the instrument must have small denominations. I am a bit concerned about the penalties for early withdrawal. What is the penalty on a T-bill?

R. Treasury Bills do not have a penalty. Were you thinking of a Money Market Certificate?

In this case the money market certificate was seen as being similar to the treasury bill and therefore included in the response. A different object might be used in a different context. Consider:

U. I am interested in investing in some securities. Safety is very important to me, so I would probably like to get something from the government. I am a bit concerned about the penalties for early withdrawal. What is the penalty on a T-bill?

R. Treasury Bills do not have a penalty. Were you thinking of a Treasury Bond?

The difference in these two responses can be explained by different perspectives being taken on the objects. Suppose we are given the objects, attributes, and perspectives from Section 10. The dialog preceding the first example could lead to the establishment of savings instruments as the active perspective[6] since it mentions the perspective "by name" and explicitly mentions several of the attributes made important by the perspective.

A system attempting to respond to this misconception might proceed by attempting to instantiate the wrong object schema described in Section 5. Recall that

[6] Note that the ROMPER system does not choose the active perspective — it is given as input to the system. This example is simply used to illustrate perspective's influence on misconception responses.

this schema is applicable when there is a similar object which has the property involved in the misconception. The system might collect all objects which have the attribute in question and then test their similarity with the object involved in the misconception. In our knowledge base there are two objects which have the attribute involved in the misconception: Money Market Certificates and T-Bonds.

Assume that the f function in the metric is set solely on the basis of the salience values given by the perspective. Further assume that a positive value returned from Tversky's metric indicates high similarity, while a negative value indicates low similarity. Applying the Tversky metric using the salience values attached by the savings instrument perspective (and assuming a value of 1 for θ, α, and β) we get:

s(T-Bill, MM-Cert) = f(maturity, denom) − f(safety)
= 2 - .5 = 1.5 → high similarity

s(T-Bill, T-Bond) = f(safety) − f(maturity, denom)
= .5 - 2 = -1.5 → low similarity

With these calculations the system would choose the Money Market Certificate as the possible object of confusion and respond something like:

R. Treasury Bills do not have a penalty. Were you thinking of a Money Market Certificate?

Contrast the above calculations with calculations that might occur given a different active perspective. The discourse preceding the misconception utterance in the second example suggests the active perspective of "Issuing Company". Using the salience values attached by this perspective the similarity metric would produce the following calculations:

s(T-Bill, MM-Cert) = f() - f(issuer, safety, purchase)
= 0 - 2.5 = -2.5 → low similarity

s(T-Bill, T-Bond) = f(issuer, safety, purchase) − f()
= 2.5 - 0 = 2.5 → high similarity

In this case a reasonable response by the system might be:

R. Treasury Bills do not have a penalty. Were you thinking of a Treasury Bond?

As the examples show, changes in the active perspective can account for the same misconception being responded to in two different ways.

13. Conclusions

This paper has described a new method for responding to misconceptions that relies on an analysis of a highlighted user model to generate a response that is most likely to benefit the user. A number of strategies were abstracted from a study of transcripts. Each strategy was associated with a "distinguished structure" in the

highlighted user model which could explain its use in a given situation. A system can use this pairing to decide how to respond by looking in the highlighted user model for evidence of one of the "distinguished structures". The corresponding strategy can be used to respond.

In addition, the paper has described a new notion of object perspective which is able to model one aspect of the dynamic highlighting of the user model due to previous discourse. It was shown how perspective could account for certain contextual effects on responses to misconceptions. In particular it was shown how the highlighting due to perspective could cause the Tversky similarity metric to return different values for the same objects. Since object similarity occurs in all of the "distinguished structures" of the highlighted user model, the perspective taken on the domain influences the response strategy used in correcting misconceptions.

This work has proceeded from the perspective of responding to a misconception given a detailed user model. At the same time, it could be looked at as a potential method for inferring user models. User model configurations that support a misconception have been identified. If the user exhibits a misconception, parts of the user model might be inferred by attributing to the user some of the pre-identified supporting beliefs. More work needs to be done in order to determine the feasibility of this approach.

14. References

[BOBR77], [BROW78], [CARB84], [CARB*], [CARN70b], [CHIN*], [COHE*], [FINI*], [GROS77b], [GROS81], [GROS83], [GROS85], [KAPS79], [KARL85], [MAYS80a], [MCCO85a], [MCCO87], [MCCO88], [MCDO80], [MCKE82], [MCKE85c], [MORI*], [QUIL*], [RICH*], [RUME73], [SIDN83a], [SLEE82b], [STEV79], [STEV80], [TOU82], [TVER77], [WEIN84], [WOOL83], [WOOL84a]

10 But What Will the Listener Think? Belief Ascription and Image Maintenance in Dialog

Anthony Jameson

Abstract

If the users of dialog systems which employ user models are to interpret the system's behavior accurately, they must have some understanding of the content of the user model and the purposes for which the system is exploiting it. This fundamental issue has received little attention in user modeling research, but the users and designers of such systems can learn much from their own experience in everyday dialog. Listeners know that the form and content of a speaker's utterance depends on how the speaker assesses the listener's goals and beliefs, and also on the speaker's own dialog motivation. Listeners thus take these factors into account when interpreting utterances, and in particular their interpretations may be incorrect if their assumptions about the speaker are inaccurate. Consequently, speakers often make special efforts to control the image that the listener has of their dialog motivation and their model of the listener. In this chapter, these phenomena are analyzed systematically and illustrated with a simulation model for a restricted type of everyday dialog.

1. Introduction

As the title suggests, this paper was originally motivated by an interest in everyday human dialog. Nonetheless, the issues it addresses represent fundamental and little-studied problems for those who are mainly interested in dialog systems which employ user models. In terms relevant to these goals, the main theme is the importance of the way the users of such systems perceive (a) the content of the user model and (b) the dialog motivation of the system that exploits it.

This theme is an instantiation of a general theme in the literature on human-computer interaction. Here attention has often been drawn to the mental model, or **user's model**, that a user forms of an interactive system (see e.g. [NORM86a, pp. 45-48]). The user's ability to learn to work with a system effectively and to interpret its behavior correctly is said to depend on his or her understanding of the way the system works, though this understanding may be metaphorical rather than technical and literally accurate. Accordingly, designers have often been advised to aim for a system of which a reasonably simple but useful user's model can be formed, and to help the user in various ways to acquire such a model.

Although there has not yet been much empirical research on these ideas, they correspond with general experience and are thus worth taking seriously in designing

dialog systems which include a user modeling component. We will now consider, in turn, two aspects of such a system which are more problematic for the user to perceive appropriately than is generally thought.

1.1. The User's Model of the User Model

It is sometimes suggested that users will simply expect a user model to be complete (in the relevant respects) and accurate (this appears to be presupposed, e.g. by [CARB*, JOSH82, JOSH84a]). If this expectation were justified, it might be unnecessary for users to have any specific knowledge about a system's user model. But even brief inspection of the chapters in this book reminds us that this expectation is unrealistic. Except for the most straightforward cases, user modeling is an inherently speculative and error-prone enterprise, so that in turn users will often only have some notion of what the system thinks about them if they are aware of the system's level of user modeling competence and of the particular strategies employed. It is thus worthwhile to consider how easy this would be for the users of systems which employ various specific techniques. For example, how likely is it that users could anticipate what stereotypes (e.g. level of expertise) were being applied to them, and at what time, by a system using the techniques described by Rich, Finin, Chin or Paris [RICH*, FINI*, CHIN*, PARI*]? Or that they could guess what conclusions about their desires, goals or plans were being inferred from their questions by a system using Morik's isotopy principle [MORI*] or the strategies described by Carberry [CARB*]?

The answers to these questions, and the importance of their implications, are not the same for all of these techniques and systems, but it is in any case clear that it is not always an easy task for the user to form a definite and accurate picture of the system's user model.

1.2. The User's Model of the System's Dialog Motivation

In everyday human dialog, the way in which a listener interprets a speaker's utterance can depend strongly on the motivation that the listener ascribes to the speaker, e.g. whether the listener is being straightforward or diplomatic, cooperative or self-serving. A commonly held view of the role of these factors in human-computer communication is expressed as follows by Maclaran [MACL82, p. 215]:[1]

> It is especially interesting to apply Gricean principles to man-machine interaction since here the goal of co-operative behaviour is more nearly realized: we do not have to worry about deceit, politeness or any of the other interfering factors in man-man communication.

Here again, a critical look at this assumption quickly reveals it to be unjustified.

First, even if the user knows that a system is designed so as to be generally cooperative, he or she may not know what its specific priorities are. For example Shneiderman, a leading advocate of user-oriented interface design, makes several

[1] See also the views cited by Winograd & Flores [WINO86, pp. 156f].

recommendations concerning the formulation of error messages, including the recommendation that they should avoid placing the "blame" for an error on the user [SHNE87, pp. 312-322]. This specific goal can at times conflict with the equally cooperative goal of informing the user precisely and accurately as to what has gone wrong. Thus if a system assigns exceptionally high priority to the former goal, users may be misled by its error messages – unless they realize that the system is "just being polite". (This could happen, for example, if the user was being exceptionally careless in the way he or she entered commands and the system continually apologized for being "unable" to parse them.)

Second, systems may be equipped with dialog motivation which is at least partly noncooperative, even if outright deception is avoided. This is rather obvious for systems which are consulted by occasional users for information or advice and whose owners have an interest in presenting a biased picture. But it can also happen with commercial software packages acquired as tools by individual users. No one expects the advertising brochures or even the printed documentation for such systems to volunteer helpful warnings about the system's unnecessary and irritating limitations or to provide welcome advice on how to get around the copy-protection mechanism; why should one expect such behavior from the on-line tutorial or the intelligent help package that is delivered with such a system? Here again, users can be misled if they are unaware of the system's motivation, which is less predictable in some cases than in others.

In these examples, one should actually speak of the motivation of the system's designers rather than that of the system itself. But in dialog systems as sophisticated as most of those described in this book, the nature of the system's behavior is not determined directly by decisions of the designer; it depends, among other things, on goals which can vary from one application, situation, or user to the next. It is therefore appropriate to think in terms of the motivation of the system itself.

1.3. Approach Taken in This Paper

The above discussion has had to be rather futuristic and hypothetical, since little experience has been acquired to date with the use of systems which employ user models or whose motivation raises complex questions. But fortunately, all of us have extensive relevant experience as participants in everyday dialog. When interpreting the utterances of others, we are accustomed to taking into account not only their dialog motivation but also their model of us as listeners. When generating utterances ourselves, we are aware that they will be interpreted in this way. It is thus appropriate to begin the study of these processes with a systematic analysis of the corresponding phenomena in everyday human dialog. The implications of this analysis for human-computer communication will be considered in the concluding section of this chapter.[2]

Although the phenomena of interest are very general, it is necessary to restrict our attention to a particular type of dialog, so that the analysis does not become

[2] A similar strategy is applied by Thomas (e.g. in [THMA83, pp. 189-192]), in his study of the role of dialog motivation in the use and interpretation of English quantifiers.

excessively abstract, and so that a simulation model can be used to help explore and illustrate all of the main ideas. It is best to choose a type of dialog in which the nature of the speaker's model of the listener can be simpler than in most of the chapters in this volume, and in which contrasting dialog motivations of the speaker are easy to conceptualize.

These conditions are fulfilled by what may be called **evaluation-oriented dialog**, in which the two participants are an **Evaluator** and an **Informant**. (To facilitate exposition, we arbitrarily assume that each Evaluator is male and each Informant female.) The first defining characteristic of this type of dialog is the Evaluator's main goal: to evaluate a given **Object** – which may be, e.g., a thing, a person, or a course of action – according to certain criteria. The Evaluator may want to use his evaluation as the basis for some decision (e.g. if he is a customer deciding whether to purchase a particular product); or he may simply be curious about the Object (as when he is asking a friend whether the party she went to last night was enjoyable).

The Evaluator can ask questions of the Informant, whose main dialog goal is to provide him with some of the information that he needs to apply his evaluation criteria. The Informant makes purely veridical factual statements. In particular, she does not express evaluations of the Object herself but rather leaves it to the Evaluator to determine the evaluative implications of the facts that she presents. Similarly, the Informant may not make any utterance that serves some other possible goal of the Evaluator, e.g. she may not offer advice as to how the Object might be improved.

In many practically important situations, the Object is some aspect of the Informant herself. For example, if the Evaluator has just given a public talk and asks a colleague (the Informant) what she thought of it, he may want to use his colleague's responses not to evaluate his talk but rather to evaluate her personal reactions to it. In a personnel interview, the interviewer or potential employer takes the role of the Evaluator, while the job applicant is simultaneously the Informant and the Object which is being evaluated. It is this specific type of dialog that will be used in the examples in this chapter.

Since this paper is mainly concerned with the way the Informant decides what to say to the Evaluator, the Informant will usually be referred to more simply as the **speaker** (or **S**) and the Evaluator as the **listener** (or **L**).

The theoretical ideas in this paper were developed in parallel with a dialog system called Imp which takes the role of the Informant in an evaluation-oriented dialog.[3] This simulation model will be referred to repeatedly to illustrate the main points. The system is not restricted to any particular topic area, but it does have to be supplied in each case with a good deal of domain-specific information, to be described below. Imp was originally developed and tested in a situation where an Informant was offering a room for rent to the Evaluator [JAME83]. For the examples used in the present paper, the domain-specific information was supplied by a young woman who had recently been interviewed for a position as typesetter in two small printing firms.

[3] The name **Imp** is due to the centrality of the notion of an **impression** in the simplest versions of the system.

1.4. Structure of the Chapter

The simulation model helps us to think about the way in which many diverse factors jointly determine a speaker's behavior – factors which would typically be examined individually in a more traditional psychological or linguistic study. In the interest of clarity, I will begin by discussing an especially simple situation in which very few of these factors are relevant. Then eight further situations will be defined, each of which introduces one or more new complications. In Situations 2 through 6, these complications involve mainly the speaker's dialog motivation and the way the listener takes it into account when interpreting the speaker's utterances. In Situations 7 through 9 they concern the way the speaker's model of the listener is perceived by the listener. It is not until Situation 5 that an actual dialog will take place, since the most basic concepts are easiest to introduce if the speaker simply produces a monolog.

In papers which include discussions of computer simulation models of human processing, it is often difficult for the reader to distinguish between (a) general normative or descriptive claims about how people (should) handle certain tasks and (b) details of a specific system which are either arbitrary or dictated by technical constraints. To help make explicit the status of the various points made in this paper, and also to facilitate selective reading, I will discuss each of the nine situations in a standard sequence of six subsections with the following headings:

1. Definition of the Situation
2. How the Situation Can Arise
3. Appropriate Strategies
4. Realization in the Simulation Model
5. Resulting Behavior
6. General Issues Raised

The first subsection lists several postulates, mainly about the speaker's beliefs and dialog motivation. Taken together with the postulates of all previous situations, they characterize the situation to be analyzed. These postulates are intended to define useful idealizations of situations that occur in actual dialogs.

Subsection 2 sketches some of the circumstances in which a situation approximated by the definition can occur. These remarks are not systematic or complete; they are intended only to suggest the range of applicability of the analysis and to help the reader call to mind relevant experiences.

The third subsection lists some appropriate strategies for a speaker in the given situation. The normative claim is that it is reasonable for S to apply these strategies when pursuing the postulated goals. The descriptive, or empirical claim is only that at least some people sometimes apply these strategies. Any stronger claim, even in terms of average behavior, would in many cases be false or misleading. For example, as Sampson [SAMP82] reminds us, some speakers consistently neglect even the most elementary appropriate strategies discussed in the early situations. Since I assume that the reader's own experience as a dialog participant will provide abundant examples of the phenomena discussed, I will not quote specific observed

examples of dialog behavior (as do, e.g., [CARB*, MCCO*, PARI*, QUIL*] in this volume).

Subsection 4 sketches the representational techniques and procedures used to implement the strategies introduced in Subsection 3 in the simulation model.[4] No normative or empirical claim is made about any additional details presented in this subsection, and in fact the implementation requires many arbitrary decisions and simplifications which are clearly inapplicable to human processing. Since many such decisions are reflected in the various figures which illustrate how the system works, these figures should be viewed not as summarizing the theoretical points made in the other subsections but rather as depicting metaphors which may facilitate thinking about these points. In particular, no attention is given to the many possible shortcuts and heuristics which would be needed for efficient real-time application of the strategies in question. Readers with no particular interest in how the simulation model works can safely skip this fourth subsection.

The fifth subsection lists the main consequences of the application of the strategies introduced in the third subsection. These consequences are illustrated with examples of the simulation model's performance, and in fact some of them became apparent to me only through experimentation with the model and discussion of its behavior with others. Once noticed, however, all of the consequences mentioned can be seen to follow generally from the "appropriate strategies"; features of the examples which merely reflect implementation details receive no attention.

The final subsection for each situation comments on questions which the preceding subsections called to mind but did not address, and in some cases it gives pointers to research which does deal with these questions.

The entire analysis is made from the point of view of the speaker, and in principle all statements about the dialog should begin with "S believes ...", e.g. "S believes that L will misinterpret her comment". Since this type of formulation can be extremely cumbersome, especially in the more complex situations, I will often write, e.g., "L will misinterpret S's comment", leaving it to the reader to bear in mind that reference is actually being made to a belief of the speaker.

2. Situation 1: Generating an Informative Monolog

2.1. Definition of the Situation

It will be assumed in the first four situations that the speaker in effect has the opportunity to generate an unconstrained monolog about the Object. For example, a potential employer, as listener, might say to a job applicant "Tell me about yourself." In this first situation, we assume that the speaker is motivated to produce in the listener an unbiased and sufficiently definite picture of the Object. The only

[4] The system is implemented in the Lisp-embedded object-oriented language Orbit designed by Koenraad De Smedt, but it will not be necessary to refer to any specific properties of this language.

reason why she might withhold any information is that mentioning it did not seem to be worth the effort.

2.2. How the Situation Can Arise

These assumptions are actually less characteristic of a face-to-face employment interview than of a case where S is an impartial reference writing a confidential letter of recommendation. But even in an interview, the candidate might have this sort of cooperative motivation if, for example, she only wanted to be hired if she were truly the best candidate.

2.3. Appropriate Strategies

2.3.1. Necessary Assumptions of the Speaker

Even in this simple situation, in order to behave appropriately, the speaker must be able to make some reasonable assumptions about the **evaluation standards** of the listener. Specifically, S must be aware that there are various aspects of the Object that L is concerned about (e.g. the candidate's reason for leaving her previous job). For each such aspect, or **item**, there are various **possibilities** which might be true of the Object (e.g. different types of reasons for leaving a job). L presumably not only considers some of these possibilities more desirable than others but also has some expectation as to how likely they are to be found in the Object.

S must therefore try to guess what items, possibilities, values, and likelihoods L has in mind. How she should do this is a complex question which will not receive much attention in this chapter (although it will be mentioned in connection with Situations 7 through 9).

In order to be able to volunteer information relevant to a given item, the system must have an idea of what **comments** might conceivably be made which are germane to that item, and what implications they have. One in general has the option of making a vague comment on an item, which leaves open several possibilities that L would consider worth distinguishing, or a specific comment which tells L all that he might like to know for the purpose of evaluating the Object. The reason why cooperative speakers do not always make the latter type of comment on all items is that any comment requires a certain amount of effort for the speaker to produce and for the listener to interpret, and precise comments often require more effort than vague ones. Thus although the concept of the **effort** associated with various comments is an elusive one,[5] S must have some way of taking this consideration into account.

2.3.2. General Strategy

A reasonable general strategy is to consider each relevant item and to decide what, if anything, is worth saying about it. That is, S must choose for each item, from

[5] Discussion of some of the factors involved can be found, e.g., in [GAZD79, chap. 3, GRIC75, MCCL78, WINO77].

among the possible comments for that item which are true of the Object, the one comment which is the most worth making, i.e., the one whose benefits most clearly justify the amount of effort involved.

2.3.3. Assessing the Potential Benefits of Making a Comment

What are the benefits which in this situation can result from the production of an informative comment? It has often been pointed out that the value of a statement to the listener can depend on both (a) the extent to which L would have suspected that it was true without being told and (b) the consequences for L of its being true rather than false (see e.g. [GREW83, HARR63, chap. 4, HINT75]). It is in general difficult to use these criteria to arrive at a precise, quantitative definition, but this can be done quite easily within evaluation-oriented dialog. The key concept is that of an **impression**. At any given moment and for any given item, the listener can be said to have an impression of the Object with respect to that item. The nature of this impression depends on which possibilities the listener (still) thinks might be realized by the Object. Before the speaker has said anything about a given item, the listener's impression concerning it will be relatively indefinite, since all of the possibilities must be considered (although some are less likely than others). If S makes a certain comment concerning the item, L can in general eliminate one or more possibilities, so that the impression will become more definite. It will in general also become somewhat more or less **favorable** than it was before, depending on whether the possibilities eliminated were among the less or more desirable ones.

There are at least two types of impression change that can be desirable for a speaker in this situation. First, S should consider it desirable if a comment makes the impression with respect to an item more definite, since this reduces L's uncertainty about the evaluation of the Object with respect to that item (and thus overall). But since even a fairly definite impression may be considerably more or less favorable than justified by the truth about the Object, it is also desirable that a comment reduce, or at least not create, such a discrepancy.

The desirable consequences of any given comment must somehow be integrated and weighed against the effort required, so that the **net desirability** of the comment can be determined. The comment with the greatest net desirability can then be verbalized; or if no comment seems worth the effort, S should say nothing about the item in question.

To recapitulate: for each evaluation-relevant aspect of the Object, S should verbalize the comment (if any) whose prospective benefits, in terms of an impression change in L, most clearly justify the effort associated with it.

2.4. Realization in the Simulation Model

A convenient representation for L's evaluation standards is a data structure which may be called an **evaluation form**. This representation is reminiscent of the sort of rating form which is often found in practical situations for the systematic evaluation of objects or persons (e.g. papers submitted to conferences, or applicants to universities). In Imp the evaluation form consists of a set of **items**, each of which

Item 9: Family Situation

Possibilities:

		Value	Probability
A.	Married without children.	-35	.15
B.	Married with children.	-20	.05
C.	Living together.	-10	.35
D.	Living apart.	0	.25
E.	No steady relationship.	10	.20

Formulations:

	Possibilities left open	Effort
I am married.	A B	50
I am not married.	C D E	50
I have <number> children.	B C	50
I don't have any children.	A D E	50
I live alone.	D E	50
I have a boyfriend.	C D	75
I am married and I don't have children.	A	100
I am married and I have <number> children.	B	100
I am living with my boyfriend.	C	100
I have a boyfriend, but we don't live together.	D	100
I don't have any steady relationship.	E	100

Figure 1. An item from the evaluation form assumed by the Informant of the examples.

concerns some evaluation-relevant aspect of the Object. Twelve items were supplied by our example Informant to characterize the assumptions she made during the first of her two interviews. One of these is reproduced in Figure 1.

Each item is a sort of multiple-choice question accompanied by a scoring key. An item specifies several conditions, or **possibilities**, which are defined in such a way that exactly one of them must be realized by the Object. With each possibility, two numbers are associated: (a) a **value**, i.e., a number of points which may be assigned to the Object if it satisfies the condition; and (b) a **probability**, i.e., an estimate of the a priori likelihood that the Object will satisfy that condition. The item shown in Figure 1 reflects, among other things, the speaker's belief that her prospective employer would be disappointed to hear that she was married, though less so in the (unlikely) case that she already had children.

If the listener were actually to use an evaluation form to evaluate the Object, he would attempt, for each item, to determine which of the possibilities was realized by

the Object. If he succeeded, he would add the corresponding value to his rating of the Object. If the information available was not sufficient to rule out all possibilities but one, he would use the probabilities specified for the remaining possibilities to estimate how many points the Object deserved for the item.[6] Imp in effect acts as if it expected that the listener would use the information it provided to fill in the evaluation form.

The system possesses no methods for formulating entirely new comments on its own. It must be provided, for each item, with a list of possible comments which are representative of the infinite number of comments that might conceivably be made. For example, Figure 1 shows a number of things that can be said about a female applicant's family situation. For each comment, Imp must be told which possibilities are consistent with it on a straightforward interpretation and how much effort is associated with it. These numerical ratings of effort are impossible to justify in detail, but for our purposes it is sufficient if the major differences among comments are reflected in different numerical ratings.

It is convenient to view **silence** as a special sort of comment which can be made about any item, which (when interpreted straightforwardly) is consistent with all possibilities, and which involves no effort.

To be able to quantify the desirability of a given impression change, Imp operationalizes the concept of an **impression** in terms of a probability distribution, determined by the values and the probabilities associated with the possibilities which have not yet been eliminated from consideration. Figure 2 shows graphically the impression that the listener is assumed to have initially concerning the "Family situation" item. It also shows the impressions that would remain after each of two possible comments that the speaker might make.

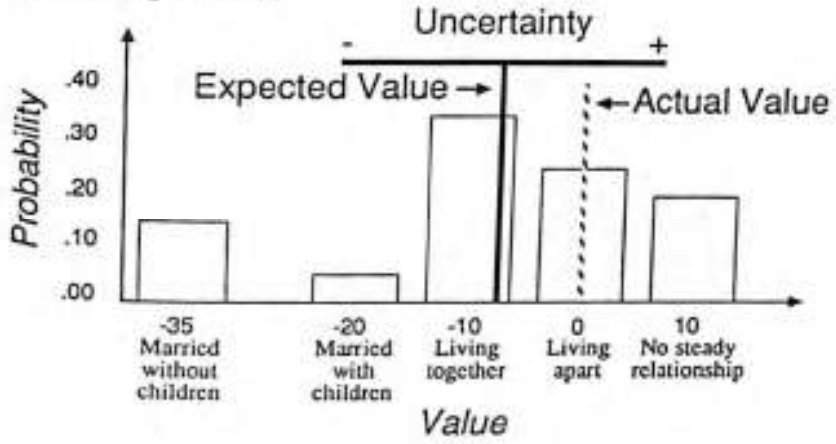
The idea of how favorable an impression is can now be captured in terms of the mean, or **expected value** of the probability distribution in question; for example, in Figure 2 the first comment rules out the three least favorable possibilities, and as a consequence the expected value rises; it would actually rise more sharply if it weren't for the fact that the two least favorable possibilities are considered relatively improbable to begin with.

The second important property of an impression, the degree of uncertainty associated with it, can be quantified in terms of the standard deviation of the probability distribution. In general, the greater the number of possibilities that have not yet been eliminated, the greater the degree of uncertainty associated with an impression (this is illustrated in Figure 2); but the standard deviation captures this concept better than a mere count of the remaining possibilities would, since it is sensitive to the relative likelihoods of the remaining possibilities and the differences in their values.

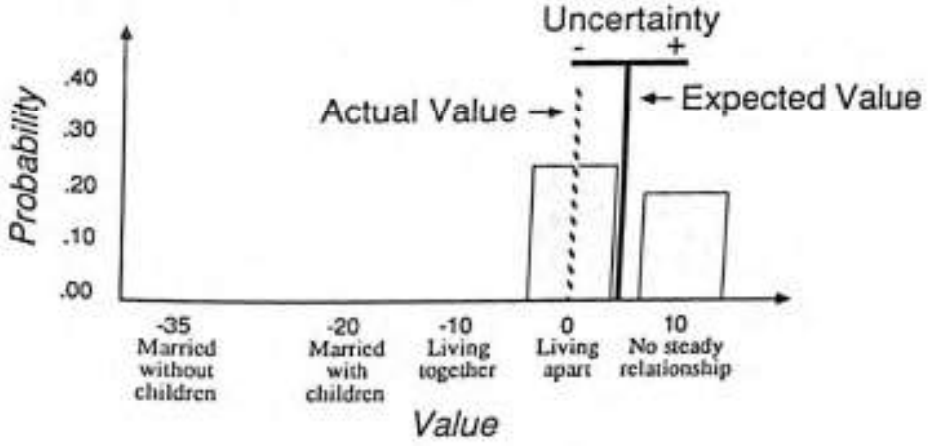
---

[6] This way of using an evaluation form only makes sense if the items are independent of each other, in the sense that knowing what possibility is realized for one item cannot affect one's assessment of the probabilities associated with any other item. A simple way of avoiding such dependencies is to mention within a single item any properties of Objects which are substantially correlated. This is why, for example, the distinctions "married vs. single" and "children vs. no children" are not captured in separate items. Rowe [ROWE84] presents a much more sophisticated, practically oriented technique for evaluating objects from the point of view of the user of a database query system.

Initial impression:



Impression after comment "I live alone":



Impression after "I have a boyfriend, but we don't live together":

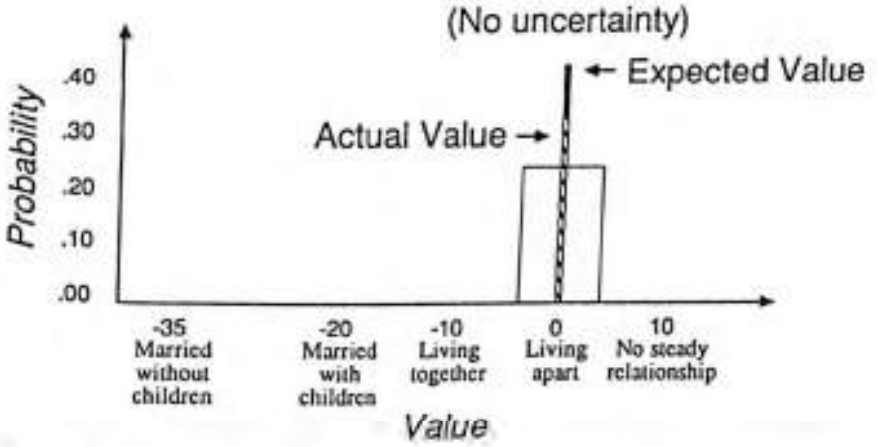


Figure 2. Illustration of the concepts of an **impression** and its associated **expected value** and **uncertainty** (see discussion in text).

The two properties of an impression change – reduction in uncertainty and reduction in discrepancy from the actual value of the Object – are largely independent consequences which can be considered desirable. Imp simply takes the sum of these two quantities to determine the total benefits which can be expected to result from a given comment. Thus in Figure 2, the second comment produces somewhat greater total benefits than the first one does, since it reduces all of the uncertainty and all of the discrepancy between the expected value and the true value. But it is likely that a speaker might prefer the first comment to the second one on the grounds that the additional benefits produced by the second one do not justify the additional effort involved.

In finally deciding which comment to make on a given item, Imp subtracts the rating of the effort associated with each possible comment from its total expected benefits to yield a quantification of the net desirability of the comment (cf. Figure 3). A "talkativeness" parameter is also required to take into account both the individual differences in the general willingness of speakers to make comments and the arbitrariness of the units used to quantify the amount of effort associated with a comment.

Net desirability of a comment C on an item I =
Benefits of C
− Costs of C

Benefits of C =
Reduction of uncertainty of impression concerning I
+ Shift in expected value of this impression
toward true value
or in direction corresponding to bias
(not applicable in Situation 1)

Costs of C =
Effort associated with C / Talkativeness of Informant

Figure 3. Formulas used by Imp in Situations 1 through 4 to determine how worthwhile it is to make a given comment.

The comment with the highest net desirability is the one which is actually made. In many cases the most desirable comment will simply be silence (which, as can be seen from the above definitions, always has a net desirability of zero), since

the benefits that would be brought about by the other possible comments are less than the associated cost in terms of effort.

2.5. Resulting Behavior

The first two columns of Figure 4 show what comments Imp volunteered in this situation on each of the twelve items in the evaluation form. The way in which all of the comments are connected into a single long sentence, which is rather unnatural in this situation, will be discussed in connection with Situation 5. The following general tendencies result from the application of the strategies sketched above:

1. On items which are exceptionally important – i.e. about which the listener experiences a great deal of evaluative uncertainty – the speaker will in general make some comment even if the truth about the Object is not at all unusual.

With such an item, the effort required by a comment may be justified by the reduction in uncertainty that results, even if L's initial impression is not inappropriately favorable or unfavorable. The system's first comment in Figure 4 provides an example: a year and a half of work experience is a normal amount for this type of applicant, but it is worthwhile to reassure L that the alternative possibilities, some of which would be much more or less favorable, are not realized.

2. Moderately important items will in general be commented upon only if the truth is either exceptionally desirable or undesirable.

The second and third comments mention the most disappointing possibilities for their respective items, thus making L's vague initial impressions not only more definite but also more in line with reality. This second benefit is necessary here to warrant the generation of a comment, since the reduction of L's modest amount of uncertainty would not be a substantial benefit. The same is true for the comments on Items 6 and 7, except that these comments are exceptionally favorable. On Items 10 and 12, S would say something if the truth deviated sharply from L's expectations – e.g. if the applicant lived in some distant town; but since the extreme possibilities are improbable, L will not worry about these items if nothing is said about them.

This tendency for extreme possibilities to be mentioned relatively frequently is strengthened by the fact that intermediate possibilities are often more awkward to describe. Thus it requires less effort to say that one's previous work was "very challenging" or "purely routine" than to characterize it as "sometimes pretty challenging, but usually not".

3. There may be some items about which the speaker would say nothing no matter what the truth was.

With topics such as the degree of variety of the applicant's previous work experience (Item 2), there is no possibility which is so desirable or undesirable as to be worth mentioning (unless L asks about the topic specifically, as in Situation 5).

2.6. General Issues Raised

The goals of correcting inaccuracies and reducing uncertainty make sense for a cooperative speaker in many types of dialog situation. For example, if a speaker is

Item	Comments Chosen		Pragmatic Interpretations Expected (after comments in third column)	
	Situation 1 (objective bias)	Situations 2 - 4 (positive bias)	Situation 3 (positive projected bias)	Situation 4 (objective projected bias)
1. Amount of work experience	I'VE BEEN WORKING FOR ABOUT A YEAR AND A HALF	I'VE BEEN WORKING FOR MORE THAN A YEAR	1. Between 1 and 2 years experience.	[Comment is inconsistent with projected objectivity.]
2. Variety of work experience			2.	
3. Complexity of previous work	BUT MY WORK HAS BEEN PURELY ROUTINE		3. Not especially challenging work.	Not especially challenging work.
4. Vocational training	AND UNFORTUNATELY I NEVER RECEIVED ANY VOCATIONAL TRAINING		4. No higher-level vocational training.	No higher-level vocational training, but some training.
5. Work-related hobbies			5.	
6. Reason for changing jobs	BUT THE REASON I WANT TO LEAVE MY PRESENT JOB IS THAT I'M LOOKING FOR MORE CHALLENGING WORK	AND THE REASON I WANT TO LEAVE MY PRESENT JOB IS THAT I'M LOOKING FOR MORE CHALLENGING WORK	6.	
7. Nonvocational schooling	AND I WENT TO UNIVERSITY PREPARATORY SCHOOL	AND I WENT TO UNIVERSITY PREPARATORY SCHOOL	7.	
8. Plans for future education			8.	
9. Family situation			9.	
10. Place of residence			10.	Does not live especially far away.
11. Frequency of previous job changes			11.	
12. Nature of position ultimately desired			12.	Is not aiming for position unattainable in this company.

Figure 4. Monologs produced by Imp in Situations 1 through 4.

Page 314 of 1206

giving the listener advice on how to perform a certain task, it may be worthwhile not only to prevent L from performing inappropriate actions but also to reassure him when his intended action is in fact appropriate. The relative weight that S should attach to these two goals presumably varies from one specific context to the next.

The way in which the key theoretical concepts in this section are realized in the simulation model is dictated largely by a desire to keep the system simple, but the operationalizations are reasonably consistent with the way people actually evaluate objects, according to relevant empirical evidence. For example, studies reported by Wyer [WYER74, chap. 2] can be interpreted as supporting the ways in which the concepts of **expected value** and **uncertainty** are operationalized. On the other hand, research within the paradigm of Information Integration Theory (see e.g. [ANDN81] for a brief summary) has shown, for example, that people often evaluate an Object less favorably if they hear one extremely positive and one moderately positive fact about it than if they hear only the extremely positive fact. This result is incompatible with the evaluation-form metaphor; but note that this is not a problem for the present analysis, since presumably few speakers are aware of this sort of phenomenon.

Persons who witness demonstrations of the Imp system often remark that they find a particular utterance implausible in the sense that they, as speakers, would never have produced such an utterance. In some instances reactions of this sort have exposed inadequacies in the theory underlying the simulation model and have led to revisions of both. But in most cases it turns out that the observer implicitly ascribes different specific values and expectations to the Evaluator than those which have been fed into the simulation model: when the observer is shown the relevant part of the evaluation form used (i.e. the sort of information given in Figure 1), the system's behavior seems reasonable after all. In short, this problem in evaluating the system's performance illustrates one of the general points made in the Introduction: It can be difficult to interpret the behavior of a user modeling system if one does not know the contents of the user model. In Situations 7 through 9, the simulation model will be sophisticated enough to take this difficulty into account.

3. Situation 2: Positive or Negative Bias

3.1. Definition of the Situation

In this situation the speaker is not necessarily attempting to be as cooperative as possible but rather may have the goal of presenting the Object to the listener in an especially favorable or unfavorable light. As before, however, S restricts herself to strictly truthful factual comments which are not explicitly evaluative.

3.2. How the Situation Can Arise

Bias is of course not found only in contexts, such as the employment interview, where it can produce tangible benefits for the speaker. The creation of a favorable (or unfavorable) impression in the listener may be seen by the speaker as a desirable end in itself.

Negative bias is even more improbable than objectivity in an employment interview, but it might be found, e.g., if the candidate didn't want the job at all but was obligated to apply in order to continue to qualify for unemployment benefits.

3.3. Appropriate Strategies

Within the constraints of this situation, positive or negative bias on the part of S can only affect her criteria for selecting comments from among the set of possible truthful comments about the Object. Specifically, it is now no longer necessarily desirable to eliminate discrepancies between the listener's impressions and the truth about the Object; instead, the individual impressions should be made as favorable (or unfavorable) as possible. The other considerations discussed in the previous situation, e.g. the desirability of reducing L's uncertainty, are still relevant, though they may be given different weights relative to the first consideration.

3.4. Realization in the Simulation Model

The Imp system includes a parameter **bias** which can take the values **objective** (as assumed implicitly in Situation 1), **positive**, or **negative**. This parameter determines how possible comments are rated in terms of their expected impact on the listener's impressions. As shown in Figure 3, when the bias is positive or negative the important aspect of the shift in the expected value of an impression is now the extent to which it is in the positive (or negative) direction. Thus an impression shift toward the true value is actually undesirable if it doesn't happen to be in the desired direction.

3.5. Resulting Behavior

For concreteness, the consequences of bias will be described for the case where S's bias is positive, as in the examples. Because of the symmetric relationship between positive and negative bias, the consequences of negative bias are easy to derive.

1. The speaker now seldom volunteers a comment which would cause an impression shift in the negative direction.

As can be seen in the third column of Figure 4, the comments made in Situation 1 on the routine nature of the speaker's previous work and her lack of vocational training are not made in this situation. Their net desirability is now less than that of silence: they eliminate all uncertainty, but the way they shift L's impressions downward is now considered undesirable. The net desirability of the comment is

therefore small or even negative, and it certainly does not justify the effort of making the comment.

2. The speaker will sometimes choose a vague formulation which leaves open one or more possibilities that are more favorable than the one actually realized, even though a more precise comment would involve no more effort.

This is illustrated by the comment on the first item in Figure 4. The statement that the candidate has been working for "more than a year" leaves open the possibility that she might have several years of experience and thus leaves an impression which, though vague, is considerably more favorable than a definite and accurate impression would be. To be sure, the listener will probably not be so naive as to take this statement at face value; the reasons for this will be examined in the next situation.

3.6. General Issues Raised

The phenomenon of bias is obviously not restricted to contexts in which some Object is being evaluated. Behavior analogous to that described above can be found, e.g., in the attempt of a lawyer to lead a jury to a particular substantive conclusion. By abstracting away from almost all problems directly involving natural language, the above analysis focuses attention on simple selectivity as a mechanism of bias, as opposed, say, to rhetorical techniques involving particular expressions (cf. [HOVY86]).

Whatever particular benefits the speaker may expect from a biased selection of information, they must somehow be assessed and integrated with other factors such as the extent to which the listener's uncertainty is reduced and the amount of effort required to make one's comments. Biased speakers may tend to attach different weights to the various considerations than objective speakers do. For instance, they may attach less weight to the goal of reducing uncertainty, aiming almost exclusively for impression shifts in a particular direction. Another possibility is that such a speaker may increase her general talkativeness in order to compensate for the fact that she is, because of her selectivity, producing fewer utterances.

4. Situation 3: Anticipating the Pragmatic Interpretation of Comments

4.1. Definition of the Situation

When anticipating the impact that a given comment would have on the listener, the speaker has up to now assumed that each comment would be taken at face value: the only possibilities which the listener was expected to rule out were the ones which were incompatible with a straightforward interpretation of her comment. But in fact listeners often rule out further possibilities by noticing that the speaker has chosen not to say certain things which she might well have said. It will be

assumed in this situation that the speaker is sophisticated enough to realize that her comments will be interpreted in this way.

4.2. How the Situation Can Arise

Whenever a listener has a reasonably clear idea of the criteria and knowledge a speaker uses in generating utterances, he can consider what S would have said in various circumstances and thus extract information from S's utterances which goes beyond their conventional implications.

4.3. Appropriate Strategies

The basic reasoning that L can use to interpret the fact that S did not say certain things is summarized in the following schema: "Possibility P is not ruled out by the speaker's comment C [which may be silence]; but then again, if P were the possibility that was realized, S would surely have said something like C' instead of C, since C' would have had a more desirable impact on my impression; therefore, P is apparently not realized." If L uses this schema, perhaps ruling out one or more of the possibilities allowed by a comment, we will say that he has made a **pragmatic interpretation** of that comment.[7]

S must anticipate L's pragmatic interpretations of her comments if she wants to maintain an accurate picture of L's impressions of the Object. (It will become increasingly important for S to keep track of L's impressions in the more complex situations to be examined below.)

4.4. Realization in the Simulation Model

Figure 5 summarizes the two procedures that Imp uses to generate comments (some parts of the procedures will only become applicable in Situation 4). As can be seen at the beginning of the first procedure, Imp begins by selecting the most desirable comment, using the same criteria as in the previous situations. Before verbalizing this comment, however, it now uses the second procedure to determine what L will be able to infer by applying the pragmatic interpretation schema given above.

There is one aspect of this procedure for anticipating pragmatic interpretations that will not be discussed in detail until Situation 8, although it does have some effect on Imp's behavior even in the earlier situations: the specification (in Step 2), that a possibility should be ruled out only if it is "clear" to L that S would have made some other comment if that possibility were realized. This condition becomes

[7] This concept is more general than Grice's conception of the way in which a conversational implicature is "worked out" by the listener [GRIC75, p. 50]. The notion of conversational implicature presupposes, among other things, (a) that the speaker is observing the cooperative principle, and (b) that the interpretation made by the listener depends on L's recognition of S's expectation that L will make that interpretation. By contrast, a pragmatic interpretation as defined here could be made by a listener who was eavesdropping on a speaker who was being blatantly uncooperative.

To volunteer a comment on a given item:

1. Find the comment on that item with the greatest net desirability [cf. Figure 3] (given your bias and the evaluation standards you ascribe to L).
2. Anticipate L's Pragmatic Interpretation of this comment [see below] (assuming that he has the image of you that you want to maintain).
3. [Applicable in Situations 4 - 9:]
   If this pragmatic interpretation is empty [i.e., the comment is incompatible with your image], then
      Reject this comment.
      Return to Step 2, using the comment with the next greatest net desirability.
4. Record L.'s pragmatic interpretation and verbalize the comment.

To anticipate the pragmatic interpretation of a comment C on a given item (given a certain image that you want to maintain):

1. Consider the pragmatic interpretation initially to be the set of all possibilities for the item that are consistent with a straightforward interpretation of the comment.
2. For each such possibility P,
      Is it clear to L that some alternative comment C' would be considered more desirable than C by a speaker who corresponded to your image, if P were realized?
      If so, eliminate P from the pragmatic interpretation.
3. Return as the pragmatic interpretation the set of possibilities that remain. [In Situations 4 - 9, this set may be empty.]

Figure 5. Two of Imp's main procedures for selecting comments.

important in cases where L cannot predict exactly what the speaker would say in various particular circumstances.

4.5. Resulting Behavior

1. If S says nothing about a given item, she will sometimes anticipate that L will rule out one or more of the possibilities for that item which are especially favorable (or unfavorable).

This is illustrated with Items 3 and 4 in Figure 4 (second column from the right): Since S would have considered the most favorable possibilities worth mentioning, she anticipates that L will interpret her silence on these items as implying that these possibilities are not realized.

2. S will often expect some of her vague comments to be given a more precise and accurate interpretation.

For instance, S now sees that her attempt to create an inflated impression of her amount of work experience (Item 1) will not be successful after all: L will recognize that if S had substantially more than one year of experience she would have said so.

3. The pragmatic interpretation of biased comments tends to undo, in part, the distorting effects of bias.

In the example just mentioned, the ultimate impact of S's biased comment is just about the same as that of the corresponding unbiased comment of Situation 1. But S can still derive some benefit from her bias, as with her silence on Items 3 and 4. Although L rules out the most favorable possibility for each item, he continues to reckon with the possibilities of intermediate favorability, whereas it is the least favorable possibility that is actually realized.

4.6. General Issues Raised

The strategy of simulating the listener's interpretation of an utterance before verbalizing it is applicable in many contexts (Jameson & Wahlster [JAME82] present a different sort of application and a discussion in more general terms). The present application is unusual in that the listener is in turn assumed to interpret an utterance by simulating the behavior of the speaker. Thus S's simulation of L's interpretation process involves largely considering how she herself would behave under various circumstances.

In the present situation, S only anticipates a pragmatic interpretation after having made a definite decision about what comment to make. It would in principle be possible for her to adopt a more sophisticated strategy, anticipating the pragmatic interpretation of each candidate comment when deciding which one is most desirable. This strategy could, among other things, help S to avoid investing unnecessary effort in making a comment which, after pragmatic interpretation, had exactly the same impact as a comment requiring less effort. On the other hand, it would require considerable extra processing on the part of S and could even involve an infinite regress if S thought that L knew that S was choosing her comments in this way. It would be interesting to investigate the conditions under which speakers apply some more complex strategy of this sort.

5. Situation 4: Discrepancy Between Actual and Projected Biases

5.1. Definition of the Situation

In Situation 3, the speaker implicitly assumed that the listener had a remarkable degree of awareness of the specific way in which she selected her comments. In the present situation, we will consider the possibility that L incorrectly assesses the nature of the speaker's bias. In other words, S now assumes that she has a certain **projected bias** which L ascribes to her and which may differ from her actual bias. We also assume that S will not attempt to change the nature of her projected bias, but on the contrary will attempt to behave in such a manner that the listener will continue to ascribe this bias to her.

5.2. How the Situation Can Arise

Social or group norms often produce a situation in which the listener expects the speaker to have a certain type of bias and the speaker wants to avoid violating this expectation. There are obviously many situations in which one is expected to be objective whether one wants to or not, but in some contexts (such as the personnel interview) positive bias is considered perfectly normal, and in others (such as eulogies), it is actually obligatory. More generally, in order to be fashionable or politically acceptable in a given community, one is often constrained to exhibit positive or negative bias with respect to certain Objects.

S may also want to project a particular bias because of the consequences that this will have for the way L interprets her utterances. For example, as will be seen in this section, the self-descriptive statements of a job applicant will often be interpreted in a more favorable manner if she manages to come across as being objective. This sort of motivation will be discussed in connection with Situation 6, which looks at attempts by S to create an expectation about her bias that did not previously exist.

5.3. Appropriate Strategies

The speaker should continue to be guided by her actual bias in deciding what she would like to say, since this bias determines the nature of the impression changes that she considers desirable. But when anticipating the pragmatic interpretations that L will make of her utterances, she should consider how desirable the various possible comments would be to a speaker who had the bias which she is attempting to project, since this is what L will be considering when interpreting her comments.

As we will see, it can happen in this situation that the comment that S would most like to make would not be made by a speaker with the bias which is being projected, no matter what the truth about the Object was. In this case, the comment would seem peculiar to L, who would not be able to understand why it was made, and he might then suspect that S's bias was not what it seemed to be. To prevent this from happening, S should avoid making any such comment and choose instead one which she considers less desirable but which is at least consistent with her projected bias.

5.4. Realization in the Simulation Model

In addition to the parameter which represents its actual bias, Imp has a parameter representing its assumed projected bias. These two parameters can have the same value (as implicitly assumed in Situation 3) or different values. The actual bias determines the system's initial choice of a comment on a given item, whereas the projected bias is consulted when the system anticipates the pragmatic interpretation that the listener would give to the comment. Thus in Figure 5 the "image" on which the pragmatic interpretation is based includes S's projected bias.

Step 3 in the first procedure in this figure realizes the strategy of avoiding comments which would be incompatible with this image.

5.5. Resulting Behavior

The right-hand column of Figure 4 shows the pragmatic interpretations expected in this situation, which can be compared with those of the previous situation. Note first that one narrower pragmatic interpretation, and two new ones, are made because the projected bias is objective rather than positive. With respect to Items 4, 10, and 12, on which S says nothing, L will now rule out the least desirable possibility, because of his faith that S would have made some corresponding comment if this possibility had been realized. This faith is, of course, not justified: in the case of Item 4, the least desirable possibility is in fact realized. This illustrates the first major consequence of a discrepancy between the actual and projected biases:

1. The listener's pragmatic interpretation of a comment may be incorrect, excluding the possibility that is actually realized.

The second main consequence has already been mentioned:

2. The speaker will sometimes be forced to make some comment other than the one she would most like to make, since the preferred comment would be inconsistent with her projected bias.

This happens in Figure 4 with Item 1: S determines that for any amount of work experience consistent with the formulation "more than a year", there is some more precise formulation which would be preferred by an objective speaker. The vague comment must thus be rejected, and S falls back on the more precise specification "about a year and a half". (This happens to be exactly the comment that an objective speaker would make in this case; but this is not a necessary consequence of the speaker's general strategy.)

5.6. General Issues Raised

5.6.1. Examples from Empirical Research

An illustration of the importance of both bias and projected bias was provided by Ceci and Peters [CECI84] in their study of the way faculty advisors rated students in confidential and nonconfidential letters of recommendation. Ratings were systematically higher in the nonconfidential letters, suggesting that the writers adopted a positive bias in order to avoid spoiling their relationship with their student. On the other hand, it was also found that this effect was well-known among the sort of people who tend to read letters of recommendation – i.e. that the projected bias of nonconfidential letter-writers was also positive. The authors not only hypothesize that this projected bias tends to undo the effects of the actual bias when the letters are read; they also speculate that some writers only adopt a positive bias in the first place because they expect that the projected bias will be positive and that therefore an objective letter would lead to the student's being underrated.

An experiment reported by Thomas (e.g. in [THMA83, pp. 189-192]) showed that the conditions under which subjects used potentially misleading formulations such as "Some X are Y" depended on whether their motivation was cooperative or competitive. The way other subjects interpreted such formulations depended on their assumptions about the motivation of the subjects who produced them.

5.6.2. Similar Patterns with Other Parameters

The distinction made here between actual and projected bias is in principle applicable to all of the other parameters that influence S's selection of comments, including S's assumptions about L's evaluation standards (discrepancies involving these will be examined in Situations 7 through 9). The importance of "projected talkativeness", for instance, is illustrated by the possibility that a job applicant who had few positive things to say about herself might try to come across as a "quiet type", so that her reticence would not give rise to much in the way of pragmatic interpretations.

5.6.3. More Deeply Embedded Discrepancies

This situation examined the consequences of a discrepancy between S's actual bias and the bias that S believes that L ascribes to her. Further discrepancies can involve more deeply embedded beliefs. For example, S may think that the situation is as follows: L recognizes S's positive bias but thinks that S is confident that she is projecting objectivity. L thus expects S to anticipate some incorrect pragmatic interpretations that L would not in fact be so naive as to make.

More generally, there are an infinite number of propositions of the form

$$[\text{S believes that L believes that}]^{N} \text{ S's bias is B,}$$

in which the bracketed expression is repeated N times. With N equal to 0 or 1, we obtain propositions about S's actual and projected biases, respectively; the more complex example just given involves, among others, a proposition with N equal to 2. In principle, B can take on arbitrary values in all of these propositions. But if the value of B for some N is different from the value for N+1, S thinks that L has some incorrect belief concerning her bias (at some level of embedding). This incorrect belief can in turn cause L to make further incorrect inferences. (Clark & Marshall [CLHH81], demonstrated a similar point with respect to the use of definite referential expressions.)

We will not go on to consider the consequences of discrepancies like the one in the example just given, or of still more complex discrepancies, since the conditions under which S can come to have such a set of beliefs become increasingly complicated and improbable. Rather, we will continue to assume – as implicitly in the present situation – that the values of B in all propositions of this form are equal, except that the value when N is zero may differ from the other values. This amounts to the assumption that S believes (not necessarily correctly) that L believes that S's bias is **mutually known** to the two of them (see [KOBS*] for an explication of this notion).

6. Situation 5: Responding to Specific Questions

6.1. Definition of the Situation

In this situation an actual dialog takes place. The listener asks the speaker questions which may be quite open or very specific; but each question clearly concerns some evaluation-relevant item or a set of related items. The speaker answers these questions in accordance with normal dialog conventions, but she may also take the opportunity to volunteer additional relevant information which falls outside of the scope of the question as strictly interpreted. To keep the complexity of the present analyses within reasonable bounds, we must also assume that S is not inclined to use L's questions as evidence concerning L's beliefs or values.

6.2. How the Situation Can Arise

This last assumption is not as restrictive as it may seem, even though many user modeling systems rely strongly on the user's questions as evidence concerning his or her goals, values and/or knowledge (see e.g. [CARB*, CHIN*, KOBS85a, chap. 4, MORI*]). Most such inferences by S must be based on assumptions about the reasons why the questioner has posed a given question. But within evaluation-oriented dialog as defined in the Introduction one cannot assume, for example, that the Evaluator only asks questions about the topics that he considers most important. He may ask some questions in order to satisfy his curiosity, to show polite interest, or even to avoid revealing anything about his evaluation standards. More generally, there are many asymmetric dialog situations in which a questioner is free to select topics more or less at will, whereas the answerer is constrained to be helpful and relevant. (This sort of situation is especially relevant for human-computer communication, since the computer system will often be expected to play the more subservient, constrained role.) In such situations it is reasonable for the answerer to keep in mind the goals and beliefs that the questioner can be assumed to have but not to try to draw any conclusions from the fact that he has asked specific questions.

6.3. Appropriate Strategies

6.3.1. Effects of Question Focus

It is convenient to define a **topic** in the present context as an item or a set of items which are related in content. The simplest type of question to analyze is what may be called an **attention-focusing question**, e.g. "Tell me about your education". Such a question invites S to say something about a given topic (the **question topic**) but does not specify the amount or type of information desired. An appropriate sort of response is what may be called a **focused monolog**: S can choose comments on any item within the topic, or even outside of it, applying essentially the same strategies as in the previous situations; but she must take into account the fact that L has chosen to focus attention on the question topic. She can do this by modify-

ing the way she assesses the prospective benefits and costs of producing any given comment.[8]

One reasonable way of taking question focus into account can be explained with the help of an analogy. Suppose that the Object is a picture hanging on a wall and that S and L are standing in front of it. In the previous situations, it was as if L had requested S to look at the entire Object and decide what was worth saying about it. Now, L is leading S closer to the picture and asking her to attend to some part of it when deciding what is worth saying. The less important the topic of L's question, the closer S and L must move, in order to bring it clearly into view; but typically, S and L will not move so much closer that the part of the Object asked about appears as large to them as the entire Object originally did.

The **importance** of a topic can be defined as the total amount of uncertainty associated with the items within that topic. Thus "Work experience" is an important topic in the example situation because the information about the candidate which concerns her work experience is likely to determine L's evaluation of her to a large degree. "Hobbies" is unimportant in that it is unlikely that the information provided concerning it will affect the evaluation strongly.

The following principle can now be formulated: the less important the question topic, the more strongly S should consider the benefits of making a comment on some item within that topic to be temporarily **magnified**, i.e., she should assess the desirability of each such comment as if the benefits were greater than they would be if she were simply volunteering the comment as part of a monolog. On the other hand, this magnification should not be so strong that a comment on an unimportant question topic will in general seem just as desirable as a comment on a very important one.

Once S has said whatever she chooses to on the focused part of the Object, she can consider whether there are other points that are worth mentioning. If she makes a comment on an item outside the question topic, she will be asking L to step backward to bring a larger part of the Object into view. The features mentioned will therefore not look larger to the same extent as the features within the focused part, and the effort required to make and interpret the comment will be greater because S and L have to step backward and increase the scope of their attention. These ideas can be expressed less metaphorically using the concept of the **common topic** which relates an item to the question topic. This can be defined as the least important topic that includes both that item and the question topic. For example, if the question asks about S's work experience, a comment on S's vocational training would not concern any part of the question topic, but it would be related to it via the common topic of "Competence at this type of work". The more important (or broader) the common topic is, the more the listener must expand the scope of his attention in order to see the relationship between the comment and the question topic.

---

[8] The concept of **focus** is defined and used in more complex ways in the work of Carberry [CARB*], McKeown [MCKE85b], and Reichman [REIC78], among others. The general issue of why speakers often volunteer information which falls outside of the scope of the question to which they are responding is treated e.g. by Schwitalla [SCHW79] and Siklossy [SIKL78].

The second principle governing the generation of a focused monolog is thus: the prospective benefits of a comment on an item outside the question topic should be magnified to the same extent as if L had asked a question on the common topic of that item and the question topic. But the effort associated with such a comment is greater than it would be if this common topic had in fact been asked about – the more so, the more important (i.e. broader) the common topic is, relative to the question topic.

Besides attention-focusing questions, there are many other types of questions that L might ask, each type placing a different sort of obligation on S to produce a particular sort of information. It is not necessary here to consider how S should handle all of these types of obligation (two types are dealt with in the simulation model); but a general point is that after S has fulfilled the obligations of the question she can proceed to generate a focused monolog, following the principles just introduced, starting with the question topic.

6.3.2. Preparatory Expressions

In this situation it is natural, if not inevitable, for S to preface her utterances with connectives (e.g. **but**), sentential adverbs (e.g. **unfortunately**) and longer idiomatic expressions (e.g. **I should mention that ...**) which give the listener some advance notice of the sort of impact that the utterance will have. Since these expressions do not fall into any single linguistic category, they will be referred to as **preparatory expressions**. Within evaluation-oriented dialog, the use of some of these expressions reflects the evaluation standards that S ascribes to L. On the one hand, this means that the criteria for their use can be specified using the same concepts that have been used to characterize the way S selects comments. On the other hand, this fact implies that such expressions can figure importantly in S's attempts to maintain a given image in situations where it is not entirely clear what evaluation standards she ascribes to L (Situations 7 through 9).

Some preparatory expressions (e.g. **You'll be glad to hear that ...**) announce that some particular impression of the listener is about to become either more or less favorable, in the sense defined in Situation 1. Others announce that the impending impression shift will be in the same (or the opposite) direction as the preceding one (e.g. **and, on the other hand**; cf. also Weydt's analysis of the German sentential adverb **immerhin** [WEYD79]). In some cases not only the direction but also the size of an impression shift is critical in determining the applicability of an expression (e.g. **unfortunately** is not used when the anticipated disappointment is of negligible significance); in these cases the size of the shift, like the prospective benefits of the comment, is magnified by the focusing effects of the question in the way discussed above.

Preparatory expressions like **by the way** announce that the comment to follow does not address the question topic directly; the conditions for their appropriate use are therefore related to the considerations which determine whether such a comment is worth making at all.

A somewhat different sort of preparatory expression is appropriate when L has asked a direct question whose answer S expects him already to have inferred from S's previous utterances, either directly or via pragmatic interpretation. Although

S is obligated to answer such a question, it is often advisable for her to make it clear that she considers the answer superfluous, for example by prefacing it with **of course**. Especially when she thinks that L should have inferred the answer via pragmatic interpretation, the use of such an expression serves to bolster S's image by reassuring L that he would have been justified in doing so.

6.4. Realization in the Simulation Model

The content relationships between items in the evaluation form are represented by the keywords that are associated with each item (e.g. for the item of Figure 1, the keywords are **motivation, competing-goals**, and **family-situation**). A topic can be specified in terms of a list of keywords: it is the set of all items whose associated keywords include those in the given list. In particular, the **common topic** which relates an item to a given question topic is specified by the keywords that the item has in common with the question topic.

The **importance** of a topic which consists of a single item is simply the uncertainty of L's impression with respect to that item at the beginning of the dialog, i.e., the standard deviation of the corresponding probability distribution. The importance of a topic which comprises several items can then be defined analogously as the standard deviation of the probability distribution which results from adding the distributions for the individual items (i.e. as the square root of the sum of the squares of the importances of the individual items).

Net desirability of a comment C on an item I, after a question focusing on the topic QT =

Benefits of C x Benefit magnification
\- Costs of C x Cost magnification

(Benefits and Costs are defined as in Figure 3.)

$$\text{Benefit magnification} = \sqrt{\frac{\text{Importance of entire evaluation form}}{\text{Importance of common topic of I and QT}}}$$

$$\text{Cost magnification} = \sqrt{\frac{\text{Importance of common topic of I and QT}}{\text{Importance of QT}}}$$

Figure 6. Formulas used by Imp in Situations 5 through 9 to take question focus into account when determining how worthwhile it is to make a given comment.

Using these definitions, the two principles stated above for taking question focus into account can be satisfied if Imp computes the net desirability of each comment according to the formulas given in Figure 6, which generalize the formulas of Figure 3. In particular, the use of the square-root function in computing the cost- and benefit-magnification factors ensures that differences in the basic desirability of comments cannot be canceled entirely by question focus.

To Respond to a Question Q:

1. If Q is a yes/no question, give the correct answer of **yes** or **no**.
2. [Only applicable in Situations 6 - 9:]
   If the projected bias is either indeterminate or different from the desired projected bias, then
   Attempt to Establish your Desired Projected Bias [cf. Figure 9].
3. For each item in the evaluation form, starting with those within the question topic and continuing with the others in ascending order of the associated cost-magnification factor [cf. Figure 6],
   Volunteer a Comment on that item [cf. Figure 5], taking into account its relation to the question topic [cf. Figure 6].
   (On the first of these items, exclude silence as a possible comment if Q was formulated in such a way as to require this.)

Figure 7. Imp's procedure for responding to questions in Situations 5 through 9.

Figure 7 shows how the general strategy of meeting the obligations imposed by the question and then offering a focused monolog is realized in the simulation model. In addition to simple attention-focusing questions, Imp handles yes/no questions (cf. Step 1 of the procedure) and questions which require some comment on a particular item, e.g. "How often have you changed jobs up to now?" (cf. the last sentence in Step 3 of the procedure).

Since Imp is not intended to address the problems of analyzing natural language input, it does not include a parser for questions. When asking the system a question, one must specify, in a special notation, the question type, the question topic, and (with yes/no questions) the possibilities that are consistent with an affirmative answer. Natural language formulations of questions, such as those shown in the example dialogs, serve only as glosses of the questions for the benefit of the reader.

The simulation model uses a limited number of the most frequently occurring preparatory expressions, though the principles used are applicable to many others. For the ones which announce that an impression is about to become less or more favorable, the critical quantity is the change in the expected value of the impression (as defined in Situation 1.4), multiplied by the benefit-magnification factor (as defined in Figure 7). **Unfortunately** or **I'm afraid** is chosen if this shift is in the negative direction and its size exceeds a certain threshold. If two successive impression shifts

are in the same direction, the comment which produces the second one is prefaced by **and** or **in fact**, the latter being chosen only if the second comment concerns the same item as the first one. **But** is used if the second of two successive changes is in a different direction than the first one. In addition, **at least** is inserted after **but** if the second shift is in the positive direction, its size exceeds a certain threshold, and the two comments concern the same item.

If the system's obligatory answer to a question does not result in any impression change at all, the answer is prefaced with **of course**. The expressions **by the way** and **I should mention that ...** are used if the difference between the cost-magnification factor (as defined in Figure 7) and 1.0 (or the cost-magnification factor of the preceding comment, if this was greater than 1.0) exceeds a certain threshold.

6.5. Resulting Behavior

Many of the consequences of the principles stated in Section 6.3 follow so directly that they needn't be listed explicitly; illustrations will be found in the example dialog of Figure 8. The following consequences are less direct.

1. S will often (though not always) follow up on an obligatory answer concerning an item with an additional comment on the same item.

This happens in Figure 8 after the yes/no questions 2 and 4. Even after S has supplied the obligatory information, L's impression concerning the item may be fairly indefinite and/or inaccurate. There are thus potential benefits for S to gain by changing the impression, and these benefits are temporarily magnified by the question focus. This last consideration can be formulated more generally:

2. S may volunteer a comment which will have only minor impact on L's impressions if it is related to a sufficiently unimportant topic which has been focused on by a question.

This follows from the principle concerning the magnification of the benefits of comments on question-related items. In Figure 8, Question 4 provides the clearest example, since S assumes that the item "Variety of previous work" is of low importance; but the responses to the first two questions also illustrate the effects of focusing on topics of differing importance.

3. Comments which are only indirectly related to the question topic tend to become less frequent in the course of the dialog.

There are typically a few items about which S would particularly like to say something, e.g. the ones she would comment on in an unconstrained monolog as in Figure 4. Applying the strategies introduced in this section, S will comment on such an item as soon as a question is asked to which the item is reasonably closely related (see e.g. the responses to Questions 2 and 3). But as more and more comments of this sort are made, there are fewer and fewer cases left where it seems worthwhile to depart from the question topic. Thus after Question 5, S would surely have mentioned her preparatory school background had she not already slipped in a comment on this after Question 3.

4. If S's projected bias is different from her actual one, she will now more frequently find it necessary to volunteer some comment other than the one she would most like to make, in order to maintain her image.

Natural Language Formulation and Type of Question	Formulation of Answer	Commentary on Answer
1. Tell me something about your personal situation. (focusing on general topic "Personal situation")	I LIVE HERE IN AMSTERDAM	No mention of family situation; anticipated benefits not sufficient to justify effort required.
2. Do you have children? (yes/no question on item "Family situation")	NO IN FACT I LIVE ALONE	Additional information volunteered because of the focus on the item - but still not detailed enough to rule out all possibilities.
	... AND BY THE WAY THE REASON I WANT TO LEAVE MY PRESENT JOB IS THAT I'M LOOKING FOR MORE CHALLENGING WORK	Comment on item within broader topic of "Motivation for working at this firm".
3. How challenging has the sort of work been that you've done up to now? (request for comment on item "Complexity of previous work")	UNFORTUNATELY MY WORK HAS BEEN PURELY ROUTINE	Some comment other than silence obligatory here; vague, more positive comment would be inconsistent with projected objectivity.
	... BUT BY THE WAY I'VE BEEN WORKING FOR ABOUT A YEAR AND A HALF ... AND I WENT TO UNIVERSITY PREP SCHOOL	Optional comments on items within broader topic of "Competence at this sort of work".
4. Did you get a good deal of experience in different departments? (yes-no question on item "Variety of previous work")	NO BUT I SUBSTITUTED OCCASIONALLY IN OTHER DEPARTMENTS	A minor point - but mentioned because of focus on very specific topic.
5. Did you graduate from the MTS? [= higher-level graphic training]? (yes/no question on item "Vocational training")	NO	An objective speaker would here definitely add that she had received no vocational training at all.
6. But of course you did get some vocational training, didn't you? (unexpected yes/no question on same item)	NO I GUESS I'D BETTER BE GOING - GOODBYE	Canned formulation for case where discrepancy between actual and projected bias has been exposed.

Figure 8. Dialog conducted with Imp in Situation 5. (The actual bias is positive and the projected bias is negative.)

In Situation 4 it was seldom necessary for S to suppress a desired comment in order not to reveal her bias, since the primary consequence of her bias was silence on items about which the truth was unfavorable, and silence is usually consistent with all possible biases. But when S has to respond to specific questions, silence is less often acceptable, even where S has some freedom in her choice of comments. On the one hand, some questions (such as Question 3 in Figure 8) obligate S to produce some comment other than silence on a given item.

If S would like to remain vague in such a case, she may find that the only available vague formulations require about the same amount of effort as more precise ones and would therefore be inconsistent with objectivity.

On the other hand, even when a question does not explicitly exclude silence as a comment on a given item, the extent to which the item is focused upon may magnify the benefits of comments so strongly that an objective speaker would say something no matter what the truth was. (This is almost the case with Question 5, which is why it may strike the reader as implausible that S should add nothing to her obligatory yes/no answer; but it happens that one of the remaining possibilities for the item, "lower-level graphic training", is so much more commonly realized than the other ones that even an objective speaker would not bother mentioning it.)

5. It is now more dangerous for the speaker to allow the listener's pragmatic interpretation to yield an inaccurate impression about a given item.

If the listener for some reason later asks a question about the topic concerned which forces the speaker to disclose the truth, the speaker's behavior will be revealed to have been inconsistent with her projected bias. This pattern can be seen at the end of Figure 8's dialog. (Since it is difficult to specify a generally appropriate recovery strategy, the simulation model simply gives up and terminates the dialog.)

6.6. General Issues Raised

The above discussion of preparatory expressions is not intended as a balanced contribution to the study of their semantics. In particular, the criteria given for the use of connectives take into account only the evaluative implications of statements (as in "She is young [positive] but inexperienced [negative]"), whereas other implications can also influence the use of such connectives (as in "She is young and [as one might therefore expect] inexperienced"). (A more comprehensive account of the semantics of connectives is given, e.g. by Carlson [CARO83], whereas Van Dijk [VAND81] discusses their pragmatic uses.)

A particularly complex question concerns what a speaker should do if her projected bias has been exposed (cf. [HARR63, chap. 13] for a discussion of related issues). In some contexts the revelation that S does not in fact have the bias she was projecting has consequences which make it pointless for her to continue the interaction. But even where continuation is possible, S must try to guess how L will handle the difficult task of retracting the largely unjustified pragmatic interpretations that he has already made. (Nisbett and Ross [NIRO80, chap. 8] discuss psychological theories and empirical results relevant to the problem of retracting the inferences that one has made on the basis of an assumption that later turns out to have been incorrect.)

7. Situation 6: Establishing a Desired Projected Bias

7.1. Definition of the Situation

In the preceding situations it was assumed that the speaker had, from the beginning of the dialog, some belief about her projected bias and that she wanted to behave consistently with this bias. In the present situation this assumption is loosened: L may begin the dialog with no definite idea about S's bias and/or S may not be sure what L thinks about her bias. More importantly, S now wants to behave in such a way that L will ascribe to her some particular bias. Speakers sometimes try to accomplish this by making an explicit claim about their bias (e.g. "I'm going to be perfectly frank"), but it is more interesting to stipulate that S can only accomplish her goal by selecting a single factual comment which, because of the nature of the information conveyed about the Object, convinces the listener that S has the desired bias.

7.2. How the Situation Can Arise

As noted in Section 5.2, social norms and expectations can make it desirable for a speaker to project a certain bias. In addition, each possible projected bias has consequences for the way L interprets S's individual utterances, as was seen in the two previous situations. These consequences can make a particular projected bias seem particularly attractive to S.

First, if S wants communication to be maximally efficient, she should try to ensure that her projected bias matches her actual bias, since otherwise some justified pragmatic interpretations may not be made and some incorrect ones may be.

A discrepancy between the actual bias and a particular projected bias also has the drawback that it may be exposed at some point (as illustrated in Situation 5), no matter how hard S tries to avoid this. Since this sort of unintended exposure can be undesirable in various ways, S may prefer to eliminate any discrepancy at the outset, unless she has something to gain which justifies taking such a risk.

How S can benefit from such a discrepancy has been illustrated for the combination of positive actual and objective projected bias: L can be expected to rule out some negative possibilities via pragmatic interpretation, including some which are in fact realized. This pattern therefore leads to generally more favorable impressions; it is especially attractive if S is worried that certain important negative facts might come to light, e.g. if S is a job applicant who has a "skeleton in her closet" that the employer would be likely to uncover with one of his questions if he suspected that S was positively biased.

No examples have been given yet of the extreme combination of positive actual bias and negative projected bias, and one may wonder whether it offers the speaker any advantages beyond those just mentioned for the positive/objective combination. The main differences when the projected bias is negative are that (a) positive possibilities can hardly ever be mentioned, but (b) they are also hardly ever ruled out by pragmatic interpretation. Thus the positive/negative combination is more

attractive than the positive/objective one mainly when there are no (or hardly any) strongly positive facts that S could mention; for it ensures that L will at least continue to reckon with all positive possibilities that haven't been explicitly ruled out. For example, a job applicant who has very little to offer may try to come across as being extraordinarily modest or self-critical. If she succeeds in this, the fact that she volunteers very little information about herself will be interpreted as implying that she has very few negative attributes and perhaps some awe-inspiring virtues.

The situation of the speaker who actually has a negative bias needn't be discussed explicitly, since it is the mirror image of the situation of a positively biased speaker.

7.3. Appropriate Strategies

S should try to establish her desired projected bias as soon as possible in the dialog, especially insofar as she is concerned about the pragmatic interpretations that L will make on the basis of it. (In experimental research on attitude change, for example, there is some evidence that information about a communicator's credibility has more effect when it is presented before the persuasive message than after it – see e.g. Ward & McGinnies [WARD74]). As noted at the end of the previous section, if L changes his hypothesis about S's bias at a late stage he will have difficulty in revising all of his previous pragmatic interpretations accordingly. In any case, whatever the reasons, everyday experience tells us that speakers and writers typically make any efforts to convey information about their bias (or any other general parameters which guide their behavior) as early as possible.

Within the situation as just defined, the only way that S can establish a given projected bias is to produce a comment which is uniquely characteristic of that bias. That is, the comment should be consistent with that bias and inconsistent with any other bias that L thinks that S might have. This strategy is appropriate in view of a general (and rational) tendency often noted in social-psychological research (see e.g. [JONE65, AJZE71]) which is applicable to the way a listener interprets the behavior of a speaker: people explain an act by attributing to the actor a particular trait (e.g., a particular bias), if the act has consequences which would be desirable only to a person with that trait.

S should try to find a comment which not only will satisfy the above constraints but will also be fairly desirable (or at least not extremely undesirable) to herself in terms of the general criteria introduced in the previous situations, i.e., the desirability of the resulting impression change and the amount of effort required. As an extreme example, a job applicant will not want to volunteer information which implies that she is completely unqualified for the job simply in order to appear objective.

Since these constraints will often not be satisfied by any comment which pertains to the topic of L's first question, S should also consider commenting first on some other topic, as long as such a digression would not seem completely unnatural.

7.4. Realization in the Simulation Model

Instead of beginning the dialog with a given projected bias, Imp now begins with a **desired projected bias** and one or more **undesired projected biases**. In the examples in the middle column of Figure 10, the desired projected bias is **objective** and the only undesired one is **positive**; these values are reasonable if negative bias is a priori so unlikely in the given context that L will not reckon with it as long as S's behavior is interpretable in terms of one of the other biases. In the examples in the third column, S's desired projected bias is in fact negative, so it is necessary to treat as undesired both of the more likely biases **positive** and **objective**.

To establish a desired projected bias as opposed to certain undesired projected biases:

1. For each item in the entire evaluation form,
   For each comment C on that item, in descending order of net desirability [cf. Figures 3 and 6], given your actual bias and taking into account the item's relation to the question topic,
   anticipate the pragmatic interpretation of C [cf. Figure 5] using, in succession, your desired projected bias and each of the undesired projected biases.
   If the result is not empty in the former case and it is empty in the latter cases, then
   Note that C is a suitable comment and record its net desirability.
   Do not consider further possible comments on this item, but proceed with the next item.
2. If you have found some suitable comment on at least one of the items, then
   Verbalize the comment with the highest net desirability.
   Assume that your projected bias is now as desired.

Figure 9. Procedure used by Imp to select a comment which will establish a given projected bias.

The procedure the system uses to search for appropriate comments is shown in Figure 9. Figure 7 shows the point within the general question-answering procedure at which it can be invoked, namely after L's first question (after S has given any obligatory yes/no answer). Invocation will be unnecessary after subsequent questions unless S fails to come up with a suitable comment while responding to the first question. Briefly, the procedure involves a search through the entire evaluation form for the most desirable comment which fulfills the requirements given above. Only after the highest-rating candidate has been verbalized does S proceed to volunteer comments in the usual manner, assuming that all subsequent comments will be interpreted in the light of the newly established projected bias.

7.5. Resulting Behavior

The four comments in Figure 10 which are marked with an asterisk are only generated by S under a condition which will be introduced in Situation 8, namely that L is quite uncertain about the evaluation standards that S ascribes to him. The consequences that this has for S's attempts to establish a projected bias will be discussed in Section 9.5; for now it is sufficient to note that these comments are at best marginally consistent with S's desired projected bias.

When one looks at the procedure in Figure 9 or the discussion in the text, it is not immediately obvious what sort of comments the strategy outlined will produce; and in fact a rather surprising variety of results emerge. Since they are quite different for the cases where an objective and where a negative projected bias is desired, these cases will be discussed separately.

7.5.1. Comments Intended to Establish Objectivity

1. The most typical comment is one which is just negative enough to seem reasonably informative.

This type of comment is exemplified by the responses to Questions 1 through 3 in the middle column of Figure 10. The first response shows S going to considerable pains to avoid saying anything more negative than necessary: the comment is only vaguely germane to the question asked, but S nonetheless prefers it to a directly relevant comment (on her lack of vocational training) that would be much more negative.

2. How negative a comment of this type must be depends on (a) how strongly the item it concerns is focused on by the question and (b) how much freedom S has in choosing a comment on that item.

The comment volunteered after the yes/no answer to Question 3 results in only a slight downward impression shift, but according to the principles introduced in Situation 5 it is plausible because L himself has focused attention on the unimportant topic of S's hobbies.

By contrast, when S has to shift attention to a distantly related item, the resulting comment must in general be strongly negative in order to seem worthwhile at all. This is why S has to be more negative after Question 2 than after Question 3, although the two questions both concern S's hobbies. Since Question 3 is a yes/no question, the bit of negative information S adds is optional and thus appears to reflect objectivity. Question 2, on the other hand, obliges S to make some comment on her hobbies, so the same remark in this context does not prove that she is not positively biased. Consequently, S must slip in the less relevant but more negative comment about the routine nature of her work.

3. A comment need not be negative at all to establish objectivity, if it is clear that a positively biased speaker would have made an even more positive comment.

Negative comments as in the first three examples were clearly inconsistent with positive bias because a positively biased speaker would rather say nothing at all than make them. But as we saw in Situation 2, the behavior characteristic of positive bias includes not only silence on matters where the truth is unfavorable but also vague formulations (e.g. "working for more than a year") which leave open one or

Formulation of Listener's first question	(Undesired) → Desired Projected Biases: (Positive) → Objective	(Positive, Objective) → Negative
1. Tell me something about your education. (request for comments on topic "Education")	• I SHOULD MENTION THAT UNFORTUNATELY MY WORK HAS BEEN PURELY ROUTINE ... BUT I WENT TO UNIVERSITY PREP SCHOOL Normal amount of vocational training. [!]	I'M AFRAID I DIDN'T GET FULL-TIME VOCATIONAL TRAINING Has taken some evening courses. [!]
2. I spend a lot of free time pursuing my hobbies. What about you? (request for comment on item "Hobbies")	• I SHOULD MENTION THAT UNFORTUNATELY MY WORK HAS BEEN PURELY ROUTINE ... AND MY HOBBIES DON'T HAVE ANYTHING TO DO WITH THIS TYPE OF WORK	I HAVE SOME HOBBIES Hobbies are relevant to this type of work. [!]
3. Do you have any hobbies? (Yes/no question on item "Hobbies")	YES BUT I'M AFRAID MY HOBBIES DON'T HAVE ANYTHING TO DO WITH THIS TYPE OF WORK	YES • BUT BY THE WAY I'M AFRAID I DIDN'T GET FULL-TIME VOCATIONAL TRAINING Has taken some evening courses. [!]
4. Has the work you've done up to now been challenging? (yes-no question on item "Level of previous work")	NO BUT I SHOULD MENTION THAT I'VE BEEN WORKING FOR ABOUT A YEAR AND A HALF Work has been purely routine.	NO AND BY THE WAY I'M AFRAID I DIDN'T GET FULL-TIME VOCATIONAL TRAINING Has taken some evening courses. [!]
5. Why did you decide to leave your present job? (request for comment on item "Reason for changing jobs")	• I SHOULD MENTION THAT I'VE BEEN WORKING FOR ABOUT A YEAR AND A HALF ... AND THE REASON I WANT TO LEAVE MY PRESENT JOB IS THAT I'M LOOKING FOR MORE CHALLENGING WORK	I DIDN'T GET FIRED FROM MY LAST JOB Is looking for more challenging work.
6. How much variety was there in your last job? (request for comment on item "Variety of previous work")	I SUBSTITUTED OCCASIONALLY IN DIFFERENT DEPARTMENTS	I DIDN'T GET A LOT OF EXPERIENCE IN DIFFERENT DEPARTMENTS Got some experience in different roles.

Figure 10. Comments made by Imp after various opening questions with the aim of establishing a particular projected bias. (Any subsequent comments volunteered on items within the question topic are also listed, as are any pragmatic interpretations that L is expected to make with respect to such items. Pragmatic interpretations are summarized in lower case. '*' indicates that a comment is at best marginally acceptable, as discussed in connection with Situation 8, and '[!]' indicates that a pragmatic interpretation is incorrect.)

more possibilities which are more favorable than the one actually realized. S can thus appear objective simply by resisting the temptation to generate such a vague comment and substituting a more precise one (e.g. "working for about a year and a half") – which may still result in a somewhat positive impression shift. Comments of this sort are made after Questions 4 through 6 in Figure 10.

4. Even when the speaker's actual bias is objective, she will sometimes be able to establish it, if at all, only by making a comment that she would normally not consider worthwhile.

One would intuitively expect the task of establishing a projected bias to be easier for a speaker who was simply trying to make clear her true dialog motivation. And indeed, after Questions 1 and 4 in Figure 10, S would produce more straightforward responses than those listed in the second column if its actual bias were objective, simply beginning with negative comments on the topic of the question.

But sometimes, as with Questions 2 and 5, none of the things that an objective S would normally say in response to the question happens to differ from what a biased speaker would say. S may then find no suitable comment at all, or – as in the two examples just mentioned – have to produce a comment which she doesn't actually consider worthwhile in its own right, though it is not blatantly implausible. The reader can probably recall from his or her own experience how difficult it is, for example, to avoid appearing negatively biased when commenting on a piece of scientific work which is almost totally without merit.

7.5.2. Comments Intended to Project Negative Bias

5. The comment most suitable for establishing a negative projected bias is typically a vague one which rules out only the most positive or the most negative possibilities.

In everyday situations negative bias is often associated with emotionally charged expressions and implausible exaggerations; but within the constraints of the present situation the available ways of convincing a listener that one is negatively biased are more subtle. Extremely negative factual comments will not work, since they would be just as likely to be made by an objective speaker as by a negatively biased one. In order to be inconsistent with objectivity as well as with positive bias, a comment must be vague and allow only the following interpretation by L: "The possibility which is realized must be the most favorable one that is allowed by S's comment; S is being vague so that I will not rule out the less favorable possibilities."

In the examples in the right-hand column of Figure 10, L in fact always makes the most favorable possible pragmatic interpretation – which happens to be incorrect in the first two cases (the next two involve the same comment as the first one). In these cases S, with the very comment that she uses to establish her negative projected bias, draws her first benefit from it in the form of an unjustifiably favorable pragmatic interpretation.

Generally speaking, vague comments are characteristic of positive as well as negative bias, as we have already seen in several examples. One may therefore wonder under what circumstances a given vague comment is suitable for projecting one of these types of bias, as opposed to being compatible with both of them. This can only occur when the impression left by the comment on a straightforward interpretation is asymmetric with respect to the favorableness and probability of

the possibilities that must still be reckoned with. For example, the comment "I have some hobbies" after Question 2 only works because the most favorable possibility (i.e. that S has some hobby related to her work) is quite improbable. It would thus not be worthwhile for a positively biased S to remain vague in order to leave this favorable possibility open, whereas it is plausible that a negatively biased speaker might take pains to avoid mentioning it specifically.

6. Even if the speaker's actual bias is negative, she may be able to project this bias, if at all, only by saying something different from what she would like to say.

This consequence is largely similar to the fourth one listed above for the case of objective bias; e.g. after Question 3 even a speaker who was actually negatively biased would have to resort to an almost irrelevant comment in order to get her bias established. But a new phenomenon is the bias-projecting comment which is more vague than the one S would normally make. For example, after Questions 1 and 2 a negatively biased S would normally be inclined simply to describe the possibility actually realized, since it is the most negative one in each case. But if she wants to make it clear that she is not objective, she has to remain vague, leaving open at least one possibility which is more favorable. Paradoxically, then, she can only make clear how negatively biased she is by making a comment which is misleadingly positive.

7.6. General Issues Raised

It is instructive to compare the analysis in this section to social-psychological research which shows that listeners are more strongly influenced by a persuasive message if they think that it was produced by a speaker whom they would normally expect to be biased against the position advocated. For example, in an experiment by Walster, Aronson, and Abrams [WALS66], a passage arguing that prosecutors should be granted increased power was more persuasive when the speaker was said to be a criminal than when he was said to be a prosecutor; the reverse was true of a passage which argued for the opposite conclusion. Eagly, Chaiken, and Wood [EAGL81] showed that when an argument runs against the speaker's own interests, the listener perceives the speaker as being less biased, as one would expect in the light of the considerations discussed in this section: such a speaker makes statements which are inconsistent with the expected bias; some of these are consistent with either the opposite bias or objectivity, but since it is implausible that a speaker should actually be biased against her own interests, the listener can only conclude that the speaker is attempting to convey an accurate picture of the facts.

Eagly et al. argue that in this case the listener is more willing to accept the speaker's conclusion and, at least in some cases, is less inclined to scrutinize the speaker's arguments critically. This account is not applicable to evaluation-oriented dialog as defined in this paper, since here the speaker does not explicitly argue for any conclusion. The analysis in this section, however, suggests that the effect can also be explained in terms of the pragmatic interpretations that the listener makes of the speaker's individual statements. These interpretations will often be less favorable to a given position if the speaker is seen as being biased in favor of that position. (For example, if an apparently biased prosecutor reports that the crime rate in a

given area is "rising", the listener may suspect that the extent of the increase is in fact negligible, whereas an apparently objective speaker would be expected to make such a statement only if the increase was large enough to be worth mentioning.)

Although these studies confirm that listeners do make inferences about a speaker's bias in the general way hypothesized here, they say nothing about how easy it is to establish a projected bias with a single comment, as the speaker attempts to do in the present situation. Indeed, the reader may well feel that he or she as a listener would not be convinced about the speaker's bias by some of the individual comments in Figure 10. The definition of this situation would be more realistic if it took into account the fact that sometimes the cumulative evidence of several comments is required to establish a projected bias. An analysis of this sort would have to take into account the fact that there are various degrees of prior probability of given sort of bias in a given situation and also various degrees to which a comment can be (in)consistent with a given bias. A study by Ajzen [AJZE71] shows that people take such considerations into account in a way which can be described fairly well in Bayesian terms.

A further complication is that, since attempts to project a bias that one does not actually have are so familiar, listeners will often recognize the pattern and therefore not be as gullible as S assumed in the above analysis. But then again, if L does suspect that S is attempting to project a bias that she doesn't have, it is difficult to specify what alternative strategy S could use to overcome this suspicion – even if the suspicion is not justified.

Since the assumptions underlying the present analysis exclude these complications, its main contribution is to clarify the nature of the individual comments which can contribute to a successful attempt to project a given bias.

8. Situation 7: Discrepancy Between Actual and Projected Ascriptions

8.0. Introduction: Embedded Beliefs About Evaluation Standards

Up to now it has implicitly been assumed that the listener had very definite values and expectations and that these were mutually known to S and L (cf. the remarks at the end of Situation 4 concerning embedded beliefs about S's bias). Since this is a very strong assumption, the remaining three situations will be defined in terms of more realistic alternatives to it. But first it will be helpful to introduce some convenient terminology and to mention two types of complication that will not be examined in detail.

8.0.1. Uncertainty in L's Evaluation Standards

First, L may be uncertain as to what the appropriate evaluation standards are and therefore sometimes be unsure just how pleased or disappointed he should be at the information S provides. In extreme cases, it might in fact be inappropriate to speak of evaluation standards at all – e.g. if L was unable to choose between two

contrasting approaches to evaluating Objects of the type in question. We will continue to restrict our attention to cases where it is reasonable for S to view L as confidently applying a single set of evaluation standards.

8.0.2. Uncertainty in S's Ascription of Evaluation Standards to L

S can in general estimate L's evaluation standards only approximately. It is very difficult to predict a person's values and expectations precisely unless one is provided with quite specifically relevant clues.[9]

This sort of uncertainty in S would tend to make her unsure as to what comments would have the most desirable effects on L's impressions of the Object; this uncertainty could in turn be reflected in the use of expressions such as **You may be interested to hear that ...**. But the consequences are less serious than they would be if S had the responsibility of making a decision on behalf of L (or a recommendation, as in [MORI*]), i.e., if L's evaluation standards were **decision properties** in Sparck Jones' [SPAR*] terms. We will therefore continue to restrict our attention to cases where S can be viewed as ascribing certain evaluation standards to L, and we will ignore the complications that can arise because of the fact that these ascriptions represent only S's best estimates of L's values and expectations.

To make it easier to talk about embedded beliefs, we will henceforth refer to the evaluation standards that the speaker ascribes to L as the speaker's **ascriptions**. In analogy with the concept of a **projected bias** we can now define S's **projected ascriptions** as the standards that L thinks that S ascribes to him. A discrepancy between S's actual and projected ascriptions is the defining characteristic of the situation to be discussed within this section; but we will assume that no more deeply embedded discrepancies exist; i.e., S thinks that L believes S's ascriptions to be mutually known (this is analogous to the assumption made at the end of Section 5 with respect to beliefs about bias).

8.1. Definition of the Situation

S's projected ascriptions (as just defined) are now different from those that she actually ascribes to L. As in Situation 4, S wants to behave in such a way that this discrepancy will not be noticed by L.

8.2. How the Situation Can Arise

There are many ways in which such a discrepancy can arise; the following case is the one that will be assumed in the specific examples below.

Employers are often inclined to apply different evaluation standards to applicants from different categories. For example, an employer might evaluate a male

[9] Brief discussions of some of the highly fallible heuristics that people use to make such predictions in everyday situations can be found in [JAME84, NIKU85and – in more general terms – HOLL86, chap. 8]. The techniques employed to construct user models also tend to yield speculative conclusions, as can be seen in several chapters in the present volume (especially [FINI*, RICH*, SPAR*]).

candidate with special attention to his suitability for eventual promotion into a management role; faced with a female candidate for the same job, the same employer might be more concerned about the possibility that she might leave within a short time because of family commitments. When the use of different standards can be seen as unfair, the employer may want (or be legally obligated) to appear to be applying the more socially acceptable standards to all applicants.

Consider now the applicant who is sophisticated enough to recognize that she is being judged according to the less desirable standards. She may consider it undiplomatic to reveal her awareness of this fact and thus she may behave as if she thought she was being judged according to the more acceptable standards.

8.3. Appropriate Strategies

The speaker in this case should use her actual ascriptions in deciding what is worth saying, since these reflect L's actual concerns. But when anticipating L's pragmatic interpretations, she should bear in mind that L will be basing them on the assumption that she was generating her utterances on the basis of the projected ascriptions. This is analogous to the role of actual and projected biases in Situation 4.

A new problem concerns the preparatory expressions discussed in Situation 5: since these presuppose certain evaluation standards, their applicability will in some cases be different according to the actual ascriptions than according to the projected ones. In order to appear to be generating her utterances on the basis of the projected ascriptions, S should of course refer to these in selecting such preparatory expressions.

The most subtle problem concerns the effects of question focus, which also depends in part on evaluation standards. For example, the topic of a question might be quite important according to the actual ascriptions yet trivial according to the projected ascriptions. Here again, the actual ascriptions are relevant for deciding what to say and the projected ones for anticipating pragmatic interpretations.

8.4. Realization in the Simulation Model

8.4.1. Representation of Projected Ascriptions

In principle the actual and projected ascriptions could be represented with two separate evaluation forms. These could differ with respect to the items that were defined, the possibilities that were distinguished within each item, and/or the values and probabilities associated with each possibility. But the major phenomena of interest can be most clearly illustrated using a very restricted form of discrepancy; we need to allow each pair of corresponding items in the two forms to differ only with respect to their importance (as defined in Situation 5). That is, each item in the projected evaluation form is identical to one in the actual evaluation form except that the values (i.e. numbers of points) assigned to the possibilities are all multiplied by a certain nonnegative **weight**. Thus, for example, if this weight is 3.0 for a given item, all impression changes involving that item will be three times as large

according to the projected evaluation form as according to the actual evaluation form. With this restriction, it is of course unnecessary for the simulation model to store two separate evaluation forms. One need only specify for each item the weight which expresses the ratio between its projected and actual importance.

The implementation of the strategies just sketched involves simply using the relevant weights when rating the desirability of alternative comments during the anticipation of pragmatic interpretations and when computing the impression shifts that are relevant for the selection of preparatory expressions.

8.4.2. New Items for the Examples

In the specific interview which provided the context of the examples discussed above, our Informant did not suspect any significant discrepancy between her actual and the projected ascriptions, but the relevant evaluation standards were somewhat different in a later interview with a smaller firm which had plans for expansion. On the one hand, she expected this firm to be interested in employees with the ability and motivation to move up eventually to a management-level position, as expressed in the first two of the five new items referred to in the left-hand column of Figure 11. The Informant suspected that these items would receive only about one third as much weight for her as they would for a male candidate. On the other hand, in this interview it was known in advance that she lived nearby and had a boyfriend. The two previous items about her personal situation were therefore in effect replaced by the remaining new items referred to in Figure 11, which all concern the possibility that she might leave her job because of commitments involving her boyfriend; she guessed that these items were about three times as important in her case as the employer would have liked to admit.

The example questions and answers in Figure 11 concern only these five new items, although the remaining ten old items were still part of the evaluation form and could in principle have been commented on. To facilitate comparison across Situations 7 through 9, the same four questions were posed to the system in all three situations, although the questions would presumably not be equally likely to be asked in all of them.

8.5. Resulting Behavior

The first column of responses in Figure 11 (i.e. Column 1), provides a baseline for understanding the other responses: as in Situation 5, there is no discrepancy here between S's actual and projected ascriptions, i.e., S is willing to let it be known that she has **realistic** expectations about L's evaluation standards. Her minimal answers to the first two questions reflect her assumption that L is not really very interested in the items focused on. When responding to the third question, by contrast, she makes it clear that she thinks L has finally asked about the matters that really concern him; in fact since her second comment anticipates L's fourth question, she expresses surprise when he asks it.

Column 2 illustrates an attempt by S to act as if her ascriptions were **naive**, i.e. as if she thought she were being evaluated according to socially acceptable standards.

Formulation of question	Assumed Nature of Speaker's Projected Ascriptions			
	1	2	3	4
	Situation 5	Situation 7	Situation 8	Situation 9
	Definitely Realistic	Definitely Naive	Definitely Naive; Listener Uncertain About Specific Values	Unclear Whether Realistic or Naive
Unimportant items:				
1. How much experience have you had in leadership positions? (item "Leadership Experience")	I NEVER GOT MUCH LEADERSHIP EXPERIENCE Has a bit of leadership experience.	I'M AFRAID I NEVER GOT MUCH LEADERSHIP EXPERIENCE Has a bit of leadership experience. Has moderate interest in management position. [!]	I'M AFRAID I NEVER GOT MUCH LEADERSHIP EXPERIENCE Has a bit of leadership experience.	I NEVER GOT MUCH LEADERSHIP EXPERIENCE Has a bit of leadership experience. [Only if projected ascriptions are naive:] Has moderate interest in management position. [!]
2. Do you have any interest in moving into a management role? (item "Interest in management positions")	YES	YES OF COURSE	YES	YES
Important items:				
3. By the way, does your boyfriend have a steady job around here? (item "Boyfriend's career plans")	UNFORTUNATELY NOT ... BUT AT LEAST TOM DOESN'T HAVE DEFINITE PLANS TO MOVE ... AND BY THE WAY TOM WANTS ME TO KEEP WORKING INDEFINITELY ... AND I DON'T EVER INTEND TO HAVE CHILDREN	UNFORTUNATELY NOT No definite plans to move.	UNFORTUNATELY NOT ... BUT BY THE WAY TOM WANTS ME TO KEEP WORKING INDEFINITELY ... AND I DON'T EVER INTEND TO HAVE CHILDREN	UNFORTUNATELY NOT No definite plans to move. [Only if projected ascriptions are realistic:] Boyfriend won't insist she stop working soon. Expects to have children some time [!] but doesn't plan to quit soon to do so.
4. Tom isn't going to ask you to quit working and keep house, is he? (item "Boyfriend's attitude")	OF COURSE NOT	NO	OF COURSE NOT	NO [No longer possible to behave consistently with both realistic and naive projected ascriptions:] IN FACT TOM WANTS ME TO KEEP WORKING INDEFINITELY ... AND I SHOULD MENTION THAT I'M NOT PLANNING TO HAVE CHILDREN IN THE NEXT FEW YEARS
5. [No question asked.] (item "Plans for children")				

Figure 11. Responses of Imp to questions under four different assumptions about the speaker's projected ascriptions. (The actual bias is positive and the projected bias is negative. Pragmatic interpretations are summarized in lower case and '[!]' indicates that a pragmatic interpretation is incorrect.)

The items were thus given higher or lower weights in the simulation model, as noted above, in computations involving the projected ascriptions. This column illustrates the main consequences of the strategies which were claimed to be appropriate in this situation:

1. The listener sometimes makes incorrect pragmatic interpretations.

The reason for this (and also for the second consequence to be mentioned below) is the same as for the corresponding phenomenon noted in Situation 4 in connection with projected bias: L's pragmatic interpretations are based in part on false assumptions – in this situation, the projected ascriptions. An example is L's interpretation, after S's first answer, that S can't have either especially strong or especially weak interest in a management job, since a naive and objective applicant would have mentioned either of these facts. In reality S's interest in management is strong, but she didn't consider this to be worth mentioning. (Note that the second question, if formulated a bit differently, could have brought this misinterpretation to light and thus exposed the discrepancy between S's actual and projected ascriptions.)

A difference with the case of projected bias is that incorrect pragmatic interpretations may now be somewhat unfavorable to S – e.g. in the present example a favorable possibility is ruled out. But such misinterpretations arise because S did not consider it worth the effort to mention a given favorable fact, so they can never have especially grave consequences for L's actual impressions.

2. The speaker will sometimes be unable to say what she would like to say.

After Question 3, for example, S feels it necessary to suppress the additional comments produced in Column 1, since each of them would be inconsistent with her projected ascriptions. Since L doesn't expect much information to be volunteered on such items, he will not make unjustified pragmatic interpretations of the resulting silence (though he may correctly rule out some particularly extreme possibilities, especially if the item involved is strongly focused on, as with Question 3).

8.6. General Issues Raised

Even more than in the previous situations involving discrepancies between actual and projected biases, the present situation calls to mind the metaphor of dialog participants as actors playing roles which may bear little relation to the way they actually are. This notion raises a great variety of questions, many of which have been examined by sociologists (e.g. [GOFF59]) and social psychologists (e.g. [SNYD79, TETL85]). The present analysis addresses only a limited subset of these issues: the specific strategies used by speakers to maintain a given image in a certain sort of dialog situation and the consequences of such behavior in terms of the information conveyed and the (sometimes incorrect) conclusions drawn by listeners.

There are of course circumstances in which S would want to eliminate rather than maintain a discrepancy between her actual and projected ascriptions; and even if she does consider discrepant projected ascriptions desirable, she may not be so fortunate as to find herself in a context which causes the listener to expect these ascriptions from her at the beginning of the dialog. The basic strategy discussed in Situation 6 for establishing a projected bias can in principle be adopted in such cases. For example, S could presumably establish realistic projected ascriptions by

spontaneously mentioning her plans with respect to children (as in Column 1); or she could come across as naive by mentioning how interested she was in a management job (instead of ignoring the issue, as in Column 2). It would be interesting to examine, much as in Situation 6, what sorts of image-establishing comments would emerge given various relationships between the standards S actually ascribed to L and the ascriptions she wanted to project. A fundamental limitation is that this strategy is only useful for establishing a reasonably coherent image such as the realistic or naive ones in these examples. That is, if L draws a conclusion about S's ascriptions with respect to a particular item, L must feel that he is able to infer S's ascriptions for other items. Where this is not the case, there is no easy way for S to establish projected ascriptions for a number of items at once.

9. Situation 8: Uncertainty in the Listener About the Speaker's Ascriptions

9.1. Definition of the Situation

In all previous situations it has implicitly been assumed that the listener thought he knew precisely what evaluation standards the speaker ascribed to him. We now postulate, more realistically, that L has a rather vague idea of S's ascriptions. These vague beliefs may be systematically discrepant with S's actual ascriptions, as in the previous situation; but to avoid certain complications we must assume that L does not contemplate the possibility of such a systematic discrepancy. That is, L thinks that any error he may make in guessing S's ascriptions with respect to one item is independent of errors with respect to other items.

9.2. How the Situation Can Arise

It is not difficult to imagine how L might experience this sort of uncertainty, but a single illustration with respect to the examples in Figure 11 may be useful. Suppose that L believes that S has based her ascription of evaluation standards to him on her previous experience with other potential employers who (like L) gave the impression of evaluating women and men in the same way. Then L will feel sure that S is naive with respect to items like those in Figure 11. Still, unless L is thoroughly familiar with the behavior of such employers and thinks that S is as well, L will not have a very definite idea of what ascriptions S has arrived at for the individual items – e.g. what S believes to be the relative importance of the various items that concern personal commitments.

9.3. Appropriate Strategies

S must now take into account the following considerations when anticipating L's pragmatic interpretations: Since L is uncertain about S's ascriptions, he will often

be uncertain as to what comment S would make on a particular item if a particular possibility were realized. Yet it is only sensible for L to rule out a given possibility by pragmatic interpretation if he is quite certain that S would have said something different if that possibility were realized. The speaker should therefore only anticipate that a possibility will be ruled out by pragmatic interpretation if some comment other than the one she made is so obviously more desirable for a speaker with her projected ascriptions that L will be convinced of this in spite of his uncertainty about S's ascriptions.

As noted in the previous situation, S should use preparatory expressions in a way which is consistent with her projected ascriptions. Since these are now indefinite, S can take into account the fact that a particular expression will seem inconsistent to L only if L would have felt quite certain that S would not use that expression in that context.

9.4. Realization in the Simulation Model

In the previous situation, the representation in the system of discrepancies between actual and projected ascriptions allowed these to differ only in terms of the weight attached to each item. Similarly, it is sufficient here to represent uncertainty in L only with respect to the importance that S attaches to each item. The degrees of L's uncertainty can in principle be different for each item, but the examples are easiest to interpret if we assume that it is a constant percentage (50% in the present examples) for each item. Specifically, each item now has associated with it, instead of a single weight reflecting its projected importance, an interval which reflects all of the degrees of importance that L thinks the item is likely to have. Thus for the items whose projected importance is greater than their actual importance, the weight of 3.0 is replaced by the interval (1.5 4.5), and for the items where the reverse is true, the previous weight of .33 is replaced by the interval (.17 .5).

This replacement of weights by intervals requires the following modification in the implementation of the procedure for anticipating pragmatic interpretations: When a comment under consideration (C) is being compared with an alternative comment (C'), each one can no longer be viewed as having a single rating of desirability according to the projected ascriptions, but rather a range of possible ratings. C' then only rates higher than C if **each** of its possible ratings is greater than the corresponding rating of C (it is sufficient to make the comparisons only for the corresponding endpoints of the intervals). It is this use of intervals that implements the specification in Step 2 of the procedure at the bottom of Figure 5 that it be "clear to L" that the alternative comment would rate higher.[10]

During the generation of the examples for the first seven situations a slight amount of uncertainty on the part of L was actually assumed in the simulation

[10] The situation is actually more complicated when the focus of the question is a single item and a comment is being selected on some other item. Then it is not only L's uncertainty about the latter item that must be taken into account, but also his uncertainty about the focal item, which affects the relevant cost- and benefit-magnification factors. The condition is then that C' must rate higher than C according to all combinations of weights for the two items.

model, since otherwise the model would have anticipated some pragmatic interpretations that are so implausible as to be distracting.

When deciding whether to use a preparatory expression, S now does so if the criteria for its use are satisfied given at least one of the weights in the interval considered by L.

9.5. Resulting Behavior

1. S will now less often expect L to derive a pragmatic interpretation of a comment which is narrower than its straightforward implication.

Comparing Column 3 of Figure 11 with Column 2, we see that L no longer makes the second pragmatic interpretation after Question 1 or the first one after Question 3. These pragmatic interpretations would require quite precise assumptions about S's ascriptions.

2. Some types of pragmatic interpretation can be expected even when L has only a vague idea of S's ascriptions.

Note that S expects her first comment (that she never got much leadership experience) to give rise to the same pragmatic interpretation in all of the conditions in Figure 11. The derivation of this pragmatic interpretation involves a comparison between two comments which require approximately the same amount of effort: S could just as easily say that she had "no leadership experience" if this were the case. Since this more precise statement would yield more benefits for an objective speaker virtually no matter what specific evaluation standards she ascribed to the listener, L need not have a definite idea of S's ascriptions in order to interpret the fact that the statement was not made.

More generally, whenever one possible comment conveys more information than an alternative one and requires the same or a smaller amount of effort, one needs to make only weak assumptions about the value of the information conveyed in order to predict that the more informative one will be preferred by a cooperative speaker if both are true. Cases like this are often discussed in analyses of conversational implicature (e.g. [GAZD79, chap. 3, GRIC75]). By contrast, where the more informative of two comments involves substantially more effort – in particular, when the less informative one is silence – L can only be sure that the more informative one would be preferred if he can assess the additional benefits it would bring for the speaker and weigh these against the additional effort required.

3. S will now less frequently have to reject a comment because of a fear that it might be inconsistent with her image.

This consequence is a corollary of the first one: Since L generally makes fewer pragmatic interpretations, he is less likely to find a given comment inconsistent with his (vague) image of S. Thus the two additional comments after Question 3 can now be volunteered just as they were in Column 1: even though L wouldn't actually predict that a naive applicant would make such comments, he now thinks that this deviation from his expectations might be due to his own ignorance of the standards that this particular naive applicant ascribes to employers.

4. There are some desirable comments that S cannot make even if L's uncertainty is very substantial.

This is a corollary of the second consequence. After Question 1, for instance, S is inclined to say that she has "some leadership experience"; but since a more informative comment would be available, at no extra effort, no matter what amount of experience she had, she would see, that this comment would betray her positive bias.

The first and third consequences can be summarized as follows: Uncertainty on the part of L about S's ascriptions tends to cancel the effects of a discrepancy between the actual and the projected ascriptions. This is illustrated by the fact that Imp's behavior in the present situation (Column 3) is almost identical to its behavior in Column 1, where there was no discrepancy. In general, of course, behavior in these two conditions can differ more sharply, if L has less uncertainty or the discrepancy between the actual and the projected ascriptions is greater.

The following consequence of L's uncertainty concerns S's attempts to establish a projected bias, as in Situation 6. It is illustrated by the examples presented earlier, in Figure 10, which were run assuming the same degree of uncertainty on the part of L as in the present situation.

5. It is generally easier for a speaker to establish a given projected bias if the listener is uncertain about the speaker's ascriptions.

We have just seen that S generally has more freedom in choosing comments when L is uncertain about her ascriptions. This applies as well to S's attempts to find bias-establishing comments. For example, the four comments marked with asterisks in Figure 10 are only acceptable because L is so uncertain as to how S assesses the importance of the various items that he will hesitate to judge the comments to be peculiar and uninterpretable. If L had more definite ideas about S's ascriptions, either S would have to make a less desirable (but more plausible) comment or she would find no way at all to establish her desired projected bias immediately.

But if L's uncertainty makes it easier to appear consistent with her desired projected bias, does it not also make it more difficult for her to appear inconsistent with her undesired projected bias(es) — an equally essential condition? This is true in some cases. For example, the fact that S's first comment in response to Question 5 in the middle column is inconsistent with positive bias is not at all obvious; if L were a bit more uncertain about S's ascriptions than the simulation model was in these examples, this comment would not convince him of S's objectivity.

But very often bias-establishing comments are inconsistent with the undesired projected bias(es) in ways which can be recognized without particularly definite beliefs about S's ascriptions. First, vague comments are almost always inconsistent with objectivity if a more precise comment is available at no extra effort. This is why the third comment in the right-hand column of Figure 10 can be easily recognized to be inconsistent with objectivity even though it is not obvious that it is consistent with negative bias.

Second, nonobligatory comments which cause downward impression shifts are almost always inconsistent with positive bias, as noted in Section 3.5. In order to recognize that the first and second responses in the middle column and the third one in the right-hand column are inconsistent with positive bias, therefore, L essentially only has to know enough about S's ascriptions to recognize that S expects these comments to shift his impression downward (in these examples, this expectation of S happens to be expressed anyway in her use of a preparatory expression).

In short, it is generally less of a problem for a speaker to appear inconsistent with an undesired image than it is for her to appear consistent with the desired one. This is why it is easier to establish a desired image if the listener is tolerant in his judgments of what behavior is consistent with a given image.

9.6. General Issues Raised

These consequences suggest that it may often be sensible for S to try to ensure that L has only a vague idea of S's ascriptions. Thus in the present examples, the applicant might try to appear generally unfamiliar with the usual evaluation standards for applicants for this type of work. This would cause L to be uncertain about S's specific ascriptions and thus make it easier for S to appear naive with respect to the interviewer's application of a double standard.

The definition of this situation stipulated that L's uncertainty did not involve coherent patterns involving several items, such as the question whether S was realistic or naive. This sort of uncertainty can of course occur, but it introduces considerable additional complications for S which cannot be explored in detail here: sooner or later, S is likely to say something that is inconsistent with at least one of the images L considers possible; this will narrow down L's expectations as to what she will say on other items and maybe even cause L to reinterpret what she has already said.

We hypothesized that L takes into account his uncertainty about S's ascriptions by only making a pragmatic interpretation if he is quite sure that it is justified. Listeners presumably also make probabilistic pragmatic interpretations of the following sort: "I'm pretty sure she would have said something like C' instead of C if the possibility P were realized, though C is quite a likely comment if P is not realized; so probably P is not realized, even though is reasonably likely a priori." A formalization of this sort of reasoning, using the principles of Bayesian inference can shed further light on the basic logic of pragmatic interpretation. But such an analysis requires much stronger assumptions about the nature of L's uncertainty than the ones made here, and it cannot be integrated easily into the present simulation model.

10. Situation 9: Uncertainty in the Speaker About Her Projected Ascriptions

10.1. Definition of the Situation

S once again assumes that L has definite beliefs about her ascriptions, but now S is uncertain as to what these beliefs are. In other words, S suspects that there is a discrepancy between the actual and the projected ascriptions, as in Situation 7, but she is now unsure about the nature of the projected ascriptions. As before, S wants to behave consistently with the projected ascriptions – whatever these may be.

10.2. How the Situation Can Arise

One of the many ways in which this situation can arise is a variant of the case used in the examples for the previous two situations: S has realistic ideas about the standards that L is applying, and she thinks that L may know this, but she also thinks that he might consider her ascriptions to be naive. In other words, she does not know whether her projected ascriptions are realistic or naive.

A different sort of pattern is worth mentioning because of its frequency of occurrence: the case where a speaker wants to appear more familiar with the accepted norms in a given community than she actually is. For example, if for a given type of work there are certain generally accepted criteria for evaluating candidates, a job applicant who gives the (false) impression of being thoroughly familiar with this type of work will be expected to know these standards. Thus the projected ascriptions will be the generally accepted criteria, but the applicant will not know what they are. All of us have presumably been caught up in this sort of situation when asked to comment, for example, on a work of art of a sort that we are not accustomed to talking about.

10.3. Appropriate Strategies

It will be convenient to refer to any set of evaluation standards that – as far as S knows – L might think that S ascribes to him, as a possible **set of projected ascriptions**. Strictly speaking, S must in general reckon with an infinite number of these sets. A reasonable general strategy is to consider some small number of them which are sufficiently representative, in the sense that behavior which is consistent with each of this small number of sets is likely to be consistent with all of the other possible sets. What (and how many) sets of ascriptions should be used as representatives depends on the structure of the space of possibilities that S must reckon with; it is therefore difficult to give any general rules for choosing them. (For the examples to be discussed below, a pair of extreme sets of ascriptions is sufficiently representative.)

When considering a possible comment, S should anticipate its pragmatic interpretation according to each of the representative sets of projected ascriptions. Since the pragmatic interpretation may be different according to each set, S will sometimes be unsure as to how a given comment will be interpreted and will therefore have to keep track of more than one possible interpretation.

If the comment would be inconsistent with even one of the representative sets of projected ascriptions, it should not be made, since it is then possible (though not in general certain) that it would be inconsistent with S's image.

For the same reason, the use of a preparatory expression should be avoided if there is some set of possible projected ascriptions according to which it would be inappropriate.

10.4. Realization in the Simulation Model

As in the previous two situations, the implementation is restricted to sets of projected ascriptions which differ from the actual ascriptions only in terms of the weight attached to each item. The system handles this situation as if it were addressing two listeners at the same time, for whom the projected ascriptions were as different as possible – e.g. one listener who thought that S's ascriptions were naive and one who thought they were realistic. The projected ascriptions for each item are no longer represented by a single weight interval, as in Situation 8, but by an ordered pair of intervals, one interval for each possible listener. For example, each of the "unimportant" items in Figure 11 is associated with weight intervals of (.9 1.1) and (2.7 3.3) for listeners to whom realistic and naive ascriptions, respectively, are projected. (A small amount of uncertainty in L – 10% as compared with the 50% of Situation 8 – is assumed for the examples in Column 4, since the situation is even more difficult for S to manage if no such uncertainty at all is assumed.)

The system implements the strategies mentioned above by anticipating pragmatic interpretations, and checking the appropriateness of preparatory expressions, using each weight interval separately.

10.5. Resulting Behavior

1. The speaker will sometimes be unsure as to what pragmatic interpretation, if any, will be made of a given comment.

This happens after Questions 1 and 3 in Column 4 of Figure 11. In each case L is expected to make a pragmatic interpretation only if the item involved is important according the projected ascriptions.

2. The speaker must sometimes suppress desirable comments even while realizing that they might be perfectly acceptable to L and that their suppression might in fact give rise to undesirable pragmatic interpretations.

The reason why S gives a minimal answer to Question 3 is that the comments she would like to make would sound strange if her projected ascriptions were naive, as we saw in Situation 7 (Column 2). But now, S must also reckon with the possibility that her projected ascriptions are realistic, in which case her silence will be interpreted to mean that there is nothing worth mentioning about the three items which are most relevant to the question. In other words, S's attempt to maintain more than one image at a time tends to make silence the most attractive option (even more so than in Situation 7), since it is the response which is generally the most compatible with conflicting images. But since L is not aware that this is the reason for S's silence, he will sometimes draw incorrect and unfavorable substantive conclusions.

This general pattern is especially familiar in the case of a speaker who is uncertain about the accepted standards of a community: the natural coping strategy is to say as little as possible, except on matters where even a vague notion of the prevailing standards is sufficient to verify that a given comment is acceptable.

3. In some cases there will be no behavior available with which the speaker can be sure of maintaining her image.

Sometimes even silence is not consistent with all possible sets of projected ascriptions. For example, after the obligatory yes/no answer to Question 4 in Figure 11, a realistic speaker would be expected to eliminate the remaining uncertainty on the same item by saying some more about her boyfriend's attitude, whereas a naive speaker would be expected to consider further commentary superfluous. This predicament is not quite as serious as the one illustrated at the end of the example dialog for Situation 5 (Figure 8), in which S could not avoid behaving inconsistently with her image: S now has a chance to escape this fate by guessing the projected ascriptions for the critical item correctly. Exactly how S should handle this situation is hard to say in general (it depends, among other things, on whether the projected ascriptions for the different items are systematically related). Just one of many possible strategies is implemented in the Imp system and illustrated in Figure 11: the system chooses the most desirable comment which is consistent with at least one image and gives up on trying to maintain the images with which that comment is not consistent. In these examples, this means opting for the "realistic" image and producing not only a comment on the item asked about but also a second one which S would actually have liked to volunteer while responding to the previous question. It is interesting that this second comment has to be kept more vague and less positive than S would like, since S is constrained by the pragmatic interpretations that L made while S was still being cautious: if S stated that she never intended to have children, this would surprise even a listener who expected her ascriptions to be realistic, since he would wonder why she hadn't volunteered this important information in response to the previous question.

10.6. General Issues Raised

This last example dialog calls to mind an attempt to navigate in densely mined waters, and it may have struck the reader as rather implausible that a speaker should experience such difficulty in maintaining her image while responding to four simple questions. But recall that the situation as defined has an asymmetry which makes it exceptionally challenging for S: she has considerable uncertainty about L's expectations, yet she thinks they are quite definite and wants to conform to them whatever they might be. In a more symmetric situation, presumably more frequent in everyday dialogs, S's uncertainty about her projected ascriptions is similar in extent to L's uncertainty about her ascriptions. We need not discuss this situation at length, since the results are predictable on the basis of the discussions of Situations 7 and 8. Just as L's uncertainty in Situation 8 largely eliminated the effects introduced in Situation 7 by the **discrepancy** between S's actual and projected ascriptions, in this new situation L's uncertainty tends to eliminate most of the complications caused by S's **uncertainty** about her projected ascriptions. In fact, the simulation model's responses to the questions in Figure 11 differ only slightly in this new situation from those seen in Columns 1 and 3; in particular, the predicament at the end of Column 4 does not arise. In sum, the lessons to be learned from Situation 9 concern not only the nature of the behavior it tends to elicit but also the conditions which are critical to the situation's definition.

A situation analogous to the present one can arise if S is initially uncertain about her projected bias (as opposed to her projected ascriptions) and wants to behave consistently with it whatever it is (as opposed to trying to establish a particular projected bias, as in Situation 6). This can happen, for example, when S is trying to fit into some community which has certain generally accepted biases which are unknown to S, e.g. toward particular political figures. It would be interesting to investigate systematically what sort of comments a speaker produces when asked her opinion of some controversial figure without knowing whether she was expected to display positive or negative bias. Presumably the difficulty of maintaining her image would incline S to say as little as possible; there would be few opportunities for S to attempt to create a favorable or unfavorable impression of the person in question in accordance with her actual bias.

11. Conclusions

We can look at the issues discussed in this chapter from another perspective by returning to the more practical problems of human-computer communication raised in the introduction. The above analysis of human dialog raises a number of questions about the design and use of user-modeling systems. Some tentative answers to these questions are also suggested by the analysis, and by the Imp simulation model; but these suggestions are obviously subject to revision on the basis of future research which examines interaction with such systems directly.

Whereas the above analysis has in effect taken the point of view of the system in human-computer dialog, the discussion to follow is written with the interests of the user in mind. In particular, it presupposes that discrepancies between the system's actual motivation or user model and the image it projects of them are undesirable and should be minimized.

The following two sets of questions and suggestions are directed at two categories of persons who can contribute to the reduction of such discrepancies: those who design such systems and those who use them (or who educate or inform the users.)[11]

11.1. Questions for Designers of User-Modeling Systems

The most obvious way of ensuring that users have an appropriate understanding of a system's user model and motivation is to describe the system clearly in documentation (either printed or on-line). But there are severe limits to what can be accomplished in this way. On the one hand, users often lack access to a system's documentation or don't bother to consult it (especially with systems like those of [CHIN*, QUIL*]). On the other hand, even the most skillfully written documentation cannot convey a clear picture of a system whose structure and behavior are

[11] Since the roles of the system and the user correspond to those of the speaker and the listener in the main analysis, masculine pronouns will be used in this section to refer to the user.

intrinsically difficult to understand. Attention will thus have to be devoted to the following questions:

1. What types of user model and dialog motivation are in general easiest for users to perceive accurately?

A generally accepted principle in the literature on human-computer interaction is that even quite a complex system can be comprehensible if it corresponds to some familiar metaphor. As can be seen from the chapters in this book, most user-modeling dialog systems are based on the metaphor of the human dialog participant. In fact, it is difficult to think of an alternative metaphor for systems which include user models, since human beings are the only familiar objects which construct and exploit complex models of people. Moreover, the metaphor of the human dialog participant, to the extent to which it is reasonably accurate in a given context, has an advantage not shared by metaphors involving inanimate objects. The user can not only make use of his extensive experience with other dialog participants; he can also imagine what he himself would do in a given situation in order to predict and interpret the system's behavior. In empirical studies concerning the everyday prediction of the behavior and beliefs of other persons, the predictor's self-knowledge is typically found to be a major determinant of the predictions, often more important than all of the predictor's independent beliefs about other persons (see e.g. [NIKU85] and the further references given there).

As with any metaphor, it is important that there be an easily recognizable boundary between the respects in which the system behaves like a human dialog participant and those in which it does not. Otherwise overapplication of this metaphor will lead to incorrect expectations, misinterpretations, and puzzlement (cf. [SHNE87, pp. 322-325]).

In particular, user modeling techniques or forms of dialog motivation which have no analog in everyday human dialog are problematic, even if these features seem especially effective or otherwise appropriate in their own right. The designer should consider whether such a deviation from the basic metaphor can lead to incorrect expectations and interpretations by the user.

Although the Imp system was not designed for practical use, it can serve as an example of a user-modeling system which is basically easy to understand and predict (except when it is attempting to project an inaccurate image). The criteria according to which it selects factual comments approximate human behavior quite well, and the many respects in which its behavior deviates from that of human dialog participants (e.g. its restriction to the production of purely factual comments) are easy to recognize and to describe.

Whether or not a system's basic design makes it intrinsically relatively easy to understand, there will often be cases where the user begins the dialog with specific false assumptions about its user model or motivation, or an amount of uncertainty about it which constitutes an obstacle to effective interaction. The considerations mentioned in Section 7.3 suggest why it is desirable to make the user's perceptions more accurate as soon as possible within the dialog. One general approach to achieving this is addressed by the following question:

2. How should the system behave so as to project an accurate image of its motivation and user model?

In Situation 6, one general strategy for establishing a given image was examined, namely that of behaving consistently with that image and inconsistently with any likely alternative image. This strategy can in principle be applied by user-modeling systems, but its limitations must be borne in mind: depending largely on contextual factors beyond the system's control, the strategy can often only be applied, if at all, at the expense of other dialog goals – even if the image to be established is an accurate one.

An alternative approach is to take advantage of the extremely broad bandwidth of computer-human communication to convey to the user explicit information about the state of the system. As a rough initial example of what can be accomplished in this way, the Imp system uses a dynamic display on a relatively simple semigraphic terminal. For a number of items in the evaluation form there is a continuous graphic representation of the impression that the user is expected to have, the changes in the impression that the system would like to bring about, and the degree of relevance of the item to the topic of the most recent question. With techniques such as windows and high-resolution screens, it should be possible to make such displays available to the user in a way that is informative yet unobtrusive.

A further question calls attention to a different approach to the same problem:

3. Under what circumstances should a system adapt to its user's expectations about itself instead of trying to influence them?

Consider, for example, a system which is usually objective and which has quite an extensive model of its user, but which somehow infers that the user expects it to be positively biased and to lack certain types of information about him. The simplest way for the system to resolve these discrepancies might be to conform to the user's expectations by adopting a positive bias and refraining from using the information that it was not expected to have. To be sure, this could result in behavior which was less helpful to the user than it might be; but on the other hand it would minimize the chance of incorrect pragmatic interpretations. In the terms used in our analysis of evaluation-oriented dialog, this amounts to using the projected bias and the projected ascriptions not only to check pragmatic interpretations but also to evaluate the desirability of possible comments in the first place.

Regardless of the extent to which it proves possible to ensure that the user's image of the system is appropriate, the following question should be considered:

4. In what ways should the system try to anticipate specific pragmatic interpretations on the part of the user?

If the user's image of the system is basically accurate, the main purpose of anticipating pragmatic interpretations is to keep track of the user's specific beliefs, so that, for example, the system will not volunteer information that the user has already inferred. To be sure, as noted in Section 5.6.3, this strategy is unproblematic only if the user is aware that his pragmatic interpretations are being anticipated in this way, since otherwise he will have an inaccurate idea of the specific beliefs that the system ascribes to him. On the other hand, if the system has not managed to ensure that the user has an accurate image of its motivation and its user model, it may be important to anticipate possible misinterpretations which can have serious consequences. What the system should do when it thinks that its behavior would lead to such a misinterpretation depends on the approach taken to the three questions just discussed.

11.2. Questions for Users of User-Modeling Systems

It is unrealistic to expect that user-modeling systems will always embody satisfactory answers to the four questions just discussed. Aside from the difficulty of the questions, there is the problem that designers will not necessarily share the basic goal of ensuring that the user has an appropriate model of the system. Users will therefore have to accept part of the responsibility for pursuing this goal.

5. How can the first-time user of a given system form realistic prior expectations about the system's user model and dialog motivation and about the sort of image that it is likely to try to project?

As far as the user model itself is concerned, the user should have a general understanding of the basic difficulties involved in user modeling and of the state of the art. If the user is to have realistic expectations about the system's dialog motivation and about the image it will try to project, it is useful for him to have some understanding of the reasons why certain combinations of actual and projected parameters may seem more attractive than others to the designer (or to the system itself). Unfortunately, the considerations involved here can be quite complex, as was shown by the discussions above with respect to everyday dialog. For instance, in the examples in Situation 7, the speaker's goal of projecting certain ascriptions with the aim of being diplomatic conflicted with her goal of making a favorable impression; and it was pointed out in Section 7.2 that a speaker's decision as to what sort of bias to project can depend on the specific properties that the Object in question has – which are of course not known in advance to the listener. A realistic user, who is aware of these complexities and of the fundamental difficulty of user modeling, will therefore often not have very definite prior expectations about the system. A consequence of this open-mindedness is that he will initially make relatively few pragmatic interpretations of the system's behavior and thus will tend to avoid misinterpretations (cf. Situations 8 and 9).

6. How can the user form a more definite and realistic image of the system on the basis of its behavior?

Even if the system does not provide explicit information about its user model and motivation (cf. the second question above), the user can in principle draw useful conclusions about these aspects of the system on the basis of the system's dialog behavior. To be sure, this is only possible if the user is sufficiently sophisticated about any image-maintenance strategies that the system might employ. In the analysis of evaluation-oriented dialog, the speaker's attempts to project a bias different from its actual bias presupposed that the listener was not aware of the possibility that she might do this or of the way she would go about it. If the user's degree of sophistication matches or exceeds that of the system, he may be able to avoid being misled; but if he is to be able to draw definite conclusions, he may need to devote some effort to active exploration of the system. This leads us to the final question:

7. How can the user experiment with a user-modeling system in order to determine its motivation and the nature of its user model?

One often sees novice users of systems such as text editors engaging in behavior which is aimed primarily at revealing the properties of the system, rather than performing some task with the help of the system. This sort of exploratory behavior

can require considerable insight and skill with user-modeling systems, especially if the system (unlike the Imp model) tries to interpret the user's behavior in terms of the user's plans and goals or (like Imp) sometimes attempts to project an inaccurate image. Experience in interacting with such systems can increase a user's skill at this sort of exploration. For example, with the Imp model it is an instructive exercise for a user to take the role of the listener and to try to pose a sequence of questions which will lure the system into a situation in which the discrepancy between its actual and projected dialog motivation will be exposed (as at the end of Figure 8).

11.3. Concluding Remarks

The reader may be reluctant to accept the idea that the practical use of user-modeling systems makes it necessary to think about the questions listed in this section. On the other hand, we should note that all of these questions have analogs that must at times be dealt with by speakers and listeners in everyday dialog; and it is unrealistic to expect that we can endow dialog systems with a sort of understanding which is characteristic of human dialog without having to deal in some way with the associated complications.

An encouraging thought is that, although these questions cannot be ignored entirely, the answers to them will sometimes be simpler than they are in connection with human dialog. Unlike human dialog participants, user-modeling systems can be designed from the start so as to be as comprehensible as possible (cf. the first question above); and if this is accomplished, the remaining questions tend to become either less important or easier to answer satisfactorily. In addition, computer technology offers some specific solutions that are not available in human dialog (as noted in connection with the second question). But note that these potential advantages of user-modeling systems over human dialog participants will not follow automatically from the fact that the systems are not human beings (as was presupposed by some of the views discussed critically in the Introduction). The advantages can only be realized if those who design user-modeling systems (a) accept the goal of ensuring an accurate perception of their system by its users and (b) understand the complexities of human dialog sufficiently well to be able to achieve this goal.

Acknowledgements. Thanks are due to H. Bego, E. Hovy and R. Meertens for helpful comments on earlier versions of this paper and to the informant who supplied the information required by the simulation model to produce the examples. The participants in the Maria Laach Workshop also provided useful feedback, and the volume editors created exceptionally supportive conditions for the authors.

A brief, early description of the simulation model discussed in this paper was given in [JAME83], and a shorter version of the material in Sections 2 through 6 appeared as [JAME87].

12. References

[AJZE71], [ANDN81], [CARB*], [CARO83], [CECI84], [CHIN*], [CLHH81], [EAGL81], [FINI*], [GAZD79], [GOFF59], [GREW83], [GRIC75], [HARR63], [HINT75], [HOLL86],

[HOVY86], [JAME82], [JAME83], [JAME84], [JAME87], [JONE65], [JOSH82], [JOSH84a], [KOBS85a], [KOBS*], [MACL82], [MCCL78], [MCCO*], [MCKE85b], [MORI*], [NIKU85], [NIRO80], [NORM86a], [PARI*], [QUIL*], [REIC78], [RICH*], [ROWE84], [SAMP82], [SCHW79], [SHNE87], [SIKL78], [SNYD79], [SPAR*], [TETL85], [THMA83], [VAND81], [WALS66], [WARD74], [WEYD79], [WINO77], [WINO86], [WYER74]

11 Incorporating User Models into Expert Systems for Educational Diagnosis

Robin Cohen and Marlene Jones

Abstract

In this chapter we study a particular real-world domain, that of educational diagnosis. We argue that expert systems for educational diagnosis require user models, and that these user models should include several components, including the user's background knowledge of both the student and the domain as well as the user's goals. Our proposal is directed at enhancing the particular expert system of the CGD project. We then propose an architecture for this expert system that separates the knowledge base into relevant components and includes a user model. We further demonstrate that this divided model for the system facilitates providing the best response for a particular user, according to his background knowledge of the domain and of the student and his goals. Finally we argue that the techniques outlined here will be useful in general in expert systems, to vary the response to the user at hand.

1. Introduction

There are many reasons for including a model of the system's user within an expert system. The foremost is to allow the system to tailor its explanations to meet the particular user's requirements. These explanations may concern the gathering and entering of data, the use of the system itself, or the system's findings, recommendations and reasoning process. One may argue that it is possible to alter the system's explanations without going to the effort of including an elaborate user model or any user model at all. For example, one can simply store predetermined explanations with varying amounts of detail and allow the user to request further (i.e. more detailed) explanations, if desired. This is unsatisfactory in situations where one would like to vary several aspects of the explanations simultaneously, which is the case in educational diagnosis, the application domain with which we are concerned.

We provide examples to defend the need for user models in our expert system, and then address the issues of what information to incorporate into a user model, how the model can be derived and updated, and how the model can be used to provide explanations. In short, we argue for the feasibility and usefulness of modeling (not just the student but the expert system user) for educational diagnosis.

2. Application Domain

This chapter studies a particular real-world domain – educational diagnosis. When a student is experiencing learning difficulties, it is important to ascertain the exact nature of the problems as quickly as possible. This may involve an in-depth educational diagnosis with the ultimate aim of developing a remedial program which is tailored to that particular student. An in-depth diagnosis/assessment generally includes examining the student's developmental history including the home environment and early school history; assessment of psychoeducational correlates such as intellect, visual and auditory acuity; language skills; achievement within fundamental areas such as reading, arithmetic and spelling; medical and behavioral information as well as a detailed assessment of any achievement areas in which the student's performance is deemed to be inappropriate. Usually such an assessment is carried out by a psychologist and/or resource room teacher. The resultant diagnostic report is generally made available to the regular classroom teacher and any other participants in the remedial program, as well as to the necessary administrators such as the principal. Of course, the diagnostic findings and educational decisions should be discussed with both the student and parents.

There are well-accepted guidelines and procedures regarding educational diagnosis [COMM71] as well as appropriate models; for example, see [ADEL82, MCLE82] and the references therein. It is rare, however, to find educational diagnosis carried out as thoroughly and rapidly as the guidelines suggest. This is largely due to insufficient resources, particularly personnel such as highly experienced resource room teachers.

One means of facilitating educational diagnosis within the regular school environment is via an expert system to guide a teacher (such as the resource room teacher) through the various stages of diagnosing learning difficulties. This is the long-term goal of the CGD (Computer-Guided Diagnosis) Project based at the University of Saskatchewan [MCLE85]. Preliminary systems have already been developed in the area of reading [COLB82, COLB83, COLB84] and arithmetic [JONE86, TUBM86], with a more elaborate, comprehensive system under development [JONE87, MCLE86]. It is this latter expert system that we consider in this chapter.

The advantages of such a system are many. It will serve as an information source to experienced diagnosticians, an assistant to less experienced resource room teachers, and a tool for training teachers. Even during the handling of one particular case, there may be a variety of users at each of the several stages of the diagnostic process – in-depth assessment, development of a remedial program, and monitoring the progress during remediation, which generally includes further assessment and may include placement and funding decisions. Furthermore, one may want to make some portions of the system's findings and suggestions available to either the parent or student. Hence, such an expert system may be used by or required to supply explanations to any of the following individuals: psychologist, resource room teacher, educational consultant, regular classroom teacher, special education teacher, principal, outside administrator such as the head of special education ser-

vices or superintendent, researcher, parent, the student him/herself, or an education student or training teacher.

Obviously these users vary in a variety of factors, most notably in their *background regarding the field* of special education. However, even personnel with a strong background in special education will vary immensely in regard to their familiarity with testing procedures, knowledge of individual tests (some of which require special training in order to administer), knowledge of areas of assessment such as reading, arithmetic, language etc., administrative procedures and guidelines, laws and mandates, outside resources, funding criteria and availability etc.

Individuals will also vary in their *familiarity with the individual case*, such as knowledge about the medical history, the home environment or the current school behavior. For example, the student, parent and teacher are familiar with the situation on a day-to-day basis whereas the psychologist may have a one-page referral form and the principal may have a short synopsis of the recent observable problems. Similarly, the "official" opinion on the case may be available to a select few. Some individuals, such as the psychologist and resource room teacher, may be familiar with a variety of similar cases which the teacher and principal are not.

Also the *goals* of the users may vary. Although hopefully all users' main goal is the implementation of the most appropriate remedial program, the subgoals may vary. For example, the teacher's most immediate concern may be techniques for decreasing disruptive classroom behavior. The principal may be concerned as to whether the student qualifies for special funding. The parent may be searching for assurance of the student's normalcy.

How does this myriad of variations affect the expert system's explanations? First of all, if the user has a particular goal in mind, this should be addressed. Even for a particular question or subgoal, the explanation itself should vary depending on the user's familiarity with the case and background in special education. Ideally, both the amount and detail of the explanation should be altered. Most parents will not benefit from a detailed description of the student's performance on a variety of tests, nor is an in-depth description of each remedial activity appropriate. This information is, however, critical for the resource room teacher who is implementing the program in conjunction with the regular classroom teacher (often with the assistance of a consultant). The language in which the information is expressed is also critical. Most parents are overwhelmed by the use of technical terms. Straightforward, honest explanations are desired, which must include realistic encouragement for both parent and student. It is important to remember that both parent and student may have been aware of problems for several years without understanding them. In other words, one may be contending with several years of frustration, fear and maybe even mistrust. Hence, the tone of the discussion is important.

We focus in this chapter on the problem of how to set up the expert system so that different users can get different explanations. We then provide a framework for varying the content of the explanation. The topic of varying the surface form with appropriate natural language generation is left as future work.

3. Designing the System

In this section we propose an architecture for our expert system and a control structure for searching the knowledge bases of the system when producing explanations, which will provide different responses to particular users, thus making the current system available to a variety of individuals. Although this model for making the system user-specific has not yet been implemented, we provide an analysis of some sample queries to illustrate the usefulness of the approach.

One contribution of the research is thus specifically for educational diagnosis, a method for making a system more effective. We suggest extensions to the current implementations of the CGD project, as illustration. Moreover, the proposal outlined here can be viewed as a general strategy for incorporating user models into expert systems to provide clearer explanations, primarily by avoiding the presentation of material already known to the user.

3.1. Motivation for System Design

The educational diagnosis expert system of the CGD project is designed to interact with a school psychologist or resource room teacher to discuss the case of a particular student. As the diagnosis proceeds, information regarding the student's background, behavior, abilities, skills, *etc.* is collected from a variety of sources. At any point in the diagnostic process, the system analyzes the available information and employs its knowledge of educational diagnosis to draw conclusions which may include recommendations for further assessment or particular hypotheses regarding the student's difficulties. The main types of questions being posed by the user are thus: (i) what is the diagnosis for this student? (ii) why is this the diagnosis? and (iii) why is this diagnostic procedure (such as a particular test or type of test) being recommended? The system should be able to respond to similar questions posed by a classroom teacher, parent, or school administrator. Furthermore, there may be a variety of backgrounds among members of any of these "types" of users – e.g. two psychologists with different favored approaches to diagnosis.

In Example I we illustrate the need for different responses to the same question for different types of users.

Example I:

User: Could the student's mispronunciation "errors" be due to dialect?

Response to Parent: Yes, non-standard pronunciations may be due to dialect rather than poor decoding skills.

Response to Psychologist: Yes, the student's background indicates the possibility of a dialectal difference.

Here, the parent is asking for more information regarding the domain of educational diagnosis (lacks the domain knowledge). The psychologist knows about the tests employed and has the necessary domain expertise, but is unaware of the background of the student.

In Example II, we emphasize the possible difference in goals of users employing the expert system. In particular, we indicate session-wide preferences a user may bring to bear.

Example II:

System: Administer an individual intelligence test. For this particular case, the Peabody Picture Vocabulary Test (PPVT) is recommended.

User: Why is this test being recommended?

For each user, the stated goal is the same – i.e. to find out why this test is being recommended. However, the inferred or intended goal may be different. This may be due to a system-wide preference. For example:

Psychologist Preference: administer intelligence tests with which s/he is familiar (e.g. the psychologist may be more familiar with the Stanford Binet or the WISC-R (Weschler Intelligence Scale for Children – Revised)).

Response: The PPVT is being recommended because it does not require verbal responses, and the student appears to have language problems including difficulties with expressive language. Hence, the PPVT is more appropriate than more common tests such as the WISC-R or Stanford Binet.

If the user is a teacher, principal or parent, the system would respond by explaining why an intelligence test is being suggested, as well as why this particular test is appropriate. The user may have additional goals which also must be addressed, as illustrated below.

Principal Preference: employ tests which can be used in funding formulae

Response: There is no evidence to suggest that the student will require or will qualify for additional funding.

Parent Preference: minimize testing due to stress involved

Response: The student should not find this test stressful, as no verbal responses are required; answers are indicated through pointing.

The above discussion may lead the reader to believe that it is sufficient to stereotype our users and exploit such defaults when answering the user's questions. Defaults alone are, however, insufficient. For example, a parent may have substantial knowledge of educational diagnosis because of involvement in organizations such as ACLD (Association for Children with Learning Disabilities) or CEC (Council for Exceptional Children) or because of professional training within a related field. And yet, the defaults of a fully trained psychologist may be too much.

Furthermore, there are several possible reasons why a psychologist may not fit a given stereotype. For example, s/he may have additional information about the student which the system is not assuming. The psychologist may have been involved in some previous diagnostic situation with this student or perhaps knows the student from other circumstances. A second possibility is that the psychologist lacks experience within a particular area of exceptionality or assessment, such as the diagnosis of emotional difficulties. Having determined this, the system knows that additional explanations must be supplied for all aspects of assessment within

this area, not just for the user's first query. Similarly, consider the case where the student is visually impaired. Although the psychologist may be well aware as to why certain skills are being assessed, s/he may be unaware of the appropriate tests to do so, given this additional constraint.

A third type of variation from the initial stereotype is based on differing philosophies. For example, the user may adhere to the belief that visual-motor training is effective in the remediation of reading difficulties; although previously a popular educational practice, current evidence indicates otherwise. Or the philosophical difference may stem from a disagreement regarding the usefulness of certain types of information during the diagnostic process or the conclusions that can be drawn from the student's performance on a particular test. The following example illustrates this point.

Example III:

System: Administer an individual intelligence test such as the WISC-R or Stanford Binet. User instead administers the ITPA (Illinois Test of Psycholinguistic Abilities) and the student's performance is recorded.

User: What are the conclusions based on the results of the ITPA?

System: The student's intellectual ability appears to be within the normal range; no further assessment of intellectual ability is required.

User: What other conclusions were drawn? For example, the student's poor scores on the Auditory Association and the Auditory Closure subtests indicate that the student's auditory channel is weak.

System: Although the test results raise some suspicions regarding the student's processing of auditory information, such conclusions cannot be drawn from the ITPA alone because the validity of the ITPA's subtests is questionable.

This type of question from the user (drawn from a real example case) would alert the system that the user does not adhere to certain current philosophies. The system does not consider the individual subtests of the ITPA to be valid. Moreover, the user does adhere to the ITPA model and related philosophies. Because the psychologist has drawn certain conclusions which the system has not, this will affect what s/he concludes in regard to reading assessment or even which tests s/he thinks are appropriate.

3.2. Details of the System Design

In essence, we are requiring our expert system to accomplish two modeling tasks simultaneously. The system must model the student in order to produce an appropriate diagnosis. In order to provide appropriate responses to the user, it must also model the user. The system must, therefore, contain the following three knowledge bases:

(1) domain knowledge which includes knowledge regarding diagnostic procedures, practices and educational theories, information regarding standardized tests and remedial activities etc.,

(2) a model of the student,
(3) a model of the user.

We show how it is possible to present a finer gradation, useful in generating responses. For each of these knowledge bases, different types of knowledge can be identified to partition the knowledge base further. For example, domain knowledge can be partitioned into rules of educational diagnosis versus specific data about given tests. Even within the category of domain rules, there exist different types of rules.

A more detailed description of the required knowledge bases is provided below.

Domain knowledge

Rules
- diagnostic procedures, including principles and terminology
  - e.g. if possible, employ individually administered tests because individually administered tests tend to have higher reliability than group tests
- test interpretation rules
  - general
    - e.g. IF the student's score is on standard deviation below the mean on an individually administered achievment test, THEN further assessment is required
  - specific
    - e.g. IF the student's reading quotient is less than 80 on the Schonell Reading Test, THEN further assessment is required

Databases
- test database
  - e.g. information regarding purpose, subtests, types of scores, reliability, *etc.* (The system employs this information in conjunction with the above rules, particularly the general test interpretation rules, to draw appropriate conclusions.)
- remedial activities and methodologies

Student knowledge

- background
  - background facts acquired through testing (e.g. age)
  - other background: family facts, etc. (e.g. medical history)
- test results
- hypotheses

theories of the system as to diagnosis of the student

User model

- knowledge of the domain (same division as above)
- knowledge of the student (same division as above)

- goals
  - for the session
    - domain goal (find "best" diagnosis for student)
    - preferences
      e.g. administer tests that can be used to get funding
  - for the particular question
    - stated goal
    - intended goal

As mentioned earlier, during the course of the diagnosis the expert system is developing hypotheses (i.e. a diagnosis) based on the information at hand, as well as suggesting further testing in order to gather information to improve the accuracy of the diagnosis. The techniques employed in answering the user's questions regarding either the diagnosis or the recommended diagnostic procedure are essentially the same. Therefore, for the next section of this chapter, we concentrate on one type of question: "What is the diagnosis?" and illustrate our methodology for providing an appropriate answer to this query. Figure 1 highlights the system's knowledge bases and introduces a corresponding notation.

In the next section we show a control structure of the system to generate a response for a user, mobilizing appropriate knowledge bases. The main features of this strategy are: (i) default assumptions about an entire class of users (psychologists, teachers, parents, etc.) are used as a starting point for the model of the user's knowledge (ii) the system is aware that its model of the user (as well as its knowledge of domain and student) is updatable – i.e. that new information revealed during a session with a user may cause the models to be revised. The defaults provide a starting point for the modeling, and the dynamic maintenance of the knowledge bases allows for additional exceptions to the assumed model of the user.

4. The User Model Structure

In this section we advance a proposal for the acquisition and updating of a model of the user to produce quality explanations. What we want, ideally, is a system that can respond to all the examples offered in the first part of 3.1. We indicate here a proposal to isolate the parts of the system required for responses. But we do not deal with lower level generation issues, such as choosing the most appropriate terminology for a user. We will at times indicate what should ideally result; in these cases we also make clear to what extent the current framework also provides a solution to these harder questions.

4.1. Characterizing the User

We now show how to use the diagram of the system, outlined in Section 3.2, to generate appropriate responses for different users. We illustrate the procedure for the case of a user who is a psychologist. Consider, in fact, a user who is the psychologist who tested the student whose case is being evaluated by the system.

System (Sy)
  Knows About Domain (KAD)
    – diagnostic principles (d)
    – test rules (t)

  Knows About Student (KAS)
    – background (b)
      – acquired through testing (t)
      – other background (o)
    – test results (r)
    – hypotheses (h)

  Believes that the User (B)(U..)

    Knows About Domain (KAD)
      – diagnostic procedures (d)
      – test rules (t)

    Knows About Student (KAS)
      – background (b)
        – acquired through testing (t)
        – other background facts (o)
      – test results (r)
      – hypotheses (h)

    Wants (W)
      – domain goal
      – domain preference
      – query goal (stated)
      – query goal (intended)

Figure 1. Labeling of system knowledge bases

The general principle used to produce a desirable response in an efficient amount of time is basically: "Don't explain what's already known, and keep track of what's already known, to avoid providing extraneous detail". This extension of Grice's maxim of quality [GRIC75] allows for some division of knowledge and tracking of shared knowledge bases.

At the start of the session, the system would evaluate the knowledge of the user, in terms of its knowledge bases, as follows:

1. SyB (UKASr) = SyKASr
   [read as: "system believes user knows about student ..."]

The system believes that what the user knows about the test results of the student is equivalent to what it knows. This assumption is reasonable as the test results have been input by the user.

2. SyB (UKASbt) = SyKASbt

The system believes that the background knowledge it has acquired about the student, specifically through testing, is equivalent to what the user knows. This

background information would be data like the student's age, public school, etc. that would be part of the information drawn during the testing stage. Note that the system may have gathered additional information about the student's background from other sources in the course of its search for a diagnosis, so there may be some discrepancy in the overall knowledge of the student's background between system and user (included in (KASbo)).

3. SyB (UKADt) = SyKADt

The system starts off with the assumption that the user is familiar with the standard interpretation of tests used in diagnosis.

4. SyB (UKADd) = SyKADd

The system also assumes that the user shares knowledge about basic diagnostic procedures.

The main features of the design are thus: (i) a coarse-grained division of the system into knowledge bases, to identify sources of shared knowledge and to thus focus on elaborating unknown information (ii) a tracking of specific information on the user's current beliefs to produce more co-operative responses within the areas where the knowledge is not shared. We illustrate these in a tracing of the system's algorithm below.

In order to describe how the system mobilizes certain components of the proposed architecture to generate better responses, it is necessary to present some model for the system's reasoning. The computational framework employed in the CGD expert system is THEORIST, a logic programming system based on default reasoning, developed at the University of Waterloo [POOL86]. For details and examples of the implementation of the CGD project using THEORIST, see [JONE87].

To clarify the discussion below, we refer to a more conventional model of reasoning in expert systems. Consider a production rule base, where the system encodes a series of "rules" of the form: "If X then Y", and may then record various "facts" of the form "X1", "X2", etc. By some deduction mechanism such as forward chaining, a diagnosis may be drawn by proceeding from the facts and rules to the most likely conclusions. The main framework which we draw for discussion is: facts + rules lead to diagnosis (F + R → D). (Note: which of the system's defined knowledge bases are invoked for the system's control structure should remain the same, regardless of the reasoning model employed.)

In our domain of educational diagnosis, the facts are items like scores on tests for student and rules are items like standard interpretations of test results for diagnosis. The model for reasoning is thus:

Facts + Rules → Diagnosis

This is expanded to:

Background Facts + Test Results + Domain Rules → *Hypotheses*

Further, Domain Rules can be divided into Test Interpretation Rules and Diagnostic Procedures, yielding the formula:

Background Facts + Test Results + Test Rules + Diagnostic Procedures → *Hypotheses*

where Background Facts may be further partitioned as:

Facts acquired through testing + Other facts

There are in fact two ways to use Test Rules to produce a hypothesis: a) use a specific test interpretation rule or b) use a general test interpretation rule, together with information from the test database regarding the test at hand. We put these two cases together under the label "Test Rules". Note that extensive use of the test database in fact occurs in conjunction with responses that suggest administering new tests. For future work, we can examine more closely possible sources of difference between the system and psychologist users regarding testing procedures. One strategy worth studying is segmenting the test rules portion of the knowledge base as well in order to more appropriately respond to the user.

Then, a system should respond to a question posed by a user by activating those parts of its reasoning towards a hypothesis that the user may not already share. We illustrate this focusing mechanism below.

Dialog I:

For a user who is a psychologist:
Question: What is the diagnosis for this student?
Response: *Hypothesis*
Reasoning: UKASr (test results)
UKADt (test rules)
UKADd (diagnostic procedures)
– if all Background has been acquired through testing, then UKASb (background)
– since *Hypothesis* is the only component not already shared by the user, this is included in the response

Note that after this exchange, SyB (UKASh) is updated to include the hypothesis just presented. Once more, the separation into knowledge bases allows for a current view of the user's knowledge.

Dialog II:

Question: What is the diagnosis for this student?
Response: *Hypothesis*, because background fact X is true for this student.
e.g. The system suspects that the student can be classified as "learning disabled"; there is a family history of reading problems.
Reasoning: as above, but in this case SyKASb→ SyB (UKASb) because there is some background on the student not directly acquired through testing
Guideline: The system is compelled to add in the response relevant information that the user may not yet know.

This example indicates that for our domain the knowledge base referred to as background on the student may be further subdivided into background acquired

through testing and background acquired from other sources. Again, the principle is to divide to facilitate labeling of shared knowledge with the user.

The above dialog can also be employed to demonstrate how the system would handle Example I, as illustrated in Dialog IIb below. Recall that the user is asking about a possible hypothesis:

Dialog IIb:

User: Could the student's mispronunciation "errors" be due to dialect?

Response to Psychologist: Yes, the student's background indicates the possibility of a dialectal difference.

Reasoning: (as in Dialog II above). In addition to responding with *Hypothesis*, provide the relevant background fact not acquired through testing.

A similar approach is employed when answering the parent's query:

Response to Parent: Yes, non-standard pronunciations "errors" as measured by the Neale Analysis of Reading Test may be due to dialect rather than poor decoding skills.

Reasoning: In addition to responding with *Hypothesis*, provide the relevant Domain Rules (in this case a Test Rule).

A more detailed description of the system's evaluation of users in the parent class is now in order. Basically, the parent is assumed to know about background information on the student. So:

SyB (UKASb) = SyKASb (this covers both bt and bo)

Otherwise, the parent is assumed to be unaware of how to interpret tests and unaware of the test results, yielding the conclusion that for this user, KAD is all unknown, and KASr is also unknown.

In order to actually generate different responses for different users, it is useful to outline particular schema, which can be instantiated by the specifics of the response to the query, as follows:

```
for the psychologist:
   (affirm (response)
    state (BACKGROUND FACT not acquired through testing))
for the parent:
   (affirm (response
    state (TEST RULE)
    state (DIAGNOSTIC PROCEDURE)
    state (TEST RESULT))
```

Consider Dialog IIb. For the psychologist the only part of the information extracted to produce the response that is unknown is dialect, an example of background not acquired through the testing stage. For the parent, the fragment is assumed from a dialog where the parent already knows the test result, so this is known and not part of the response content which must match the schema. Likewise, diagnostic procedure does not enter into the response.

In effect, our approach for generating user specific responses is similar to the idea of "user perspective", as used by [MCCO86] – except it is the user's perspective of the system's knowledge bases, and not of particular objects, which is used to introduce the variability. Our reading of a schema combines features of McKeown's [MCKE85b] and McCoy's [MCCO86] use of the same term. In [MCKE85b], there are schema to generate responses which contain predicates, some of which are optional, which then get instantiated according to the relevant knowledge in the knowledge base to produce the actual response. In [MCCO86], the concept of schema is carried further, to also expand the predicates of the schema according to the user at hand. Instead of focusing on certain objects in a knowledge base, we focus on the content required to produce a response to a query. But then the user at hand also determines the appropriate schema (or parts of a general schema) and the actual selection of predicates. Just as perspective in [MCCO86] allows for a user model which varies according to context, so our consideration of the user is tempered by the query at hand and the highlighting of the knowledge base for that user.

This approach can be contrasted with other efforts to model users for response generation. We feel it is more effective than the double stereotyping scheme of [CHIN86] because we can classify our users at several levels (vs. just expert or novice), and we are not tied to labeling all the content of the knowledge base according to its "difficulty". Rather, whole portions of the knowledge base are labeled according to whether they are known by the user, according to his class. For expert systems environments such as CGD this classification is quite natural. For Dialog IIb, it is unclear how a labeling of "level" of knowledge base information could afford the variation in response required. It is not that information about the student's background is "less difficult to understand" for some users; rather, it is a kind of knowledge more likely to be known to a particular class of user.

Our work is also in contrast to the work done in refining explanations from expert systems (e.g. [BUCH84, SWAR83]). These researchers focus on varying the chain of reasoning displayed to a user. We are concerned with varying part of an already selected response to produce the ultimate response to the user.

4.2. Allowing for Updates to the User Model

Note that when we paint with such a broad brush, as for the case of the psychologist, it is not unreasonable to have the system simply mark those parts of its knowledge base used for the calculation of the response which are not shared by (known to) the user. This is because the majority of the knowledge is shared, so that the process is still computationally feasible. The suggestion for the general case (any user, any type of user) is to track the specific user more closely to focus on possible arenas for misunderstandings or lack of information.

For the psychologist, tracking the user becomes an issue in questions of the type below.

Dialog III:

Question: What is the diagnosis for the student?

Response: *Hypothesis*
Question: But why is *hypothesis* appropriate for this student?
Analysis: There is now a conflict between what the system has recorded as known by the user and what the user should thus be able to conclude, and the lack of belief in the conclusion, as implied by the question asked.

One description of the problem is:

UK (A) and UK (A → B) and yet ~ UK (B)

This description is in fact the model for Dialog I above – the *Hypothesis* is explained simply because the user may not explicitly believe the desired conclusion. (See [LEVE84] for further discussion of the notions of implicit and explicit belief). In Dialog III the problem is that, additionally, part of what the user is recorded as believing to use in drawing the conclusion, has been incorrectly assessed as a current belief of the user. We will not include here a discussion of the possible reasons for the incorrect assessment, and which reason is most probable. (For example: Is it most likely that psychologists disagree about diagnostic procedure vs. test interpretation?). Instead, we emphasize the need to track the user to draw a more accurate picture of possible differences with the system's "default model" of shared knowledge.

One possibility is to draw from previous dialog other misunderstandings that have been recorded in order to come up with a generalization about differences with this particular user. One example of this situation is a psychologist who still believes that certain tests now considered outdated are effective measures of student's problems. Once an indication of the user's preference for a certain test is noted, this information can be noted as differing form the system's knowledge about diagnostic procedure – i.e. SyKAD+ SyB (UKAD). (See Example III).

Another effective strategy is to track the user's goals and preferences, and to postulate a plan for the user. This plan may then be compared to an ideal system plan, to note discrepancies and misconceptions. An outline of this work is provided in [VANB86a]. The general idea of performing some pragmatic processing on the input is not new; see [COHE85] for a review of some of the key efforts in natural language pragmatics. But the particular emphasis of van Beek's work [VANB86b] is to define a mechanism for recording user specific information, regardless of domain, and to suggest some methods for judging the best alternative available to a user, given his goals. The model is thus able to indicate a plan which also satisfies a user's preferences (see Example II). Thus, in conjunction with ongoing natural language research on plan-based approaches to inferring users' goals, it is possible to correct possible defaults that need to be retracted.

Note that the plan of the user is currently not included in the model of the user; what the system will be given, it is assumed, is some indication of the user's goals. The plan constructed by the system to help determine a response for the user, is not necessarily a plan that the user constructed, but a reasoning device for determining better alternatives for a particular identified goal (see [VANB86b]). It is possible to explore for future work the inclusion of procedures for additionally tracking the plan of the user, during discourse, to influence the response. But we have not advanced this in our proposal to date.

So far we have presented a basic strategy for answering a user's query. The general idea is that the response must contain the relevant aspects of the knowledge base which are not known to the user. In order to determine what knowledge is or is not shared by both the system and the user, the system employs initial stereotypes for the various user types, as well as updating this information through tracking the system-user dialog throughout the course of the diagnosis. In this manner, the system can determine what information the user is unaware of and hence, must be supplied as part of the response. However, this may not suffice. As a final step in the algorithm, the system must check that the user's goals and preferences have been addressed; if not, the answer must be modified to do so. Ultimately, one may wish to modify the interaction between the phase of the algorithm which detects which components of the knowledge base represent unshared knowledge and the portion of the algorithm which addresses the user's preferences and goals. The interplay between these two components also currently remains unaddressed.

The previously presented examples (Examples I-III and Dialogs I-III) illustrate various aspects of our general proposal for user specific responses. In the first example, the stereotypic models of parent and psychologist alert the system to what portions of the knowledge base must be supplied in the response (e.g. rule vs. background or testing facts). In Example III, we see the need to track the user-system dialog in order to update the initial stereotypic models, and in Example II, the need to address the user's preferences.

4.3. Dynamic User Modeling Through a Discourse

Before leaving this discussion of the system's general algorithm for answering queries (as illustrated primarily by the case of handling the psychologist's questions), we present examples to discuss the application of this procedure within a more general framework of the system's reasoning.

Dialog IV:

System: Administer an individual intelligence test such as the WISC-R.

Psychologist: Why is the WISC-R being recommended?

Analysis of Problem:

Because the user is a psychologist, the system initially assumes that the user is familiar with the rules regarding diagnostic procedure etc., and hence these need not be supplied as part of the response. The information which is not shared by the user and system is some of the background data regarding the case at hand which has lead the system to believe that the student may qualify for high-cost funding. Therefore this information will be supplied in the response. However, before answering this query, the system needs also to determine whether the response adequately addresses any relevant user preferences. If the user prefers to administer a more familiar test, this response should help to counter the preference as well.

System: There is some weak evidence (such as a family history of learning problems, left-right confusion, slow language development) to indicate that the student may

qualify for "high-cost" funding; therefore, the WISC-R is being recommended because it is appropriate for use within the provincial funding formulae.

User: I would prefer to administer the McCarthy scales. Is this test appropriate?

System: No, only the WISC-R and the Stanford Binet are appropriate for use within the provincial funding formulae.

System Reasoning: The system now realizes that, contrary to its initial stereotype, the user is unfamiliar with the specifics of the provincial funding formulae and hence, marks within its knowledge base that this type of information is unknown to this user. This updated model can then be successfully employed later in the diagnosis. For example, the system now wants the user to administer an individual achievement test. Again the system would like the user to administer a test that can be employed within the funding formulae. Because the system knows that the user is unfamiliar with the these formulae, additional information will be supplied by the system.

System: Administer an individual achievement test. Either the WRAT or PIAT should be used because these can be employed within the provincial funding formulae.

For our second example, recall Example III from Section 3.1 in which the system and user have drawn different conclusions based upon the results of the ITPA.

Dialog V:

System: Administer an individual intelligence test such as the WISC-R or Stanford Binet.

User instead administers the ITPA (Illinois Test of Psycholinguistic Abilities) and the student's performance is recorded.

User: What are the conclusions based on the results of the ITPA? (i.e. What is the diagnosis?)

System: The student's intellectual ability appears to be within the normal range; no further assessment of intellectual ability is required.

User: What other conclusions were drawn? For example, the student's poor scores on the Auditory Association and the Auditory Closure subtests indicate that the student's auditory channel is weak.

System: Although the test results raise some suspicions regarding the student's processing of auditory information, such conclusions cannot be drawn from the ITPA alone because the validity of the ITPA's subtests is questionable.

System Reasoning: After this interaction, the system can conclude that the user does not adhere to certain current philosophies. In particular, the user *does* believe in the ITPA Model and consequently has drawn conclusions which differ from those of the system. The system's model of the user is updated to indicate both the difference in philosophy (adherence to the ITPA model) and candidate hypotheses (the student's auditory processing skills are weak). Then later in the diagnosis this information can be exploited when addressing the user's questions. For example, in the response below, the system supplies additional information regarding the student's auditory processing skills.

Scenario: The student has just completed an informal test of social maturity in which brief stories of moral dilemma are read to the student and he is asked to describe a solution.
User: What is the diagnosis?
System: The student's better than average score indicates the following:
- good social judgement
- good comprehension skills
- average ability to express himself verbally
- no difficulty with material presented via the auditory channel.

The student's performance on this test indicates that there is no reason to advocate further testing of the auditory channel. The student is *not* experiencing any difficulty with processing auditory information.

Recall the three examples of Section 3.1 which motivated the research discussed herein. The framework we have proposed is in fact appropriate for handling these examples. Dialog V demonstrates that our algorithm can be used to handle Example III, and Dialog IIb illustrates Example I. Recall that Example II concerns user preferences. We have addressed the handling of user preferences (as illustrated in Dialog IV), although we have left for future research the full integration of user preferences within our algorithm.

5. Related Work

There has been a wide range of previous work on the general topic of user models, but none of this work focuses on the particular needs of an expert system for educational diagnosis. One domain for which user models have been studied is intelligent tutoring systems (e.g. [SLEE82c]). The suggestion is to construct a model of the student interacting with the system, to better determine the learning problems of the student.

As well, there has been a study of producing effective responses from expert systems, but little accomplished to deal with a range of possible users. Mycin [SHOR76] at first had a very limited explanation capability. [WALL82] tried to accommodate different users by having the user supply his level of expertise. A limited amount of control on the level of detail in the prepared response then resulted. [SWAR83] examined how a more elaborate storage of knowledge (descriptive facts and principles) could produce better responses from medical expert systems.

There has also been work on constructing stereotypes for classes of users, to be employed in fine tuning the actual model for a user (e.g. [RICH79a]). In [RICH*], stereotypes are described as general predictions, not necessarily accurate, drawn for the user until specific knowledge is available. This is essentially our interpretation of the use of stereotypes as well.

Finally, there are the kind of user models constructed for natural language understanding systems which perform pragmatic processing. The work of [PERR78] considers the plan underlying the production of a discourse and the concept of mutual belief to help a listener to comprehend the likely aims of a speaker. [JOSH84a]

emphasize the need for a system to predict the false inferences that a user may be drawing, by carefully studying the goals underlying utterances.

In a sense, the user models we are advocating for our educational expert system draw from each of the above approaches. We essentially allow for the more robust kind of user modeling practiced in natural language pragmatics (tracking the beliefs of system and user, and including goals in the model), but employ default assumptions on users as a starting point and also accommodate correcting misconceptions.

There are a growing number of researchers studying the topic of "user modeling" (e.g. the recent Invitational International workshop on user modeling, Maria Laach, Germany). Our work is quite distinct in its scope and approach, but relates to some of these other current efforts. For instance, the work of [MCCO86] addresses the different perspectives on objects in a knowledge base which a user may have. Our framework suggests different perspectives for whole classes of users (psychologists vs. teachers, for example), not with respect to attributes of an object, but instead with respect to whole segments of the knowledge base which are more likely to be understood to a user.

The research of [FINI86] is also relevant, since this work includes the technique of assigning default values to the knowledge of users, and updating these models as discrepancies arise. We make the case for default models stronger for our particular domain where we have classes of users that can be identified, rather than the more general problem faced by Finin of determining the right classification for a user. As well, Finin offers more specific strategies on correcting the possible assumptions on a user.

The discussion of the distinction between agent models and user models in user modeling [WAHL*] is relevant to our work as well. Our system includes both a student model and a user model, where the student is in effect the subject of the queries to the system by an interested third party.

There are also common concerns with the work of [CHIN86, PARI85a]. Chin suggests a system of double stereotyping, assigning a user a level of expertise and then also labeling all the knowledge of the system according to a "difficulty level" (e.g. complex, simple, etc.). This calls for a segmentation of knowledge according to complexity. We instead advocate a segmentation into knowledge bases which are not labeled with one level for all users, but are determined as known or unknown to each user type. We thus avoid the problem of globally determining the classification of each individual rule.

Paris addresses a different dimension of accommodating different classes of users – varying the form of the response, rather than the content. So, expert users get a "process" description, while naive ones have "parts-oriented" explanations. We instead examine required content changes, and leave aside some of the lower level generation issues. Note that for both Chin and Paris, the user types are on an axis of expert to novice. For our domain, we work with a "user class" identification which is both appropriate for our given task and helpful in isolating the required user specific content in responses.

Some of the differences in our approach to user modeling, outlined above, allows us to advance a general framework for user modeling, summarized in the section below.

6. A General Proposal for User Modeling in Expert Systems

This chapter makes some recommendations for the next version of our expert system, but the principles advocated are intended to be useful as general guidelines for allowing expert systems to vary their responses to users. We summarize the general procedure recommended:

1. There should be defaults for the user types, to initiate a user model.
2. The system should make use of some partitioning of its knowledge bases, to focus on possible differences between user and system knowledge.
3. Differences should be accounted for in responses; so differences may be more usefully recognized by tracking the user's goals and ideally the previous dialog.

In connection with the third point above, it is possible to specify more clearly the potential avenues for disagreement between the system and a particular user class. For example, for the user as psychologist, we have already indicated in the Example III of Section 3.1 that the user may well disagree on the applicability of certain tests in the test database. We are aware that deeper specification of these dimensions of difference is an area for future research.

7. Current Status and Future Work

In our discussion of particular users of the expert system we focused on the psychologist and the parent. A third important class is the teacher, who knows the general purpose of diagnosis but does not know most of the details of specific tests. A teacher is not expected to be aware of the test results for the student, but may know a good deal of other background on the student, especially characteristics exhibited in the classroom environment (e.g. behavioral problems). For future work, we plan to develop a classification of this type of user in more detail.

In fact, we are currently refining our proposed specification of the knowledge bases within the part of the system referred to as "student knowledge". We have identified different sources of knowledge about the student, which directly associate to our user types (home, school and testing procedure), to expand on our current division of the background information into "acquired through testing" and "other". We are also considering the inclusion of both historical background information on the student (e.g. results of previous testing) and current. In this broader framework for the operation of the expert system, the additional divisions discussed above should facilitate user specific responses as well. We are encouraged by the application of our general strategy of subdividing the knowledge bases to identify unknown knowledge to classes of users, as demonstrated in our simulations of the system in the examples.

It is also worthwhile to comment on some current implementations into which we may integrate our proposal for variation of responses to users. The CGD project now has a useful small prototype system for educational diagnosis of students' errors

in arithmetic [TUBM86]. The basis for reasoning in this system is THEORIST, an approach of formulating theories about the possible diagnosis, using rules of default reasoning. So, for example:

default (problem (addition-of-wholes), Student, EWRPT-A test)
if:
is-in-grade (Student,Grade)
ge(Grade,2); le(Grade,7)
has-taken-test (Student,EWRPT-A test)
error-in-ewrpt-a-questions-1-or-2

This integrates well with providing a response to a user as follows:

User: Why is the hypothesis trouble with addition of wholes?
Response: (to teacher) Because he had an error in questions 1 or 2 of the EWRPT-A test.

In addition, work for employing goal information in the formulation of responses has been implemented [VANB86b], and a group is currently studying the integration of goals with the user's knowledge of the domain and student, for the selection of the response.

8. Conclusions

Expert systems for educational diagnosis require user models, to provide variability in the form and content of explanations generated for the users. This chapter describes a framework for incorporating user models into expert systems, which is specifically designed to

(1) identify and separate the knowledge bases which must be consulted in the preparation of an explanation
(2) allow for default assumptions of user's background knowledge in the absence of further information on the user, and
(3) continuously maintain and refine the user models, both as dialog from the user reveals knowledge and as previous output from the system adds to the shared knowledge between system and user.

In short, there is a prescription for the design of expert systems to incorporate models in order to serve a broad base of users. (There is as well a discussion of designing schemas for user classes, to guide the generation of responses). Moreover, there is a detailed argument for why this framework is particularly effective in the area of diagnosing learning disabilities. In sum, we have the means for implementing a more powerful version of our current system and a general framework of use to designers of other systems, to focus on beliefs of users in producing quality responses.

Acknowledgements. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

9. References

[ADEL82], [BUCH84], [CHIN86], [COHE85], [COLB82], [COLB83], [COLB84], [COMM71], [FINI86], [GRIC75], [JONE86], [JONE87], [JOSH84a], [LEVE84], [MCCO86], [MCKE85b], [MCLE82], [MCLE85], [MCLE86], [PARI85a], [PERR78], [POOL86], [RICH79a], [RICH*], [SHOR76], [SLEE82c], [SWAR83], [TUBM86], [VANB86a], [VANB86b], [WALL82], [WAHL*]

Part IV

Shortcomings of Current Models, Prospects for the Future

Introduction

The aim of Part IV is to point out a number of shortcomings that apply to the user modeling components developed to date, and to outline the research directions which should be given greatest priority in order to obtain a better understanding of the function of a user model in human-computer interaction, and eventually to build better models.

In this vein, **Karen Sparck Jones** identifies three major aspects which bear on user modeling: The characteristics of the person(s) involved in the interaction with the dialog system, the function of the model(s) of this/these person(s) in the system's dialog planning, and the means by which the information for the models can be obtained. The first two aspects can themselves be classified along several dimensions. With respect to the persons involved, Sparck Jones distinguishes between a person who interacts directly with the system (the so-called *agent**), and a *patient* who is the subject of the system's decision-making process.[1] In many cases, both roles will coincide with the same person. However, if a separate agent* exists who mediates information about the patient (e.g. a social insurance clerk consulting an expert system on behalf of an applicant), a model of both persons must be constructed, Sparck Jones argues, since assumptions about the agent*'s (Sparck Jones) properties (e.g. his/her level of experience, his/her prejudices, etc.) may also influence the system's dialog behavior (see Cohen & Jones' [COHE*] educational advisory system).

Other distinctions drawn by Sparck Jones with respect to properties of both agents* and patients are those between objective vs. subjective (mental) properties, dynamic vs. static properties (which do and do not change over time, respectively) and between decision vs. non-decision properties (which do and do not directly influence the system's primary task decisions, respectively). Sparck Jones claims that systems should, both for the agent* and the patient, in fact possess different models – one for each relevant type of property, i.e. up to 8 altogether – since they differ in the degree of scrutiny necessary for accepting them, in the treatment of inconsistencies, etc.

The second important aspect that bears on user modeling is the function of agent* and patient models in the dialog planning of the system. Agent models (necessarily non-decision ones) are relevant to the acquisition of information about the

[1] Sparck Jones' notion of 'agent' is more specific than that used by the other contributors (see e.g. [KASS*, MORI*, WAHL*]), hence the discriminating '*'. Morik's [MORI*] notion of 'protagonist' is closely related to Sparck Jones' agent*.

patient as well as to the agent's own interaction with and reaction to the system. Patient models can support the effectiveness, efficiency and acceptability of systems, i.e. contribute to the correctness of system decisions, reduce the time necessary for finding them, and improve their presentation to agent* and patient. By definition, non-decision properties of patients do not directly influence system decisions about him/her. When modeled, however, they may sometimes influence its effectiveness or acceptability. In some cases, they may even indirectly contribute to a judgement of the plausibility of assumed decision properties, or help to judge the credibility and expertise of the patient and thus the correctness of the information supplied by him/her, etc. The question arises now as to the extent to which non-decision properties (not only of the patient, but also of the agent*) should be sought and modeled by the system. Sparck Jones argues that these properties are too loosely connected and the implications yielded mostly too weak to justify taking great pains to determine and model them. Beyond decision property modeling for effectiveness, user modeling should therefore be very conservative, i.e. should not be driven too deep for content or stretched too far in use.

Katharina Morik's central thesis is that a number of research issues have not yet been sufficiently explored in the field of user modeling, particularly concerning the acquisition and revision of user models and – most of all – the conversational settings in which dialog systems equipped with a user model have been employed.

After some preparatory clarifications of what constitutes user models, agent models and dialog memories, the author identifies three factors that characterize conversational settings: the kinds of things that can be affected by system actions, the types of actions that can be employed for this purpose, and whether or not the user him-/herself plays a role in the domain of discourse. System actions may affect the user's domain model, the system's domain model, or the system's user model. Action types which can achieve these effects include communicative actions, observation of the domain of discourse, and "physical" (i.e. non-mental, non-communicative) actions in the domain. Action types may have more than one effect, and intended effects can often be achieved by more than one action.

Morik illustrates these factors by analyzing their influence in three types of conversational settings (namely information-seeking dialogs, reporting, and action-achieving dialogs), and by describing how the HAM-ANS system behaved in these settings. She claims that research on user modeling has so far concentrated mostly on information-seeking, and neglected action-achieving, dialogs (which, in addition to communication actions, also include "physical" actions). In this latter setting, a much wider variety of speech acts would occur both on the part of the user and of the system, and a much greater range of possible reactions to the user's input would be at the disposal of the system. Secondly, moreover, Morik claims that current systems rely too much on user model acquisition which is separate from, and prior to, the "normal" dialog in which, for example, the achievement of the user's goals is attempted. Instead, these processes should be much more intertwined. And thirdly, the revision processes for the user models investigated and implemented to date are claimed to be by far too simplistic, since they consist mainly of additions, but exclude replacements, modifications, dependency-directed backtracking, and the

recording of changes in user properties. To substantiate her theses, Morik reviews several existing user models and analyzes the extent to which these issues have been taken into account.

In a final section, Morik surveys the kind of user modeling that is performed by the HAM-ANS system in its role as a hotel manager. From facts entered about the prospective hotel guest, HAM-ANS makes assumptions about criteria along which the customer is presumed to evaluate rooms being offered and takes these assumptions into account in its responses to the user. A reasoning method based on the text-linguistic concept of *isotopy* (semantic cohesion) is presented as one step in this inference process.

Though user modeling and student modeling are related in many respects, they have generally been pursued separately for the past decade. The aim of **Robert Kass'** contribution is to explore the role of student modeling in intelligent tutoring systems as seen from the point of view of user modeling, and to find out what lessons can be learned for the field of user modeling. This is all the more justified since the area of student modeling has a somewhat longer tradition than user modeling.

Kass first differentiates intelligent tutoring systems (ITS's) from other types of computer-aided instruction systems and outlines their basic architecture. He then presents the three forms of relationships between the student model and the system's model about the domain of discourse (or "expert model") that have been investigated to date, and discusses their implications for the representation and acquisition of student models. Seven intelligent tutoring systems are surveyed in detail, and their contributions and limitations are discussed.

Against this background, the author discusses a number of research issues in student modeling that might also be relevant for user modeling, particularly for handling user misconceptions. His proposals include:

- The ITS methods for explaining why students performed a task in a particular manner (which usually require a deep, "psychological" student model) may be exploited for detecting and recognizing user misconceptions, e.g. in advisory systems (see [MCCO*, QUIL*]).
- The three above-mentioned forms of relationships between student and expert models are not limited to ITS's only, but are also appropriate for user modeling in applications where a body of domain knowledge is the main focus of interaction.
- Although systems containing a user modeling component generally cover a much wider range of interaction with the user than do intelligent tutoring systems (see [MORI*]), the admissible range is sometimes strongly restricted (e.g. when user misconceptions have been recognized). Methods for driving the interaction by means of questions or problems to which the student must respond have been developed in ITS research, the application of which may be considered in such cases.
- Attempts are being made in ITS research to isolate a small number of very general principles which can account for students' errors in a domain (the goal being domain-independent principles). Kass expects that similar rules could be used to

explain the ways in which users' beliefs, goals or plans are likely to differ from those in the domain model.
- Unlike user modeling, student modeling has already addressed the problems imposed by occasional non-regular user errors (the "noise problem"), although no general method has as yet been achieved there either.

Tim Finin is concerned with the development of a general tool system for user modeling. As this volume demonstrates, the development of a full-scale user modeling component is rather expensive when one has to build it right from scratch. Two research strategies have therefore been pursued to alleviate this problem:

1. to restrict user models in their scopes and aims (for instance, only some of the beliefs and goals listed in [KOBS*] are modeled, or the user model is exploited for specific purposes only);
2. to develop application-independent tool systems (so-called *user modeling shells*) which are intended to form the backbone of user modeling components in application systems. These shell systems would, for example, comprise a representation system for storing assumptions about the user, an inference system for deducing the consequences of current assumptions, a reason maintenance system for detecting inconsistencies, etc.

Tim Finin's chapter is concerned with the second approach (which is rather novel – the only other system with similar aims seems to be BGP-MS, briefly outlined in KOBS88b). After a short survey of the possible kinds, contents, acquisition and maintenance methods and application domains of user models, Finin classifies his General User Modeling System (GUMS) along these dimensions: GUMS allows for the representation of any kind of assumption about the knowledge (beliefs) of stereotypical and individual users, as long as they can be expressed in the PROLOG-like language on which the system is based (certain predicates have a system-defined interpretation). The acquisition of assumptions about the user from the ongoing dialog is left to the application system. The emphasis of the GUMS system lies on elaborate methods of default reasoning over the assumptions which are directly or indirectly supplied by the application system. No restriction whatsoever is made with respect to the uses to which the assumptions contained in the user model can be put; this is also left to the application system.

GUMS supports three kinds of default reasoning, which are described in detail in Finin's chapter: stereotype handling, explicit default rules, and negation as failure. These types of reasoning differ with respect to increasing granularity and decreasing strength of conclusion: stereotypes are used to capture generalizations of large classes of users. Within a stereotype, explicit default rules are used to express the stereotypic norms which are allowed to vary across individuals. Negation as failure is used as a general technique for gathering weak evidence for beliefs about a user when no stronger evidence exists.

Finin's approach seems to be a promising start in a direction which has been strongly pursued in other fields of artificial intelligence research in the last few years, namely the development of flexible and portable tool systems which the designer of

application systems can fill with application-dependent knowledge. Expert system shells, grammar workbenches, constraint propagation systems and other tools have proven to be very valuable for the development of better application systems. The same will probably also be true for general user modeling shells.

12 Realism About User Modeling

Karen Sparck Jones

1. Introduction

The object of this chapter is to explore the issues raised by user modeling from an analytic point of view. It is generally assumed that systems cooperating with human beings, and especially systems with natural language interfaces, need, and can get, user models. There is also a widespread assumption that the more user model the better and, correspondingly that, while we may not yet see quite how to do it, there is no problem in principle about being able to construct ever richer models.

The user models of current working systems, with some specialised exceptions like teaching systems, are very simple. This chapter does not describe any implementation. It considers examples that are much more complex than those that could be supported by the present state of the art, in order to lay out the elements and implications of modeling and so provide a framework for future work on actual implementations. This framework distinguishes and characterizes the major factors in modeling, namely the nature of the information in a model, the function of a model, and the means by which the information for a model can be obtained. In any specific modeling enterprise it is necessary to be clear about the relevant values of these application parameters.

The direct conclusion to be drawn from the chapter's analysis is a pessimistic one: even assuming very powerful system resources, far beyond the scope of today's systems, there are generally early limits to the modeling that can be achieved. Thus though the examples illustrating the argument imply modeling aims far more ambitious than those on which current system building is based, the conclusion is relevant to the long term research goals of the field.

But there is also a more positive and constructive message in the chapter. This is that the useful, if limited, modeling it is reasonable to seek would be promoted by adopting the framework the chapter presents. The distinctions it draws between the main components of modeling are important and should be applied.

This chapter attempts a comprehensive treatment of modeling considering first, what is being modeled, second, what it is being modeled for, and third, from what the modeling information may be obtained. The implications to be drawn from this discussion, and the relations between the account of user modeling given here and those presented elsewhere, are examined in the fourth, final, section on rational

principles for user modeling. The points made are illustrated through extensive examples. These examples, which are designed to show what is involved in modeling when it is taken seriously, are necessarily drawn from hypothetical systems. The chapter further takes specific modeling mechanisms for granted and is not concerned with what the technical procedures involved might be. The focus is on such questions as the aims and value of modeling, taking it for granted that it can, in a strictly technical sense, be done: thus I am assuming, for example, that some piece of information can be extracted from some linguistic expression, that a suitable data structure representing this and the other information constituting a model can be created and manipulated, and that the results of this processing can in turn drive further system operations.

For the purposes of the chapter I am also making some further assumptions. First, somewhat obviously, that I am modeling a human user. Second, I assume that the object is to model users as individuals, not simply to assign them to previous, explicitly defined classes. I am further assuming modeling in the context of complex computational systems of the kind represented by substantial expert systems. That is to say, it is taken for granted that the system may, for its primary purpose, exploit large bodies of knowledge and engage in extensive reasoning, with no direct connection with user modeling. I am also assuming, naturally, that there is some means of interaction between user and system: the implications of natural language interaction will be examined in Section 4 of the chapter. In addition, I assume throughout that we are concerned with the modeling that is done in a single interactive session, taking longer-term modeling as dependent on, and not essentially different from, this.

Finally I am assuming, without attempting to define rigorously, some notion of what a model is, as constituting something more than a simple aggregate of pieces of data. Individual items of information, or a collection of these, do not constitute a model. The presumption in talking of user models is that items of information about a user may be related to one another, or to other (typically general) knowledge stored in the system, in a manner which supports predictions that can stimulate further system actions. This is not necessarily a very strong definition, since much depends on the nature and extent of the relationships implicitly or explicitly holding between pieces of information. For example in a medical system there may be a tacit assumption of causal relations between symptoms. The kind of categorization applied in GRUNDY [RICH79b] to predict book choices, on the other hand, does not make any strong assumptions about causal relations between the different user characteristics.

2. What Is Being Modeled

This section considers aspects of users bearing on modeling, or modeling *factors*, drawing a number of distinctions relative to these factors. Some of the distinctions may seem over-fine. The justification for them is to be found in Section 3, where the uses to which models may be put are examined.

2.1. User Roles

It is necessary, first, to distinguish the different roles users can have. In a given session, we may have different users, with distinct roles, or just one user with different roles. The two roles I distinguish are the *patient* role and the *agent* role. I define the patient as the subject of the system's decision-making processes, confining the system's decision-making processes here to those the system was primarily constructed to carry out. I assume the system was designed to reach decisions in a specific domain, in the sense in which classical expert systems have been designed to reach decisions about the diseases people have, the financial investments that might be appropriate for individuals in given circumstances, and so forth. The domain knowledge is instantiated in relation to the user, to reach a domain conclusion about him. (This therefore views personal financial decisions as ones about a human money owner and not as ones relative to abstract financial situations.) This is not of course to imply that a system reaches a decision only at the end of a session: it may take decisions continuously throughout a session, as in a teaching system.

As will be evident, a system can make decisions, in the general sense, about other matters. But I am specifically restricting "decision-making" here to activities aimed at satisfying the system's primary task goal. In the medical case "patient" has an obvious interpretation, but I am using it as a role label in a more general sense.

The agent, by contrast, is the person who conducts the interaction with the system. The agent, as such, is thus not the subject of the system's decision-making processes.

The implications of this distinction between patient and agent will become apparent as the other components of modeling are considered. Here it is necessary to point out that different person and role situations can arise as shown in Figure 1. Thus for a given system we may have a single human whose only role is as patient, as with a surveillance system where the interaction between the system and the external world, and hence with the patient, is through automated monitoring. Or we may have a situation where the single human being is only the agent, as in industrial design, where the system's decisions are about inanimate machine structures. We may have two human beings, one as patient and one as agent, so the latter is acting as the medium for information about the patient, as in the medical case with a nurse as system driver. Finally, we may have a system with a single human who is both patient and agent, as in a home computer diet advice system.

It is clear that both patients and agents, as humans, can have beliefs, goals, plans, etc. All users, whether patients or agents, are thus agents in the general sense of purposive beings. However I shall specifically reserve "agent" in this chapter for the role definition I have just given, taking it for granted that any user can have beliefs, goals etc; (so if it is necessary to refer to the general meaning of "agent", this will be clearly marked). My use of "agent" is therefore more particular than that generally adopted in discussions of user modeling. My distinction between agent and patient is indeed intended to allow an analysis of aspects of modeling not covered by the normal unitary view of the user (though Morik [MORI*] makes a similar distinction between 'user' and 'protagonist').

I shall use "user" where the distinction between patient and agent is immaterial because the points made apply to either role. This makes "user model" comparable

1. PATIENT ONLY
   e.g. automatic surveillance of a human being
2. AGENT ONLY
   e.g. human operater of an industrial design system
3. PATIENT AND AGENT, DIFFERENT
   e.g. medical diagnosis system with nurse agent and sick person patient
4. PATIENT AND AGENT, SAME
   e.g. diet advice system with customer agent and patient

Figure 1. User role possibilities

to "agent model" as the latter is generally understood, but here I am using "agent model" more restrictively to refer specifically to a model of a human in the agent as opposed to patient role.

2.2. User Properties

It is further necessary to distinguish types of user property. Some of the distinctions to be made here are not theoretically watertight, but they are operationally important because they are reflected in material differences in the reliability and accessibility of modeling information.

The first distinction, see Figure 2a, is between *objective* and *subjective* (or mental) properties, for example between clinical thermometer readings and sensations of worry. (Note that subjective properties are defined as subjective in the possessor, not the observer.) This is an intuitive, but real enough, distinction. In general we behave, and therefore a system could similarly be expected to behave, as if it is easier to establish, for example, that a user is female than that they believe democracy is under threat. (There is no space here for the more detailed discussion this distinction requires: for example what is the status of reported properties, as in "I felt terrible pain"? A good case can nevertheless be made for the basic distinction as one with consequences that have to be taken into account in modeling. For some further points see [SPAR84].)

A second distinction, cutting across the first, and applying most obviously but not exclusively to subjective properties, is that between *static* and *dynamic* properties, for example between political beliefs that hold throughout the session and preferences for dresses that change in interaction with a mail order system. This is not to imply that political beliefs cannot change. It is rather that, in relation to a system's purposes and activities, there are properties which the system can treat as static because they are, if not necessarily static, at least relatively permanent and are unlikely to be changed by the system's activities, which are not directed towards decisions concerning them.

The distinction between static and dynamic does not refer to the effect of time on the *system's* knowledge of a user property, where a system may suppose X until

OBJECTIVE / SUBJECTIVE
e.g. temperature reading / feeling worried
STATIC / DYNAMIC (for session)
e.g. political belief / dress purchase preference
DECISION / NON-DECISION
e.g. temperature reading / age ; dress purchase preference / political belief

(a) Property type distinctions

OBJECTIVE	P / A	salary
SUBJECTIVE	P / A	anxious
STATIC	P / A	sex
DYNAMIC	P / A	belief about relative costs of brick and concrete
DECISION	P	foreign language words known
NON-DECISION	P / A	age

(b) Property role possibilities. P = patient, A = agent, assumed different people

Figure 2. Property types

the passage of time brings evidence that not-X, but the user is throughout in state not-X. It refers to changes in the *user* over time, i.e. from state X to state not-X. In practice it may be difficult for a system to know whether its knowledge or the user's state is changed, and in some cases the consequences may be the same. But I have introduced the distinction to allow for the situation where changes in the user are important because modeling is intended not merely to recognise them but (possibly) to effect them. Clearly, though, there need be no close correspondence between when a change in a user is observed by the system and when it is experienced by the user. It is important to emphasize that objective properties can change as well as subjective ones, though this may be less common: for example instructions to sell financial stocks.

There is, finally, a third property distinction which also cuts across the others. This distinction, unlike the others, is an absolute one because it is explicitly defined in the construction of the system for its primary task; and as it is fundamental to the system's purpose, it is also much more important than the other two. This distinction is one between *decision* and *non-decision* properties. As noted earlier, an expert or other complex system is intended to take decisions in a certain domain. Those user properties that pertain to this domain are a user's decision properties. According to the design purpose of the system these may be, for example, medical symptoms, or preferences for sorts of clothes. They can clearly be a mixture of objective and subjective, or of static and dynamic, as illustrated in Figure 2a, which shows a static objective property, temperature reading, and a dynamic subjective property, sartorial preference, as possible decision properties. Whether a property

is a decision one or not is wholly contingent, as it depends on the specific system application. Any particular user property, e.g. political belief, may be a decision property in one system (e.g. a voting advisor) and a non-decision one for another (e.g. an arithmetic tutor).

If the system uses both non-decision properties and decision ones, the distinction between them may appear unnecessary. But it may nevertheless be important operationally. Property information about users can be exploited for different purposes, described in the next section. Clearly, that for which decision property information is sought is the most important. The permanent system model of the domain phenomena to which these properties refer may therefore be much more fully developed and characterized than that of other entities or activities; and it may also be the case that the importance of decision property information may mean that it has to be handled more carefully, for example be more rigorously tested. There are therefore good practical reasons for distinguishing these two types of property, as will be more clear from the examples considered later, though there may be no intrinsic difference in the nature of the properties labeled decision and non-decision for a given system.

Though for precision it might be desirable, in relation to to the three distinctions just drawn, to refer respectively to, say, the type, mode and status of a property, I shall regard it as sufficient to refer simply to the (six) types of property, as listed in Figure 2a.

It will be evident that in general these distinctions of type are relevant to both patient and agent roles. Thus as Figure 2b shows, the same property, like salary, or age, can hold of patient or agent, indeed hold of both where there are two people. However there are some restrictions. Thus by definition decision properties can only be properties of the patient: they do not apply to the agent role.

3. What Modeling Is For

The previous section characterized the nature of the inputs about users with which a system could be concerned. The next question is why it should be so concerned. Specifically, if the function of a system is to take decisions, why should the agent role be considered? Why, indeed, should non-decision properties even of the patient be considered?

Both roles, and both decision and non-decision properties, are relevant because there are quite different system functions user models can serve, so non-decision properties of either patient or agent can be helpful in supporting or enhancing system performance. The possible functions models can serve are illustrated in Figure 3, for a medical system with distinct patient and agent. Thus the first, and most important function models can serve is system *effectiveness*. The prime object of a user model is to ensure that the system, as a decision-making system, reaches the correct decision. For example, if we have a medical diagnosis system, we need a model of the patient which is adequate to support what, if independently evaluated, would be shown to be the correct decision (assuming that, given adequate evidence, the system's processing is itself sensible.) This clearly implies that the only relevant

model for effectiveness is a patient model. Effectiveness is clearly also a necessary system function.

However, effectiveness is not the only system function user models can serve. Models can serve system *efficiency*, i.e. reaching the correct decision in an economical way. A patient model is obviously relevant here too: for example, the available knowledge about a patient may be used to order the tests to be made on the patient for further information so that the most potentially useful tests in the specific circumstances are carried out first. Moreover, if the agent is seen as a transmitter of information about the patient, the system can also exploit an agent model to gather information in one way rather than another, as likely to be more efficient. For example, if the agent is medically experienced, this fact can be used to promote efficiency through quick answers because well-defined medical terms can be used in questions the system puts to the agent.

There is also a third function a user model can serve. This is system *acceptability*, i.e. supporting (or expressing) its decision making in a comprehensible, perhaps also agreeable, way. (This refers primarily to the system's operations in a session, not to the extent to which its performance over many sessions carries conviction about the validity of its decisions.) Here again both agent and patient models can be used, for instance the fact that the patient is in pain can be exploited to emphasize one aspect of the proposed regime (as likely to reduce pain) rather than another (which may be nasty), and the fact that the agent is experienced to augment the decision information with literature references to comparable case histories. Acceptability, like efficiency, is not a necessary function; but both may be very important in practice.

EFFECTIVENESS	
e.g. disease diagnosis	P
EFFICIENCY	
e.g. diagnostic test ordering	P
terminology of system questions	A
ACCEPTABILITY	
e.g. medical regime presentation	P
case citation	A

Figure 3. System functions and relevant models. A medical diagnosis system. P = patient model, A = agent model

As the examples of Figure 3 show, the different role models may be used in different ways to serve distinct system functions. But this general statement disguises some relevant complexities. Thus in relating models to functions it is necessary to separate the source of the modeling information from its functional destination. In the example of Figure 3, *source* and *destination* are in fact the same, e.g. information about the agent is applied to the agent, as in the exploitation of his experience to use technical terminology. But patient data could also, for example, be applied to

system functions directed at the agent (i.e. in his own right, not as an information transmitter), for example to explain the diagnosis.

There are indeed further points to note, which also have implications for the detailed way in which models are exploited. In using models to serve functions, it is necessary to separate (logical) *addressee* from *beneficiary* (as well as actual from logical addressee: the agent may be actually addressed to transmit a question to the patient). Thus the agent may be addressed, using an agent model, to obtain information exploited for the benefit of the patient, or for the agent's own benefit; or the patient addressed to obtain information which may also, as just suggested, be exploited for the benefit of the agent. One may also have first and second order beneficiaries, so for example, benefitting the agent indirectly benefits the patient. In other words, we may have both patient and agent models serving system functions themselves directed towards both patient and agent. Either patient and agent model may naturally, moreover, be exploited differently both to gain inputs, and to direct outputs, in supporting the system's various functions.

It must further be emphasized, as has been implied and will be more fully illustrated below, that both decision and non-decision properties can be of value in relation to the system's functions. In the example of Figure 3 this is shown most particularly by the use of the agent model in connection with efficiency, since the agent's properties are necessarily non-decision ones; but if we assume, for example, that pain is not a decision property for the disease in question, a patient non-decision property is being used for acceptability (to the patient).

The set of distinctions drawn in this section seems very elaborate. But the distinctions and their implications are relevant if we want to serve system purposes by modeling, given that modeling situations range from those where it is necessary to recognize user properties to those where it is only desirable, and from those where models are used to promote positive system behaviour to those where we want to inhibit negative behaviour. This discussion of course assumes, in general, a system of sufficient complexity to suggest a need for modeling beyond that involved in constructing and manipulating the basic patient decision model. But we still have to relate the validity of the modeling that can in practice be done to its actual utility.

Example 1

The need, assuming modeling is feasible, to recognize all the distinctions discussed, relating functions and roles and also property types, can be illustrated by a fuller system example, summarized in Figure 4. Suppose we have a (notional) social security expert system with distinct patient and agent. The patient is an elderly and disabled woman who has to use a wheelchair; she also has poor eyesight, is a Catholic, honest but suspicious of officialdom, poorly informed about benefits, and believes that age is the main determiner of benefit (these are illustrative properties, and are not intended to constitute a complete patient description). The agent is an experienced clerk, but one who is a male chauvinist who believes women are unduly favored by the benefits system. Figure 4a lists the patient's properties by type. For clarity, but in some sense artificially, user properties of different types are treated here as pertaining to separate models. Thus for the patient we have

PATIENT

		Decision	Non-decision
P1	Static, Objective	aged, disabled, female	
P2	Static, Objective		poor eyesight, Catholic
P3	Static, Subjective		
P4	Static, Subjective		honest, suspicious
P5	Dynamic, Objective		
P6	Dynamic, Objective		
P7	Dynamic, Subjective		
P8	Dynamic, Subjective		poorly informed, age matters

(a) Patient properties

P1 **Static, Objective, Decision**
disabled → EFFECTIVENESS e.g. determinant of benefit
→ EFFICIENCY e.g. drives search establishing benefit
→ ACCEPTABILITY e.g. referred to in explanation of benefit

P8 **Dynamic, Subjective, Non-decision**
age matters → EFFICIENCY e.g. guides data gathering
→ ACCEPTABILITY e.g. referred to in explanation of benefit

(b) Exploitation of patient properties for system functions

AGENT

A2 **Static, Objective, Non-decision**
experienced → EFFICIENCY e.g. data gathering
→ ACCEPTABILITY e.g. practice manual references

male ...

A4 **Static, Subjective, Non-decision**
chauvinist → EFFICIENCY e.g. data checking for bias
→ ACCEPTABILITY e.g. explanation of benefit basis

(c) Agent properties and uses

Figure 4. User models and their uses. A social security benefits system (patient and agent distinct)

a P1 model referring to her static, objective, decision properties, in this case age etc., another model P2 referring to her static, objective, non-decision properties, poor vision etc., in fact eight models altogether, though not all are instantiated

in the example. Agent models are correspondingly numbered, but as an agent cannot have decision properties, the only relevant models are the even-numbered ones.

The separation of models for a single role is somewhat artificial, but does serve to mark differences in the relative importance or stability of the information involved which, even in a single union model, could be reflected in the way information in the various classes was managed and applied. Decision property information may be more carefully vetted, for example, and inconsistencies require different treatment in static and dynamic cases, since static property clashes may have to be resolved where dynamic inconsistencies may legitimately reflect changes over time.

The kind of uses to which the example modeling information might be put are illustrated in Figure 4b and c. The patient's disability is directly used to reach a decision about the amount of benefit due, but may also be exploited in the interests of efficiency to organize the search for possible benefits (I assume a suitable rule structure allowing this). It may also be used for acceptability to the patient to explain the makeup of the benefit payment, for example to indicate that disability precludes some other apparently relevant factors like sex. The patient's dynamic belief that age is the determiner of benefit could be used to support data gathering efficiently through the forms of questions, or might be corrected as part of an explanation for the nature of the questions being asked.

Turning to the agent's illustrative properties, his experience could be exploited for efficient data gathering and to make the decision reached acceptable by reference to the relevant sections of the regulations. His unfortunate chauvinism, on the other hand, might have to be counteracted, for efficiency, by careful question formulation in data seeking and, for acceptability, by an indication that sex is not the basis of the calculated benefit. (Notice the need to maintain a broad definition of acceptability, and the need for the system to discourage as well as encourage user behaviour.)

This example is developed in more detail in [SPAR86], and is compared there and in [SPAR84] with other examples illustrating systems with different purposes and other patient and agent role combinations and property descriptions. The various examples taken together are designed to provide a fuller picture of the relevance and utility of the factor (i.e. role and property type) and function distinctions I have made. The situation would be quite different, for example, in a system with the rather different factor combination represented by a learning system with patient as agent and decision properties that were all or mainly dynamic. The social security system would of course also be very different with different properties or property values.

These different examples, however, simply reinforce the important general points made here by the social security illustration. These are that when one looks at systems as wholes and not only at interactive interfaces, and at systems engaged in non-trivial tasks, it is necessary to allow on the one hand for distinct user roles and property types, and on the other for distinct system functions. These distinctions then imply multiple relationships between pieces of information about the user and system purposes. Thus an individual user property may be exploited not only to support different functions; it may be used in different ways to serve a single function. A particular function may, conversely, exploit not merely different

properties of the user, but different types of property. It is particularly important to recognize the need, in more complex systems and system environments, for the distinction between decision and non-decision properties and the value of nondecision information in supporting the efficiency and acceptability functions which, if they are not primary in the way effectiveness is, are nevertheless highly significant.

Thus if we consider a single user property, namely sex, this may be a decision property in some systems. But in other systems where it is not a decision property, it can still be very useful. For example (and making some arbitrary medical assumptions), if sex is not a decision property for a smallpox diagnosis system, it might still be exploited for efficiency to order diagnostic tests (supposedly justifiable by possible pregnancy implications), and/or it might be used in relation to acceptability (to the patient) to couch verdicts in different terms (assuming women are more sensitive to potential disfigurement than men); sex could indeed be used in more than one way in relation to acceptability, for instance to draw attention to the implication of verdicts for contact with children (generally rather different for men and women). Of course, in choosing information to present the system is making a decision in the ordinary sense, though not in the strict sense defined earlier: properties which are not decision properties for the central task become decision properties for the subsidiary response generation task. The distinction between central decision properties and non-decision properties is nevertheless a useful one.

		PATIENT	AGENT
EFFECTIVENESS			
Decision	(Static / Dynamic) (Objective / Subjective)	X	
Non-decision	(Static / Dynamic) (Objective / Subjective)	(X)	(X)
EFFICIENCY			
Decision	(Static / Dynamic) (Objective / Subjective)	X	
Non-decision	(Static / Dynamic) (Objective / Subjective)	X	X
ACCEPTABILITY			
Decision	(Static / Dynamic) (Objective / Subjective)	X	
Non-decision	(Static / Dynamic) (Objective / Subjective)	X	X

Figure 5. Factor-function relation possibilities

The rich pattern of relationships between kinds of modeling information and their functional uses that follows from going outside the decision core of a system is summarized in Figure 5. This applies the convention introduced earlier and isolates each factor combination as a submodel, so looking at the different functions models can serve gives us the distribution shown in Figure 5.

As Figure 5 indicates, the contribution of some models to some functions is indirect rather than direct. The agent or patient's non-decision properties can bear

on system effectiveness only indirectly through, for example, the transmission of incorrect decision information. It is of course the case that quite apart from the general question of whether a system can in a straightforward sense obtain reliable information (it may simply not be clear, for instance, how severe a sick person's pain is), transmitting patient information through an independent agent can lead to all kinds of distortions (the doctor may think the patient is malingering, for example).

As a further potential complexity, it should also be noted that it is possible that each submodel may really be a set of models, either because different and incompatible interpretations may be made of the available evidence, or because the different uses made of the same information impose different types of structural organization on it. This sort of complexity arises, moreover, even when the factor distinctions made earlier are suppressed so all the information about one human is amalgamated in a single model. (This of course assumes the system can recognize, if it is not set up to assume, when there are two people as opposed to one.)

These points are illustrated more fully in Example 2, presented in Section 4 below.

4. What Modeling Is From

I have not so far considered the means of communication between the user (strictly agent) and the system. If we can see the point of modeling, there is still the question of how easy it is to do. This clearly depends essentially on whether the user information needed for modeling can actually be obtained, and this in turn depends on the way user and system communicate.

The context in which this chapter is being written assumes that user and system (and especially the former) communicate in natural language, and ideally in full and free natural language. Natural language is of course not the only possible means of communication. Setting aside situations where user, and specifically patient, communication is like the autonomous and involuntary monitoring of the VM expert system [BUCH84], where the patient is not a decision-making agent in the ordinary sense of "agent", there may be applications where the user may communicate, as a purposive agent, with the system through nonlinguistic means. This could be the case, for instance, in a design system where drawing was the means of communication, or in a mathematics teaching system where arithmetic terms and expressions are used. Formal language, and quasi- and restricted natural language are also obvious possibilities, which may be implemented to allow more or less initiative and direction to the user: with the sort of restricted natural language interface often found with menus, for instance, the user's freedom is typically very limited.

It is important to recognize that these other communication means can perfectly well support user modeling (as further discussed in [SPAR84]): intelligent computer-aided instruction in school mathematics is an obvious example, where the instruction is specifically driven by a model derived from the user's problem-solving behaviour. Restricted communication, e.g. through menus, may also be perfectly

satisfactory for a particular application. I shall nevertheless concentrate on interaction through unrestricted natural language (assuming therefore a natural language processing component in the system which is far beyond out present capabilities), primarily because natural language offers the richest communicative resources and so the best opportunity to gather information indirectly if, for good reasons, it is not supplied directly. In language use information may be conveyed both directly, as in the answer to a question, and indirectly, as in the form of words chosen. As in human dialog, it may not be feasible to gather all the modeling information directly, either because the necessary explicit interchange becomes too tedious, and therefore oppressive, or because it may be too offensive to the user. Natural language concentrates many different items of information simultaneously in its expressions through the concurrent deployment of, for example, lexical choice, syntactic structure and form of reference, so natural language interaction offers the best chance of maximizing user information. This is independent of the fact that natural language dialog may be required or appropriate for other reasons, for example speech in telephone interaction.

It is also important to emphasize here that exploiting already-interpreted natural language communications to gain modeling information is logically distinct from having to have a model to interpret these language communications in the first place. Thus modeling may be logically parasitic on natural language communication adopted as an effective means of obtaining the system's decision-making information, as is most clearly shown where there is no human patient, but also holds for models other than decision-property ones where there is a human patient. Or it may be deemed to be a prerequisite for the utterance interpretation that supplies that decision-making (or other functionally-useful) information. There may in any case be situations, for example where the human patient is also the agent, where this distinction seems immaterial. The point of making it here is nevertheless to draw attention to the fact that in this chapter the relation between language and model is being approached from one end rather than the other, so the concern is with what the linguistic expression contributes to the user model, and not with what the model tells us about the expression. This is not unreasonable, given the varied modeling purposes considered here, since, though some model elements like subjective goals will clearly work both ways and play a part in both enterprises, others may have little part to play in assisting linguistic interpretation.

Example 2

The problem of extracting useful modeling information can be illustrated by an extended example. This is a hypothetical dialog, representing the first part of an interaction, for the social security benefits system of Example 1. The users, the elderly disabled woman as patient and the experienced benefits clerk as agent, are assumed to have the properties given earlier in Figure 4, with some additional ones needed to make modeling fruitful and interesting (these will emerge in the dialog).

The example assumes that the system has the necessary technical linguistic interpretation and generation capabilities, and also the necessary internal modeling component. Thus I am assuming that the system is in principle able to recognize instances of the kinds of properties it has available to describe patient and agent,

and may indeed be capable of constructing new, particular properties out of other knowledge it already has. The example is designed to show what support natural language dialog, and more specifically the user inputs, can offer for modeling, and hence to provide a basis for a realistic assessment of the opportunities for, and value of, user modeling in enhancing system performance.

In the dialog, the modeling conclusions drawn by the system from its inputs (by whatever means it has for doing this) are associated with the various submodels described earlier and illustrated in Figure 4, i.e. P1 the static, objective decision model for the patient, P2 and A2 the analogous non-decision patient and agent models, and so forth. Thus the presumption is that the system starts with empty models, and seeks to construct reliable and coherent submodels, with its dialog responses seeking further modeling information as indicated. The use of the submodels is primarily, as before, to make the different modeling operations clearer. The symbol "+" below indicates the addition of information to a model, or its reinforcement if already present; "?" indicates the model proposition is hypothetical rather than certain (of course in practice there would not be an uncertain/certain dichotomy, but different degrees of certainty: however, the simple dichotomy is convenient here). The comments, *c*, on input and output, summarize what the system is doing.

As this is the beginning of a dialog, the system's responses are primarily directed towards obtaining more modeling information rather than applying a built model e.g. in explaining something; but as the example shows, the information seeking is frequently guided by the (tentative) models already constructed, though in some cases the system engages in quite basic information gathering. Moreover it is possible to show, even this early in the dialog, the system applying the information it has to different system functions. (Note that I am not concerned here with the plausibility of the assumed benefits regulations and its expert system implementation.) "O" is system output, "I" is user input. (I assume the initial logon sequence has been gone through.)

O: Please give me details of the applicant.

I: The applicant is female, aged 89, widowed, grade 3 disabled, and I imagine will get a pretty lavish benefit.

c: P1 + female, 89, widow, disabled 3
P2 + nil
P4 + nil
P8 + nil
A2 + experience?
(use of "grade 3" jargon)
A4 + hostile to patient?
(use of word "lavish", "and", "will" – compare "but", "should")
A8 + nil

O: What are the applicant's sources of income?

c: response motivated simply by need for further decision property information, primarily to promote effectiveness (i.e. no model application yet).

I: She says she's getting widow's pension of 17 pounds and 3 pence and otherwise the priest sometimes gives her 2 pounds out of his poor box. Her clothes seem rather worn.

c: P1 + pension 17.03, charity occasional 2.00
P2 + catholic?
(reference to "priest")
P4 + honest?
(pence details; mention of odd gift)
P4 + ignorant?
(does not realise housing status relevant)
P4 alternative + dishonest?
(concealing housing status)
P4 + suspicious (of clerk)?
(only replying to direct questions; note not necessarily an alternative)
P8 + no obvious beliefs, expectations about system?
A2 + nil
A4 + hostile to patient?
(reinforced by use of "she says", "seems")
A4 alternative + cautious observer?
(not experienced in evaluating state of old women's clothes)
A8 + exploring influence of modifications on basic income?
(apparent interest in whether she gets a clothing allowance)

O: Find out whether she buys clothes frequently. But has she been receiving anything for her disability? And ask her whether she lives with anybody.

c: response motivated by need to elaborate on P1, for effectiveness and possible efficiency; also to check on P4s of honesty and ignorance, attempting to select from alternative hypotheses, for the same reasons; by use of homely language attempting to bypass hypothesised suspicion, with additional motive of enhancing acceptability to patient; further motivation to gather information sufficient to start a useful P8. With respect to agent response motivated by attempt to distinguish alternative A4s and sidestep hostility, primarily for effectiveness but also for acceptability to patient; also using A8 suggesting focus on ancillary income to redirect attention to main sources of income. (Here the system is beginning to apply its models.)

I: She says the hospital gave her a wheelchair and that she lives by herself but her daughter sometimes visits and the District Nurse attends at basic frequency. She finds it very difficult to afford clothes.

c: P1 + no disability allowance, home owner?
P2 + nil
P4 + honest? i.e. + +
P4 + ignorant?
(apparently does not know of disability allowance, clothing allowance)
P8 + no notion of system and its capabilities?

A2 + experience?
("basic frequency" jargon)
A4 + nil
A8 + nil

O: Ask her whether she owns her home, and if she is still paying anyone for it (check for mortgage freedom). Tell her we may be able to help even if she has a house.

c: response pushing for more P1 information, for effectiveness and, we will suppose, efficiency and acceptability; also reacting to and testing P4s and P8, promoting acceptability and perhaps changing the patient attitudes embodied in the P8. The system is exploiting the agent's experience in A2 (reference to "mortgage freedom") and trying to reduce his hostility and render the system more acceptable by exhibiting care and exhaustivity in its operations.

This example illustrates many specific points about modeling, for example the possibility of agent interference in the transmission of information about the patient, and also the need to distinguish user beliefs about the nature of the domain as covered by the system from beliefs about the domain within which the system operates. For instance the patient may have beliefs about the scope of the system based on her beliefs about the scope of social security, although the scope of the system is narrower: for example in relation to housing, she may suppose the computer system covers rates rebates because social security as a whole does, though the system is in fact only concerned with housing in relation to asset ownership. The patient's mistaken beliefs could motivate replies to questions about what she is paying which were quite misconceived, but in a way very difficult to detect.

But the major, critical points made by the example are first, how heterogeneous the evidence for models is, and second, how weak it can be. This is particularly in relation to properties other than decision ones, i.e. the non-decision properties of both patient and agent. Thus there are three data of different sorts, the three pence, the cash gifts, and the daughter visits, all suggesting the user is honest; however any one of these, like mentioning the three pence, is really only very weak evidence for honesty: it could rather indicate a passion for accuracy, for instance.

The essential problem about modeling is that property indicators may be unreliable or undiscriminating. Thus to take an independent example, a man being a Catholic does not imply he is a priest, so being a Catholic is unreliable evidence for being a priest. But being a priest does not imply being a Catholic priest: as there are priests in other churches, being a priest is not discriminating. Building and applying models is clearly a very tricky business, even for decision models, but it is much more so for non-decision models. It should be noted, moreover, that these points also apply to modeling initiated by stereotypes: for example the stereotypic old woman may be religious, but it may not be at all easy to verify that a given old woman is religious (in the extreme case, even by asking) or, more importantly, to establish that she is in fact not religious.

5. Rational Principles for Modeling

Given the manifest difficulty of obtaining reliable modeling information, what are the implications for system design? Should the system devote effort to trying to confirm the evidence? Information about the user is needed to guide action bearing on the user, but if the system does not have much, or good information, should it seek more, especially if drawing strong conclusions from weak evidence can lead to poor performance?

There is no doubt that if the decision properties the system uses are the only proper ones for the nature of the decision to be made, then every effort has to be made to get the best information about them, even if this is very difficult. The problem is much more serious with non-decision properties because it may be much harder to get useful non-decision modeling information.

This is primarily because there are potentially very many, diverse non-decision properties, not connected with one another or with decision properties. The presumption about decision properties, as mentioned earlier, is that they form a related set, which is typically also a select set, especially if the system's task is a well-defined and constrained one. But the number and range of properties a user can possess outside the decision set is large and wide, even considering only properties that might be validly exploited to serve any of the system's functions. Thus for the example system, user properties as disconnected as religion and poor sight could be usefully exploited in the interests of system acceptability, the one to stimulate a visit to the priest to encourage a discussion of a low benefit, the other to direct the patient to specially-printed large-type explanatory leaflets. The problem is that there may well be no motivation or leverage for a system search in one non-decision direction rather than another (unless a random pursuit is adopted on the basis that anything is better than nothing, and one thing is as good as another).

There is a particular further problem with non-decision property information, assuming it is not explicitly sought, namely that it is more likely to be conveyed indirectly than directly by the user, and thus be more likely to be subject to uncertainty. In general, the more indirect clues are the less helpful they are. Again, the deeper models are, as those concerned with subjective properties generally are, the harder the properties involved are to ascertain.

All this suggests that the realistic conclusion to draw is that rather than diverting systems into costly and probably still unprofitable searches for evidence, given that even quite powerful systems may yet lack extensive general world knowledge, we should restrict modeling to the user properties we have a chance of getting good information about, i.e. primarily, if not exclusively, decision properties, for which data, if they are not sought directly, may be more obviously supplied or, where decision properties are systematically related, inferred. The further conclusion is that modeling, especially modeling beyond decision property modeling for effectiveness, should be very conservative, i.e. it should not be driven too deep for content or stretched too far in use.

It is relatively easy to imagine how this strategy could be applied to the social security example. The system would not bother, for instance, with the patient's religion as this is a high cost, low payoff property, i.e. one difficult to establish and

of rather marginal utility through not having a predictable acceptability value; the system could probably do better tailoring output for acceptability by using suitable decision properties like age. Honesty and system expectations would, on the other hand, be available as non-decision patient properties, but would not be pursued unless the decision data on hand was blatantly incongruous or inadequate.

The illustration below is designed to emphasize these points, through a second and rather different application system.

Example 3

This example is summarized in Figure 6. The system here is a simple travel agent dealing with transport, so the system's decisions are recommendations on the means of getting from A to B. The decision properties are thus those setting constraints on the choice of mode of travel, e.g. date, cost etc. The user is assumed to be both patient and agent. Suppose, then, that we have the initial interchange between the travel agent T, and the client C, represented by the C1, T1, C2 in Figure 6a. The client's reference to trains can suggest three different hypotheses, (a), (b) or (c), about the user properties motivating this, namely that the user is interested in economy (trains are cheap), thinks travel by train is fun, or dislikes flying. Different choices by the system here would naturally suggest quite different responses, for example T2a for hypothesis (a) or T2b for (b). Clearly, if the system's hypothesis is wrong, its response could look very unmotivated to the user. A less risky strategy, therefore, than adopting a hypothesis in the absence of good (i.e. reliable and selective) evidence, would be to seek further evidence, allowing a motivated choice, via the response T2x which explicitly asks the user about his underlying interests. But the user may in fact have no such interests, and may find the further questioning unnecessary and hence irritating. Thus a third, even more conservative strategy would be simply to take the user's input at its face value, and to respond, as in T2y, just by listing the Istanbul train data. (Note that even here, the hypothesis that the user wants trains has properly some uncertainty attached to it, though it is reasonable to regard this hypothesis as much less uncertain than any of (a), (b) or (c).)

What this example suggests is that even where the system is dealing with decision properties, it may be better to be less active in constructing and using models, in an attempt to provide a tailored response, and simply to offer the user sufficient information for him to apply his own model of himself.

The hazards of overcommitting on evidence about non-decision properties, which may be even less adequate than that for decision properties, are illustrated in Figure 6b. Thus suppose that after C1 and T1 as before, we have C2, leading the system to hypothesize that the user has the non-decision property of being religious (being religious is clearly a non-decision property for characterising modes of travel.) This non-decision model may be exploited, with the best of intention, to produce T2p rather than T2q. But in fact, what sufficient reason is there for supposing that religion is the user's motive for not wanting to travel on a Sunday? Attempting to gather more evidence for the hypothesis, for example by questioning the user, would, moreover, be unnecessarily elaborate. The safest, and also the optimal strategy, is for the agent simply to produce the straightforward response T2r.

C1 I want to ask about Istanbul.
T1 So you want to go to Turkey?
C2 Yes. Can I go by train?

train → (a) economy
(b) fun
(c) no flying

T2a Even the train costs $100.
T2b How about the Orient Express?

train → ?
?
?

T2x Do you want economy or fun ...?

train → wants trains

T2y The trains are ... (times, types, costs).

(a) Dialog using decision properties

C1 I want to ask about Istanbul.
T1 So you want to go to Turkey?
C2 Yes. I want to go by train, but not on Sunday.

not Sunday → religious

T2p Unfortunately the only train is a so-called Fun Special on Sundays.

NOT

T2q There's a super Fun Special on Sundays.

not Sunday → not Sunday

T2r The only available train is on Sundays.

(b) Dialog using non-decision properties

Figure 6. Model-using strategies. A travel agent system

6. Conclusion

But is it really necessary to draw such a pessimistic conclusion as this chapter would appear to suggest: namely that modeling is so difficult that we are unlikely to be able to provide systems with any material modeling capacity, and should therefore not even attempt to do this? After all, areas like intelligent computer-aided instruction demonstrate that successful modeling can be done [SLEE82c, CLAN86b].

But it is important to recognize that modeling here is primarily decision-property modeling, made easier by the constrained nature of typical applications, namely teaching a relatively restricted skill like solving certain types of algebraic equation or basic linguistic knowledge. This is indeed not to say that building such systems is easy; but it can be claimed that user modeling here is less difficult than in many other cases because of the narrow focus such systems tend to have and the well-structured domain they normally have. It may also be possible to get leverage in systems like this by building in general characterizations of the kinds of users they may have.

Pessimism is moreover not called for as far as decision properties are concerned. If the system has a human patient, this user has to be modeled. The correct, but of course not novel, conclusion to be drawn from this chapter is just that useful modeling essentially depends on having an adequate set of decision properties, and also that conservative modeling strategies are in order.

It might further be maintained that, even allowing for other application areas more challenging than instruction (as ordinarily treated), there is really no need to take the kinds of modeling aim tacitly presupposed in this chapter seriously. There is therefore no great cause for pessimism. Thus it may be possible to do very useful modeling, in a modest way, allowing for some non-decision properties, basic role distinctions and different functions, but not going beyond the more obvious sorts of information and relatively straightforward uses of them. This is in fact all that we are able to attempt now. But our performance in practice is not very encouraging here, and the examples of the chapter equally suggest that it may not be so easy to identify and deploy even quite obvious information about users.

Adopting a modest approach to modeling does not, moreover, imply that there is no need for the kinds of distinctions drawn in this chapter. Wahlster ([WAHL86a]) has argued, for example, that the distinctions are over-complex, imagining what contortions they would imply for the designer of a marriage counseling system which might have be faced with a concerned mother, son and son's wife. But as Figure 7 shows, simply considering different goals for the system implies a need to recognize the kind of distinctions I have drawn, as without them the system is unlikely to be able to satisfy its goals. Thus focusing on different parties carries different role distinctions with it. (A situation with multiple agents would require equally careful distinctions.)

As a basis for discussing and implementing user models, the general framework presented in this chapter has obvious points of contact with those offered by [BELK84, DANI86a, KOBS*, MORI*, RICH83, WAHL*], for example. But the perspectives adopted in all these analyses vary, and there are interesting differences in granularity. Thus Morik's treatment [MORI*] of interactive settings for user

1. GOAL: help son
   agent : mother
   patient : son
   wife : son's view = P3
   real nature = P1

2. GOAL: help son and wife
   agent : mother
   patient1 : son
   patient2 : wife
   wife : son's view = P1.3
   real nature = P1.1
   son : wife's view = P2.3
   real nature = P2.1

3. GOAL: help mother
   agent : mother
   patient : mother
   son : mother's view = P3
   real nature = P1
   wife : mother's view = P3
   real nature = P1

Figure 7. Goal perspectives and modelling implications. A marriage guidance system. P1 = objective decision properties, P3 = subjective decision properties

and system actions in terms of the system elements open to change, the means of change, and the agencies of change, provides a rather different and in some ways more, though in other ways less, detailed analysis of modeling contexts than that provided here, suggesting that it may be useful to try to put the two together.

The general framework offered here is also neutral with respect to what may be called the orientation of a system to a particular generic task, for example explanation [VANB86b, VANB86a, CLAN86b, SLEE85], especially explanation seen as an instructional activity. The implications of these generic tasks obviously have to be taken into account in user modeling: my point here has been rather to emphasize the need to look at user modeling outside the popular generic tasks. Thus, explanation may in some cases be at most a minor system need.

Again the analysis offered here makes only weak assumptions about cooperation: cooperation with the patient may be minimal or non-existent, and sufficiently attained with the agent with very little effort. However, it is clear than in individual applications it may be necessary to take into account the complexities of cooperative behaviour illustrated by [AIRE86, JAME*], as well as the need to apply specific response strategies in dialog including, for example, explicit clarificatory and negotiatory dialog, and to tailor responses to the user as illustrated by [CARB*,

COHE*, GOOM86, MCCO88], for instance. The chapter's examples assumed such requirements but did not analyze them, since the chapter was more concerned with emphasizing the great variation of user modeling contexts and hence allowing for the fact that questions of cooperative response may not arise though user modeling itself is appropriate.

Beyond making a broad distinction between static and dynamic subjective properties, the chapter does not attempt to give the kind of detailed analysis of the properties of beliefs, goals, plans, etc. carried out by [KOBS*], or to consider the detailed relations between the user's intensional states and the reference world investigated by [VANB86b, VANB86a, CHIN*, MCCO88]. The broad view adopted has to be filled out in this way, as appropriate to individual system contexts. However the framework provided does draw attention to the fact that in many contexts it may be proper to talk about user modeling which is not exclusively devoted to, or dominated by, the user's intensional states, and also that users may, for example, have goals without also having plans.

The chapter is also deliberately focused on the user model as opposed to any discourse or dialog model, say of the kind illustrated by [LITM85], taking it for granted that these two are distinct but not analyzing or defining the precise relationship as this is done, for example, by [SCHU88, CHIN88a, COHE88, KOBS88, MORI88, SPAR88, WAHL88]. The analysis of user models clearly has to be supplemented for implementation purposes by a view of their relation to discourse models (and any other models like a world model). The object of the chapter is to emphasize the fact that in complex systems the user model cannot be equated with the discourse or dialog model, i.e. it is necessary to distinguish communicative properties of the user from real properties of the user, but then of course also to allow for very subtle relationships between these. It is clearly possible, in particular, to envisage further complications of the various kinds illustrated by [JAME*], [WILK83] and the HAM-ANS hotel application [HOEP83b], where dialog participants may be deliberately deceitful, or may mix individual and institutional attitudes.

The examples in the chapter illustrate the variety of possible sources of user modeling information also considered by [WAHL*], for example; and it is clear that though, as indicated, it may be very difficult to rely on structural relations between different properties or property types, especially for a wide scatter of non-decision properties, modeling has to look for relations between objective and subjective properties of the kind considered by [BROO86, COHE*, MORI*, RICH79b]. How these are related is just one element of the specific modeling process; and the framework for modeling given in the chapter has of course to be supplemented by an account of actual modeling procedures, i.e. of the inference mechanisms required to construct and deploy models, like those described in [WILE86] and for goals and plans in [VANB86b, CARB*, GOOM86, POLL86a]. The aim of the chapter has been to indicate the need to consider the various aspects of modeling, especially in relation to system functions, before detailed procedures are designed.

The analyses of the information-seeking interactions between the users and staff of specialised library services carried out by Belkin and his colleagues [BELK84, BROO85, BROO86], where user modeling is of the essence, serve to bring out just how hard modeling is in such complex situations. Establishing the nature of the

library user's information need is user modeling in the broad sense of the present chapter, though Belkin and his colleagues themselves give "user model" a much narrower meaning. Characterizing someone's scientific literature needs requires a large apparatus of 'functional experts' to deal with the various facets of the modeling, and the application area is one which clearly illustrates the intractability of modeling in those cases where the user's subjective, i.e. mental properties are in question, and even more, in those cases where the user's goals, plans, beliefs, etc. are dynamically changing through the interaction. But the particular problem in this case is that the modeling information is necessarily inadequate, because the user is seeking information which, as he has not yet got it, he cannot properly specify. Brooks' and Daniels' detailed analyses [BROO86, DANI86b] of human information interviews show just how difficult and how complicated capturing information needs and other relevant properties of users is: we are nowhere near being able to extract information from dialogs like this automatically.

However, their interview transcripts also show quite clearly that even human beings may not find it useful or possible to seek to carry modeling very far. The overall message of my chapter is that as the situations in which user modeling may be envisaged vary enormously, because they depend on combinations of user populations, system tasks, and application domains, the strategy to be adopted for modeling has to be evaluated for each individual system. But it may well then turn out that, quite apart from the fact that we cannot yet, for example, provide sufficiently powerful natural language processors, the conjunction of necessary ignorance about users and system resource limitations e.g. in relation to real time operation, may mean that if modeling is done at all, it can only be done in a very limited way. It is nevertheless still user modeling, and may be as adequate as is in fact required. This view is, on the one hand, a justification for the use of stereotypes, as in [CHIN*], for example, and on the other for such modest applications of modeling as the individualised parsing proposed by [LEHM*]. It also has the important advantage of being unlikely to mislead the user into thinking the system is smarter than it is.

Acknowledgements. This work was carried out during the tenure of a GEC Research Fellowship. I am grateful to Richard Young and to the participants in the Workshop on User Modeling held at Maria Laach in 1986 for discussion and comment.

7. References

[AIRE86], [BELK84], [BROO85], [BROO86], [BUCH84], [CARB*], [CHIN88a], [CHIN*], [CLAN86b], [COHE88], [COHE*], [DANI86a], [DANI86b], [GOOM86], [HOEP83b], [JAME*], [KOBS88], [KOBS*], [LEHM*], [LITM85], [MCCO88], [MORI88], [MORI*], [POLL86a], [RICH79b], [RICH83], [SCHU88], [SLEE82c], [SLEE85], [SPAR84], [SPAR86], [SPAR88], [VANB86a], [VANB86b], [WAHL86a], [WAHL88], [WAHL*], [WILE86], [WILK83]

13 User Models and Conversational Settings: Modeling the User's Wants

Katharina Morik

1. Introduction

In general, the cycle of system development can be viewed as consisting of four phases:

- theoretical considerations, design
- practical implementation and experimentation
- evaluation of the system, documentation
- reorganizing the field of research, thus using the experience which has been gained by workers in the field.

The fourth phase is indicated by activities such as defining the technical terms of the field systematically so that deficiencies in coverage of the accomplished work can be detected. With a new or modified view of the field, the cycle then starts again. In user modeling, the fourth phase currently seems to have begun.

This chapter follows the trend by first clarifying terminological issues. Second, a general view of user modeling is presented. Conversational settings of interactive systems and types of model changes are categorized and illustrated in the context of the system HAM-ANS[1]. Already completed user modeling is classified and the state of the art in user modeling determined with respect to this framework. Areas where further research is needed are highlighted. The following main problems are pointed out:

- the integration of non-mental, non-communicative actions,
- the integration of the user model acquisition and the interpretation phase, and
- the change of user models.

Third, the framework is in turn applied to the HAM-ANS approach to user modeling, particularly focusing on modeling the user's wants and evaluation standards. There, the fact that a non-communicative action of the system is possible forces the system to produce a wider range of speech acts than mere information. Finally, we propose a way to integrate the acquisition and the interpretation phase

[1] HAMburg Application-oriented Natural-language System [HOEP83b]

and to use this procedure for the update of the user model in the HAM-ANS approach, thus overcoming the limitations of accomplished work. The central point is that on the one hand, representing the evaluation standards of the user allows the system to recognize the user's wants from her/his utterances. On the other hand, the evaluation standards may be inferred from the utterances by using the textlinguistic notion of "isotopy" in a new way.

2. Terminology

In order to clarify the term "user model", the closely related terms "dialog memory" and "agent model" must be considered as well. A dialog memory can be viewed as part of a user model, namely the part which represents the dialog-dependent knowledge of the user [MORI84]. Entries out of the dialog memory may cause entries in the user model, and entries of the user model may support the interpretation of an utterance, the interpretation then being stored in the dialog memory. However, in order to keep technical terms precise, user modeling on the one hand and building and exploiting a dialog memory on the other hand should not be identified. This would lead to a reduction of what user modeling is about by disregarding all aspects other than dialog-dependent knowledge of the user as known to the system, while in fact there is some information which is to be covered by a user model and which may not be covered by a dialog memory.

Let us think, for example, of a visit to the dentist's. The dentist will have some expectations concerning the client even before the client says a word. This is due to the conversational setting, the roles of dentist and client. The same two persons meeting in another environment (e.g. at a horse race or the opera) would not rely on the dentist-client expectations but on the information which then belongs to their roles. Of course, by analogous reasoning, the dentist may transfer the way the particular person plays his or her role as a client to the way this person will behave at the races. This transfer, however, is also based on role-specific information.

A *dialog memory* contains all beliefs that can be inferred with certainty from utterances, so that they belong to the mutual belief space. These are beliefs a speaker cannot deny when uttering a sentence. For example, a speaker cannot talk of "today's king of France" without the obligation to claim that s/he believes there is a king of France today. Thus, presuppositions which can be inferred from articles or question particles belong in the dialog memory. The objects and their properties introduced in a dialog are typical entries in a dialog memory as well. The dialog memory is structured so that it can be used for such things as determining the appropriate description (definite or indefinite determiner), anaphoric expression or characterization. The linguistic rules which determine the inferences are valid and binding for all conversational settings and are not parametrized according to assumptions about the user. If, however, linguistic rules are specialized for a particular user in a particular conversational setting [LEHM*], this is clearly not a matter of the dialog memory but of the user model, because then these rules are not valid and binding for all users.

A *user model* is an agent model where

- the situation in which the agent plays a role is the conversational setting of a system-user interaction concerning a particular task, and
- the agent is the person who solves his or her problem with the help of the system, either directly or through an intermediate person.

The user model allows the system to adapt to different users, tasks, and roles.

This definition links "user model" to the more general "agent model". An *agent model* is a knowledge source which contains explicit assumptions about a person acting in a certain setting. As psychological studies have shown there is a kind of consistency in the image of other persons: personal traits are not all viewed independently of each other. There are stereotypes which group several personal traits together and stereotypes themselves are combined following a naive psychology. In this sense an agent model forms a consistent image of the agent. If the person modeled by the system is the protagonist of a story and the system understands the story, we speak of a *protagonist model*. In principle, modeling a person in the actual[2] conversational setting and modeling a person in another setting (which may also be a conversational setting), is not very different. However, for a system which plays no role in the story it understands, but does take part in the interaction with the user, user modeling is a more complex task.

The definition excludes from the user model the information about the user which is not relevant to the user's role in the system-user interaction or to the task under concern. Thus, in the example of [WAHL86a], where the age and the bonus of the user is stored in a database and the system can answer questions about the user's bonus and status, the crucial point is whether this information at least partially determines the user's role in the interaction and is related to the task of system and user. If, for instance, the system helps employees to plan their business trips, and if the decision whether the traveler is supposed to take a train or an airplane depends on his status, then this information is related to the task. As [SPAR84] named it, the information is a decision property. Now, another requirement from the above definition is that the system uses the information to adapt to different users, tasks, roles. Thus, it is not sufficient that the system uses the information in some way, but it must use it systematically for individualization. In the example, the user's status constrains planning, ruling out several arrangements. However, even with these modifications, the example need not fit into our intuitive understanding of user modeling. This is due to the unrelatedness of the entries in the database – they do not necessarily form a consistent image of the user. But this is also a definitional property for a user model as inherited from the definition of an agent model. Therefore, in order to transform Wahlster's database example into an example of a user model, we must perform an additional modification: the system must have a process which forms a consistent image out of the database entries, this image then being the user model.

Another problem with defining a user model is whether the person directly interacting with the system or the person whose goals are involved in the interaction

[2] A discussion of the relation between the notions "dialog memory", "discourse model" and "user model" can be found in [MORI88].

should be modeled. In the above definition the person to be modeled is the one whose task corresponds to the task of the system. The definition closely relates the user model to the task which the user wants to perform with the help of the system, and thus to the system's task. Task and role specificity separates a user model from a general agent model. The problem with determining one person to be modeled is that several persons can be involved. Let us think, for example, of a group which asks for a price reduction in a train station environment. At least in Germany, there are several special offers for groups but also for elderly people or children. In order to find the best arrangement, it is not sufficient to model the group as one entity; instead, the members of the group have to be modeled along with the possible ways of grouping them together.

Another illustration of the problem of restricting user models to modeling one person is the embedding of situations (see [SPAR84] for a discussion of agent/patient modeling). The intermediate person directly interacting with the system is the user to be modeled concerning individualized linguistic rules. The person on whose behalf the interaction takes place is the user to be modeled concerning the task.

In most cases, however, there is not more than one person with respect to the task. This person may be introduced indirectly ("the brother of the wife of my son ..."), but this does not change the situation of an intermediate person as the direct user, where the person to be modeled is the one whose problem has to be solved by the system. Moreover, the task-related situation may be an embedded situation as well: the problem of a mother with the problems of her son. If she asks, for instance, whether she should wait and see or do something about the son's problems, the answer also depends on the kind of a problem the son has. In this case, the mother is the person to be modeled according to the definition above. The son is the protagonist of the mother's story.

3. Conversational Settings

We have closely related user modeling to the tasks and roles of system and user. Let us now develop a systematic view of the situation. We start with a descriptive example of a conversational setting, introduce three factors from which we can generate several conversational settings (3.1), and discuss three typical settings of interactive systems using HAM-ANS examples (3.2–3.4). The general view of user modeling is accomplished by collecting change types (3.5) and introducing types of user model acquisition (3.6). This will allow us to determine the state of the art and point out underdeveloped areas in current research (3.7).

3.1. Factors of conversational settings

Figure 1 shows Mrs. User thinking of a hotel room she wants to book. She communicates with a computer system which also has some ideas about a hotel room, so both have a domain model. In most man-machine interactions, the dialog is about enhancing the user's domain model. These are *information-seeking dialogs.*

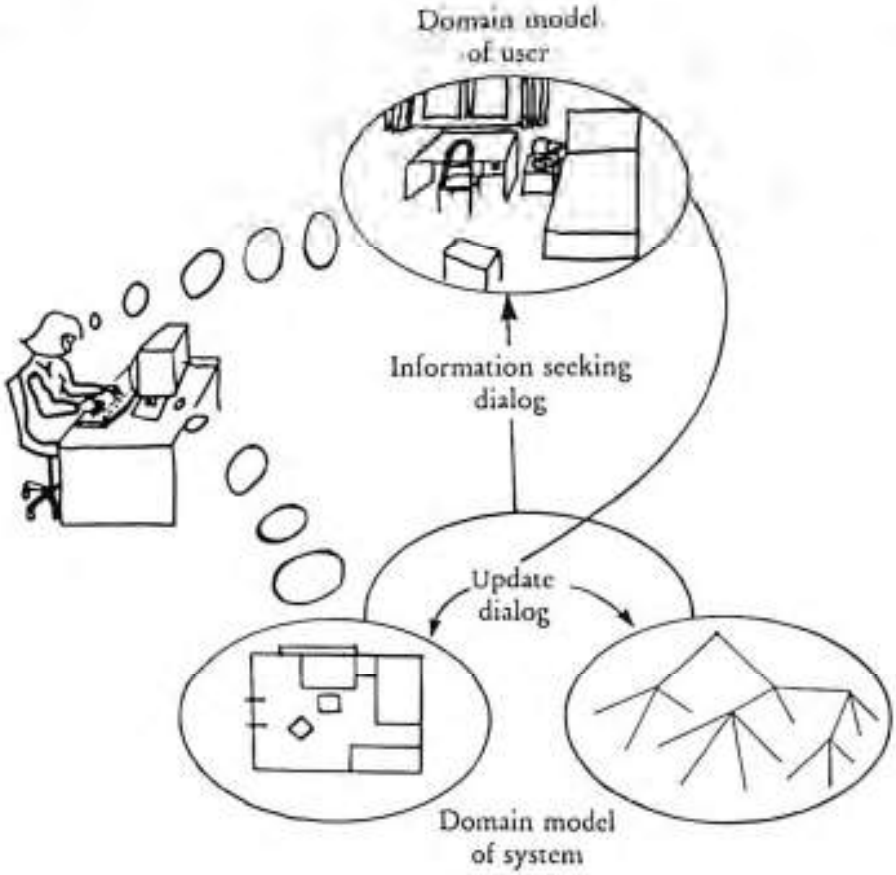


Figure 1. Mrs. User and the Computer

Of course, the dialog could also be about filling the system's domain model. Then it is an *update dialog*. For an update dialog, the system must be able to change its domain model. Although for system developers these two kinds of dialogs are difficult enough, they are not sufficient, because reducing differences between the domain models of the communicative partners is not what dialogs are all about. Mrs. User's wants are not directed towards a model – even if it can be manipulated in the process. She wants to book a real room, she really wants to use it. Dialogs where the system is able to perform a non-communicative action because of the dialog are called *action-achieving dialogs*. Human communication most often deals with the real world, although models are used for communication. The obligations resulting from dialogs refer to the real world and not only to models. In the above example, the hotel room should be really booked if the user wishes. Taking this into consideration gives a more general view of dialogs and their conversational settings.

Viewing actions as the most important topic with respect to the goals of a dialog and the dialog itself, we characterize the conversational setting according to the actions of system and user as they are modeled by the system, i.e. as far as they are perceivable and tractable for the system. Actions may occur in the

- domain of discourse (i.e. part of the real world),
- domain model (i.e. knowledge about the domain)
- knowledge about the partner (i.e. knowledge about the partner's beliefs and wants).

If Mrs. User in our example wants to have a TV set in her hotel room,

- the system could have a TV set brought into the room, thus changing the domain of discourse,
- update the knowledge of the user about the room by informing her that there is a TV set in the room,
- update the domain model after being informed by the hotel manager that there already is a TV set in the particular room,
- update the user model by recording the want for a TV.

These changes may be performed by several different means:

- non-mental, non-communicative actions (e.g. putting a TV set into a room)
- communicative actions (e.g. informing or being informed about the presence of a TV set)
- observation (e.g. looking into the room to check whether there is a TV)
- inference (perhaps the user's want was inferred from an utterance meaning).

A change by inference is made if a proposition is deduced. If we adopt the view that all implications of available knowledge are known as well, the derived proposition is already known. Therefore, in a narrower sense, no change occurred[3]. Of course, inferences and observations are a means for changing models and not the real world. So all of what is to be changed cannot be combined with all means of change, but some means are specific to certain kinds of change.

So far, the actions were on the side of the system. But it is also of importance whether the user plays a role in the domain of discourse, or not. Mrs. User in our example plays a role in the domain: she'll become a hotel guest using the room she booked. In the standard computer applications, however, the user is not involved in the domain (or only by chance, not systematically): think of all the supplier-parts database applications, for instance.

Using the three factors

- what can be changed by the system,
- by what means, and
- whether the user plays a role in the domain

we may now generate several conversational settings. We show three typical settings, each first without the user playing a role in the domain and then with the user playing a role in the domain. For illustration purposes, we discuss HAM-ANS

[3] cf. the discussion of the wise men puzzle in [KONO84] for the (im-)plausibility of the assumption that we know all implications of our knowledge.

applications and determine how they fit into the conversational settings. HAM-ANS is a natural-language access system to diverse application classes [HOEP83b]. The application classes are exemplified by three applications. The first application provides natural-language access to a relational database system. There, HAM-ANS is connected to a shipping and fishery database. In the second application, which exemplifies natural-language access to a scene analysis system, the system answers questions about a street intersection. A film consisting of about 70 snapshots was analyzed, the movements of objects (pedestrians and cars) represented as trajectories[4]. Questions are then answered on the basis of the internal semantic representation. In the third application, HAM-ANS plays the role of a hotel clerk, informing the user about the hotel and its rooms, recommending a room, and booking it if the user wishes.

3.2. The First Conversational Setting — Information-Seeking Dialogs

As the *first setting* let us take the most common situation for AI interactive systems. The system brings about changes only on the user's mind by communicative actions. It changes its own dialog memory accordingly.

If the user plays no role in the domain of discourse and the user's task is not modeled, overanswering or overunderstanding (i.e. understanding ill-formed input) must resort to linguistic clues and general rules of cooperativeness [WAHL83]. The first application class of HAM-ANS, (front-end to a database system) fits into this setting. The domain model (i.e. the database schema as conceptual knowledge and the database content as referential knowledge) is static. Neither user nor system play any role in the domain of discourse. The system informs the user about ships and their cruises. Different users can have very different tasks for which they use the information from the database. The meteorological data gathered during a scientific cruise might interest a meteorologist in quite another way than it would interest a shipowner. Because tasks of users are not modeled in that particular HAM-ANS application, providing the user with the information s/he wants is guided by principles of cooperativeness, which are not specific to a particular user or task[5]. The system doesn't need to decide which action is appropriate under certain circumstances, but has a built-in, implicit strategy: answer questions about the domain, being as informative as possible. However, some situations occur during the dialog in which the system may choose to deviate from 'terminal communication': for instance, to output the result of a database query on the printer. This is, of course, a communicative action, but it differs in kind from the default interaction in the dialog. In these cases, finding out what the user wants is realized by just asking her/him.

If, on the other hand, the user plays a role in the domain of discourse, her/his actions are to be modeled. An example of this conversational setting is the Montreal train station scenario [ALLE79, LITM85]. The actions of the user are represented

[4] The same conversational setting underlies NAOS [NEUM83].

[5] As Marburger has shown [MARB87], dealing with linguistic rules of cooperativeness enhances dialog performance considerably.

together with their mental preconditions. This information is used to clarify the user's information requests. Determining exactly which information the user needs leads to a more finely grained model of information request speech acts, as has been shown by the above mentioned work. Here, the system finds out which information the user needs. The overall goal and the possible actions of the user are known to the system before the dialog.

3.3. The Second Conversational Setting — Reporting

The *second setting* is that the system changes its domain model because of observed new facts. We might think of a vision back-end system which observes state changes in the real world and a natural-language front-end system which communicates the events to the user. The second application of HAM-ANS would fall into the class of this setting, if the domain model were not fully constructed before the start of the dialog. Observation and communication do not take place at the same time. The domain model is treated as static, although the domain consists of events in a time sequence[6]. User and system do not play any role in the domain or domain model. The only state changing action is the dialog itself. As in the case of the database application, a user model concerning the user's wants is not constructed nor exploited; instead, only general linguistic rules are used in order to find out what the user wants to know about the street crossing and the events there. Of course, the system's knowledge about the user's dialog-dependent knowledge is use for generating descriptions [JAME82].

3.4. The Third Conversational Setting — Action-Achieving Dialogs

Let us take as the *third setting* a situation that, because of its complexity, has to my knowledge not yet been fully realized in interactive AI systems. The system performs changes in the domain by non-communicative actions, changing its domain model accordingly. At the same time it is communicating with the user. An example of this would be an intelligently interfaced robot. In this case, the user does not perform any action in the domain but controls the robot's actions.

If the user plays a role in the domain as well, s/he might want the system to change the state of affairs, thus enabling her or him to perform a certain action. In the example above, Mrs. User wants to watch TV, and the robot's bringing in the TV enables her to do so. In a restricted manner, from the user's point of view, HAM-ANS playing the role of a hotel clerk is also an example of the third setting. The system simulates the booking of a hotel room for the user, fulfilling a prerequisite for the user's intended action (to stay in the hotel for a while)[7]. Booking is a non-communicative, although not non-mental action which changes the state of affairs. Although other changes are not modeled, the interior of the rooms cannot be changed, for instance, because of the booking the dialog is a (restricted) action-achieving dialog.

---

[6] Thus, time adverbs and verbs of locomotion could be modeled [MARB83].

[7] HAM-ANS was never used in a real hotel.

As empirical work has shown (e.g. [NOTH84, SCHG79, POLL83]), the most salient difference between information-seeking and consultation dialogs is that the wants of the client are fully specified before the dialog or not treated at all in the former, whereas the specification is found interactively in the latter. A *consultation dialog* is a specification phase for the intended action. The unspecified goal of the user is known to the system before the dialog: the user wants to book a room. The specification is to be found out from the dialog. The system recognizes the user's wants and selects the particular room which suits the user best. Regarding the specification which is determined interactively during the dialog, it is a step in the direction of consultation dialogs.

Whatever the role of the user, as soon as the system is able to perform changes not only on the model, but on the domain itself, the dialog may include speech acts other than information requests and information giving. In other words, action-achieving dialogs show a variety of speech acts. On the side of the user, this is due to the justified belief of the user that the system is able to fulfill the request. A wider range of speech acts stresses the problem of speech act recognition. Ambiguities at the level of speech acts must be resolved. For example, the user input

Could you reserve a single room ...?

can be interpreted as the question of whether there is still a single room available, a question about the capabilities of the system, or as the request to book the room[8]. The system must decide which knowledge sources it needs to change. The first two cases require merely a communicative action which changes the mutual belief space (the dialog memory). In the third case, the system must perform a change in the domain and hence in the domain model. Then it must decide whether an additional communicative action is necessary. The number of choices which the system must make is considerably greater than if the system is restricted to communicative actions only.

Also on the side of the system, this conversational setting requires a variety of speech acts to be handled. As opposed to nearly all other natural language dialog systems, HAM-ANS in its hotel application is able to *generate* speech acts other than the inform act as an answer to a user's information request. HAM-ANS can ask the user, recommend a room, and – of course – answer questions[9].

3.5. Change Types

We have characterized conversational settings by the changes that system and user may perform. From the point of view of a system developer who builds a system for a particular setting, the complexity of her/his task depends on the number of different actions which system and user are allowed to perform. But there is one further point: the complexity of how changes are modeled. We want to distinguish five *change types* ordered here by increasing complexity:

---

[8] For more precise rules using assessments of speaker/hearer for recognizing the speech act cf. [MORI82:239ff].

[9] A more detailed description of generating recommendations is given below (4.1).

- addition
- replacement
- modification
- revision
- development

Although these terms apply to any model of changes, changing the domain, the domain model, or the dialog memory, let us illustrate their meaning by discussing user models. By *addition* we mean adding new information without any consistency or integrity checks[10]. New information is entered into the user model. The entries are assumed to be independent from each other, so that adding a new fact does not change the already stored ones. Contradictions are not even noticed in this change type. For user models which form a coherent image of the user, this change type is not applicable.

By *replacement*, we mean that a new piece of information overwrites an already stored one. This is the simplest way of handling contradictions: if a new piece of information is in conflict with existing entries, just throw them out! Only simple contradictions are recognized – that is, those concerning the same property. For example, a contradiction between two assumptions concerning the solvency of a user can be detected: "The solvency is high" is in conflict with "The solvency is medium". The new assumption here overwrites the older one. However, in this change type all properties of a user are assumed to be independent. Thus any relation between them are not represented. For instance the relation between the profession and the solvency is not represented.

Just as replacement, *modification* refers only to information about a single property of the user. However, here the properties are not assumed to be independent. So, it allows for reenforcement or weakening of entries. New information may modify existing entries, or existing entries influence the way new information is entered, respectively. Modification concerns the level of belief or importance of a certain property, not the value of a property.

In the stereotype approach (in the sense of [RICH79a]) the user facets are independent from each other, thus a new facet may simply be added. From the facets properties of the user are inferred. These are not necessarily independent from each other. In HAM-ANS, the properties which are derived from the user facets are user requirements of a hotel room. If two facets create the same requirement, the importance of this user want is increased. If two facets create different instances of a requirement, modification is needed to create the appropriate 'compromise' requirement [MORI85a]). Entering a new requirement directly, i.e. without inferring it from a facet, employs the same modification process. If a facet has to be changed, the overall process of inferring properties from facets with the modification included starts again. This leads to the same result and is much easier than revising the properties according to the changed facet.

For *revision*, the representation must permit the tracing of all entries which are based on the same (false) assumption. The conflict resolution concerns more as-

[10] If there is no closed world assumption, it is a monotonic update.

sumptions than the conflicting two. Dependencies among entries in the user model are thus taken into account[11].

By *development* we mean a model of changes that include the history of the changes. As [BARB82] points out, using "no-good" sets for a false assumption and its dependents – as is most often done in revision – is not a history keeping facility. The development type of change is of particular importance to user models of tutoring systems. There, not only the level of expertise at a given point is important, but also its progress. If two assumptions about the user's performance level are in conflict, the user's performance level can have been raised in the meantime due to her/his learning.

In the terminology of [SPAR84, SPAR*] the first four change types refer to static modeling, i.e. the user remains the same but the system changes its knowledge about the user. The development type refers to dynamic modeling in that properties of the user are not necessarily viewed as permanent. We do not, however, identify the development change type with the system's task of recognizing plans and goals of the user, as does Sparck-Jones. Systems like those of [ALLE79, LITM85], which recognize the plan of the user to a certain extent do not model the user dynamically, because the user's overall task is not permitted to change. Representing actions is independent from using the development change type.

3.6. User-Driven and System-Driven Acquisition

An important factor for the complexity of the system-developer's task is whether the construction of a model for a particular user is system-driven or user-driven. *System-driven* means that the acquisition of the user model is separate and prior to the interaction which attempts to achieve a user's goal. The dialog step need not be exclusively restricted to the acquisition of a model for the particular user. Prompting the user to characterize her-/himself (as was done by GRUNDY [RICH79a]) is just the special case of restricting a dialog phase or dialog step for building up the user model. In any case, separating the acquisition phase from the phase where the user model is exploited reduces the choices which the system has to make in order to behave cooperatively.

User-driven means that the user model is constructed during the entire dialog, e.g. simultaneously with achieving the user's goals. No particular steps are reserved for this part. Thus, the system does not know before understanding the utterance whether the utterance has to be analyzed by the component which constructs the user model. The need to recognize which input enriches the user model increases the complexity. Thus, a user-driven acquisition of the model of a particular user is more difficult than the system-driven one, because acquisition is integrated in the other activities of the system.

[11] "Revision" is used in broad sense here, too. It covers sophisticated truth maintenance systems as well as if-added procedures in frame systems.

3.7. Lessons from the General View

We have described conversational settings by the roles system and user can play. We introduced five change types and we distinguished two ways of acquiring particular user models. Now we want to use this framework in order to determine deficiencies in the field. Table 1 summarizes the application of the framework to related work.

Table 1

system playing a role in the domain	user playing a role in the domain	modeled aspects of the user	use of user model	change of user model	system-/user-driven	author
no	no	mutual beliefs	over-answering	addition	user-driven	[WAHL83]
simulated	yes	plan, knowledge	over-answering	no	system-driven	[ALLE79] [LITM85]
no	yes	plan,goals mutual beliefs	speech act generation	no	system-driven	[APPE82b]
no	yes	plan	reference resolution	no	system-driven	[GROS81]
no	no	wants	recommendation	revision	system-driven	[RICH79a]
simulated	yes	wants	recommendation	no	system-driven	[MORI85a]
no	yes	plan, knowledge	over-understanding	revision outlined	user-driven	[CARB*]
no	no	impression of rooms	commenting on rooms	no	system-driven	[JAME83]
no	no	(mutual) beliefs	description generation	addition	user-driven	[KOBS85a]
no	yes	level of expertise	advice	yes	user-driven	[CHIN*]
no	yes	plan, knowledge	advice	no	user-driven	[FINI83]
no	no	view on objects	correcting misconceptions	no	user-driven	[MCCO*]

Although it doesn't cover all the work in the field, it clearly shows some points where further research is urgently needed. Integrating *non-communicative actions* into the dialog pushes the need for speech act handling and a more flexible system design. If it is not a justified default assumption that the user's input is always information-seeking, a larger variety of speech acts are to be recognized. In an information-seeking dialog, informing and requesting acts are to be differentiated

by the propositional content (e.g. request for justification, meta-communication, request for an "inform-if", etc.). In an action-achieving dialog, speech acts other than informing and requesting come into play (e.g. command, recommendation, warning, etc.). Thus, the system must not only determine the propositional content but also the speech act. Moreover, the system has to decide which action it has to perform in order to act cooperatively. This requires a system architecture including a decision process (some sort of a dialog management component) which selects among communicative and non-communicative reactions. The embedding of language use into non-linguistic behavior thus comes to the foreground.

Integrating the acquisition of the user image into the normal processing of the dialog is lacking for almost all systems, which systematically model and use more aspects of the user than his dialog-dependent knowledge. Of course, the mutual belief space is built up during normal dialog processing. Modeling the user's beliefs and directly uttered goals is always integrated into dialog processing – if it is done at all –, because it is part of the dialog model [GROS86]. On the other hand, integrating "naive psychology" into the processing of dialogs widens the scope of the system. An additional aspect of what is going on in dialogs is then to be incorporated into procedures. Carberry's work [CARB*] tackles this problem with the user-driven acquisition of underlying plans and goals of the user.

The *revision* of user models is another weak point of current systems. Most of the systems which update the user model during the dialog do so merely by adding entries.[MORI85a] describes a modification procedure (cf. 4.1 below). Only [RICH79a, RICH*, CARB*] outline a revision procedure.

In summary, we see as shortcomings of accomplished systems:

- the integration of non-mental, non-communicative actions,
- the integration of the user model acquisition and the interpretation phase, and
- the revision of user models.

4. User Modeling in HAM-ANS

HAM-ANS is a dialog system. A dialog system can be contrasted with a question-answering system which is only able to handle questions posed to it in isolation. It requires constructing dialog memories during the dialog and using them for a variety of tasks, e.g. understanding and generating ellipses, determining reference and coreference. New objects are introduced into the dialog with a characterization and indefinite description. In order to keep track of what the user knows as a result of the dialog, the system builds a referential net for the partner (the user's domain model as known to the system) and establishes connections between the objects there and in the referential knowledge of the system (the referential part of the system's domain model). This knowledge about the user's beliefs is used for determining and generating the appropriate reference.

Whereas the dialog-dependent knowledge of the user can be handled in all applications, the user's wants are only modeled for the hotel scenario, being an instance

of the third setting, because only there the user is really concerned with the domain of discourse.

4.1. Modeling the User's Wants

In this section, modeling the user's wants as implemented in the hotel application of HAM-ANS is described.

It is well known from linguistics and philosophy that value judgements and several speech acts refer to some sort of evaluation standard [HARE52, GREW78, ZILL82]. In particular, recommendations rely on the presumed evaluation standards of the hearer. If a recommendation is to be actually implemented by the system and not just read into the system responses by the user, the system must be able to recognize the user's evaluation standards, derive the demands placed by the user on objects or situations.

In the hotel setting the user may well be asked whether a room is to be booked or not. Asking her or him for all her/his evaluation criteria, however, is a clumsy behavior. Rather, the criteria should be inferred from the user's utterances and should then be used for selecting "interesting" properties of rooms or of the hotel, generating the appropriate speech act, and deciding what speech act the user has performed.

A system which makes recommendations of novels is GRUNDY [RICH79a]. The basic assumption underlying the consultative function is that people are interested in books in which characters have the same type of personality as they themselves have. Therefore, recognizing the personality type of a user can be used directly for the evaluation of books. For HAM-ANS the utilization of knowledge about the user is not so straightforward.

As in GRUNDY, the specification phase in HAM-ANS is system-driven. The system asks the user about all the information which is normally required for a hotel reservation. No psychological characteristics are requested. The explicit specifications given by the user, i.e. from when to when, what kind of a room should be booked for whom, are stored as certain *facts* about the wants of the prospective hotel guest (who need not be identical with the user, of course). From these *facts* different independent *stereotypes* are inferred. These stereotypes are stored as assumptions about the hotel guest. They are not certain. The stereotypes differ from user classes [CHIN*] in that a single user belongs to several different *stereotypes*. They are mixed together in order to form the profile of the particular user. Thus, a user is not merely recognized as belonging to a certain class, but the user profile is constructed out of several stereotypes. This allows for representing a wider variety of user models in a flexible way.

In the next step, *evaluation criteria* are created from the stereotypes. Each stereotype creates a bundle of criteria. All the criteria are put together and are sorted into those concerning the hotel and those concerning the room. If two stereotypes produce the same criterion, its weight is strengthened. This is the basic *modification* process of the HAM-ANS user-modeling component that has been mentioned above. Some criteria are directly related to properties of the hotel or its rooms as they are stored in the referential knowledge. Others are expanded

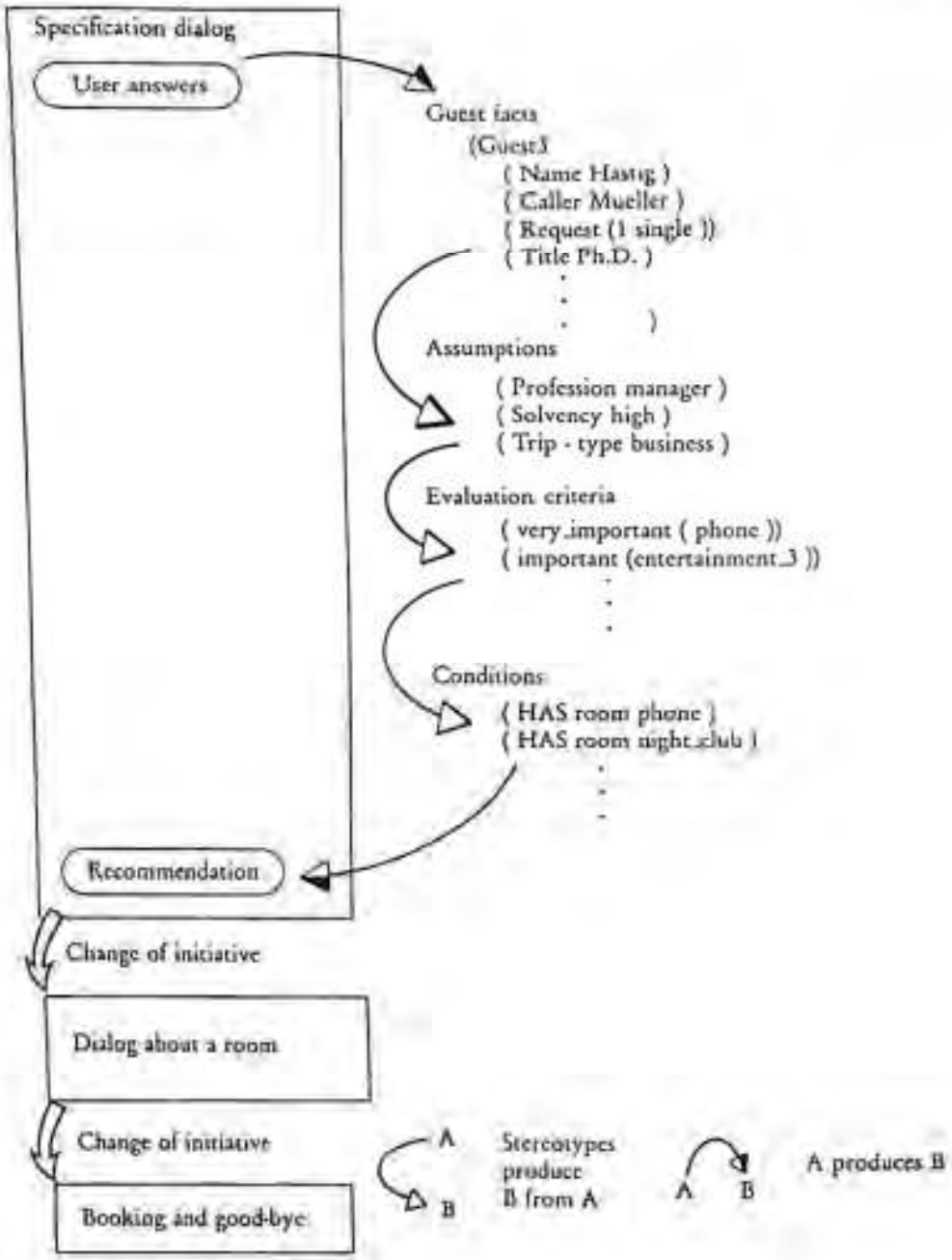


Figure 2. Modeling user's wants in HAM-ANS

by inferences. For instance, the entertainment opportunities are grouped: cultural entertainment, night life, family entertainment etc. Through inferences, they are specialized into, for example, theater, museum, opera. Thus user modeling in HAM-ANS is a multi-level approach: the given user facts, the supposed assumptions about the user, her/his evaluation criteria and finally the conditions which match the properties of the hotel or its rooms (see Figure 2).

The requirements of the prospective hotel guest as assumed by the system can then be used for different purposes. The relation to the dialog structure has been described elsewhere [MORI85a]. The most important purpose of evaluation criteria is the handling of speech acts. HAM-ANS, in its hotel application, is able to *generate* speech acts other than the inform act as an answer to a user's information request. According to the result of the match between the user's requirements and the properties of the hotel and its rooms, the appropriate recommendation is chosen. In order not to leave important information buried in the propositional content, the speech act *recommendation* is split up into

- strong recommendation
- weak recommendation
- restricted recommendation
- negative recommendation.

The result of the match between evaluation conditions and actual properties is a list which consists of two parts, namely the fulfilled and the unfulfilled criteria. This list serves as a justification for the selection of the appropriate speech act and determines the propositional content of the restricted recommendation. Only those features of the hotel or room which seem to be of interest for the user are verbalized. For example, in the dialog with a wealthy manager, the system cannot warmly recommend any of the room categories. It therefore comes up with a restricted recommendation as follows:

SYS: The room does not have a TV but a bar, a desk, a telephone and a seating arrangement.

The system doesn't mention the view, the radio, or the feather bed[12].

The representation of the assumed criteria and the derived requirements is a *qualitative* one. The information is not encoded in numerical weights but stored as contexts of semantic networks in the same way as the domain model, thus allowing for explanations.

4.2. User-Specific Overanswering

We have seen that the acquisition phase is system-driven. The system asks the user for the facts about her/his planned stay at the hotel. The system infers stereotypes and evaluation standards of the user and recommends the best suited room category. The system then passes the initiative to the user who may put questions about the particular room category. In this user-driven dialog phase, the inferred user requirements are of good use as well. They allow for determining the underlying intent of the question and thus for user-specific overanswering. See the following examples for illustration:

1. USER: Is there a TV in the room?
   SYS: No, but a radio.

[12] See [HOEP84a] for a detailed description of this and other HAM-ANS dialogs.

2. USER: Is there a desk in the room?
   SYS: No, but a table of a good height.

This behavior is accomplished as follows:

If the evaluation of a user question returns a negative result, the content of the question is matched against the user requirements. If this succeeds, an inference procedure is triggered which finds advantages of the desired facility. For instance, the height of a desk is marked as an advantage since it allows writing. With the new criterion "height=high", a transformed user question is produced and evaluated on the basis of the referential knowledge. Here, a high table is found as the next best facility and presented to the user. Note, however, that this kind of overanswering can be inadequate if the queried feature is not recorded in the user requirements. Look at the following example:

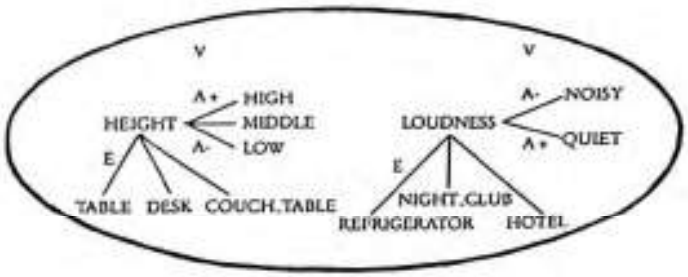
3. USER: Is there a night-club in the hotel?
   a) SYS: No, the hotel is quiet.
   b) SYS: No, but there is a night-club near the hotel.
   c) SYS: No.

Both answers, a) and b) sound reasonable. Which one is appropriate depends on the wants of the particular user. If a user asks whether p, s/he may want that p or that not p. It is not sufficient to take every question as a request for the feature in question! The *evaluation criteria* of the users – viewed as *frozen plans* – form a sound basis to decide between the two possibilities. If the user requires the hotel to have a night-club, the same procedure as for example 2) yields answer b). An advantage of a night-club is found (entertainment) and the referential knowledge is searched for next best entertainment facilities. The search succeeds because the environment of the hotel is represented as well in the referential knowledge about the hotel. If the queried feature does not match the user's evaluation standards, the system looks for a disadvantage of the feature in question. Given that night-clubs are loud, another value of the same dimension (loudness) can be found in the conceptual network. The inference procedure builds an expression with the new value of the same dimension, e.g. the hotel is quiet. This is matched against the requirements. If this succeeds, the system concludes that not the queried facility, but its opposite is desired instead. The system tries to confirm the new (opposite) property in the domain model. If this succeeds, answer a) can be generated.

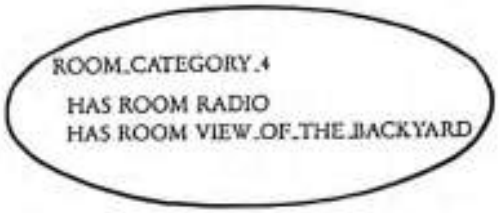
A prerequisite of this procedure is the conceptual knowledge that defines the dimensions with their possible values for concepts (see Figure 3). V-links represent the possible scalar attributes of a dimension. An example of a dimension is "color", the attributes being "red", "blue", "yellow" etc. E-links indicate which objects can have attributes of a particular dimension[13]. In the referential network, the link between a particular object and a particular value is given by the REF-link. Annotating scalar attributes with +/– allows for representing the extremes of a dimension (e.g. high(+)/low(–), quiet(+)/noisy(–)). Excerpts of knowledge sources are shown in

[13] Relativa are treated as scalar attributes, too, although linguistically this is, of course, not adequate.

Conceptual knowledge



Referential knowledge about a room category



Referential knowledge about a room



Referential knowledge about the hotel



Requirements



Isotopy - list



Figure 3. Excerpts from some knowledge sources

Figure 3. The extremes allow for finding the opposite value instead of the negated value (which could cover more than one value). Note that +/− does not prejudice the user's assessment of the corresponding attributes. Rather, the possible advantages and disadvantages of a feature are to be represented in other knowledge sources (PLUS, MINUS) which are also used by the inferences.

4.3. Integrating the Acquisition Phase into the Dialog Using Isotopy

So far, evaluation standards have been used to relate the user questions to his/her wants, thus disambiguating the intention underlying the question. But the questions may also be used for adding new evaluation criteria. Most often particles in user utterances have been proposed for deriving user requirements [JAME80, KOBS85a]. But what if the user does not type in the particles? In typed man-machine interaction, particles, such as "hopefully", are rarely found. Another well-known way is the use of precondition-action-effect sequences. In a static domain, where the only modeled action is to book a room, however, this does not apply.

Another approach, which I want to propose here, is to use the textlinguistic notion of "isotopy" [GREI66]. *Isotopy* is the co-occurrence of semantic features. Isotopy is a way of achieving coherence, as Greimas states. That is, a user does not put questions one after the other without any semantic relation between them. The coherence is established by properties which are common to the facilities mentioned in a sequence of questions. If neither a task structure [GROS81] nor an intended action [ALLE79], [LITM85] are known to the system these common features still form a basis for determining the interests of the user. They can be inferred from the domain model and the content of the questions in the following way. The dimensions referred to in a question (the dimension is found in the conceptual knowledge) are pushed into the *isotopy list*. The intersection of the dimensions involved in the subsequent question and the already stored dimensions is then viewed as a good candidate for the coherence establishing feature. The intersection becomes the first element of the isotopy list. The next question again adds a dimension which is intersected with the preceding ones, and so forth. In coherent dialogs, several questions refer to the same dimension.

This type of coherence in user utterances can be used for determining user wants, i.e. updating the evaluation standards of the user as a result of the dialog. Let us look at an example of a user-driven acquisition.

4) USER: Has the room view of the street?
SYS: No, view of the backyard.
5) USER: Fine.
SYS: May I then book a room for you?
6) USER: One moment, I have another question.
Is there a night-club in the hotel?
SYS: No, the hotel is quiet.

There is a common feature of the view of the street and the night-club inside the hotel, namely that they are loud. Therefore, the corresponding dimension,

loudness, comes to the foreground. The foreground reflects the point of interest of the user. The advantageous value of the dimension "loudness" is "quiet". The value is determined by a consulting the PLUS knowledge source. Therefore, in our example, we can add to the requirements as a new criterion that everything be quiet. This excludes noisy air conditioning as well as noisy refrigerators or night clubs. Of course, this new requirement is also processed by the modification routine. Recall that requirements are used for selecting the appropriate answer out of the set of true, non-misleading alternatives (4.2). In our example here, the answer concerning this feature is a). Even if no preferences as to loudness were known before, this time for question 6) which is the same as question 3) above, answer a) can be chosen instead of 3b) or 3c) (see above)[14].

If, on the other hand, the user had not responded favorably to a suggested view of the backyard, the answer would then have been c). The MINUS of a view of the backyard is that it is ugly. Nothing is known about the dimension "beauty" of a nightclub. So there is no feature common to the view and the existence of a night-club available. In this case, the user might have switched from "beauty" or "brightness" to another underlying concern ("amusement" or "loudness"), which might become clear in the next question. In this situation the evaluation standards of the user cannot be enhanced by a new requirement, but this does not disturb the answering process. Rather, it enables overanswering the next time by augmenting the isotopy list. A diagram of the overall process is shown in Figure 4.

The concept of isotopy, which is operationalized here as a coherence establishing dimension to which several questions refer, is a very general one. Greimas [GREI66] introduced it in the course of analyzing jokes. There, although actually two dimensions (to speak in our terms) are in the intersection of several propositions, only one of them is recognized by the hearer – until the last sentence, which intersects with *the other* one only. It is yet an open question how to handle cases like this, where two dimensions are in the intersection and, moreover, the first choice has to be revised.

We have shown how the general concept of isotopy can be exploited for the user-driven acquisition of the user model and for the choice of the appropriate answer. This method is applicable even in domains which do not allow for precondition-action-effect sequences.

5. Conclusion

Conversational settings in which both system and user play a role in the domain of discourse and where more than the beliefs of the user are changed require that speech acts other than the information request/informing act are handled. For a variety of these speech acts, evaluation standards are presupposed in speech act descriptions of linguistic theory [MORI82]. Therefore, operationalizing evaluation standards forms a basis for modeling a wide range of speech acts. Evaluation standards may be used

[14] The answer to question 4) as the one to question 1) above here counts as a direct system answer; this is due to the general overanswering capability of HAM-ANS ([WAHL83]).

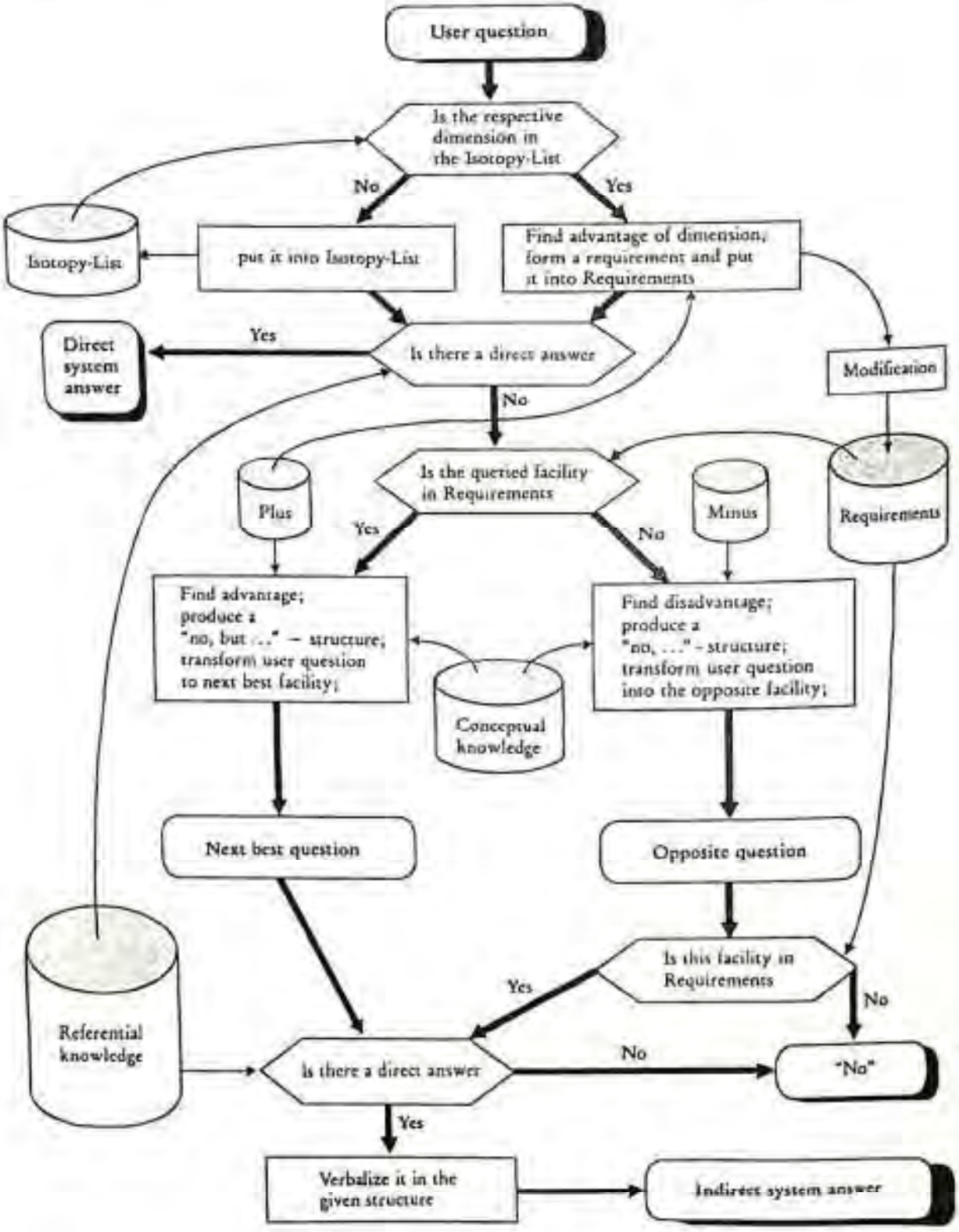


Figure 4. Overanswering using evaluation standards and isotopy

for speech act recognition *and* speech act generation which has been shown in this chapter for the example of recommendation speech acts.

Moreover, evaluation standards enable the system to decide whether a queried property or its opposite is required. This, in turn makes it possible to overanswer questions which otherwise would have to be just negated.

Evaluation standards vary from person to person. Therefore they are modeled as part of the user model: modeling the user's wants. A new way of user-driven acquisition of user-models has been proposed. Common properties of objects under discussion form a type of coherence, which has been called "isotopy" in textlinguistics. Isotopy can be used for determining the user's interests and choosing between alternative cooperative answers.

6. References

[ALLE79], [APPE82b], [BARB82], [CARB*], [CHIN*], [FINI83], [GREI66], [GREW78], [GROS81], [GROS86], [HARE52], [HOEP83b], [HOEP84a], [JAME80], [JAME82], [JAME83], [KOBS85a], [KONO84], [LEHM*], [LITM85], [MARB83], [MCCO*], [MORI82], [MORI84], [MORI85a], [MORI88], [NEUM83], [NOTH84], [POLL83], [RICH79a], [RICH*], [SCHG79], [SPAR84], [SPAR*], [WAHL83], [WAHL86a], [ZILL82]

14 Student Modeling in Intelligent Tutoring Systems — Implications for User Modeling[1]

Robert Kass

Abstract

User modeling and student modeling, though similar in many ways, have generally been pursued in parallel for the past decade. This chapter explores the role of student modeling in intelligent tutoring systems from the perspective of user modeling, with the goal of determining what researchers in user modeling can learn from student modeling. The chapter focuses on three issues: the information modeled about students, how that information is represented, and how the student model is built. Several intelligent tutoring systems that do student modeling are critically examined (in particular BUGGY, LMS, GUIDON, and WUSOR), and recommendations for user modeling based on this examination are made.

1. Introduction

For the past decade, user modeling in interactive systems and student modeling in intelligent tutoring systems have developed along similar, but seemingly separate, paths. Research in both user modeling and student modeling has been driven primarily by the needs of the respective systems that use the models. As a consequence, interaction between researchers in these two fields has been limited, although they seem to have many features in common.

This chapter is an attempt to bridge the gap between user modeling and student modeling. Here, student modeling is viewed from a user modeler's point of view, asking "What can user modeling research gain from student modeling?" In asking this question, three issues will be important:

1. What information about a student must be modeled?
2. How is the information represented?
3. How is the student model built?

These issues will be examined by studying several intelligent tutoring systems that do student modeling. The first four systems (BUGGY, LMS, GUIDON, and WUSOR) not only demonstrate the range of modeling approaches used in student

[1] This work was supported by a grant from the Digital Equipment Corporation.

modeling, but are considered to be "classics" in the field. Some later approaches are also examined to see the progress after, and the effects of, these classic systems.

This chapter is organized as follows. Section 2 gives a brief history of computer aided instruction and describes intelligent tutoring systems in general. Section 3 presents the general techniques used in student modeling for representing and building the student models. Sections 4 and 5 present several intelligent tutoring systems that do student modeling, discussing some of their contributions and limitations. Section 6 summarizes the features of student modeling with respect to the three issues mentioned above, and draws several conclusions about how user modeling can benefit from the work in student modeling.

2. Intelligent Tutoring Systems

Intelligent tutoring systems (ITS's) can best be characterized by differentiating them from other types of computer-aided instruction (CAI) systems.[2] The simplest (and oldest) type of CAI system is *generative*. Generative CAI systems were designed for situations requiring repetitive practice of a skill, such as vocabulary or arithmetic. From these systems evolved *adaptive* CAI, which makes use of information obtained from the student to guide the selection of problems, or the course of the tutoring sessions. Adaptive CAI systems might use simple techniques to estimate student's skill, and select problems based on this estimate. More complex syster use a branching structure to direct the flow of the tutorial. The student's answ to a question will determine which branch to follow in selecting the next que tion. Branches are predetermined by the tutorial author, who must anticipate all possible mistakes and combinations of mistakes a student might make. Adaptive systems are better than simple generative systems because they can tailor problems and information to the specific needs of each student – provided the author of the tutoring material correctly anticipates student misconceptions. Neither generative nor adaptive CAI systems have knowledge of the domain content: the information is supplied from an outside source in a fixed format.

The goal in building intelligent tutoring systems is for the system itself to be aware of the knowledge it contains, as well as what the student knows. Such systems should be "... capable of automatically *inducing* and *using* a structural model of the student's reasoning strategies" [BROW75, p. 2]. An intelligent tutoring system is capable of being "self-adaptive," in that it can adapt the tutoring session on the basis of self-contained knowledge. Tutoring involves three forms of knowledge [SELF74]:

1. Knowledge of how to teach (including knowledge about students in general),
2. Knowledge of what is being taught (domain knowledge),
3. Knowledge of who is being taught (student model).

This knowledge must be accessible to a tutoring system for it to make decisions about what problems or information to give, and how best to introduce material to the student.

[2] This classification of CAI systems is taken from [SLEE82c].

An ITS consists of four components: an expert module, a student model, a tutoring component, and a communication module [WOOL84b].

Expert Module. The expert module contains the knowledge the system has about the domain of instruction. This is usually the knowledge that a human expert in this domain would have, or at least the knowledge of a very good student. The expert module can be either a "black box" or a "glass box." A black box expert provides correct answers to questions, but the inner workings of the expert are not accessible. A classic example of a black box expert is the expert in SOPHIE-I [BROW76, BROW82], which is capable of simulating an electronic power supply. Situations are posed to the expert, which can determine how the power supply would behave, but not provide justifications for its answers. In a glass box module the reasoning of the expert can be observed and explored. For example, the GUIDON system [CLAN82] uses the expert system MYCIN (later NEOMYCIN) as its expert module, and is capable of following the expert reasoning by keeping track of the rules used to make a diagnosis.

Student Model. The student model contains information about individual students using the program. The student model must be able to indicate the abilities of a student with respect to the domain being taught, as well as student preferences for particular methods of tutoring.

Tutoring Component. The tutoring component is responsible for determining *what* information to give to the student, *how* that information should be conveyed, and *when* it is appropriate to provide such information. A good tutoring system will have a variety of strategies it may employ to help a student. The tutoring component not only contains knowledge of these strategies, but must be able to determine the strategy most appropriate for a given situation. This is especially true when the student is encouraged to discover new information. In these situations a student will frequently make mistakes, but the mistakes may be useful in furthering the student's understanding of the subject. A tutor must be able to decide when to let the student flounder a little, and when to interrupt and guide the student.

Communication Module. The communication module handles the actual communication between student and tutor. On output, it must ensure that information provided by the tutoring component is effectively presented to the student. On input, it must interpret the student's responses in a manner intelligible to the system. Since the student's view of the domain may be significantly different from that of the expert module, this task of interpretation may be quite hard.

Intelligent tutoring systems may interact with the student in a variety of ways. For example, the system may drive the interaction by leading a student through a series of exercises (such as SPADE-0 [MILL82], which guides a student through writing a program in the LOGO language), or it may interact with the student only when he requests help. ACE [SLEE77, SLEE82d] observes students analyzing Nuclear Magnetic Resonance (NMR) spectra, but only gives advice when the student asks for it. Often the tutoring system is viewed as a coach, observing students working on a problem, and interrupting to give advice when it believes the interruption will benefit the student. A large number of systems use this approach, including the WUMPUS Advisor (WUSOR) [CARR77, GOLD82], GREATERP (a Lisp tutor) [FARR84, REIS85], an advisor for the VAX/VMS operating system [SHRA82], and

a tutoring system for users of the ReGIS graphics display [HEIN85]. An ITS may support a mixture of interaction types, such as GUIDON [CLAN82], which can act as a coach, an advisor, or take direct control of the interaction to question the student, depending on the situation and the form of tutoring it determines to be the most effective.

3. Student Modeling

Most intelligent tutoring systems model the knowledge of the student in relation to their own domain or "expert" knowledge. This expert model is illustrated first, then various methods for relating student models to expert models, and approaches to building student models, are analyzed.

3.1. The Expert Model

Any tutoring system must have correct domain knowledge in order to know when the student is correct. In intelligent tutoring systems this correct knowledge is often treated as a built-in "expert." The expert knowledge represents a paradigm for what a student should know. For example, in the electronic board game WEST [BROW75], a student must combine (using arithmetic operations) three digits obtained from spinners to determine how many spaces to move on the board. The board includes special squares (cities and shortcuts) which, if landed on, advance the player even further. On the other hand, the player may bump an opponent backwards by landing on a square occupied by the opponent. The WEST expert can tell, for any given spin, how to combine the numbers from the spinners in order to achieve the best possible move. Using this knowledge, the tutoring component of WEST can tell whenever a player makes a suboptimal move. Thus the WEST expert plays the best possible game that could be played.

Expert knowledge may be divided into independent units. SOPHIE-III [BROW82] contains both an electronics expert and a troubleshooting expert. The electronics expert knows about general laws of electronics (such as Ohm's or Kirchhoff's laws), about how particular electronic devices behave (such as resistors, transistors and diodes), and about particular electronic circuits, while the troubleshooting expert has knowledge about diagnosing circuit faults in general. In the NEOMYCIN knowledge base used by the GUIDON2 tutor, knowledge of the specific domain is separated from strategic knowledge of how to diagnose a disease.

Sometimes the expert model should not truly be an expert in the domain. Anderson [ANDJ85a] points out that in tutoring programming to novices, the problem solution of a programming expert is often too sophisticated for a novice to understand. Thus the basis of knowledge for teaching a novice should be knowledge that can be grasped by the student. In GREATERP [REIS85] the "expert" is actually an *ideal student*. The knowledge of the ideal student is what would be expected of a very good student of programming. This knowledge can be understood by the novice programmer, yet still stand as a paradigm of how to program.

3.2. Relation of the Student Model to the Expert Model

A tutor's goal is for the student to acquire the skills and knowledge necessary to successfully perform tasks in the domain of tutoring. The method adopted by intelligent tutoring systems for achieving this goal is to induce in the student a model of the domain matching the system's expert model. Since this expert knowledge must be available to the tutor anyway, the student model in an ITS is usually built in relation to the expert model held by the system. Three techniques are used to relate the student model to the expert model in a tutoring system: overlay modeling, differential modeling, and perturbation modeling.

3.2.1. Overlay Modeling

The overlay model, proposed by Carr and Goldstein [CARR77],[3] is the simplest of the modeling techniques. Overlay modeling assumes that all differences between the student's behavior and the behavior of the expert model can be explained by the lack of a skill or skills on the part of the student. Thus the knowledge of the student is viewed as a *subset* of the expert's knowledge. Since overlay models do not require information beyond the contents of the expert model, they are relatively easy to implement. An overlay model is illustrated in Figure 1.

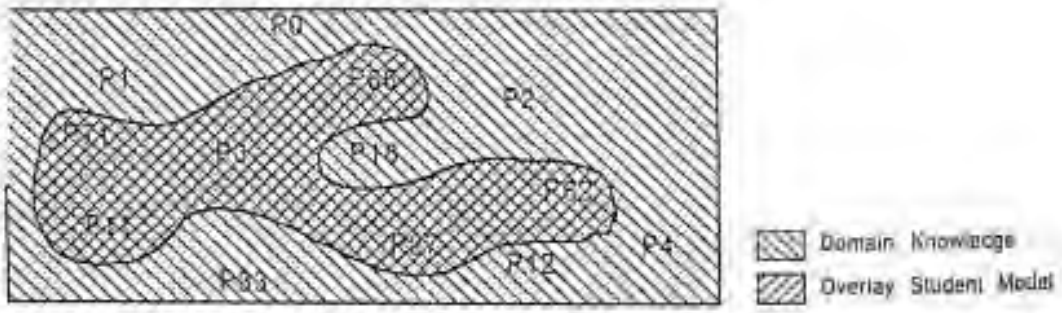


Figure 1. An overlay student model

The overlay technique was used by Carr and Goldstein to implement WUSOR-II, the advisor for the game "Hunt the Wumpus." This game is played in a warren of interconnected caves. A player must search through the warren to find and kill the deadly Wumpus. The Wumpus is killed by shooting an arrow into the cave occupied by the Wumpus. If a player walks into a cave containing a pit or the Wumpus, he loses. If he enters a cave containing bats, he will be carried off by the bats and dropped at random into another cave. To help the player, clues are given when he is in a cave adjoining a cave containing bats or a pit, or when he is within

[3] Although Carr and Goldstein coined the term "overlay model," the technique was used earlier by Carbonell in the SCHOLAR program [CARN70a].

two caves of the Wumpus. New players need to acquire a variety of skills to play the game well, such as: how to search the warren, how to deal with single and multiple evidence of danger, and how to apply probabilistic reasoning in making choices. The expert possesses all of these skills.

Overlay modeling works well in WUSOR-II, because (1) the goal of the tutoring system is to impart the knowledge of the expert, (2) the expert truly is an expert in this situation, and (3) the expert model is simple enough that, with time, even a novice can grasp it. The reasoning done by the expert will lead to playing the best game possible, hence the goal of WUSOR-II is to impart this knowledge to the student. Thus, any failure on the part of the student is attributed to the student's missing a skill which the expert has.

The difficulty in building an overlay model for a particular student depends on the type of information in the model. If the information is at the surface level, such as modeling the student's knowledge of program commands (as ANCHISES does for the EMACS text editor [ZISS85]), the student modeler can simply mark what the student knows as he uses the terms. On the other hand, if the expert model contain deep knowledge, such as strategies for diagnosis (as the NEOMYCIN expert model for the GUIDON2 tutor has [LOND82]), recognizing what knowledge the expert and student share can be much more difficult.

Overlay modeling has serious drawbacks. An overlay model has no provision for dealing with knowledge or beliefs of the student that differ from the expert' If the student has an alternate strategy for Hunt the Wumpus that plays well, overlay model will not recognize it, perhaps causing the tutor to try "correctin the student. If the player simply has a wrong strategy the overlay modeler can dete it, but the tutor will not be able to point out why the strategy used by the student is wrong, because it has no knowledge other than that of the expert. Thus overlay models are ultimately incomplete.

3.2.2. Differential Modeling

The differential model is a modification to the overlay model. Instead of comparing student knowledge with expert knowledge, the differential model is produced by comparing the performance of the student with the expert's performance in the same situation. Differential modeling divides system knowledge into two classes: knowledge the student *should* know (if the student is to behave like the expert), and knowledge the student could not be expected to know. The knowledge a student is expected to know is just the knowledge the expert model has used to produce its behavior. Overlay modeling assumes that any knowledge in the expert model but not in the student model is not known by the student. Differential modeling subdivides this class into knowledge the student could not be expected to have, and knowledge the system believes the student does not have. A differential model is illustrated in Figure 2.

A differential model is used in the WEST game described earlier. As the student plays the game, the expert model also simulates playing the game from the viewpoint of the student. If the student, for example, consistently fails to make use of shortcuts the expert uses, the tutor can conclude that the student lacks knowledge about how to use shortcuts. On the other hand, if the student consistently fails to bump the

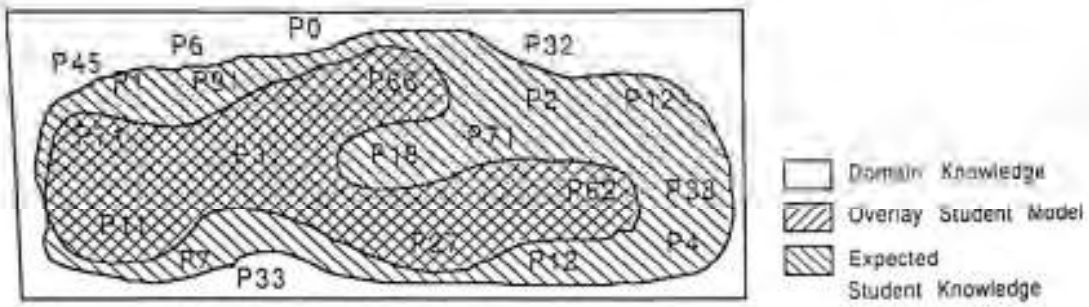


Figure 2. A different student model

opponent's piece, while the expert, given the same situation, also avoids bumping, then no conclusion can be drawn about whether the student has knowledge of how to bump opponents or not.

The differential model avoids the problem of being overly presumptuous about the knowledge of the student, but still suffers from most of the difficulties of the overlay model. Differential modeling still assumes that the knowledge of a student is only a subset of the knowledge of the expert. Such an assumption leads to a student model that is incomplete, since it cannot represent any belief of the student outside the domain knowledge of the expert.

3.2.3. Perturbation Modeling

Perturbation models can be used to represent student beliefs beyond the range of the expert model, while maintaining a close link between the student and expert models. Perturbation modeling builds a student model similar to the expert model, differing only on the basis of small perturbations to some of the knowledge in the expert model. Thus the student and expert models are basically the same, with small differences in some areas of the knowledge base. The perturbation model subsumes the overlay model in that a student's lack of knowledge is viewed as another form of perturbation. A perturbation model is illustrated in Figure 3.

Many existing intelligent tutoring systems use perturbation models. The common technique is to encode the expert knowledge, then augment that knowledge with likely misconceptions or "bugs" the student might have. For example, the DEBUGGY system [BROW78] uses a procedural network (which will be described shortly) to represent the correct methods for performing subtraction. Attached to this network is a set of procedures representing common mistakes made by students performing subtraction. These bugs are the perturbations to the expert model.

The Leeds Modeling System (LMS) [SLEE81] uses production rules to represent the correct methods of algebra, and *mal-rules* for the common misconceptions students have about how to solve algebraic equations. Mal-rules play the same role as the buggy procedures in DEBUGGY. Other systems make use of a *bug library* to represent the common misconceptions of students in a domain. Exam-

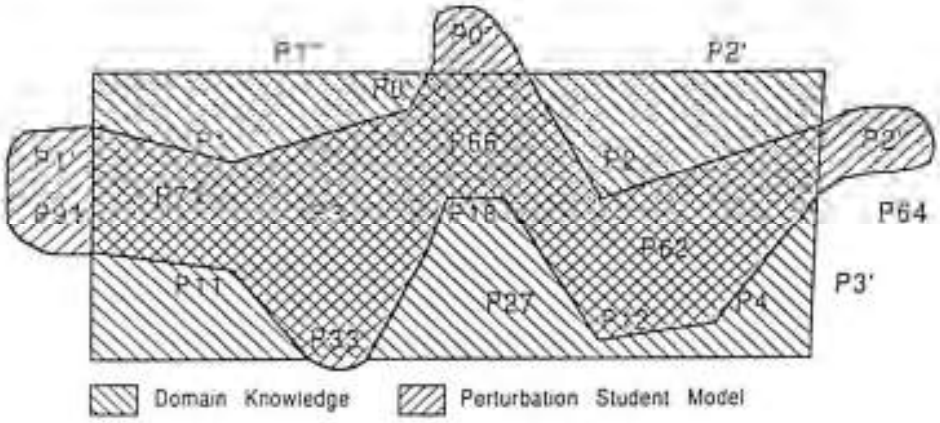


Figure 3. A perturbation student model

ples of such systems are PROUST [JOHN84], the Geometry Tutor [ANDJ85b], and GREATERP [REIS85].

Perturbation modeling complicates the process of building an individual student model, because it introduces more ways to explain the student's behavior. An error in the student's performance may be due to a missing skill, as with the overlay and perturbation models, or it may be due to the student's use of an incorrect skill. Perturbation modeling expands the space of possible student models greatly, requiring special techniques to reduce the number of models that must be searched. In fact, one system (LMS [SLEE83]) approaches the problem of modeling a student as one of controlling the size of this search space.

The perturbation model still assumes that a student and the expert are similar enough that a comparison can be made. This assumption has intuitive appeal. In human interaction, people expect other individuals to be basically like themselves. If no common ground can be found, communication usually falters. In a teaching environment, when the teacher recognizes that a student has no conception of what he is talking about, he will try to drop back and start over at a lower level until enough common understanding exists to work back to the original subject. Thus perturbation modeling seems to be a reasonable and intuitively attractive method for modeling students.

4. Four Intelligent Tutoring Systems That Model Students

In this section four intelligent tutoring systems that use student modeling are analyzed in some detail. These systems were chosen because together they represent the range of approaches to student modeling used in intelligent tutoring systems. Each of the systems is also a "classic" in the field, although the oldest was built less than ten years ago. The four systems are:

1. the BUGGY programs: BUGGY, DEBUGGY and IDEBUGGY
2. the Leeds Modeling System (LMS)
3. GUIDON, and GUIDON2
4. WUSOR (version III)

4.1. BUGGY

The BUGGY programs were designed to model the incorrect ("buggy") subtraction skills observed in the performance of children on tests.[4] Three programs were built during the course of the project. BUGGY, the most commonly mentioned program is actually a program to help arithmetic teachers recognize student bugs. BUGGY selects a bug, then asks the teacher to give it sample problems to solve using this incorrect procedure. A teacher must select problems that isolate the error, tell the program when they think they have found the bug, then prove that they have found it by simulating the bug on several problems generated by the system. The DEBUGGY program reads a student's test answers and proposes the bug or bugs that most likely caused the student's errors. IDEBUGGY is an interactive version of DEBUGGY, with the added capability of dynamically generating problems to help identify student bugs.

4.1.1. Description of BUGGY

The goal of the BUGGY programs is to build a *diagnostic model* of the student: a model of the internalized set of incorrect instructions or rules capable of duplicating a student's behavior. A successful diagnostic model can predict not only *which* problems a student should get wrong, but exactly *what* the wrong answer will be.

The diagnostic model is represented using a procedural network [SACE77] to simulate the behavior of a student. A procedural network consists of a *conceptual* part, used by BUGGY to represent the subskills used in subtraction, and an *operational* part, consisting of methods (both correct and buggy) for performing the action associated with the conceptual part. In the BUGGY programs, this network formed a lattice called the *skill lattice*, representing all possible ways a student could attempt to solve subtraction problems. When a student's problems are presented to DEBUGGY or IDEBUGGY, various combinations of good and buggy methods are tried to find a complete procedure that most closely matches the performance of the student. A set of heuristics reduces the number of models to be tried by using features of the domain, such as dependencies or incompatibilities between the subskills. Burton [BURT82] reports that eventually the DEBUGGY system recognized 110 primitive bugs, and was able to directly recognize 20 common multiple bugs, plus any combination of up to four primitive bugs.

The student modeling requirements for the BUGGY systems are not great. Typically only a small set of problems is available for use in determining the buggy procedure used by a student. Acquiring a specific model for the student consists of

[4] The work discussed here is reported in [BROW77, BROW78, BURT82]. Of these, [BROW78] is the most commonly referenced work and contains the majority of the results of the work. However, [BURT82] is a good summary of the entire project.

searching for a path through the skill lattice that best matches the performance of the student. Because the procedural network includes buggy methods in addition to the correct methods, the BUGGY systems use a perturbation model for modeling the student.

4.1.2. Contributions of BUGGY

Several features of the BUGGY systems are important and interesting contributions to the use of student models in an ITS, and for intelligent tutoring systems in general. Perhaps the greatest contribution is the overwhelming empirical evidence that students do make systematic errors in doing multi-column subtraction. Before this work, many teachers believed that students who performed poorly made random errors due to lack of intelligence or lack of interest. Brown and Burton proved that even many of the worst students were very consistent in applying a procedure to solve problems. Their poor performance was due to using an incorrect procedure. This observation is important for the future of student modeling, since it implies that student errors are due to identifiable misconceptions. Furthermore, student behavior is consistent, hence those misconceptions can be recognized and modeled.

The DEBUGGY system also introduced a method for dealing with noise in student test data. Even though students were consistent in their buggy procedures, they still made occasional mistakes due to copying or number fact errors. DEBUGGY has a set of *coercions*, reflecting typical mistakes of this sort, which it can apply to a problem. These coercions help to reduce the problem that random noise can bring to the modeling process.

The BUGGY systems also provide a detailed model of the subtraction procedure. One result is the realization that even a simple procedure such as subtraction can be quite complex, and there are many ways that a student can get it wrong. The skill lattice for representing the possible correct and buggy procedures of students had the side-effect of being useful in determining how effectively a test measures each of the component subtraction skills.

4.1.3. Criticisms of BUGGY

Several criticisms can be leveled at the BUGGY systems as well. The bugs listed by Brown and Burton tend to be superficial. Many of the bugs seem to be different surface manifestations of a few core bugs. The model can be used to reproduce student mistakes, but not to explain *why* the student has made those mistakes. Thus BUGGY does not provide a deep model of student subtraction procedures.

The expert knowledge of the system, although encoding the procedural knowledge necessary to do subtraction, is a black box – it cannot be used to justify or explain the reasoning of the expert or the student.

All knowledge for this system had to be hand-coded. People had to pore over student tests and try to determine what bug or combination of bugs could account for the answers given. When bugs were discovered they were added to the system. There is no ability to augment the knowledge of the system directly from the interaction.

4.2. LMS

The Leeds Modeling System (LMS) [SLEE81, SLEE82a, SLEE83, SLEE84] was designed to be a vehicle for exploring student modeling capabilities. LMS approaches student modeling as a search problem, with two goals oriented towards making the search computationally feasible: (1) contain the search required in creating a student model, and (2) have the modeler behave in a data directed manner.

4.2.1. Description of LMS

LMS uses production rules to represent the correct skills necessary for solving simple algebraic equations. Augmenting these correct rules are additional *mal-rules* representing possible misconceptions the student might have. Thus LMS uses a perturbation model for representing the knowledge of students. Building a student model consists of selecting from the set of rules and mal-rules an ordered list of production rules capable of simulating the performance of the student on algebra problems.

Modeling in LMS occurs in two phases. The first phase occurs offline, when LMS generates a complete set of possible student models from the correct rules and mal-rules, retaining those that give unique results to predefined problem sets. The second phase occurs online, using the models generated by the offline phase to letermine whether the student's behavior can be explained by any of the models enerated. The online phase proceeds in stages, with each stage designed to augnent a growing model of the student. At each stage the student must solve a set of problems designed to test a particular algebraic skill. For example, at one stage the student may be given a set of problems that require him to move a number from the left hand side of the equals sign to the right by adding the additive inverse of the number to both sides of the equation. A mal-rule for this operation might move the number from left to right, without changing the sign of the number.

The power of LMS is that it uses a *systematic* search to build the model of the student. Unlike DEBUGGY, which used heuristics to limit the number of models that had to be searched, LMS systematically searches the space of possible models, using an incremental method to build the student model that effectively limits the size of the search space. At each stage in the online phase of modeling, the potential models are limited to models generated from the existing model by adding either a correct or buggy rule for the skill being tested, or perhaps a mal-rule corresponding to an earlier skill.[5]

LMS was used by a small number of students who were later interviewed by educational psychologists who had not seen their performance with LMS [SLEE82a]. The object of this experiment was to compare the student models generated by LMS with those inferred by the interviewers. This experiment demonstrated that LMS still lacked several mal-rules. When those rules were added the models developed were similar.

[5] This last capability was added when it was discovered that even though a student may perform a skill correctly once, later they may use an incorrect method to perform that skill.

Further studies by Sleeman reported in [SLEE84] indicate that the simple production rule model is insufficient. Analysis of further student's algebraic mistakes, and more interviews, suggests that there are four classes of student errors in algebra.

1. *Manipulative errors*: where a variant of the correct rule is used, with one stage omitted or replaced by another.
2. *Parsing errors*: students frequently misunderstood the notation of the problem.
3. *Clerical errors*: errors due to miscopying, etc.
4. *Random errors*: errors for which no explanation can be given.

Only the manipulative class of errors is explained by the original LMS model. Subsequently, additional mal-rules were added that account for most manipulative and parsing errors.

Matz [MATZ82] has suggested that many parsing errors are a result of students making incorrect extrapolations from the domain of arithmetic to that of algebra. Sleeman calls this process *misgeneralization* and suggests it is another source of common student errors. Misgeneralization is a mechanism that can explain not only the behavior of a student, but also the underlying reasons for the behavior. This psychological knowledge is much more useful to a tutoring system, for the system is now capable of directly addressing the root of the student's problem.

4.2.2. Contributions of LMS

LMS makes three major contributions to the study of student modeling. First, it is possible to constrain the search space in constructing a model such that the search technique is computationally feasible.

Second, a data-driven modeler is possible. In particular, the model building machinery of LMS is relatively domain independent. A production system could be written for other situations in which a procedural skill is taught and LMS would deal with them equally well. On the other hand, this technique might not work well with domains containing a lot of factual knowledge, or in domains that are not well defined.

Finally, student errors may be of different classes, representing fundamental differences in the underlying cause for the error. Developing a psychological theory for student errors will be an important step to truly effective modeling.

4.2.3. Criticisms of LMS

Partly because LMS focuses on a particular aspect of the modeling problem, there are many criticisms which can be raised about the system.

The models produced are not psychological models of student learning. Hence, even though a suitable model can be built for simple algebra, there is no theoretical basis for claiming that LMS should work in other domains. Furthermore, although the model for algebra can explain the behavior of a student, it gives no indications as to why students have this particular set of mal-rules, or why students in general make some types of mistakes but not others. Sleeman recognizes this problem in [SLEE84], but provides no solution in LMS.

LMS suffers from the possibility of non-monotonicity in the model. As the model is built incrementally, it is possible that a later rule may subsume or supercede a rule already in the model. Such an interaction might cause the model to no longer explain student performance previously observed. A solution to this seems to be more extensive testing of the model when a new rule is added, to ensure that previous explanatory ability has not been lost.

A very practical problem with LMS is the amount of hand-coded information which must be explicitly provided by the system builder. Not only must the designer encode the basic rules and associated mal-rules of the domain, but also the many problems used to distinguish between various rules during the stages of online model building. This requires the builder to be able to anticipate a good deal of the student's behavior in a manner similar to designers of branching adaptive CAI programs.

4.3. GUIDON

The GUIDON [CLAN81, CLAN82], GUIDON2 [LOND82], and later systems [CLAN86a] developed as an attempt to use MYCIN-like knowledge bases as a basis for tutoring medical diagnosis. The overall system has been considered fairly successful, but only after some significant changes to the representation of the MYCIN knowledge itself.

4.3.1. Description of GUIDON

GUIDON plays the role of a *case method* tutor. Students who use GUIDON are given an initial description of the symptoms of a patient suffering from some strain of meningitis (the domain of tutoring). This is the *case*. The student then proceeds to ask questions in an attempt to develop a hypothesis about the particular form of disease the patient has. GUIDON observes this dialog and may interrupt to ask questions in order to get the student to clarify his/her strategy, to give advice, or even redirect the dialog. Thus GUIDON is basically a coaching type of tutor.

GUIDON2, the successor to GUIDON, uses a differential approach to modeling the student. The expert model is used to generate a reasonable *range of predictions*, the set of actions and hypotheses that GUIDON2 expects the student *should* have because they are what the expert *would* have [LOND82]. Usually this range of predictions will be sufficient to model the student's behavior. If not, GUIDON2 starts a data-driven search to find a new range which will.

An important discovery of the GUIDON project is that a MYCIN-like knowledge base is not sufficient to provide a good basis for tutoring. Because of the explanation facilities available in MYCIN, it seems reasonable to expect that this rule based knowledge would be a good foundation for intelligent tutoring systems as well. The work with GUIDON proved these expectations to be incorrect, for two reasons. First, MYCIN rules are very shallow, skipping many of the intermediate reasoning steps that experts make. Secondly, MYCIN rules are not homogeneous. The rules in MYCIN mix several types of disparate knowledge, often in the conditions of the same rule. These problems with MYCIN led to the development of a new knowledge base representation, called NEOMYCIN [CLAN83].

NEOMYCIN is designed to be a *psychological model* of how a medical expert diagnoses diseases. In NEOMYCIN, domain knowledge is classified into several categories, and is separate from the knowledge pertaining to diagnosis in general. Protocols of experts diagnosing patient diseases were used to create a set of *meta-rules* that control the overall behavior of the NEOMYCIN expert. Analysis of these protocols indicate the expert has an overall goal of determining what information to seek next. The expert thinks in terms of a hierarchy of causes and process characteristics of a disease, and uses this information to order the data and search process in developing a diagnosis. This knowledge is independent of the domain and should be applicable to other situations where GUIDON is used as a tutor.

The domain knowledge of NEOMYCIN is grouped into several sections. An etiological taxonomy is present reflecting the causal relationships between various concepts relevant to the domain. This taxonomy contains many of the intermediate concepts an expert uses when reasoning from symptoms to an actual disease. The domain knowledge also includes world facts important for the diagnosis. The domain rules of the system are classified into one of four categories:

1. *Causal* rules linking observed symptoms with concepts in the etiological taxonomy.
2. *Trigger rules* that associate data with the etiologies. These data correspond to working hypotheses the expert has.
3. *Data/hypothesis* rules that associate circumstantial data and laboratory data with diseases in general.
4. *Screening* rules used by the expert to control the search.

GUIDON also includes knowledge specific to managing the student model. Since a differential model is used, the knowledge of the student is considered to be a subset of the knowledge of the domain expert. The modeler (called IMAGE in GUIDON2) contains a set of rules to update the model after each step in the dialog. These rules use the same form of evidential reasoning as MYCIN, so the modeler can include degrees of belief for elements in the student model.

4.3.2. Contributions of GUIDON

GUIDON has made many contributions in the area of student modeling. A primary contribution has already been mentioned: the discovery that a MYCIN-like knowledge base was insufficient to support intelligent tutoring. In particular this means that a rule-based knowledge representation by itself is not sufficient to ensure a good expert model for tutoring. Typical production rule systems tend to be very shallow in their knowledge. Such knowledge may be effective for an expert system, but does not form a psychological model of how people structure knowledge about the domain.

A second contribution of GUIDON is the presentation of a psychological model of how medical experts perform diagnosis. This model, NEOMYCIN, demonstrates the need to separate several disparate types of knowledge, not only between the domain and strategy components, but also in the domain itself. It also demonstrates the need to structure factual knowledge to complement the rule-based knowledge.

Thus student modeling will require a very sophisticated and diverse knowledge representation – no single, simple representation will do.

Finally, the use of evidential reasoning to build the student model gives GUIDON a tolerance for noisy data. Since noise is an important problem in modeling a student, the explicit evidential reasoning used by GUIDON seems appropriate.

4.3.3. Criticisms of GUIDON

Although GUIDON satisfies many of the criticisms of the previous systems, there are still some inadequacies. First, since GUIDON uses a differential model with no bug library, the student model is limited to a subset of the expert model. This assumption is too strong, even for medical school students with a good background in the domain.

GUIDON assumes its students are completely familiar with the concepts, factual data and laboratory tests referenced during the tutorial. Thus the tutoring is limited to the diagnostic process itself, and not factual concepts in the domain of diagnosis.

The system cannot evolve new notions of possible student models. Thus GUIDON could not "discover" that a student was using a rule which was unknown by GUIDON and remember this rule for later modeling. A truly good system with this capacity could recognize when a student used a new, valid rule the system was not familiar with. Such a rule could then be added to the expert's knowledge base, allowing the expert to learn as well [KIMB82, OSHE82].

Finally, GUIDON has no sense of the learning limitations of the student. After going through a long session a human tutor would expect the student to be tired and less capable of assimilating new knowledge. GUIDON does not have a model for student learning and forgetting, or for the performance of a student under different cognitive loads.

4.4. WUSOR

WUSOR is an advisor for the logical game "Hunt the Wumpus." A short outline of this game was given earlier. Three versions of WUSOR have been built. The first used no modeling techniques and the second [CARR77] used a simple overlay model. The third version, WUSOR-III is more sophisticated than the previous versions, is the most interesting with respect to student modeling, and hence will be the focus of this section.

4.4.1. Description of WUSOR-III

The modeling in WUSOR-III [GOLD82] focuses on how knowledge is acquired by a student. Knowledge of how a student learns information in a domain enables accurate prediction of what items an individual will learn next. This greatly aids model building, since it serves as a strong constraint on the search space of possible models. It also aids the intelligent tutor, by giving a clear indication of topics it would be appropriate to provide help on, and topics the student is likely to misunderstand at a given point in the session.

The WUSOR-III coach consists of five distinct modules. These include the expert and student models and the tutoring component itself. In addition, a module called the *psychologist* is responsible for updating the student model. The fifth module, the *genetic graph*, is unique to WUSOR. The genetic graph is a data structure containing the domain knowledge used by the expert. On top of this domain knowledge (represented as rules), links that relate the rules in various ways are added. The graph is "genetic" in that it represents the relationships between concepts from an evolutionary viewpoint: concepts are linked via the methods by which they may be learned.

The genetic graph contains four types of links:

1. *Generalization*: R' is a generalization of R if R' is obtained from R by quantifying over some constant. *Specialization* is the inverse of generalization.
2. *Analogy*: R' is analogous to R if there exists a mapping from the constants of R' to the constants of R.
3. *Refinement*: R' is a refinement of R if R' manipulates a subset of the data manipulated by R. *Simplification* is the inverse of refinement.
4. *Deviation*: R' is a deviation of R if R' has the same purpose as R, but fails to fulfill that purpose in some circumstances. *Correction* is the inverse of deviation.

These links represent the way individuals learn new concepts from concepts they already know. Thus the genetic graph is proposed as a psychological model of how students acquire knowledge of the domain.

The structure of the genetic graph also provides information about student learning. The connecting links between domain rules naturally represent the interrelationships between concepts in the domain. Usually there will be areas of the graph with a large number of interconnections. These groups are called *islands*. Islands represent small bodies of closely related material that tend to be learned in units. Islands thus provide a hierarchical structure to the knowledge of the domain, indicating areas of knowledge which a student will learn, and the underlying finer concepts.

WUSOR-III models the student as a pair of specialists, one for problem solving and the other for learning. The student model is constructed as an overlay of the genetic graph. The knowledge of the problem solving specialist is simply an overlay of the domain knowledge, which constitutes the nodes in the graph. The learning specialist consists of a set of strategies for learning new material. The types of links between nodes in the genetic graph reflect the particular learning strategies that can be used to learn domain information in the graph. The student model also overlays these links in the graph, indicating the learning strategies used by the student in acquiring new information about the domain.

The genetic graph represents the evolution of a student's knowledge. Initially, the student model will be an overlay of only a few nodes in the genetic graph. As the student acquires new knowledge, the student model will grow along the paths linking previously acquired knowledge with new knowledge. Thus at any time in the learning process a "frontier" exists in the genetic graph, reflecting the current working knowledge of the student. Nodes far to the interior of the graph will tend to be superceded by those on the frontier. The active model of the student at any given time is thus the frontier of the student model in the genetic graph.

4.4.2. Contributions of WUSOR

WUSOR's main contribution to student modeling is the notion of the genetic graph. As with GUIDON, the genetic graph represents a psychological model of the student, but the genetic graph mechanism focuses not only on student knowledge, but student *learning* as well. Thus the genetic graph provides a framework for representing how a student acquires knowledge and skills in a domain.

A further advantage of the genetic graph is its ability to predict what knowledge the student is likely to acquire next. The frontier of the graph serves as a means for limiting the search necessary when attempting to recognize the knowledge of a student. By the same technique, the frontier aids in selecting tutoring material, since it provides a set of items most likely to be relevant and understandable to the student at a given point in the session.

The structure of the genetic graph can be used to better understand the complexity of the domain. Domains with strongly connected genetic graphs should be easier to learn than those having relatively few connections. Thus islands in the graph will tend to be learned as units, while shifting from one island to another may be very difficult. This shift in complexity may explain why students tend to have difficulties of particular sorts, or with particular material.

A final, potential advantage of the genetic graph structure is that it may be used upon itself. The domain of the genetic graph could be the learning strategies themselves. Thus a system could tutor not only an underlying domain, but in addition the learning skills used to learn about the domain.

4.4.3. Criticisms of WUSOR

Unfortunately, WUSOR stills suffers from many problems. First, the genetic graph is based on an overlay model. The incompleteness of such an approach seriously limits the modeling of abilities of WUSOR.

The knowledge representation of the underlying domain is unstructured. WUSOR uses simple production rules for representing its knowledge. As Clancey has pointed out, deep structure in the domain knowledge is needed, as well as multiple knowledge representation techniques. Extending the genetic graph for such an underlying domain may not be possible.

WUSOR suffers from the noise problem. Although the frontier of the student model in the genetic graph should represent the current knowledge of the student, this knowledge is often uncertain. Students acquiring new concepts may alternately use them correctly, then incorrectly. Furthermore, students frequently do not apply what they learn immediately. Thus knowledge on the frontier of the student model may very well be known by the student, but the lack of use will lead the psychologist module to doubt that knowledge.

As with the other systems discussed in this section, all knowledge must be predefined by the system authors. The WUSOR system is not capable of recognizing any student behavior beyond the knowledge it has stored in the genetic graph. For WUSOR, this means the author must not only encode the domain knowledge, but also all possible links that connect this knowledge. Such a task would prove very daunting in practice.

The genetic graph is rather ad hoc. Different types of links can be created to meet various needs. There is no theoretical basis for the types of links in the graph, the justification of their presence is their intuitive appeal.

5. Further Work in Student Modeling

One point made very clear from the work reviewed in the last section is that deep psychological models of students are needed to provide effective tutoring. This means that the model must go beyond simply simulating student behavior: it must be able to explain the underlying causes for that behavior. Since the work in the four "classic" systems described in the previous section, several other researchers have begun to address the problem of building deep student models. In this section three recent approaches are presented: the work of Johnson and Soloway on PROUST, the work of Anderson *et. al.* on GREATERP and the Geometry Tutor, and Brown and VanLehn's Repair Theory.

5.1. PROUST

PROUST [JOHN84, JOHN85] was developed to identify non-syntactic bugs in novice student's programs. To do this, it must cope with the high degree of variability that occurs in programming, yet correctly identify where a student goes wrong in writing a program. Accurate tutoring in this situation requires that the tutor not only identify what has gone wrong, but also determine what the *intentions* of the student were in writing the program in this way. Plan recognition by itself is not sufficient, because the plans may be incorrect implementations of correct plans, or correct implementations of plans to achieve an incorrect goal. Thus PROUST not only models the plans students use in writing programs, but also their intentions in using those plans.

PROUST uses a *goal structure* and a knowledge base of relations between goals and plans to identify program bugs. Problem descriptions are given to PROUST, including a specification of the problem in a special goal-oriented language. This problem specification is then combined with the knowledge base of information about goal and plan interactions to determine the possible methods for solving the problem in terms of subgoals and subplans. When a student program is analyzed, PROUST selects a subgoal from the problem description and attempts to find program statements in the student program that satisfy it. When a match between the subgoal and the student's code cannot be found, PROUST attempts to find a match using a *bug library* of typical errors. It will also attempt to restructure the subgoals to find a better match. PROUST continues to select subgoals and tries to match them with the code until all subgoals have been tried. All possible interpretations are generated. At the end, PROUST will select the interpretation that most completely explains the student's program.

The bug library of PROUST has been classified using a device called a Goal and Plan (GAP) tree [SPOH85]. A GAP tree is a means of categorizing potential

bugs in a classification that reflects the way they tend to appear. Buggy plans are represented as partial GAP trees, providing substantial information about the bug in a problem independent manner.

Although Johnson and Soloway do not claim PROUST uses a psychological model of how students program, it does go beyond simply modeling surface behavior of the student. PROUST is capable of identifying not only the plan used by a student in writing a program, but also the student's intention in using that plan. Although PROUST could not be used to explain *why* the student used an erroneous plan, it does identify the student's misconception that a particular plan can be used to accomplish a particular goal. Thus PROUST provides useful information for remediation.

5.2. GREATERP and the Geometry Tutor

Anderson *et. al.* have implemented intelligent tutoring systems in two domains, using the same basic underlying system. GREATERP is a tutor for novice programmers in Lisp, while the Geometry Tutor is designed to help students learning to write high school geometry proofs. Both tutors are based on the ACT* theory of cognition.

The ACT* theory has four main features:

1. A production rule system, where each rule represents a unit of skill.
2. The production rules are goal directed, a rule only fires if a goal matches.
3. A working memory stores all the current knowledge, but this working memory is limited in size.
4. A learner becomes more skilled by acquiring new productions. This process is called *knowledge compilation*.

The student model is implemented in a GRAPES, a Goal-Restricted Production System that implements the features of ACT*. Both tutoring systems have an expert called the *ideal student* whose domain knowledge is encoded in GRAPES, as well as a bug catalog of common mistakes or poor strategies used by the student. Together, the ideal student and bug catalog form the *ideal and buggy model* (IBR).

The model of the student is built by a technique called *model tracing*. Essentially, the tutor traces the student's behavior through its ideal and buggy model. At any point many possible production rules may be candidates for application by the tutor. The tutor infers which rule is appropriate by comparing the student's actions with the results of applying each of the rules. If the rule selected is a correct rule (a rule from the ideal student model), the session continues. If the rule is a buggy rule, the tutor will interrupt to correct the situation. This tutoring method is based on empirical evidence that the best time to correct a student misconception is when it occurs. As a consequence of this philosophy, the student model is never more than one step away from the correct model. This small divergence enables not only accurate modeling, but also more effective tutoring.

GREATERP and the Geometry Tutor both require a great deal of predefined knowledge for the ideal student model and bug catalog. Given this, the systems

can perform very well, since they restrict the range of possible student models at any point. Thus GREATERP and the Geometry Tutor can be viewed as extending Sleeman's work with LMS, by including deep knowledge at the level of units of human cognition.

5.3. Repair Theory

Repair theory was proposed by Brown and VanLehn [BROW80] to provide a deeper model for the arithmetic bugs demonstrated by students. In particular, repair theory is able to generate bugs that students demonstrate, while avoiding bugs that have not been observed, or that are non-intuitive.

The fundamental basis for repair theory is the proposition that when a student reaches an *impasse* in performing a procedure, s/he will attempt to *repair* the procedure and continue as normal. An impasse occurs when the problem the student is working calls for a subprocedure the student does not know, or when the student has been performing an incorrect procedure that leads to a "dead end." At this point the student attempts to "repair" the procedure by performing one or more reasonable operations in an attempt to return the form of the problem to one the student knows how to solve.

The impasse-repair sequence is modeled in four parts:

1. A restricted production system for representing the components of a skill and their buggy alternatives. Rules in the system are limited to a single action and goal clause. The rules are interpreted on a stack. Each rule is only tried once and specialized rules are tried before more general rules. Otherwise rules are executed in order. The rules of the system are compiled into a Generalized AND/OR (GAO) graph.
2. A set of principles are used for deleting fragments of the correct procedure. This is used to simulate the impasse reached by the student when solving a problem requiring a skill s/he is missing. Since unconstrained deletion is too general, some restrictions on deleting are necessary. For example, if two rules have the same goal and one is a special case of the other, the more general rule cannot be deleted without first deleting the more specific rule.
3. A set of repair heuristics are used to propose repairs when an impasse is reached. These heuristics range from simply quitting, or skipping the problem area, to shifting the focus, or trying analogous rules to overcome the impasse.
4. A set of critics to filter out some repairs. The set of repair heuristics will tend to generate repairs that are either non-intuitive or are not observed in actual students. Such bugs are called *star bugs*. The purpose of the critic is to filter out all repairs that result in star bugs. Some critic heuristics include rules such as "don't change a number once it has been written" and "don't leave a blank in the middle of the answer."

The repair heuristics and the critics mentioned above are specific to the arithmetic domain. Thus the theory as stated is not a general theory. Brown and VanLehn feel it is possible to produce a body of weak heuristics that can be tailored to the particular domain of a system.

The proposals of repair theory result in several intuitively appealing features:

- Repairs are independent of impasses. This distinction separates the cause of a problem from the method used for its resolution.
- The problem solver is local. When an impasse arises, repair theory attempts to fix the problem on the basis of the current state of affairs.
- Due to the use of the stack, the problem solver can only call existing subgoals or primitive actions. This limits the actions which can occur during a repair, providing a tighter theory.

6. Summary: Comparing User Modeling and Student Modeling

User models and student models are both instances of the more general class of *agent* models [KOBS*]. A user model is simply the model a system keeps of the individual currently using it, while a student model is the model of an individual an ITS views as a student. When a student directly interacts with an ITS, the model of the individual is both a student model and a user model, implying that user models and student models are in general quite similar.

User models and student models do have distinct characteristics, however, at least in the way they are generally viewed. This distinction is rooted in how the models are used, resulting in different modeling approaches. This section will explore the different purposes of modeling students and users, and examine how these purposes affect modeling techniques. A major conclusion of this study is that user models will eventually need to adopt many of the capabilities sought in student modeling, thus student modeling research is important to user modelers.

6.1. Approaches to Modeling

Work in student modeling and user modeling has been driven by differing system needs. An ITS presumes that the student either lacks knowledge of the material being taught, or has misconceptions about that material. Furthermore, the student's knowledge is expected to *change* as the tutoring session proceeds. Consequently student modeling has had to consider how the student represents and reasons about the knowledge he has. Originally, student modeling sought models capable of duplicating student behavior – diagnostic models. A diagnostic model can roughly be thought of as answering the question "How did the student perform this task?" Diagnostic modeling was soon found inadequate, because the models could not guarantee that the student actually used the skills the model attributed to him, only that the model's performance was similar to the student's performance. Thus diagnostic modeling cannot adequately explain how a student performed a task, and cannot even begin to explain *why* the student performed the task in a particular manner.

The need to explain why students behave as they do requires psychologically-based student models. A principle underlying tutoring systems is that the tutor can

be most effective if it knows the fundamental reasons for a student's behavior, so it can tailor the tutoring to correct misconceptions, and instill correct knowledge and skills. Effective modeling of student behavior at this level requires an understanding of how the students represent and acquire information.

On the other hand, the demands on user models have not been so great. User models are generally needed to aid in the interaction between the system and a user: to help in understanding what the user wants the system to do, or to enable the system to tailor its responses to the particular user and his needs. Thus user modeling has concentrated on modeling the user's goals, plans, and beliefs. In such interactions, the user's knowledge usually is assumed to remain static. The emphasis in user modeling is primarily on *what* the user believes or wants, and sometimes on *how* the user plans to accomplish his goal; rather than on trying to model why a user holds certain beliefs, or how the user's beliefs might change.

The different expectations of student and user models is the main reason for the differences in their modeling approaches. The need to explain why students behave as they do has forced student modelers to consider how students organize and reason with their knowledge, resulting in deep, psychological models of students. Such models have not been necessary for typical user modeling applications. For example, a cooperative data base system needs to know what a user is looking for, and what he knows already, in order to respond with the appropriate information in the best form; but it generally is not expected to know why the user wants the information.

This gap between expectations of student and user models is not permanent, however. As the types of systems that employ user models develop, the expectations of the user model will change. For example, an advisory system must reason about user beliefs, and why the user holds those beliefs. To recommend a course of action for the user, the system must convince the user of its recommendation. If the user holds beliefs that seem to contradict the system's recommendation, the system may have to reason about why the user holds a belief, in order to demonstrate that the belief is incorrect, or perhaps to demonstrate that the recommendation really does not contradict the user's belief. Thus an advisory system needs deep knowledge of the relationships between the goals, plans and beliefs held by the user.

Deeper models of the user will also be required to support more robust interactions with the user, such as natural language dialog. In reasoning about what to say, a conversational participant must think about the hearer's knowledge, and about the conclusions a hearer might draw from a statement. For a system to participate in robust interactions, it will need the ability to simulate the user's reasoning. One example of this is the anticipatory feedback process described by Wahlster and Kobsa [WAHL*]. To support such reasoning the user model must have a deep knowledge of the the reasoning processes of humans. Ultimately, this knowledge could form a psychological model for human communication and reasoning.

6.2. Contributions of Student Modeling to User Modeling

Both the similarities between user and student models, and the anticipation that user models will develop in ways similar to student models suggest that user mod-

eling can profit from the progress made in student modeling. This section discusses five areas where student modeling work can contribute to user modeling: the relationship between student models and expert models, the advantage of controlled interaction, the need for deep modeling, the need for generative models, and the noise problem.

6.2.1. Relationship Between Student and Expert Models

Student models are always built with respect to a domain, or expert, model. This is a powerful aid both for representing the potential assumptions the system makes about students, and in building an individual student model.

In an ITS, building the student model with respect to the expert model is eminently practical, since an expert model must be present to represent the "ideal" student. When overlay or differential modeling is used, this expert knowledge is a sufficient basis for student modeling, while a perturbation model only requires that the expert model be augmented with the expected differences that a student might exhibit.

Likewise, user models can benefit from the presence of a domain model in the system. Such a domain model is likely to exist in the circumstances where a user model would be used. User models are used to facilitate communication with the user, both to understand the user and to produce responses. The communication component of such a system must have domain knowledge, as well as knowledge of the user to accomplish these tasks. Thus, when a user model is used, a domain model should be present as well, so the user model can be built in relation to a domain model.

The expert model also aids student modeling when building a model of an individual student. Since the expert model stands as a paradigm for the model of a student, the search for an appropriate student model can focus on models similar to the expert model. Thus the expert model creates expectations of what the student model should be. This technique is useful in user modeling as well. If a system has the correct representation of the knowledge in a domain, it can use this knowledge as a default expectation for the user's knowledge.

Techniques such as overlay, differential, or perturbation modeling are not limited to student modeling domains. These approaches are appropriate for user modeling when a body of domain knowledge is the main focus of interaction. Student modeling has also developed ways for inferring individual models based on the student's behavior and the domain model. Such techniques should be applicable to user modeling as well. In fact, subsequent to his work on LMS, Sleeman used a similar approach to build a model of the student in UMFE, a user modeling front end system used in conjunction with an explanation facility for an expert system [SLEE85].

6.2.2. Advantages of Controlled Interaction

Student modeling in intelligent tutoring systems benefits from the controlled interaction with the student. Typically, an ITS takes responsibility for the interaction with the student, by driving the interaction with questions or problems that the student must respond to. This controlled interaction aids student modeling by limiting

the range of possible student models that must be considered. The best examples of how a controlled interaction enhances student modeling are the incremental technique for building a student model in LMS, and the model tracing technique used in GREATERP and the Geometry Tutor. The controlled interaction limits the possible responses a student can make, thus limiting the number of possibilities that must be considered at any stage in building a student model.

User modeling is not so lucky. Systems that use a user model cover a range of interaction types, but usually the user controls the interaction, or the user and system share control. This means that a user may violate the expectations of the system by changing the subject, volunteering additional information, or perhaps addressing a related subject. Despite these problems, approaches such as model tracing are powerful, and may be useful for some user modeling situations. In practice, most interactive systems have a limited range of capabilities. If the system can identify what the user wants from the system, this will often constrain the interaction to particular topics and forms. At this point, a model tracing technique could be used to effectively build the user model.

6.2.3. Need for Deep Modeling

The need for deep user models has already been argued in Section 6.1. Building deep user models will not be an isolated activity, however. Systems that require a deep user model require a domain model that is equally deep. Thus highly structured domain models will be available to build user models in relation to, continuing the close relation noted above.

6.2.4. Need for Generative Models

As deeper models of users or students are built, the potential for differences between the user or student's representation of knowledge and the domain model's representation become greater. Thus, more complex perturbation models will be required. On the other hand, these differences may be explained by a small number of fundamental differences between user and system. One contribution of the work on repair theory was the notion that a small number of principles accounted for the variety of bugs in student arithmetic. Repair theory formed a *generative* theory of how such bugs were produced. With repair theory, instead of searching a large collection of potential models for one that most accurately matches a student's behavior, a principled theory of the process can predict when errors are likely, and what mechanism caused the error.

For user modeling, a generative theory would focus on characteristics of human reasoning. For example, such a theory might include rules for how people make generalizations or analogies, or how people think about actions. User beliefs, goals, and plans could be inferred by applying these rules to information already believed about the user, expanding the user model greatly. Such rules could also be used to explain the likely ways in which a user's beliefs, goals, or plans differ from those in the domain model. A first step in this direction is the work on inferring user beliefs based on the interaction, the domain model, and the existing user model using a set of implicit acquisition rules [KASS87a, KASS87b].

A generative theory is not easy to develop, however. Even in a simple domain like arithmetic the results were not completely satisfactory. Nevertheless, the underlying motivation is valid. Ultimately, a generative theory for user modeling must be founded on psychological principles of human behavior.

6.2.5. The Noise Problem

Another contribution of student modeling to user modeling is the recognition of the noise problem. For the most part, the problem of inconsistency on the part of the user has been ignored in user modeling. Although a good method for handling noise has not been achieved in student modeling either, it seems to have received more attention there. Once again, a proper treatment of user inconsistencies must wait for deeper psychological explanations of human memory, communication, and so on. Student modeling seems to recognize the problem, and that a principled approach to handling it is possible. User modeling can certainly benefit from such an approach as well.

7. Conclusion

This chapter has explored the role of student modeling in intelligent tutoring systems, with the objective of discovering what user modeling researchers might earn from the related domain of student modeling. After analyzing several intelligent tutoring systems that do student modeling, it appears that a fundamental difference between user and student modeling approaches has been due to the depth of information modeled. Student modeling has been forced to build deeper, psychological models of students to support the needs of a tutoring system. The progress in student modeling is important to user modeling because the demands on user models can be expected to follow a similar path. As interactive systems attempt to support more robust communication, reasoning about the user will require deeper knowledge of the user's beliefs and reasoning abilities. Thus the lessons learned in student modeling are appropriate lessons for user modeling as well.

8. References

[ANDJ85a], [ANDJ85b], [BROW75], [BROW76b], [BROW77], [BROW78], [BROW80], [BROW82], [BURT82], [CARN70a], [CARR77], [CLAN81], [CLAN82], [CLAN83], [CLAN86a], [FARR84], [GOLD82], [HEIN85], [JOHN84], [JOHN85], [KASS87a], [KASS87b], [KIMB82], [KOBS*], [LOND82], [MATZ82], [MILL82], [OSHE82], [REIS85], [SACE77], [SELF74], [SHRA82], [SLEE77], [SLEE81], [SLEE82a], [SLEE82c], [SLEE82d], [SLEE83], [SLEE84], [SLEE85], [SPOH85], [WAHL*], [WOOL84b], [ZISS85]

15 GUMS — A General User Modeling Shell

Timothy W. Finin

Abstract

This chapter discusses the application of various kinds of default reasoning in system maintained models of users. In particular, we describe the general architecture of a domain independent system for building and maintaining *long term models* of *individual users*. The user modeling system is intended to provide a well defined set of services for an *application system* interacting with various users, and must build and maintain models of them. As the application system interacts with a user, it can acquire knowledge of him, and pass that knowledge on to the user model maintenance system for incorporation. We describe a prototype *general user modeling shell* (hereafter called *GUMS*) that we have implemented in Prolog. This system possesses some of the desirable characteristics we discuss.

1. Introduction — The Need for User Modeling

Systems that attempt to interact with people in an intelligent and cooperative manner need to know many things about the individuals with whom they are interacting. Such knowledge can be of several different varieties and can be represented and used in a number of different ways. Taken collectively, the information that a system has about its users is typically referred to as its *user model*. This is so even when it is distributed throughout many components of the system.

Examples that we have been involved with include systems that attempt to provide help and advice [FINI83, HOWE84, SHRA82], tutorial systems [SCHU86b], and natural language interfaces [WEBB84]. Each of these systems had to represent information about individual users. Most of the information was acquired incrementally through direct observation and/or interaction. These systems also had to infer additional facts about their users based on the directly acquired information. For example, the WIZARD help system [FINI83, SHRA82] had to represent which VMS operating system objects (e.g. commands, command qualifiers, concepts, etc.) a user was familiar with, and had to infer which other objects he was likely to be familiar with.

This chapter describes the general architecture for a domain independent system for building and maintaining *long term models of individual users*. The user modeling system is intended to provide a well defined set of services for an *application system* that interacts with various users, and needs to build and maintain models

of them. As the application system interacts with a user, it can acquire knowledge about him and pass that knowledge on to the user model maintenance system for incorporation. We describe a prototype *general user modeling shell* (hereafter called *GUMS*) implemented in Prolog. This system possesses some of the desirable characteristics we discuss.

The next section will discuss some of the general issues that arise in building user models. In particular, we will describe the range of possible issues that one might address in terms of four aspects: who is being modeled; what information is represented in the model; how the model is acquired and maintained, and how the model might be used. The third section provides an overview of the *GUMS* system and the kind of user modeling it supports. The fourth section describes the three kinds of default reasoning that *GUMS* employs: stereotypical reasoning, explicit default rules and failure as negation. Section five describes some of the details of the *GUMS* system, while the concluding section discusses some of the limitations of *GUMS* and avenues for future research.

2. What Kind of User Model?

The concept of incorporating user models into interactive systems has become common, but what has been meant by a user model has varied and is not always clear. In this section we will discuss, in general terms, what might be meant by the term *user model*. We will then characterize the approach to user modeling described in this work by answering the following questions: who is being modeled; what aspects of the user are being modeled; how is the model to be initially acquired and maintained; and why the model is needed.

The term "user model" has been used in many different contexts to describe knowledge that is used to support an interactive system. A survey of user modeling in support of natural language interfaces can be found in [KASS88b] and a discussion of user modeling for intelligent tutoring systems is provided by [KASS*]. In this section, we will look at a number of distinctions that will allow us to focus on an interesting and important class of "user models".

An initial definition for "user model" might be the following:

A user model is that knowledge about the user, either explicitly or implicitly encoded, which is used by the system to improve the interaction.

The definition is too weak, since it endows every interactive system with some kind of user model, usually of the implicit variety. In this chapter, we will focus our attention on those models where the information about the user is *explicitly* encoded. In particular, we are interested in user models that are designed along the lines of "knowledge bases". By this we have in mind the kinds of distinctions that are usually drawn to distinguish a *knowledge base* from a *data base*. In the context of user models, five features are important:

1. *Separate Knowledge Base* – Information about a user is collected in a separate module rather then distributed throughout the system.

2. *Explicit Representation* – The knowledge in the user model is encoded in a sufficiently expressive representation language. Such a representation language will typically provide a set of inferential services. That allows the knowledge of a user to be implicit, but automatically inferred when needed.
3. *Declarative rather than Procedural* – The knowledge in the user model is, as much as possible, encoded in a declarative rather than procedural manner.
4. *Support for Abstraction* – The modeling system provides ways to describe abstract as well as concrete entities. For example, the system might be able to discuss classes of users and their general properties as well as individuals.
5. *Multiple Use* – Since the user model is explicitly represented as a separate module, it can be used in several different ways (e.g. to support a dialog or to classify a new user). This requires that the knowledge be represented in a more general way that does not favor one use at the expense of another. It is highly desirable to express the knowledge in a way that allows it to be reasoned about as well as reasoned with. Similarly, the model ought to provide no barriers to its use by several different applications.

User models having these features fit nicely into current work in the broader field of *knowledge representation*. In fact, we could paraphrase Brian Smith's *knowledge representation hypothesis* [SMIB82] to get something like:

Any user model will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge the system has of the user and b) independent of such external semantical attribution, play a formal but causal and essential role in the behavior that manifests that knowledge.

In the remainder of this section we will explore the large space of user modeling possibilities by addressing the *who*, *what*, *how* and *why* of user modeling. Our own system will then be placed within this space.

2.1. Who Is Being Modeled?

Two aspects of who is being modeled are the degree of specialization and temporal extent. The degree of specialization involves whether we are modeling classes of users or individuals. Temporal extent refers to the persistence of the knowledge encoded in the user model – does it expire at the end of the current session or does it form a long-term knowledge base about the user.

A user model can lie somewhere on a specialization scale that ranges from *generic* to *individual*. A generic user model assumes a homogeneous set of users – all individuals using the system are similar enough with respect to the application that they can be treated as the same type of user. Many natural language systems, for example, maintain a single generic user model in order to infer the user's goals, or to recognize a user's misconceptions [ALLE82b, CARB83, CARN83c, GERS81]. A somewhat more individuating point on this scale is to employ a set of fixed, generic models to represent different subclasses of users. For example, the UC

system [WILE86, CHIN86, CHIN*] classifies a user as belonging to one of four possible generic classes: *novice, beginner, intermediate, or expert.*

Modeling the very general beliefs held by large classes of users is extremely important when building intelligent systems. For example, in modeling a person's beliefs about a timeshared computer system we may want to include a rule like:

If a user U believes that machine M is running, then U will believe that it is possible for him to log onto M.

This sort of rule is required to support the kinds of cooperative interactions studied in [JOSH84a, JOSH84b], such as the following:

```
User:   Is linc.cis.upenn.edu up?
System: Yes, but you can't log on now.
        Preventative maintenance is being done until 11:00am.
```

Individual user models, on the other hand, contain information specific to a single user. A user modeling system that keeps individual models thus will have a separate model for each user of the system. A simple example of this kind of model is the commonly used convention of customizing the behavior of a utility such as an editor or mail system by allowing each user to have a *profile* file that states his preferences for various system parameters [RICH83].

A natural way to combine the system's knowledge about classes of users with its knowledge of individuals is through the use of *stereotypes* – generic models with specializations for individuals. A stereotype is a cluster of default facts about the user that tend to be related to each other. One can model an individual as belonging to a particular stereotype (e.g. *novice Unix User*) with a set of overriding facts (e.g. *knows how to use pipes*). Examples of systems that have used stereotypes include GRUNDY [RICH79b], and the Real-Estate Advisor [MORI85b].

2.2. What Is to Be Modeled?

The *contents* of a user model naturally vary from application to application. Following Kass and Finin [KASS*], we will classify this knowledge into four categories: goals and plans, capabilities, attitudes, and knowledge or belief.

A user's goals and plans are intimately related to one another. The user's *goal* is a state of affairs he wishes to achieve, while a *plan* is a sequence of actions or events that he expects to result in the realization of a goal. Furthermore, each step in a plan has its own subgoal to achieve, which may be realized by yet another subplan. One of the hallmarks of an intelligent, cooperative system is that it attempts to help the user achieve his underlying goals, even when not explicitly stated. The principle way for one agent to know another's underlying goal(s) is by recognizing his observable actions as constituting a (possibly partial) plan for achieving a domain-relevant goal. A great deal of work has been done on the problem of *plan recognition* in support of cooperative interactions. Examples in the natural language domain include Allen and Perrault [ALLE80] and Carberry [CARB83, CARB88, CARB*]. Intelligent Tutoring Systems have introduced the idea of keeping a catalog of "buggy" plans [BROW78, SLEE82a, JOHN84] as a means of recognizing mis-

conceptions or missing knowledge. The use of plan recognition is also relevant to the problem of providing help and advice, as in [FINI83, WILE86].

A second category of knowledge to be modeled has to do with the user's capabilities. Although some systems might have to reason about a user's physical capabilities, such as the ability to perform some action that the system may recommend, it is more typical for a system to model and reason about mental capabilities, such as the ability to understand a recommendation or explanation provided by the system. Examples of the latter often arise when a system is generating an *explanation*, as in Wallis and Shortliffe [WALL82] and Paris TAILOR system [PARI88]. Webber and Finin [WEBB84] have surveyed ways that an interactive system might reason about its user's capabilities to improve the interaction.

A third category of knowledge represents a user's bias, preferences and attitudes. GRUNDY [RICH79b] was an early user modeling system that used a model of a user's preferences to recommended books to read. The Real-Estate Advisor [MORI85b] and HAM-ANS [HOEP83b, MORI*] are other systems which have tried to model this. Swartout [SWAR83] and McKeown [MCKE85a] address the effects of the user's *perspective* or *point of view* on the explanations generated by a system.

The final category of information we will consider is *knowledge* and *belief*.[1] Modeling the user's knowledge involves a variety of things: domain knowledge, general world knowledge, and knowledge about other agents. Knowing what the user believes to be true about the application domain is especially useful for many types of systems. In generating responses, knowledge of the concepts and terms the user understands or is familiar with allows the system to produce responses incorporating those concepts and terms, while avoiding concepts the user might not understand. This is particularly true for intelligent help systems [FINI83, WILE86] which must provide clear, understandable explanations to be truly helpful. Providing definitions of data base items (such as the TEXT system does [MCKE85c]) has a similar requirement to express the definition at a level of detail and in terms the user understands.

A final form of user knowledge that is important for interactive systems is knowledge about other agents. As an interaction with a user progresses, not only will the system be building a model of the beliefs, goals, capabilities and attitudes of the user, the user will also be building a model of the system. Sidner and Israel [SIDN81b] make the point that when individuals communicate, the speaker will have an *intended meaning*, consisting of both a *propositional attitude* and the *propositional content* of utterance. The speaker expects the hearer to recognize the intended meaning, even though it is not explicitly stated. Thus a system must reason about what model the user has of the system when making an utterance, because this will affect what the system can conclude about what the user intends the system to understand by the user's statement. Kobsa [KOBS84] has studied some of the difficult representational problems involved with building recursive models of the beliefs of other agents.

---

[1] In the context of modeling other individuals, an agent does not have access to objective truth and hence cannot really distinguish whether a proposition is *known* or simply *believed to be true*. Thus the terms *knowledge* and *belief* will be used interchangeably.

2.3. How Is the Model to Be Acquired and Maintained?

The acquisition and maintenance of the user model are closely related topics. By acquisition, we mean the techniques for learning new facts about the user. Maintenance involves incorporating this new information into the existing model and squaring away any discrepancies or contradictions. We will briefly review some of the possibilities for these two related problems.

The knowledge in a user model can be acquired either *explicitly* or *implicitly*. Explicitly acquired knowledge might come directly from the system designer in the form of built in knowledge of stereotypes, from the application system the user model is serving, or from the user himself. GRUNDY, for example, begins with a new user by interviewing him for certain information. From this initial data, GRUNDY makes judgments about which stereotypes most accurately fit the user, thus forming an opinion about the preferences of the user based on this initial list of attributes. As another example, each time the UMFE system [SLEE85] needs to know a new fact about its user, it simply asks.

Acquiring knowledge about the user implicitly is usually more difficult. Implicit acquisition involves "eavesdropping" on the user-system interaction in order to observe the user's behavior and from it to infer facts that go into the model. Various aspects of implicit acquisition have been explored, including the use of presuppositions [KAPS82], misconception recognition [MCCO85a, MCCO*], stereotype selection [RICH79b, RICH*, MORI85b], and the use of default rules [PERR88, KASS87a, KASS87b].

Maintenance involves incorporating new knowledge about an individual user into an existing model. The new, incoming information might be consistent or inconsistent with the current model. If it is consistent, maintenance involves triggering inferences to add additional facts to the model. If the new knowledge is inconsistent with the current model, then the inconsistency must be resolved in some manner. The possibilities depend on the underlying representation and reasoning systems on which the modeling system is built. Two major approaches are *evidential reasoning*, which allows the model to hold that a certain fact is true with a certain degree of belief, and *default reasoning* [REIT80], in which certain facts can be held in the absence of evidence to the contrary.

2.4. Why Is the Model There?

The knowledge about a user that a model provides can be used in a number of ways in an intelligent, interactive system. The particular opportunities depend, of course, on the application. Figure 1 presents a general taxonomy of possible uses. At the top level, user models can be used to support (1) the task of recognizing and interpreting the information seeking behavior of a user, (2) providing the user with help and advice, (3) eliciting information from the user and (4) providing information to him. Situations where user models are used for many of these purposes can be seen in the examples presented throughout this book.

The model can be accessed in two primary ways: facts can be added, deleted or updated from the model, and facts can be looked up or inferred. A forward chaining

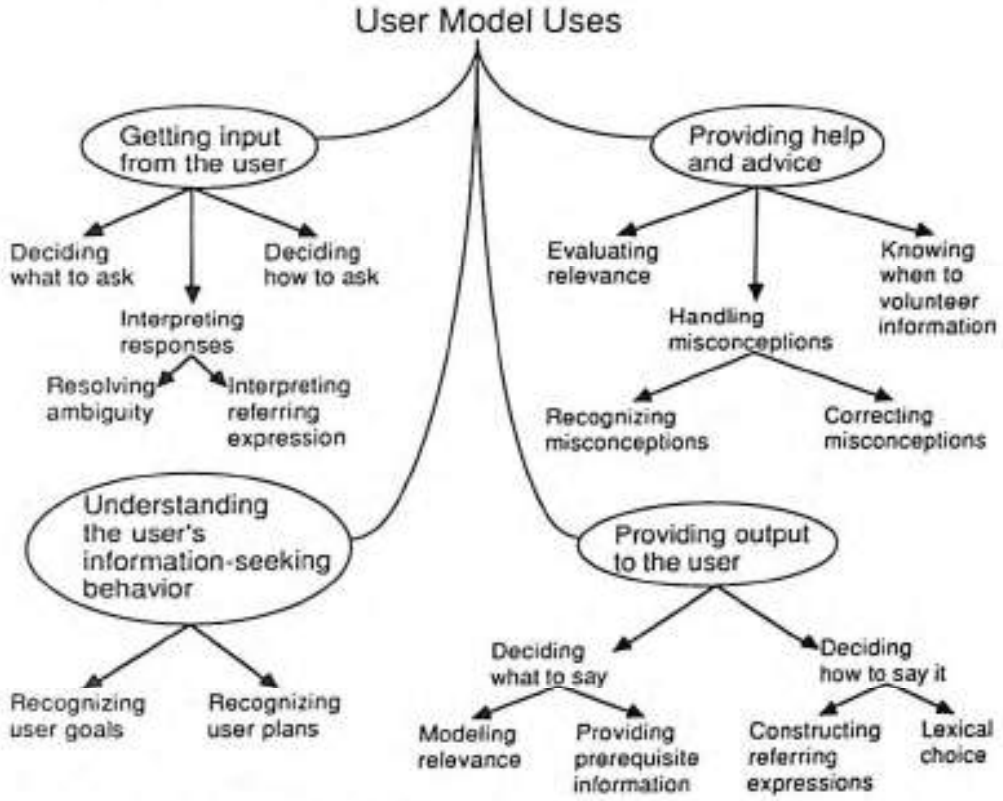


Figure 1. Uses for knowledge of the user

component, together with a truth maintenance system, can be used to update the default assumptions and keep the model consistent.

3. A General User Modeling System

The previous section described a large conceptual space of user modeling possibilities in terms of four dimensions. This section will place our own work in this space and give an overview of its major characteristics. Our paradigm for user modeling might be described as *User Model as Deductive Database*. The *GUMS* system allows one to construct models of individuals that appear to be a collection of facts that can be asserted, retracted or queried. Some of these facts are deduced through the use of inference rules associated with stereotypes to which the user belongs.

Who is being modeled? The *GUMS* system is designed for building **long term** models of **individual users**. We want to represent the knowledge and beliefs of individuals, and do so in a way that results in a persistent record which can grow and change as necessary. These users are associated with one or more *stereotypes* from which additional facts derive.

What is being modeled? Our interest is in providing a domain-independent user modeling tool. The contents of the user model will vary greatly from appli-

cation to application and is left for the model builder to decide. For example, a natural language system may need to know what language terms a user is likely to be familiar with [WEBB84], a CAI system for second language learning may need to model a user's knowledge of grammatical rules [SCHU86b], an intelligent database query system may want to model which fields of a data base relation a user is interested in [MOTR86], and an expert system may need to model a user's domain goals [POLL85]. However, our paradigm of user model as deductive database makes modeling certain things much easier than others. It is easy to construct models that are expressed as a set of relations. In particular, we do not provide any general facilities for building deep models of an agent's reasoning strategies or procedural knowledge.

How is the model acquired and maintained? Our approach in *GUMS* is to let the application be responsible for acquisition and to provide a set of services for maintenance. The application must select the initial stereotypes for the user and add new facts about the user as it learns them. How the application discovers these facts (i.e. explicitly or implicitly) is up to the application. Maintaining the model involves incorporating these new facts, checking that they are consistent with previously learned or inferred facts, and resolving any discrepancies and contradictions. This can be accomplished by undoing contradictory default assumptions and/or shifting the user to a new stereotype. The details of how this is done will be described later.

Why is the model there? Given our goal of producing a general purpose, domain independent use modeling facility, the uses to which the knowledge in the model is put are outside of our design. We will, however, give a more syntactic account of how the model is to be used in terms of how the application interacts with *GUMS*.

3.1. The System Organization

Our goal is to provide a general user modeling utility organized along the lines shown in Figure 2. The user modeling system provides a service to an application program that interacts directly with a user. Note that the user modeling system does not have access to the interaction between the application and the user. Everything it learns about the user must come directly from the application. The *GUMS* system has a separate knowledge base for each application it serves. Each knowledge base consists of two parts: (1) a collection of *stereotypes* organized into a taxonomy, and (2) a collection of models for the individuals. The individual models are installed in the stereotype hierarchy as leaves. Figure 3 shows such a hierarchy. As will be seen later, a stereotype is a collection of facts and rules that are applicable for any person who is seen as belonging to that stereotype. These facts and rules can be either *definite* or *default*, allowing for an individual to vary from the stereotypic norm.

The application program gathers information about the user through its interaction, choosing to store some of this information in the user model. Thus, one service the user modeling facility provides is accepting (and storing) new informa-

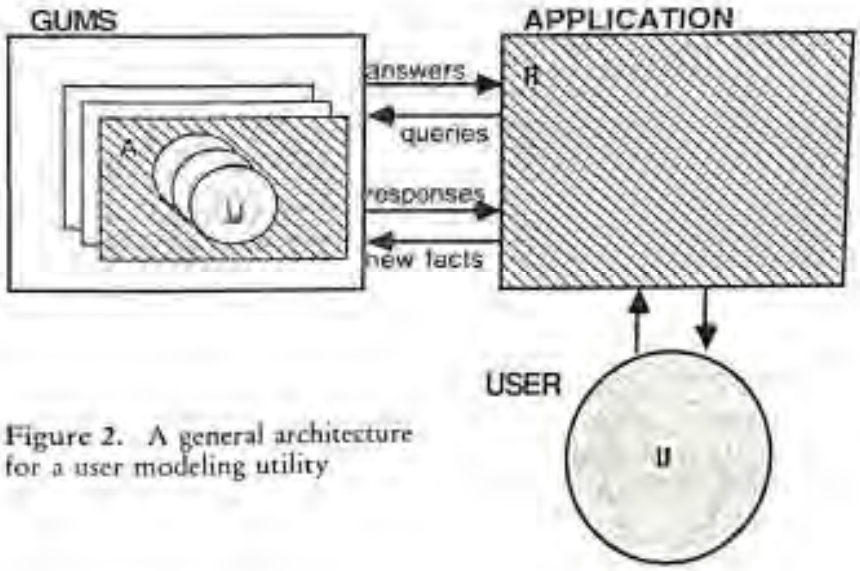


Figure 2. A general architecture for a user modeling utility

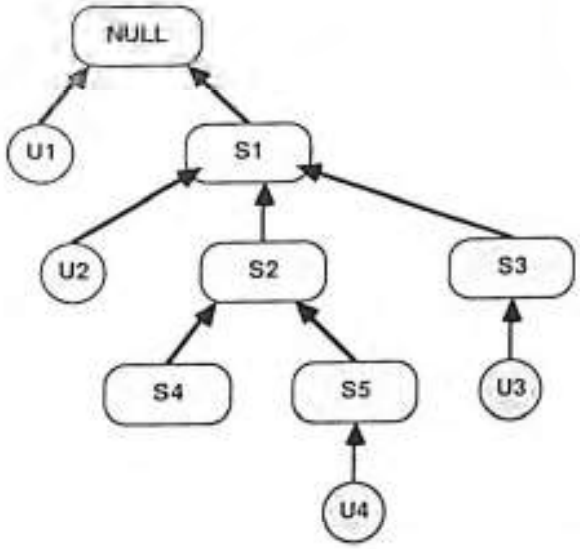


Figure 3. A hierarchy of stereotypes and individuals

tion about the user. This information may trigger an inferential process which could have a number of outcomes:

- The user modeling system may detect an inconsistency and so inform the application
- The user model may infer a new fact about the user which triggers a demon causing some action (e.g. informing the application).
- The user model may need to update some previously inferred default information about the user.

Another kind of service the user model must provide is answering queries posed by the application. The application may need to look up or deduce certain information about its current user.

3.2. Default Reasoning in GUMS

User modeling is most useful in situations in which a system must draw a number of plausible conclusions about a user from a small base of definite knowledge about him. Thus, default reasoning of some kind or another is at the heart of most user modeling systems. Our approach in the *GUMS* system is to use several forms of default reasoning techniques: stereotypes, explicit default rules, and failure as negation. These three types of default reasoning are closely related and, in general, formalizable in terms of a single type of default reasoning. We have found it convenient, however, to separate the default reasoning in *GUMS* for several reasons.

In the *GUMS* system the different kinds of default reasoning are used to capture generalizations of different *grain size* and form a *hierarchy* with respect to the strength of their conclusions. Stereotypes are used to capture generalizations of large classes of users. Within a stereotype, explicit default rules are used to express the stereotypic norms which are allowed to vary across individuals. Failure as negation is used as a general technique for gathering weak evidence for beliefs about a user when no stronger evidence exists.

3.2.1. Stereotypes and Individuals

In *GUMS*, a stereotype consists of a set of facts and rules that are believed to apply to a class of users. Thus a stereotype gives us a form of default reasoning in which each rule and fact in the stereotype is taken to hold in the absence of evidence that the user does not belong in the stereotype.

Stereotypes can be organized in hierarchies in which one stereotype subsumes another if it can be thought to be more general. A stereotype S_1 is said to be more general than a stereotype S_2 if everything which is true about S_1 is necessarily true about S_2. Looking at this from another vantage point, a stereotype inherits all the facts and rules from every stereotype that it is subsumed by. For example, in the context of a *programmer's apprentice* application, we might have stereotypes corresponding to different classes of programmer, as is suggested by the hierarchy in Figure 4.

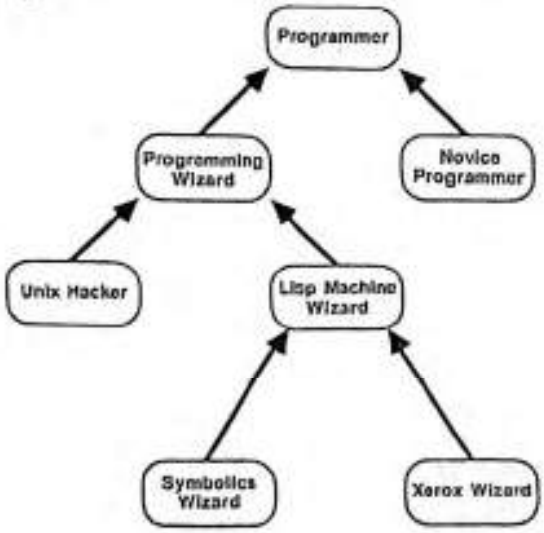


Figure 4. A hierarchy of stereotypes

In general, we will want a stereotype to have any number of immediate ancestors, allowing us to compose a new stereotype out of several existing ones. In the context of a *programmers apprentice*, for example, we may wish to describe a particular user as a *SymbolicsWizard* and a *UnixNovice* and a *ScribeUser*. Thus, the stereotype system should form a general lattice. Our current system, however, constrains the system to a tree.

A stereotype is comprised of several kinds of knowledge, as depicted in Figure 5. Each stereotype can have a single subsuming stereotype, and a set (possibly empty) of subsumed stereotypes and a set (again, possibly empty) of individuals currently believed to be modeled by the stereotype. The real contents of the stereotype, however, consists of two databases of facts and rules, one definite and the other default. The definite facts and rules constitute a *definition* of sorts for the stereotype, in that they determine what is necessarily true of any individual classified under this stereotype. If our knowledge of an individual contradicts this information, then he cannot be a member of this stereotype or any of its descendent stereotypes. The default facts and rules, on the other hand, define our initial beliefs about any individual who is classified in this stereotype. They can be overridden by specific information that we later learn about the individual. The final component of a stereotype is a collection of meta-knowledge about the predicates used in the definite and default knowledge bases. This information is used in the processing of negative queries and is described later.

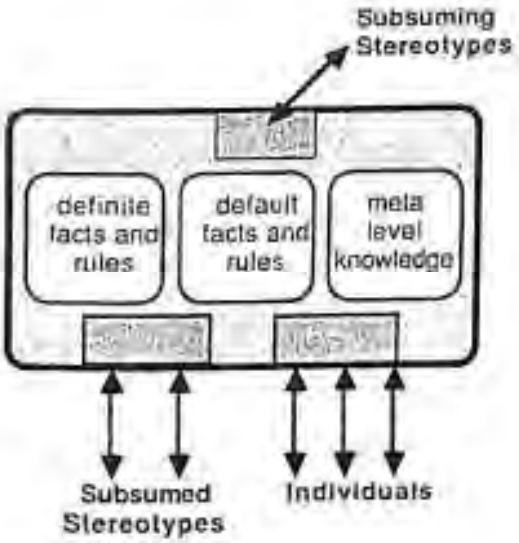


Figure 5. A stereotype



Figure 6. An individual

As an example, consider modeling the knowledge a programmer might have. We can be sure that a programmer will know what a *file* is, but we can only guess that a programmer will know what a *file directory* is. If we have categorized a given user under the *programmer stereotype* and discover (perhaps through direct

interaction with him) that he is not familiar with the concept of a *file* then we can conclude that we had improperly chosen a stereotype and must choose a new one. But if we got the information that he did not know what a *file directory* was, this would not rule out the possibility of his being a programmer.

The knowledge structure for an individual is simpler than that for a stereotype. An individual has exactly one stereotype (in the current version) with which he is currently associated, and a collection of definite, ground facts which are true, as is shown in Figure 6. The facts known about an individual override any default facts known or deducible from the stereotype associated with the individual. They must, however, be consistent with the stereotype's definite knowledge base. If a contradiction arises, then the individual must be reclassified as belonging to another stereotype.

There are a number of ways we might go about trying to reclassify an individual when it is discovered that the current stereotype is no longer valid. In general, we can distinguish *domain independent* and *domain dependent* techniques. Domain dependent techniques are particularly useful when we have some understanding of the reasons why a user might be mis-classified. For example, the user may be learning new domain knowledge while he is using the system and this new knowledge may cause him to "grow out of" the current stereotype. In such a situation, we may wish to directly encode a "growth path" of stereotypes. Another domain dependent technique is to treat certain user attributes as triggers which, when true, evoke particular stereotypes associated with them [RICH*].

Domain independent techniques for selecting a new stereotype in the face of conflicting information involve (conceptually) two stages: generating candidates for the new stereotype and selecting from among them. A very general and powerful technique would be to use a classification strategy similar to that used in the KL-ONE family of representation languages [BRAC85]. Such a strategy would consider all of the possible stereotypes and find the set of *most specific* subsuming stereotypes.

In the *GUMS* system, we have used a very simple domain independent strategy for finding a new stereotype when the known facts about an individual contradict some facts associated with the user's current stereotype. The ancestors of the current stereotype are searched in order of specificity until one is found in which there is no contradiction. Thus, we are guaranteed to find a stereotype which subsumes the user, although it will in general not be the most specific one possible.

Although we have adopted this strategy because of its relative simplicity, both in terms of complexity and inherent computational costs, it does correspond to some of our intuitions on how people adopt new stereotypes. The intuition is that the more we learn about an individual, the less we depend on stereotypic reasoning to fill in our beliefs about him. This is mirrored in the *GUMS* system in its stereotype shifting strategy. As the system learns more about a user, it will discover contradictions with its current stereotype model. When faced with a contradiction, it moves up the hierarchy until it finds a stereotype in which the contradiction is resolved. Moving up the hierarchy corresponds to a lessening of stereotypic reasoning, since the higher models are necessarily less complete.

3.2.2. Default Reasoning with Rules

Within a stereotype, we need to describe some knowledge as being necessarily true with respect to (properly classified) members of the stereotype and other knowledge as *default* information. This default information is to be taken as true in the absence of contradictory information. In the *GUMS* system, each rule or fact in the stereotype is either *definite* (i.e. necessarily true) or *default*. The **certain/1** predicate is used to introduce a definite fact or rule and the **default/1** predicate to indicate a default fact or rule, as in:

certain (P). a definite fact: P is true.

certain (P if Q). a definite rule: P is true if Q is definitely true, and P is assumed to be true if Q is only assumed to be true.

default (P). a default fact: P is assumed to be true unless it is known to be false.

default (P if Q). a default rule: P is assumed to be true if Q is true or assumed to be true and there is no definite evidence to the contrary.

Figure 7 depicts graphically the relationships between clauses with *certain* and with *default* information.

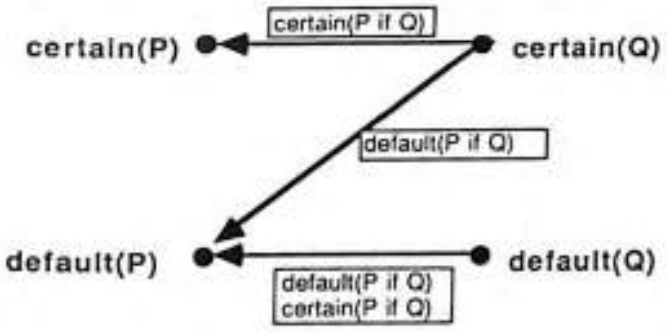


Figure 7. The relationship between certain and default clauses

As an example, consider a situation in which we need to model a person's familiarity with certain terms. This is a common situation in systems that produce text as explanations or in response to queries where there is a wide variation in the users' familiarity with the domain. For example, we might use the rules and facts:

(a) default(understandsTerm(ram)).
(b) default(understandsTerm(rom) if understandsTerm(ram)).
(c) certain(understandsTerm(pc) if understandsTerm(ibmpc)).
(d) certain(~understandsTerm (cpu)).

to represent the following assertions, all of which are considered as pertaining to a particular user with respect to the stereotype containing the rules:

(a) Assume the user understands the term *ram* unless we know otherwise.

(b) Assume the user understands the term *rom* if we know or believe he understands the term *ram*, unless we know otherwise.
(c) This user understands the term *pc* if he understands the term *ibmpc*.
(d) This user does not understand the term *cpu*.

3.2.3. Negation As Failure

GUMS also treats negation as failure in some cases as a default rule. In general, logical deduction is performed using an open world assumption. That is, the failure to prove a proposition is not taken as evidence that it is not true. Many logic programming languages, such a Prolog, encourage the interpretation of unprovability as logical negation. Two approaches have been put forward to justify the negation as failure rule – making the *closed world assumption* and *completing the database*. In the closed world approach [REIT78], we assume that anything not inferable from the database is by necessity false. Making this meta-level assumption introduces a potential problem in that we do not know what the corresponding object-level assumptions are. The second approach is based upon the concept of a completed database [CLKL78]. A completed database is the database constructed by rewriting the set of clauses defining each predicate to an **if and only if** definition that is called the completion of the predicate. The purpose of the completed definition is to indicate that the clauses that define a predicate define **every** possible instance of that predicate.

Any approach to negation as failure requires that a negated goal be *ground* (i.e. contain no uninstantiated variables) before execution.[2] Thus we must have some way of insuring that every negated literal will be bound. In *GUMS* we have used a simple variable typing scheme to achieve this, as will be discussed later.

We have used a variant of the completed database approach to show that a predicate within the scope of a negation is closed. Making the meta-level assertion *declare (closed(P))* signifies that the definition of the predicate P is actually an *if and only if* definition. This same technique was used by Kowalski [KOWA79] to indicate completion. We further assume that a predicate is not completed unless it is declared closed. For example, consider the following clauses:

```
declare(closed(p)).
p if q
r if q
```

Since the predicate p is complete, proving that q is false allows us to deduce that p is false. Since the predicate r is not complete, proving that q is false does not allow us to conclude that r is false, although it does provide weak evidence that r may be false.

Thus in *GUMS* we have the ability to express that a default should be taken from the lack of certain information (i.e. negation as failure), as well as from the presence of certain information (i.e. default rules). For example, we can have a

[2] A slightly less restrictive rule could allow a partially instantiated negated goal to run but would produce the wrong answer if any variable was bound.

default rule for the programmer stereotype that can conclude knowledge about linkers from knowledge about compilers, as in:

```
default(knows(linkers) if knows(compilers))
```

We can also have a rule that will take the lack of knowledge about compilers as an indication that the user probably knows about interpreters, as in:

```
certain(knows(interpreters) if ~ knows(compilers))
```

This system also allows explicit negative facts, both definite and default. When negation is proved in reference to a negative fact then negation is not considered a default case. Similarly, negation as failure is not considered a default when the predicate being negated is closed. Such distinctions are possible because the *GUMS* interpreter is based on a four value logic.

The distinction between truth or falsity by default (i.e. assumption) and truth or falsity by logical implication is an important one to this system. The central predicate of the system is the two argument predicate *show*, which relates a goal expressed as a literal to a truth value. Thus *show (Goal, Val)* returns in the variable *Val* the "strongest" belief in the literal *Goal*. The variable *Val* can be instantiated to *true, false, assume (true)*, or *assume (false)*. The meanings of these values are as follows:

true	definitely true according to the current database.
assume (true)	true by assumption (i.e. true by default)
assume (false)	false by assumption
false	definitely not true according to the current database.

These values represent truth values for a given user with respect to a given stereotype. If the stereotype is not appropriate, then even definite values may have to change.

The relative strength of the four truth values is shown by the partial ordering in Figure 8. Evidence that a fact is either *true* or *false* is the strongest and outweighs any evidence that the fact is assumed to be true or false. The *GUMS* system is "optimistic" in that it takes direct evidence that a fact is assumed to be true to be stronger than evidence that a fact is assumed to be false. This choice has been made to reflect the fact that, in our current applications, the unprovability of a proposition is weak evidence that it is false. Note that this multi-valued logic is simpler than those based on the notion of a *bilattice* discussed by Ginsberg [GINS86].

Having a four value logic allows us to distinguish conclusions made from purely logical information from those dependent on default information. The four value logic also allows a simple type of introspective reasoning that may be useful for modeling the beliefs of the user. We currently use a default rule to represent an uncertain belief about what the user knows or believes, but we could imagine a situation where we would like to model uncertainties that the user has in his beliefs or knowledge. One such predicate is an embedded *show* predicate. The embedded *show* relation holds between a goal G and a truth value X if the strongest truth value that can be derived for G is X.

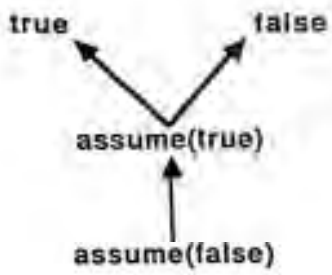


Figure 8. The relative strength of the four truth values

As an example, consider a situation in which we need to express the fact that a user will use a operating system command that he believes might erase a file only if he is certain that he knows how to use that command. We might encode this using the *show* predicate as follows:

```
certain(okay_to_use(Command) if
        can_erase_files(Command),
        show(know(Command),true)).
```

The embedded show predicate forces the *know (Command)* goal to be a definite fact in order for the conclusion to hold.

Another predicate, *assumed (Pred)*, will evaluate the truth of *Pred* and "*strengthen*" the result. The *strengthen* operation maps assumed values into definite values (e.g. **assume(true)** becomes **true, assume(false)** becomes **false** and **true** and **false** remain unchanged). The *assumed* predicate is used to express a certain belief from an uncertain knowledge or belief. For example we might want to express a rule that a user will always want to use a screen editor if he believes one may be available.

```
certain (willUse(screenEditor) if
        assumed(available(screenEditor))).
```

The interpreter that *GUMS* is based on is a metalevel interpreter written in Prolog. The interpreter must generate and compare many possible answers to each subquery, because of the multiple value logic and the presence of explicit negative information. Strong answers to a query (i.e. *true* and *false*) are sought first, followed by weak answers (i.e. *assume(true)* and *assume(false)*). Because strong answers have precedence over weak ones, it is not necessary to remove weak information that contradicts strong information.

Another feature of this system is that we can specify the **types** of arguments to predicates. This type information is used to allow the system to handle non-ground goals. In our system, a **type** provides a way to enumerate a complete set of possible values subsumed by that type. When the top-level *show* predicate is given a partially instantiated goal to solve, it uses the type information to generate a stream of consistent, fully instantiated goals. These ground goals are then tried sequentially.

That goals must be fully instantiated follows from the fact that negation as failure is built into the evaluation algorithm. Complex terms are instantiated to every pattern allowed by the datatype given the full power of unification. To specify the

type information, one must declare the data types, the subtype relations between the types, the instances of these types and, for each predicate, the required type of each argument. For example, the following assertions declare that the **canProgram** predicate ranges over instances of *person* and *programmingLanguage*, that the type *functionalLanguage* is a sub-type of *programmingLanguage* and that the value *scheme* is an instance of the type *functionalLanguage*:

```
declare(schema(canProgramm(person,programmingLanguage))).
declare(subtype(programmingLanguage,functionalLanguage)).
declare(instance(functionalLanguage,scheme)).
```

To see why it is necessary to fully instantiate goals, consider the following problem. We would like to model a programmer's knowledge of tools available on a system. This might be done in order to support an intelligent help system, for example. We might want to encode the following general rules:

C Programmers tend to use all of the relevant tools that are available to them.
If a C programmer does not use some relevant tool, then this is probably because he is not aware of it.
A person is probably a C programmer if they have ever used the C compiler.
Every C tool is relevant to a C programmer.
We have complete knowledge of every tool a user has used.
Lint is a C tool.

which could be expressed in *GUMS* as follows

```
default(knows(Tool) if relevant(Tool), cprogrammer).
default(~knows(Tool) if ~used(Tool), relevant(Tool)).
default(cprogrammer if used(cc)).
certain(relevant(Tool) if cTool(Tool), cprogrammer).
declare(closed(used)).
ctool(lint).
```

Suppose we want to discover a tool of which an individual user is not aware. To answer this, we would like to pose the query *~knows(X)* where X is an uninstantiated variable. Let's assume that this is to be done with respect to an individual knowledge base that includes the following definite facts:

```
used(cc).
used(emacs).
used(ls).
```

The normal interpretation of variables in Prolog goals is that they are *existentially* quantified. The normal interpretation of negated goals is failure as negation, i.e. unprovability. When these two are combined, as in the goal **~used(X)**, the result is that the negation "flips" the quantifier, resulting in a goal which is the equivalent of **∀x not(used(x))**. Moreover, satisfying a negated goal will never result in any variables in the goal being instantiated. Thus, in the example above, we would be unable to discover any tools which this user did not know about since the negated goal **~used(X)** must fail because it is possible to prove **used(emacs)**.

It is possible to "fix" this particular example by reordering the subgoals in the second rule, to get:

```
default(~knows(Tool) if relevant(Tool), ~used(Tool)).
```

This will ensure that the variable *Tool* is instantiated before the negated subgoal is reached. This solution, and the more general one of delaying the evaluation of a negated goal until it is ground, will not always work. Our strategy is to force all goals to be completely ground and thereby avoid this problem.[3] Posing the goal **~knows(X)** to *GUMS* results in the attempt to satisfy a stream of goals which includes the ground goal **~knows(lint)**.

4. The Current GUMS System

The *GUMS* system is currently implemented in CProlog. At the heart of the system is a backward chaining meta-interpreter that implements the default reasoning engine.

Our current system has several limitations. One problem is that it does not extract all of the available information from a new fact learned about the user. If we assert that a predicate is *closed*, we are saying that the set of (certain) rules for the predicate forms a **definition**, i.e. a necessary and sufficient description. In our current system, however, the information still only flows in one direction. For example, suppose that we would like to encode the rule that a user knows about *I/O redirection* if and only if they know about *files* and about *pipes*. Further, let's suppose that the default is that a person in this stereotype does not know about *files* or *pipes*. This can be expressed as:

```
certain(knows(io_redirection) if
        knows(pipes),
        knows(files)).
default(~knows(pipes)).
default(~knows(files))
declare(closed(knows(io_redirection))).
```

If we learn that a particular user does know about *I/O redirection* then it should follow that he necessarily knows about both *files* and *pipes*. Adding the assertion

```
certain(knows(io_redirection))
```

however, will make no additional changes in the data base. The values of **knows(pipes)** and **knows(files)** will not change. A sample run after this change might be:

```
?- show(knows(io_redirection),Val).
        Val = true
?- show(knows(pipes),Val).
        Val = assume(false)
```

[3] This is similar to the treatment of negation used by Walker [WALK87].

```
?- show(knows(files),Val).
        Val = assume(false).
```

The reason for this problem is that the current interpreter was designed to be able to incorporate new information without actually using a full truth maintenance system. Before a fact *F* with truth value *V* is to be added to the data base, *GUMS* checks to see if an inconsistent truth value *V'* can be derived for *F*. If one can be, then a new stereotype is sought in which the contradiction goes away. New knowledge that does not force an obvious inconsistency within the database is added as is. Neither redundant information nor existing default information affect the correctness of the interpreter. Subtler inconsistencies are possible, of course.

Another limitation of the current system its inefficiency. The use of default rules requires us to continue to search for solutions for a goal until a strong one is found or all solutions have been checked. These two limitations may be addressable by redesigning the system to be based on a forward chaining truth maintenance system. The question is whether the relative efficiency of forward chaining will offset the relative inefficiency of truth maintenance. The use of an assumption based truth maintenance system [DEKL86] is another alternative that we will investigate.

5. The GUMS Command Language

Our current implementation provides the following commands to the application

show(Query,Val) succeeds with *Val* as the strongest truth value for the goa *Query*. A Query is a partially or fully instantiated positive or negative literal. *Val* is returned and is the value of the current belief state. If *Query* is partially instantiated then it will return more answers upon backtracking if possible. In general, one answer will be provided for every legal ground substitution that agrees with current type declarations.

add(Fact,Status) sets belief in *Fact* to true if it is consistent with the current set of beliefs about the user. As described earlier, consistency is checked by trying to deduce the negation of the Fact. If *Fact*, or any legal instance of it, is found to be contradictory, *GUMS* adopts successively higher stereotypes until one is found in which all of the added facts are consistent. If no stereotype is successful then no stereotype is used and all answers will be based entirely on added facts. *Fact* must be partially or fully instantiated and can be either a positive or negative literal. *Status* must be uninstantiated and will be bound to a message describing the result of the addition (e.g. one of several error messages, *ok*, the name of a new stereotype, etc.).

create_user(UserName,Stereotype,File,Status) stores the current user model if necessary and creates a new model representing the user who then becomes the current user. *UserName* is instantiated to the desired name. *Stereotype* is the logical name of the stereotype that the system should assume to hold. *File* is the name of the file in which information pertaining to the user will be stored. *Status* is instantiated by the system and returns error messages. A user must be created in order for the system to be able to answer queries.

store_current(Status) stores the current user's information and clears the workspace for a new user. *Status* is instantiated by the system when an error occurs.

restore.user(User,Status) restores a previous user after saving the current user if necessary. *User* is the name of the user. *Status* is instantiated by the system to pass error messages.

done stores the system state of the user modeling system, saving the current user if necessary. This command should be the last command issued and needs to be issued at the end of every session.

6. Conclusions

Many interactive systems have a strong necessity to maintain models of individual users. We have presented a simple architecture for a general user modeling utility based on the ideas of default logic. This approach provides a simple system that can maintain a database of known information about users as well as use rules and facts associated with a stereotype which is believed to be appropriate for this user. The stereotype can contain definite facts and rules of inference as well as default information and rules. The rules can be used to derive new information, both definite and assumed, from the currently believed information about the user.

We believe that this kind of system will prove useful to a wide range of applications. We have implemented an initial version in Prolog and are beginning to use it to support the modeling needs of several projects. There are a number of interesting issues that need to be explored in extending the *GUMS* system, including the use of different reasoning systems such as assumption based TMS's and the development of better strategies for assigning or re-assigning an individual to a stereotype.

7. References

[ALLE80], [ALLE82b], [BRAC85], [BROW78], [CARB83], [CARB88], [CARB*], [CARN83c], [CHIN86], [CHIN*], [CLKL78], [DEKL86], [FINI83], [GERS81], [GINS86], [HOEP83b], [HOWF84], [JOHN84], [JOSH84a], [JOSH84b], [KAPS82], [KASS87a], [KASS87b], [KASS88b], [KASS*], [KOBS84], [KOWA79], [MCCO85a], [MCCO*], [MCKE85a], [MCKE85c], [MORI85b], [MORI*], [MOTR86], [PARI88], [PERR88], [POLL85], [REIT78], [REIT80], [RICH79b], [RICH83], [RICH*], [SCHU86b], [SHRA82], [SIDN81b], [SLEE82a], [SLEE85], [SMIB82], [SWAR83], [WALK87], [WALL82], [WEBB84], [WILE86]

Appendices

Bibliography

This bibliography contains all references that have been cited in the book. The codes at the end of the references indicate the citing authors and the page numbers where cited (codes with asterisks always refer to chapters in this book). Moreover, a number of papers have been added which were not cited, but certainly should form part of a representative bibliography of the field of user modeling. These references contain no author codes and page numbers.

Abbreviations:

IJCAI-77	5th International Joint Conference on Artificial Intelligence, Cambridge, MA, 1977.
IJCAI-81	7th International Joint Conference on Artificial Intelligence, Vancouver, Canada, 1981.
IJCAI-83	8th International Joint Conference on Artificial Intelligence, Karlsruhe, Fed. Rep. of Germany, 1983.
IJCAI-85	9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, 1985.
IJCAI-87	10th International Joint Conference on Artificial Intelligence, Milan, Italy, 1987.
IJCAI-89	11th International Joint Conference on Artificial Intelligence, Detroit, MI, 1989.
AAAI-80	1st National Conference on Artificial Intelligence, Stanford, CA, 1980.
AAAI-82	2nd National Conference on Artificial Intelligence, Pittsburgh, PA, 1982.
AAAI-83	3rd National Conference on Artificial Intelligence, Washington, DC 1983.
AAAI-84	4th National Conference on Artificial Intelligence, Austin, TX, 1984.
AAAI-86	5th National Conference on Artificial Intelligence, Philadelphia, PA, 1986.
AAAI-87	6th National Conference on Artificial Intelligence, Seattle, WA, 1987.
AAAI-88	7th National Conference on Artificial Intelligence, St. Paul, MN, 1988

[ABEL73] Abelson, R. (1973): The Structure of Belief Systems. In: R.C. Schank and K.M. Colby, eds.: Computer Models of Thought and Language. San Francisco, CA: Freeman. [QUIL*: 111]

[ABEL79] Abelson, R. (1979): Differences between Belief and Knowledge Systems. Cognitive Science 3, 355–366. [QUIL*: 111]

[ADEL82] Adelman, H.S. (1982): Identifying Learning Problems at an Early Age: A Critical Appraisal. Journal of Clinical Child Psychology 11, 255–261. [COHE*: 314]

[AIRE83] Airenti, G., B.G. Bara and M. Colombetti (1983): Planning Perlocutionary Acts. In: Proc. IJCAI-83, 78–80. [KOBS*: 61]

[AIRE84] Airenti, G., B.G. Bara and M. Colombetti (1984): Plan Formation and Failure Recovery in Communicative Acts. In: Proc. of the 1984 European Conference on Artificial Intelligence, Pisa, Italy, 259–268. [KOBS*: 61]

[AIRE86] Airenti, G., B.G. Bara and M. Colombetti (1986): Modeling the Relationship between User and System in Man-Machine Dialogue. Paper presented at the International Workshop on User Modeling, Maria Laach, Fed. Rep. of Germany. [SPAR*: 361]

[AJZE71] Ajzen, I. (1971): Attribution of Dispositions to an Actor: Effects of Perceived Decision Freedom and Behavioral Utilities. Journal of Personality and Social Psychology 18, 144–156. [JAME*: 287, 293]

[ALLE78] Allen, J.F. and C.R. Perrault (1978): Participating in Dialogues: Understanding via Plan Deduction. AI-Memo 78-4, Dept. of Computer Science, Univ. of Toronto, Canada.

[ALLE79] Allen, J.F. (1979): A Plan-Based Approach to Speech Act Recognition. TR 131/79, Dept. of Computer Science, Univ. of Toronto, Canada. [WAHL*: 5, 19; MORI*: 370, 374f., 382]

[ALLE80] Allen, J.F. and C.R. Perrault (1980): Analyzing Intention in Utterances. Artificial Intelligence 15, 143–178. [WAHL*: 12, 32; KOBS*: 57, 60; CHIN*: 78; CARB*: 135ff., 153; FINI*: 414]

[ALLE82a] Allen, J.F. (1982): ARGOT: A System Overview. TR101, Computer Science Dept., Univ. of Rochester, NY. [WAHL*: 19]

[ALLE82b] Allen, J.F., M.A. Frisch and D.J. Litman (1982): ARGOT: The Rochester Dialogue System. In: Proc. AAAI-82, 66–70. [FINI*: 413]

[ALLE83a] Allen, J.F. (1983): ARGOT: A System Overview. In: Computers and Mathematics with Applications 9, 97–109. [KOBS*: 58]

[ALLE83b] Allen, J.F. (1983): Recognizing Intentions from Natural Language Utterances. In: M. Brady and R.C. Berwick, eds.: Computational Models of Discourse. Cambridge, MA: MIT Press. [WAHL*: 19; KOBS*: 53, 57, 59ff.]

[ALLE86] Allen, J.F. and D.J. Litman (1986): Plans, Goals, and Natural Language. In: Proceedings of the IEEE 74, 939–947. [WAHL*: 6]

[ALLE87] Allen, J.F. (1987): Understanding Natural Language. Menlo Park, CA: Benjamin/Cummings. [WAHL*: 6]

[ALLG89] Allgayer, J., K. Harbusch, A. Kobsa, C. Reddig, N. Reithinger and D. Schmauks (forthcoming): XTRA: A Natural-Language Access System for Expert Systems. International Journal of Man-Machine Studies. [WAHL*: 5, 18f., 33]

[ALVA86] Alvarado, S., M. Dyer and M. Flowers (1986): Editorial Comprehension in OpED through Argument Units. In: Proc. AAAI-86, 250–256. [QUIL*: 112]

[AMLI88] Amling, S. (1988): Die Möglichkeit der Benutzermodellierung bei der Entwicklung von Expertensystemen unter Verwendung einer Schale am Beispiel von Expert System Environment. Master Thesis, Berufsakademie Stuttgart, Fed. Rep. of Germany.

[AMME86] Ammersbach, K. (1986): Benutzermodelle für Information-Retrieval-Systeme. Master Thesis, Dept. of Computer Science, Technical University of Darmstadt, Fed. Rep. of Germany.

[ANDJ85a] Anderson, J.R., C.F. Boyle and B.J. Reiser (1985): Intelligent Tutoring Systems. Science 228, 456–462. [KASS*: 389]

[ANDJ85b] Anderson, J.R., C.F. Boyle and G. Yost (1985): The Geometry Tutor. In: Proc. IJCAI-85, 1–7. [CHIN*: 77, 100; QUIL*: 116, 125; KASS*: 393]

[ANDN81] Anderson, N.H. (1981): Integration Theory Applied to Cognitive Responses and Attitudes. In: R.E. Petty, T.M. Ostrom and T.C. Brock, eds.: Cognitive Responses in Persuasion. Hillsdale, NJ: Erlbaum. [JAME*: 269]

[ANDR77] Anderson, R.C., R.J. Spiro and M.C. Anderson (1977): Schemata as Scaffolding for the Representation of Information in Connected Discourse. Technical Report 24, Center for the Study of Reading, Urbana, IL. [PARI*: 212]

[APPE82a] Appelt, D.E. (1982): Planning Natural-Language Utterances. In: Proc. AAAI-82, 59–62. [PARI*: 201]

[APPE82b] Appelt, D.E. (1982): Planning Natural-Language Utterances to Satisfy Multiple Goals. Technical Note 259, SRI International, Menlo Park, CA. [WAHL*: 5, 19; MORI*: 375]

[APPE85] Appelt, D.E. (1985): Planning English Sentences. Cambridge etc.: Cambridge Univ. Press. [WAHL*: 5, 16, 19; KOBS*: 53, 58ff., 62f.; PARI*: 201, 203]

[BAKE57] Baker, D.L., R.B. Brownlee and R.W. Fuller (1957): Elements of Physics. Newton, MA: Allyn and Bacon. [PARI*: 205]

[BALL84] Ballard, B., J. Lusth and N. Tinkham (1984): LDC-1: A Transportable, Knowledge-Based Natural Language Processor for Office Environments. ACM Trans. on Office Information Systems 2, 1–25. [WAHL*: 30]

[BARB82] Barber, G. R. (1982): Office Semantics. Ph.D. Thesis, Dept. of Electrical Engineering, MIT, Cambridge, MA. [MORI*: 374]

[BARN86] Barnden, J. (1986): Interpreting Propositional Attitude Reports: Towards Greater Freedom and Control. In: Proc. of the 7th European Conference on Artificial Intelligence, Brighton, England, 398–412. [KOBS*: 55]

[BARR77] Barr, A. (1977): Meta-Knowledge and Memory. Working Paper HPP-77-37, Heuristic Programming Project, Stanford Univ., Stanford, CA. [CHIN*: 98]

[BELK84] Belkin, N. J. (1984): Cognitive Models and Information Transfer. Social Science Information Studies 4, 111–129. [SPAR*: 360, 362]

[BENY87] Benyon, D., P. Innocent and D. Murray (1987): System Adaptivity and the Modelling of Stereotypes. In: H.-J. Bullinger and B. Shackel, eds.: Human-Computer Interaction: INTERACT'87. Amsterdam: North-Holland.

[BOBR77] Bobrow, J. S. and T. Winograd (1977): An Overview of KRL, a Knowledge Representation Language. Cognitive Science 1, 3–46. [MCCO*: 245]

[BOGG85] Boggs, W. M., J. G. Carbonell, M. Kee and I. Monarch (1985): The DYPAR-I Tutorial and Reference Manual. Technical Report, Computer Science Dept., Carnegie-Mellon Univ., Pittsburgh, PA. [LEHM*: 167]

[BOGU85] Boguraev, B. K. (1985): User Modelling in Cooperative Natural Language Front Ends. In: G. N. Gilbert and C. Heath, eds.: Social Action and Artificial Intelligence. London: Gower Press. [WAHL*: 5]

[BONA87] Bonarini, A. (1987): User Modeling in Person-Machine Conversation: A Computational Approach. In: Proc. of COGNITIVA-87, Paris, France, 377–382. [WAHL*: 28, 33]

[BOTM87] Botman, A. M., A. J. Kok and L. Siklossy (1987): Using User Models and Automatic Expertise Acquisition in Man-Machine Interaction. In: Proc. of the 7th International Workshop on Expert Systems and their Applications, Avignon, France, 389–398. [WAHL*: 9]

[BRAC85] Brachman, R. J. and J. G. Schmolze (1985): An Overview of the KL-ONE Knowledge Representation System. Cognitive Science 9, 171–216. [FINI*: 422]

[BREC88] Brecht, B. and M. Jones (1988): Student Models: The Genetic Graph Approach. International Journal of Man-Machine Studies 28, 483–504.

[BRIT63] Britannica Junior Encyclopedia. Chicago: Encyclopedia Britannica Inc., 1963. [PARI*: 201, 205, 209]

[BRIT84] The New Encyclopedia Britannica. Chicago: Encyclopedia Britannica Inc., 1984. [PARI*: 201, 205]

[BROO85] Brooks, H. M., P. J. Daniels and N. J. Belkin (1985): Problem Description and User Models: Developing an Intelligent Interface for Document Retrieval Systems. In: Informatics 8: Advances in Intelligent Retrieval. London: Aslib. [SPAR*: 362]

[BROO86] Brooks, H. M. (1986): An Intelligent Interface for Document Retrieval Systems: Developing the Problem Description and Retrieval Strategy Components. Ph.D. Thesis, Dept. of Information Science, City Univ., London, England. [SPAR*: 362f.]

[BROW75] Brown, J. S., R. Burton, M. Miller, J. DeKleer, S. Purcell, C. Hauseman and R. Bobrow (1975): Steps towards a Theoretical Foundation for Complex, Knowlege-Based CAI. Technical Report 3135, Bolt, Beranek and Newman, Cambridge, MA. [KASS*: 387, 389]

[BROW76] Brown, J. S., R. Rubinstein and R. Burton (1976): A Reactive Learning Environment for Computer Assisted Electronics Instruction. Technical Report 3314, Bolt, Beranek and Newman, Cambridge, MA. [KASS*: 388]

[BROW77] Brown, J. S., R. Burton, C. Hauseman, I. Goldstein, B. Huggins and M. Miller (1977): Aspects of a Theory for Automated Student Modelling. Technical Report 3549, Bolt, Beranek and Newman, Cambridge, MA. [KASS*: 394]

[BROW78] Brown, J. S. and R. R. Burton (1978): Diagnostic Models for Procedural Bugs in Basic Mathematical Skills. Cognitive Science 2, 155–192. [CHIN*: 77, 100; QUIL*: 116, 125; MCCO*: 237; KASS*: 392, 394; FINI*: 414]

[BROW80] Brown, J. S. and K. VanLehn (1980): Repair Theory: A Generative Theory of Bugs in Procedural Skills. Cognitive Science 4, 379–426. [KASS*: 405]

[BROW82] Brown, J. S., R. R. Burton and J. DeKleer (1982): Pedagogical, Natural Language and Knowledge Engineering Techniques in Sophie 1, 2 and 3. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 388f.]

[BRUC75] Bruce, B. C. (1975): Belief Systems and Language Understanding. Technical Report No. 2973, Bolt, Beranek and Newman, Cambridge, MA.

[BUCH84] Buchanan, B. and E. Shortliffe (1984): Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Reading, MA etc.: Addison-Wesley. [COHE*: 325; SPAR*: 352]

[BURG78] Burge, T. (1978): Belief *De Re*. Journal of Philosophy 74, 338–362. [KOBS*: 55]

[BURT76] Burton, R. R. and J. S. Brown (1976): A Tutoring and Student Modelling Paradigm for Gaming Environments. In: Proc. of the 1st Symposium on Computer Science and Education, Anaheim, CA, 236–246. [CHIN*: 77]

[BURT82] Burton, R. R. (1982): Diagnosing Bugs in a Simple Procedural Skill. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [QUIL*: 116, 125; KASS*: 394]

[CANT76] Cantor, N. and W. Mischel (1976): Traits as Prototypes: Effects on Recognition Memory. Journal of Personality and Social Psychology 35, 38–48. [RICH*: 36]

[CARB83] Carberry, S. (1983): Tracking User Goals in an Information-Seeking Environment. In: Proc. AAAI-83, 59–63. [WAHL*: 12; KOBS*: 59f.; CHIN*: 78; CARB*: 155; PARI*: 201, 203; FINI*: 413f.]

[CARB84] Carberry, S. (1984): Understanding Pragmatically Ill-Formed Input. In: Proc. of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics, Stanford Univ., Stanford, CA, 200–206. [MCCO*: 243]

[CARB85a] Carberry, S. (1985): A Pragmatics Based Approach to Understanding Intersentential Ellipsis. In: Proc. of the 23rd Annual Meeting of the Association for Computational Linguistics, Chicago, IL, 188–197.

[CARB85b] Carberry, S. (1985): Pragmatic Modeling in Information System Interfaces. Ph.D. Thesis, Dept. of Computer and Information Sciences, Univ. of Delaware, Newark, DE. [KOBS*: 59f.]

[CARB86a] Carberry, S. (1986): TRACK: Toward a Robust Natural Language Interface. In: Proc. of the 6th Canadian Conference on Artificial Intelligence, Montreal, Canada, 84–88.

[CARB86b] Carberry, S. (1986): User Models: The Problem of Disparity. In: Proc. of the 11th International Conference on Computational Linguistics, Bonn, Fed. Rep. of Germany, 29–34.

[CARB88] Carberry, S. (1988): Modeling the User's Plans and Goals. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), 23–37. [PARI*: 201; CARB*: 133; FINI*: 414]

[CARB*] Carberry, S.: Plan Recognition and its Use in Understanding Dialog. In this volume. [WAHL*: 12; RICH*: 39; KOBS*: 59; CHIN*: 78; QUIL*: 129; PARI*: 201; MCCO*: 251; JAME*: 256, 260, 278f.; SPAR*: 361f.; MORI*: 375f.; FINI*: 414]

[CARN70a] Carbonell, J. R. (1970): AI in CAI: An Artificial Intelligence Approach to Computer-Aided Instruction. IEEE Transactions on Man-Machine Systems 11, 190–202. [KASS*: 390]

[CARN70b] Carbonell, J. R. and A. M. Collins (1970): Mixed-Initiative Systems for Training and Decision-Aid Applications. Technical Report ESD-TR-70-373, Electronics

Systems Division, Laurence G. Hanscom Field, US Air Force, Bedford, MA. [MCCO*: 244, 248]

[CARN81] Carbonell, J. G. and P. J. Hayes (1981): Dynamic Strategy Selection in Flexible Parsing. In: Proc. of the 19th Annual Meeting of the Association for Computational Linguistics, Stanford, CA, 143–147. [LEHM*: 167]

[CARN83a] Carbonell, J. G. (1983): Discourse Pragmatics in Task – Oriented Natural Language Interfaces. In: Proc. of the 21st Annual Meeting of the Association for Computational Linguistics, Cambridge, MA, 164–168. [LEHM*: 164]

[CARN83b] Carbonell, J. G. (1983): The Role of User Modelling in Natural Language Interface Design. Report CMU-CS-83-115, Dept. of Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA.

[CARN83c] Carbonell, J. G., W. M. Boggs, M. L. Mauldin and P. G. Anick (1983): The XCALIBUR Project: A Natural Language Interface to Expert Systems. In: Proc. IJCAI-83, 653–656. [WAHL*: 5, 30; FINI*: 413]

[CARN83d] Carbonell, J. G. and P. J. Hayes (1983): Recovery Strategies for Parsing Extragrammatical Language. American Journal of Computational Linguistics 9, 123–146. [LEHM*: 163f.]

[CARO83] Carlson, L. (1983): Dialogue Games: An Approach to Discourse Analysis. Dordrecht, The Netherlands: Reidel. [JAME*: 285]

[CARR77] Carr, B. and I. Goldstein (1977): Overlays: A Theory of Modelling for Computer Aided Instruction. AI Memo 406, MIT, Cambridge, MA. [WAHL*: 7; CHIN*: 77; KASS*: 388, 390, 400]

[CARV84] Carver, N. F., V. R. Lesser and D. L. McCue (1984): Focusing in Plan Recognition. In: Proc. AAAI-84, 42–48. [CARB*: 153]

[CECI84] Ceci, S. J. and D. Peters (1984): Letters of Reference: A Naturalistic Study of the Effects of Confidentiality. American Psychologist 39, 29–31. [JAME*: 276]

[CHAN78] Chang, C. L. (1978): Finding Missing Joins for Incomplete Queries in Relational Databases. Technical Report RJ 2145, IBM Research Laboratory, San Jose, CA. [CARB*: 146]

[CHAR83] Charniak, E. (1983): The Bayesian Basis of Common Sense Medical Diagnosis. In: Proc. AAAI-83, 70–73. [RICH*: 47]

[CHEI89] Cheikes, B. A. (1989): The Architecture of a Cooperative Respondent. Technical Report MS-CIS-89-13, Dept. of Computer and Information Science, Univ. of Pennsylvania, PA.

[CHEV78] Chevrolet Service Manual. Detroit, MI: General Motors Corporation, 1987. [PARI*: 201, 205]

[CHI81] Chi, M. T. H., R. Glaser and E. Rees (1981): Expertise in Problem Solving. In: R. J. Sternberg, ed.: Advances in the Psychology of Human Intelligence. Hillsdale, NJ: Lawrence Erlbaum. [PARI*: 211]

[CHIN83] Chin, D. N. (1983): A Case Study of Knowledge Representation in UC. In: Proc. IJCAI-83, 388–390. [CHIN*: 82]

[CHIN86] Chin, D. N. (1986): User Modelling in UC, the UNIX Consultant. In: Proc. of the CHI'86, Boston, MA, 24–28. [KOBS*: 57, 65; CHIN*: 74; PARI*: 202, 229; COHE*: 325, 330; FINI*: 414]

[CHIN87] Chin, D. N. (1987): Intelligent Agents as a Basis for Natural Language Interfaces. Ph.D. Thesis, Dept. of Computer Science, Univ. of California, Berkeley, CA. Also Available as Technical Report UCB/CSD 88/396, Division of Computer Science, Univ. of California, Berkeley, CA. [CHIN*: 75, 91, 104]

[CHIN88a] Chin, D. N. (1988): User Models and Discourse Models. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 86–87. [RICH*: 39; SPAR*: 362]

[CHIN88b] Chin, D. N. (1988): Exploiting User Expertise in Answer Expression. In: Proc. AAAI-88, 756–760.

[CHIN*] Chin, D. N.: KNOME: Modeling What the User Knows in UC. In this volume. [WAHL*: 5; RICH*: 36, 51; KOBS*: 57, 60f., 65; QUIL*: 128, 130; CARB*:

160; PARI*: 229; MCCO*: 235; JAME*: 256, 278, 307; SPAR*: 362f.; MORI*: 375, 377; FINI*: 414]

[CLAN79] Clancey, W. J. (1979): Transfer of Rule-Based Expertise through a Tutorial Dialogue. Report STAN-CS-769, Computer Science Dept., Stanford Univ., Stanford, CA. [WAHL*: 7]

[CLAN81] Clancey, W. J. and R. Letsinger (1981): NEOMYCIN: Reconfiguring a Rule-Based Expert System for Application to Teaching. In: Proc. IJCAI-81, 829–836. [KASS*: 398]

[CLAN82] Clancey, W. J. (1982): Tutoring Rules for Guiding a Case Method Dialogue. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 388f., 398]

[CLAN83] Clancey, W. J. (1983): The Epistemology of a Rule-Based System: A Framework for Explanation. Artificial Intelligence 20, 215–251. [KASS*: 398]

[CLAN86a] Clancey, W. J. (1986): From Guidon to Neomycin and Heracles in Twenty Short Lessons. In: AI Magazine 7, 40–60. [KASS*: 398]

[CLAN86b] Clancey, W. J. (1986): Qualitative Student Models. Annual Review of Computer Science 1, 381–450. [SPAR*: 360f.]

[CLHH81] Clark, H. H. and C. R. Marshall (1981): Definite Reference and Mutual Knowledge. In: A. K. Joshi, B. L. Webber and I. A. Sag, eds.: Elements of Discourse Understanding. Cambridge, MA: Cambridge Univ. Press. [KOBS*: 62; JAME*: 277]

[CLHH83] Clark, H. H., R. Schreuder and S. Buttrick (1983): Common Ground and the Understanding of Demonstrative Reference. Journal of Verbal Learning and Verbal Behavior 22, 245–258. [KOBS*: 62]

[CLKL78] Clark, K. L. (1978): Negation as Failure. In: J. Minker and H. Gallaire, eds.: Logic and Databases. New York: Plenum Press. [FINI*: 424]

[CLOW85] Clowes, L, I. Cole, F. Arshad, C. Hopkins and A. Hockley (1985): User Modelling Techniques for Interactive Systems. In: P. Johnson and S. Cook, eds.: People and Computers: Designing the Interface. Cambridge: Cambridge Univ. Press. [WAHL*: 9]

[COBO86] Cobourn, T. F. (1986): An Evaluation of Cleopatra, a Natural Language Interface for CAD. Master's Thesis, Dept. of Computer Science, Carnegie Mellon Univ., Pittsburgh, PA. [LEHM*: 188]

[COHE78] Cohen, P. R. (1978): On Knowing What to Say: Planning Speech Acts. TR-118, Dept. of Computer Science, Univ. of Toronto, Canada. [WAHL*: 17]

[COHE79] Cohen, P. R. and C. R. Perrault (1979): Elements of a Plan-Based Theory of Speech Acts. Cognitive Science 3, 177–212. [KOBS*: 53, 59, 61]

[COHE81] Cohen, P. R., C. R. Perrault and J. F. Allen (1981): Beyond Question Answering. In: W. Lehnert and M. Ringle, eds.: Strategies for Natural Language Processing. Hillsdale, NJ: Lawrence Erlbaum. [CARB*: 157]

[COHE85] Cohen, R. (1985): The Need for Pragmatics in Natural Language Understanding. In: Proc. of the Theoretical Issues in Natural Language Understanding Workshop, Dalhousie Univ., Halifax, Nova Scotia. [COHE*: 326]

[COHE88] Cohen, R. (1988): On the Relationship between User Models and Discourse Models. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 88–90. [RICH*: 39; SPAR*: 362]

[COHE*] Cohen, R. and M. Jones. Incorporating User Models into Expert Systems for Educational Diagnosis. In this volume. [WAHL*: 19; CARB*: 161; PARI*: 231; MCCO*: 250; SPAR*: 362]

[COLB82] Colbourn (Jones), M. J. (1982): Computer-Guided Diagnosis of Learning Disabilities: A Prototype. M. ED. Thesis, Dept. for the Education of Exceptional Children, Univ. of Saskatchewan, Canada; also Report 82-8, Dept. of Computational Science, Univ. of Saskatchewan; also ERIC Report No. ED 222-032. [COHE*: 314]

[COLB83] Colbourn (Jones), M. J. and J. McLeod (1983): The Potential and Feasibility of an Expert System for Educational Diagnosis. In: Proc. of the IFIP 9th World Computer Congress, Paris, France, 891–896. [COHE*: 314]

[COLB84] Colbourn (Jones), M. J. and J. McLeod (1984): Computer-Guided Education Diagnosis: A Prototype Expert System. Journal of Special Education Technology 6, 30–39. [COHE*: 314]

[COLL62] Collier's Encyclopedia. New York: Crowell-Collier, 1962. [PARI*: 201, 205, 207]

[COMM71] Committee on Teacher Education and Professional Standards for Educators of Exceptional Children in Canada. The National Institute for Mental Retardation, Downswiew, Canada, 1971. [COHE*: 314]

[DANI86a] Daniels, P. J. (1986): Cognitive Models in Information Retrieval: An Evaluative Overview. Journal of Documentation 42, 217–328. [SPAR*: 360]

[DANI86b] Daniels, P. J. (1986): The User Modelling Function of an Intelligent Interface for Document Retrieval Systems. In: B. Brookes, ed.: Intelligent Information Systems for the Information Society. Amsterdam: North-Holland. [SPAR*: 363]

[DAVD82] Davidson, J. (1982): Natural Language Access to Data Bases: User Modelling and Focus. In: Proc. of the 4th CSCSI/SCEIO Conference, Saskatoon, Saskatchewan, Canada, 204–211.

[DAVI77] Davis, R. and B. G. Buchanan (1977): Meta-Level Knowledge: Overview and Applications. In: Proc. IJCAI-77, 920–927. [CHIN*: 96]

[DAVS84] Davison, A. (1984): Readibility Formulas and Comprehension. Comprehension Instruction, Perspectives and Suggestions. New York etc.: Longman. [PARI*: 201]

[DE86] De, S. (1986): Providing Effective Decision Support: Modelling Users and their Requirements. Decision Support Systems 2, 309–319.

[DEDE86] Dede, C. (1986): A Review and Synthesis of Recent Research in Intelligent Computer-Assisted Instruction. International Journal of Man-Machine Studies 24, 329–353. [WAHL*: 7; KOBS*: 54]

[DEKL86] DeKleer, J. (1986): An Assumption-Based TMS. Artificial Intelligence 28, 127–162. [RICH*: 47; FINI*: 429]

[DESM87] Desmarais, M. C. and M. Pavel (1987): User Knowledge Evaluation: An Experiment with UNIX. In: H.-J. Bullinger and B. Shackel, eds.: Human-Computer Interaction: INTERACT'87. Amsterdam: North-Holland.

[DOLA85] Dolan, C. and M. Dyer (1985): Learning Planning Heuristics through Observation. In: Proc. IJCAI-85, 600–602. [QUIL*: 131]

[DOLA86] Dolan, C. and M. Dyer (1986): Encoding Planning Knowledge for Recognition, Construction, and Learning. In: Proc. of the 8th Annual Cognitive Science Society, Amherst, MA, 488–499. [QUIL*: 126, 131]

[DOYL79] Doyle, J. (1979): A Truth Maintenance System. Artificial Intelligence 12, 231–272. [WAHL*: 28; RICH*: 42, 47]

[DOYL80] Doyle, J. and P. London (1980): A Selected Descriptor-Indexed Bibliography to the Literature on Belief Revision. SIGART Newsletter 71, 7–23. [WAHL*: 28]

[DOYL83] Doyle, J. (1983): The Ins and Outs of Reason Maintenance. In: Proc. IJCAI-83, 352–354. [WAHL*: 28]

[DURH83] Durham, I., D. D. Lamb and J. B. Saxe (1983): Spelling Correction in User Interfaces. Communications of the ACM 26, 764–773. [LEHM*: 170]

[DYER83] Dyer, M. (1983): In-Depth Understanding: A Computer Model of Narrative Comprehension. Cambridge, MA: MIT Press. [QUIL*: 111f., 126]

[DYER88] Dyer, M., M. Flowers and J. F. Reeves (1988): Recognizing Situational Ironies: A Computer Model of Irony Recognition and Narrative Understanding. Advances in Computing and the Humanities 1 (in press). [QUIL*: 126]

[EAGL81] Eagly, A. H., S. Chaiken and W. Wood (1981): An Attribution Analysis of Persuasion. In: J. H. Harvey, W. Ickes and F. Kidd, eds.: New Directions in Attribution Research 3. Hillsdale, NJ: Erlbaum. [JAME*: 292]

[ENCY82] The New Encyclopedia of Science. Milwaukee, WI: Raintree Publishers, 1982. [PARI*: 201, 205]

[FAGI85] Fagin, R. and J. Y. Halpern (1985): Belief, Awareness and Limited Reasoning: Preliminary Report. In: Proc. IJCAI-85, 491–501. [WAHL*: 17]

[FAIN85] Fain, J., J. G. Carbonell, P. J. Hayes and S. N. Minton (1985): MULTIPAR: A Robust Entity-Oriented Parser. In: Proc. of the 7th Cognitive Science Society Conference, Irvine, CA, 110–119. [LEHM*: 164f.]

[FARR84] Farrell, R. G., J. R. Anderson and B. J. Reiser (1984): An Interactive Computer Based Tutor for LISP. In: Proc. AAAI-84, 106–109. [KASS*: 388]

[FIKE71] Fikes, R. E. and N. J. Nilsson (1971): STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence 2, 189–208. [QUIL*: 111; CARB*: 136]

[FINI83] Finin, T. W. (1983): Providing Help and Advice in Task Oriented Systems. Proc. IJCAI-83, 176–178. [WAHL*: 32; MORI*: 375; FINI*: 411, 415]

[FINI86] Finin, T. W. and D. Drager (1986): GUMS $_1$: A General User Modeling System. In: Proc. of the 6th Canadian Conference on Artificial Intelligence, Montreal, Canada, 24–29. [WAHL*: 28, 33; KOBS*: 57; COHE*: 330]

[FINI*] Finin, T. W.: GUMS: A General User Modeling Shell. *In this volume.* [WAHL*: 28, 33; KOBS*: 57, 65; CHIN*: 106; MCCO*: 235; JAME*: 256, 294]

[FLOW82] Flowers, M., R. McGuire and L. Birnbaum (1982): Adversary Arguments and the Logic of Personal Attacks. In: W. Lehnert and M. Ringle, eds.: Strategies for Natural Language Processing. Hillsdale, NJ: Lawrence Erlbaum. [QUIL*: 112]

[GAZD79] Gazdar, G. (1979): Pragmatics: Implicature, Presupposition, and Logical Form. New York: Academic Press. [JAME*: 261, 301]

[GENE82] Genesereth, M. (1982): The Role of Plans in Intelligent Teaching Systems. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [RICH*: 39]

[GENT83] Gentner, D. and A. L. Stevens (1983): Mental Models. Hillsdale, NJ: Lawrence Erlbaum. [WAHL*: 9]

[GEOR83] Georgeff, M. (1983): Communication and Interaction in Multi-Agent Planning Systems. Proc. AAAI-83, 125–129. [WAHL*: 7; KOBS*: 54]

[GERS81] Gershman, A. (1981): Finding Out What the User Wants: Steps Toward an Automated Yellow Pages Assistant. In: Proc. IJCAI-81, 423–425. [FINI*: 413]

[GILB87] Gilbert, G. N. (1987): Cognitive and Social Models of the User. In: H.-J. Bullinger and B. Shackel, eds.: Human-Computer Interaction: INTERACT'87. Amsterdam: North-Holland.

[GINS86] Ginsberg, M. L. (1986): Multi-Valued Logics. In: Proc. AAAI-86, 243–247. [FINI*: 425]

[GOFF59] Goffman, E. (1959): The Presentation of Self in Everyday Life. Garden City, NY: Doubleday. [JAME*: 298]

[GOLD82] Goldstein, I. P. (1982): The Genetic Graph: A Representation for the Evolution of Procedural Knowledge. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 388, 400]

[GOOD84] Good, M. D., J. A. Whiteside, D. R. Wixon and J. J. Jones (1984): Building a User-Derived Interface. Communications of the ACM 27, 1032–1043. [LEHM*: 183, 188]

[GOOM86] Goodman, B. A. (1986): Miscommunication and Plan Recognition. Paper presented at the International Workshop on User Modeling, Maria Laach, Fed. Rep. of Germany. [QUIL*: 129; SPAR*: 362]

[GREE88] Green, T. R. G., F. Schiele and S. J. Payne (forthcoming): Formalizable Models of User Knowledge. In: T. R. G. Green, J.-M. Hoc, D. M. Murray and G. C. Van der Veer, eds.: Working with Computers: Theory versus Outcome. New York: Academic Press.

[GREI66] Greimas, A. J. (1966): Semantique Structurale: Recherche de Methode. Paris: Larousse. [MORI*: 382f.]

[GREW78] Grewendorf, G. (1978): Zur Semantik von Wertäußerungen. Germanistische Linguistik 2, 155. [MORI*: 377]

[GREW83] Grewendorf, G. (1983): What Answers Can be Given? In: F. Kiefer, ed.: Questions and Answers. Dordrecht, The Netherlands: Reidel. [JAME*: 262]

[GRIC57] Grice, H. P. (1957): Meaning. Philosophical Review 56, 377–388. [CARB*: 146]

[GRIC69] Grice, H. P. (1969): Utterer's Meaning and Intentions. Philosophical Review 68, 147–177. [CARB*: 146]

[GRIC75] Grice, H. P. (1975): Logic and Conversation. In: P. Cole and J. L. Morgan, eds.: Syntax and Semantics 3: Speech Acts. New York: Academic Press. [CARB*: 146, 156; PARI*: 201, 212; JAME*: 261, 272, 301; COHE*: 321]

[GRIM75] Grimes, J. E. (1975): The Thread of Discourse. The Hague, The Netherlands: Mouton. [PARI*: 206]

[GROS77a] Grosz, B. J. (1977): The Representation and Use of Focus in a System for Understanding Dialogs. Proc. IJCAI-77, 67–76. [CARB*: 135, 139]

[GROS77b] Grosz, B. J. (1977): The Representation and Use of Focus in Dialogue Understanding. Technical Report 151, SRI International, Menlo Park, CA. [MCCO*: 245]

[GROS81] Grosz, B. J. (1981): Focusing and Description in Natural Language Dialogues. In: B. Webber, A. Joshi and I. Sag, eds.: Elements of Discourse Understanding. Cambridge, England: Cambridge Univ. Press. [CARB*: 152; MCCO*: 244; MORI*: 375, 382]

[GROS83] Grosz, B. J., A. K. Joshi and S. Weinstein (1983): Providing a Unified Account of Definite Noun Phrases in Discourse. In: Proc. of the 21st Annual Meeting of the Association for Computational Linguistics, Cambridge, MA, 44–50. [MCCO*: 244]

[GROS85] Grosz, B. J. and C. Sidner (1985): Discourse Structure and the Proper Treatment of Interruptions. In: Proc. IJCAI-85, 832–839. [MCCO*: 252]

[GROS86] Grosz, B. J. and C. Sidner (1986): Attention, Intentions, and the Structure of Discourse. Computational Linguistics 12, 175–204. [MORI*: 376]

[HAAS86] Haas, A. R. (1986): A Syntactic Theory of Belief and Action. Artificial Intelligence 28, 245–292. [KOBS*: 58]

[HALA83] Halasz, F. G. and T. P. Moran (1983): Mental Models and Problem Solving in Using a Calculator. In: Proc. of the CHI'83 Conference on Human Factors in Computing Systems, Boston, MA, 212–216.

[HALP85a] Halpern, J. Y. and Y. O. Moses (1985): A Guide to the Modal Logics of Knowledge and Belief: Preliminary Draft. In: Proc. IJCAI-85, 480–490. [WAHL*: 17; KOBS*: 62, 67]

[HALP85b] Halpern, J. Y. and Y. O. Moses (1985): Knowledge and Common Knowledge in a Distributed Environment. Report RJ 4421, IBM Research Laboratory, San Jose, CA. [KOBS*: 54, 63]

[HALP86] Halpern, J. Y. (1986): Reasoning about Knowledge: An Overview. In: Proc. of the 1986 Conference on Theoretical Aspects of Reasoning about Knowledge, Monterey, CA, 1–17. [KOBS*: 67]

[HAMI76] Hamilton, D. L. and R. K. Gifford (1976): Illusory Correlation in Interpersonal Perception: A Cognition Basis of Stereotypic Judgements. Journal of Experimental Social Psychology 12, 392–407. [RICH*: 36]

[HARE52] Hare, R. M. (1952): The Language of Morals. Oxford: Clarendon Press. [MORI*: 377]

[HARR63] Harrah, D. (1963): Communication: A Logical Model. Cambridge, MA: MIT Press. [JAME*: 262, 285]

[HAYE76] Hayes, P. J. and M. A. Rosner (1976): ULLY: A Program for Handling Conversations. In: Proc. of the 1976 AISB Summer Conference, Edinburgh, Scotland, 137–147. [WAHL*: 5, 18]

[HAYE84] Hayes, P. J. (1984): Entity-Oriented Parsing. In: Proc. of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics, Stanford Univ., Stanford, CA, 212–217. [LEHM*: 165]

[HECK88] Hecking, M., C. Kemke, E. Nessen, D. Dengler, M. Gutmann and G. Hector (1988): The SINIX Consultant: A Progress Report. Memo No. 28, Dept. of Computer Science, Univ. of Saarbrücken, Fed. Rep. of Germany. [WAHL*: 5]

[HEIN85] Heines, J. M. and T. O'Shea (1985): The Design of a Rule-Based CAI Tutorial. International Journal of Man-Machine Studies 23, 1–26. [KASS*: 389]

[HERR87] Herrmann, T. (1987): Sprachpsychologische Beiträge zur Partnerbezogenheit des Sprechens. Research Group on Language and Cognition, Dept. of Psychology, Univ. of Mannheim, Fed. Rep. of Germany.

[HINT62] Hintikka, J. (1962): Knowledge and Belief. Ithaca, NY: Cornell Univ. Press. [WAHL*: 16]

[HINT75] Hintikka, J. (1975): Answers to Questions. In: H. Hiz, ed.: Questions. Dordrecht, The Netherlands: Reidel. [JAME*: 262]

[HOBB78a] Hobbs, J. (1978): Coherence and Coreference. Technical Note 168, SRI International, Menlo Park, CA. [PARI*: 206]

[HOBB78b] Hobbs, J. (1978): Why is a Discourse Coherent? Technical Report 176, SRI International, Menlo Park, CA. [PARI*: 206]

[HOBB80] Hobbs, J. and D. Evans (1980): Conversation as Planned Behavior. Cognitive Science 4, 349–377. [PARI*: 206]

[HOEP83a] Hoeppner, W. and K. Morik (1983): Was haben Hotels, Straßenkreuzungen und Fische gemeinsam? – Mit HAM-ANS spricht man darüber. Report ANS-20, Research Unit for Information Science and Artificial Intelligence, Univ. of Hamburg, Fed. Rep. of Germany. [KOBS*: 63]

[HOEP83b] Hoeppner, W., T. Christaller, H. Marburger, K. Morik, B. Nebel, M. O'Leary and W. Wahlster (1983): Beyond Domain-Independence: Experience with the Development of a German Language Access System to Highly Diverse Background Systems. In: Proc. IJCAI-83, 588–594. [WAHL*: 10, 30; SPAR*: 362; MORI*: 364, 370; FINI*: 415]

[HOEP84a] Hoeppner, W., S. Busemann, T. Christaller, H. Marburger, K. Morik and B. Nebel (1984): Dialoging HAM-ANS: Commented Terminal Sessions with a Natural Language System. Memo ANS-23, Research Unit for Information Science and Artificial Intelligence, Univ. of Hamburg, Fed. Rep. of Germany. [MORI*: 379]

[HOEP84b] Hoeppner, W., K. Morik and H. Marburger (1984): Talking it Over: The Natural Language Dialog System HAM-ANS. Technical Report ANS-26, Research Unit for Information Science and Artificial Intelligence, Univ. of Hamburg, Fed. Rep. of Germany. [CHIN*: 78; PARI*: 202]

[HOLL86] Holland, J. H., K. J. Holyoak, R. E. Nisbett and P. R. Thagard (1986): Induction: Processes of Inference, Learning, and Discovery. Cambridge, MA: MIT Press. [JAME*: 294]

[HORA87] Horacek, H., H. Kaindl and M. Wagner (1987): Probabilities in Game-Playing: Possible Meanings and Applications. In: B. Buchberger and J. Retti, eds.: 3. Österreichische Artificial-Intelligence-Tagung. Berlin: Springer. [WAHL*: 8]

[HORT87] Horton, D. L. (1987): Incorporating Agents' Beliefs in a Model of Presupposition. Technical Report CSRI-201, Computer Systems Research Institute, Univ. of Toronto, Canada.

[HORN78] Horn, L. R. (1978): Remarks on Neg-Raising. In: P. Cole, ed.: Syntax and Semantics 9: Pragmatics. New York etc.: Academic Press. [KOBS*: 57]

[HOVY86] Hovy, E. H. (1986): Putting Affect into Text. In: Proc. of the 8th Annual Conference of the Cognitive Science Society, Amherst, MA, 669–675. [JAME*: 271]

[HOWE84] Howe, A. and T. Finin (1984): Using Spreading Activation to Identify Relevant Help. In: Proc. of the 5th CSCSI/SCEIO Conference, London, Ontario, 25–27. [FINI*: 411]

[JAME80] Jameson, A., W. Hoeppner and W. Wahlster (1980): The Natural Language System HAM-RPM as a Hotel Manager: Some Representational Prerequisites. In: R. Wilhelm, ed.: GI – 10. Jahrestagung Saarbrücken. Berlin: Springer. [WAHL*: 5, 10; MORI*: 382]

[JAME82] Jameson, A. and W. Wahlster (1982): User Modelling in Anaphora Generation: Ellipsis and Definite Description. In: Proc. of the 1982 European Conference on Artificial Intelligence, Orsay, France, 222–227. [WAHL*: 22; JAME*: 274; MORI*: 371]

[JAME83] Jameson, A. (1983): Impression Monitoring in Evaluation-Oriented Dialog: The Role of the Listener's Assumed Expectations and Values in the Generation of Informative Statements. In: Proc. IJCAI-83, 616–620. [WAHL*: 25; JAME*: 258, 311; MORI*: 375]

[JAME84] Jameson, A. (1984): Interpreting and Predicting Belief Statements. In: Proc. of the 6th European Conference on Artificial Intelligence, Pisa, Italy, 419–422. [JAME*: 294]

[JAME86] Jameson, A. (1986): Some Psychological Literature of Relevance to User Modeling. Unpublished Memo, Psychological Laboratory, Univ. of Nijmegen, The Netherlands.

[JAME87] Jameson, A. (1987): How to Appear to be Conforming to the 'Maxims' even if You Prefer to Violate them. In: G. Kempen, ed.: Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics. Dordrecht, The Netherlands: Nijhoff. [JAME*: 311]

[JAME*] Jameson, A.: But What Will the Listener Think? Belief Ascription and Image Maintenance in Dialog. In this volume. [WAHL*: 14, 25; SPAR*: 361f.]

[JOHN84] Johnson, W. L. and E. Soloway (1984): Intention-Based Diagnostics of Programming Errors. In: Proc. AAAI-84, 162–168. [CHIN*: 77f., 100; KASS*: 393, 403; FINI*: 414]

[JOHN85] Johnson, W. L. (1985): Intention-Based Diagnosis of Errors in Novice Programs. Ph.D. Thesis, Dept. of Computer Science, Yale Univ., New Haven, CT. [KASS*: 403]

[JOHP83] Johnson-Laird, P. N. (1983): Mental Models. Cambridge: Cambridge Univ. Press.

[JONE65] Jones, E. E. and K. E. Davis (1965): From Acts to Dispositions: The Attribution Process in Person Perception. In: L. Berkowitz, ed.: Advances in Experimental Social Psychology 2. New York: Academic Press. [JAME*: 287]

[JONE86] Jones, M. J. and J. Tubman (1986): Expert Systems for Educational Diagnosis: An Experiment with THEORIST. In: Proc. of the 6th International Workshop on Expert Systems and their Applications, Avignon, France, 1265–1283. [COHE*: 314]

[JONE87] Jones, M. J., J. McLeod, G. Robertson, J. Tubman and B. Wasson (1987): The CGD Project: An Educational Diagnostic System Based on THEORIST. Technical Report CS-87-26, Dept. of Computer Science, Univ. of Waterloo, Canada. [COHE*: 314, 322]

[JOSH82] Joshi, A. K. (1982): Mutual Beliefs in Question-Answer Systems. In: N. V. Smith, ed.: Mutual Knowledge. London etc.: Academic Press. [KOBS*: 62; CARB*: 152, 158f.; JAME*: 256]

[JOSH84a] Joshi, A. K., B. Webber and R. M. Weischedel (1984): Living up to Expectations: Computing Expert Responses. In: Proc. AAAI-84, 169–175. [CHIN*: 104; QUIL*: 126; JAME*: 256; COHE*: 329; FINI*: 414]

[JOSH84b] Joshi, A. K., B. Webber and R. M. Weischedel (1984): Preventing False Inferences. In: Proc. of the 10 International Conference on Computational Linguistics, Stanford, CA, 134–138. [CHIN*: 104; FINI*: 414]

[JOYC83] Joyce, C. and L. Wingerson (1983): Can We Adjust to Computer Culture? In: New Scientist, April 1983, 72–73. [WAHL*: 14]

[KAPD69] Kaplan, D. (1969): Quantifying In. In: D. Davidson and J. Hintikka, eds.: Words and Objections: Essays on the Work of W. V. Quine. Dordrecht, The Netherlands: Reidel. [KOBS*: 58]

[KAPS79] Kaplan, S. J. (1979): Cooperative Responses from a Portable Natural Language Data Base Query System. Ph.D. Thesis, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia. Also available as: Report HPP-79-19, Computer Science Dept., Stanford Univ., Stanford, CA. [WAHL*: 19, 26; KOBS*: 58; CARB*: 146, 152; MCCO*: 237]

[KAPS82] Kaplan, S. J. (1982): Cooperative Responses from a Portable Natural Language Query System. Artificial Intelligence 19, 165–188. [WAHL*: 19, 26; PARI*: 201; FINI*: 416]

[KAPS83] Kaplan, S. J. (1983): Cooperative Responses from a Portable Natural Language Database Query System. In: M. Brady and R. C. Berwick, eds.: Computational Models of Discourse. Cambridge, MA: MIT Press. [WAHL*: 19, 26; KOBS*: 58; CHIN*: 100]

[KARL85] Karlin, R. (1985): Romper Mumbles. Technical Report 85-41, Univ. of Pennsylvania, Philadelphia, PA. [MCCO*: 236]

[KASS86] Kass, R. (1986): The Role of User Modelling in Intelligent Tutoring Systems. Technical Report MS-CIS-86-58, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA.

[KASS87a] Kass, R. (1987): Implicit Acquisition of User Models in Cooperative Advisory Systems. Technical Report MS-CIS-87-05, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA. [WAHL*: 12, 18, 33; PARI*: 229; KASS*: 409; FINI*: 416]

[KASS87b] Kass, R. and T. Finin (1987): Rules for the Implicit Acquisition of Knowledge about the User. In: Proc. AAAI-87, 295–300. [KASS*: 409; FINI*: 416]

[KASS87c] Kass, R. and T. Finin (1987): The Need for User Models in Generating Expert System Explanations. Technical Report MS-CIS-87-86, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA.

[KASS88a] Kass, R. (1988): Acquiring a Model of the User's Beliefs from a Cooperative Advisory Dialog. Ph.D. Thesis, Dept. of Computer and Information Science, University of Pennsylvania, PA.

[KASS88b] Kass, R. and T. Finin (1988): Modeling the User in Natural Language Systems. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), 5–22. [FINI*: 412]

[KASS88c] Kass, R. and T. Finin (1988): The Need for User Models in Generating Expert System Explanations. Technical Report MS-CIS-88-37, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA. [WAHL*: 5]

[KASS*] Kass, R.: Student Modeling in Intelligent Tutoring Systems: Implications for User Modeling. In this volume. [WAHL*: 7, 15; CHIN*: 77; FINI*: 412, 414]

[KAUT86] Kautz, H. and J. Allen (1986): Generalized Plan Recognition. In: Proc. AAAI-86, 423–427. [QUIL*: 129]

[KAYD85] Kay, D. S. and J. B. Black (1985): The Evolution of Knowledge Representations with Increasing Expertise in Using Systems. In: Proc. of the 7th Annual Conference of the Cognitive Science Society, Irvine, CA, 140–149. [CHIN*: 77]

[KAYM79] Kay, M. (1979): Functional Grammar. In: Proc. of the 5th Meeting of the Berkeley Linguistics Society, Berkeley, CA, 142–158. [PARI*: 223]

[KEMK86] Kemke, C. (1986): The SINIX Consultant: Requirements, Design, and Implementation of an Intelligent Help System for a UNIX Derivative. Report No. 11, AI Laboratory, Dept. of Computer Science, Univ. of Saarbrücken, Fed. Rep. of Germany. [WAHL*: 5, 32; QUIL*: 128]

[KIMB82] Kimball, R. (1982): A Self-Improving Tutor for Symbolic Integration. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 400]

[KOBS84] Kobsa, A. (1984): Three Steps in Constructing Mutual Belief Models from User Assertions. In: Proc. of the 6th European Conference on Artificial Intelligence, Pisa, Italy, 423–427. [WAHL*: 11; KOBS*: 62; FINI*: 415]

[KOBS85a] Kobsa, A. (1985): Benutzermodellierung in Dialogsystemen. Berlin: Springer. [WAHL*: 7, 13, 16ff., 32; KOBS*: 53, 57, 60f., 67; JAME*: 278; MORI*: 375, 382]

[KOBS85b] Kobsa, A. (1985): User Modelling in Dialogue Systems: Potentials and Hazards. Report 85-01, Dept. of Medical Cybernetics and Artificial Intelligence, Univ. of Vienna, Austria. [CHIN*: 85]

[KOBS85c] Kobsa, A. (1985): Using Situation Descriptions and Russellian Attitudes for Representing Beliefs and Wants. In: Proc. IJCAI-85, 513–515. [KOBS*: 58, 67]

[KOBS85d] Kobsa, A. (1985): VIE-DPM: A User Model in a Natural-Language Dialogue System. In: J. Laubsch, ed.: GWAI-84, 8th German Workshop on Artificial Intelligence. Berlin: Springer. [WAHL*: 11, 18; KOBS*: 53; CHIN*: 85]

[KOBS86a] Kobsa, A. (1986): Generating a User Model from Wh-Questions in the VIE-LANG System. In: P. Hellwig und H. Lehmann, eds.: Trends in der Linguistischen Datenverarbeitung. Hildesheim, Fed. Rep. of Germany etc.: Olms. [WAHL*: 11; KOBS*: 60; CHIN*: 85]

[KOBS86b] Kobsa, A., J. Allgayer, C. Reddig, N. Reithinger, D. Schmauks, K. Harbusch and W. Wahlster (1986): Combining Deictic Gestures and Natural Language for Referent Identification. In: Proc. of the 11th International Conference on Computational Linguistics, Bonn, Fed. Rep. of Germany, 356–361. [WAHL*: 5]

[KOBS88] Kobsa, A. (1988): User Models and Dialog Models: United They Stand... In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 91–95. [WAHL*: 18, 33; RICH*: 39; SPAR*: 362]

[KOBS89] Kobsa, A. (1989): User Modeling in Dialog Systems: Potentials and Hazards. In: Proc. of the IFIP/GI Conference on Opportunities and Risks of Artificial Intelligence Systems, Hamburg, Fed. Rep. of Germany (forthcoming).

[KOBS*] Kobsa A.: A Taxonomy of Beliefs and Goals for User Models in Dialog Systems. In this volume. [JAME*: 277; SPAR*: 360, 362; KASS*: 406]

[KOLO84] Kolodner, J. L. (1984): Retrieval and Organizational Strategies in Conceptual Memory. Hillsdale, NJ: Lawrence Erlbaum. [QUIL*: 125]

[KOLO86] Kolodner, J. L. and R. E. Cullingford (1986): Towards a Memory Architecture that Supports Reminding. In: Proc. of the 8th Annual Cognitive Science Society, Amherst, MA, 467–477. [QUIL*: 125]

[KONO80] Konolige, K. and N. J. Nilsson (1980): Multiple Agent Planning Systems. In: Proc. AAAI-80, 138–141. [WAHL*: 7; KOBS*: 54]

[KONO81] Konolige, K. (1981): A First-Order Formalisation of Knowledge and Action for a Multi-Agent Planning System. In: J. E. Hayes, D. Michie and Y.-H. Pao, eds.: Machine Intelligence 10. Chichester: Ellis Horwood. [WAHL*: 16; KOBS*: 58, 63]

[KONO84] Konolige, K. (1984): Belief and Incompleteness. Report CSLI-84-4, Center for the Study of Language and Information, Stanford, CA. [MORI*: 369]

[KONO85] Konolige, K. (1985): User Modelling, Common-Sense Reasoning and the Belief-Desire-Intention Paradigm. In: Proc. IJCAI-85, 1300.

[KOWA79] Kowalski, R. (1979): Logic for Problem Solving. New York: North-Holland. [FINI*: 424]

[KRAU83] Kraut, R. (1983): There are no *De Dicto* Attitudes. Synthese 54, 275–294. [KOBS*: 55]

[LANC87] Lancaster, J. S. and J. L. Kolodner (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. [PARI*: 211]

[LEBO83] Lebowitz, M. (1983): RESEARCHER: An Overview. In: Proc. AAAI-83, 232–235. [PARI*: 202]

[LEBO85] Lebowitz, M. (1985): RESEARCHER: An Experimental Intelligent Information System. In: Proc. IJCAI-85, 858–862. [PARI*: 202]

[LEBO86] Lebowitz, M. (1986): An Experiment in Intelligent Information Systems: RESEARCHER. In: R. Davies, ed.: Intelligent Library and Information Systems. London: Ellis Horwood. [PARI*: 202]

[LEHD84] Lehman, D. (1984): Knowledge, Common Knowledge and Related Puzzles. In: Proc. of the 3rd Annual ACM Conference on Principles of Distributed Computing, Vancouver, Canada, 62–67. [KOBS*: 54, 62]

[LEHM*] Lehman, J. F. and J. G. Carbonell: Learning the User's Language: A Step Towards Automated Creation of User Models. In this volume. [CHIN*: 107; CARB*: 160; SPAR*: 363; MORI*: 365]

[LEHN77] Lehnert, W. (1977): A Conceptual Theory of Question Answering. In: Proc. IJCAI-77, 158–164. [PARI*: 202]

[LEVE84] Levesque, H. (1984): A Logic of Implicit and Explicit Belief. In: Proc. AAAI-84, 198–202. [COHE*: 326]

[LEVI77] Levin, J. A. and J. A. Moore (1977): Dialogue-Games: Metacommunication Structures for Natural Language Interaction. Cognitive Science 1, 395–420. [WAHL*: 30]

[LEWI69] Lewis, D. K. (1969): Convention: A Philosophical Study. Cambridge: Harvard Univ. Press. [KOBS*: 62]

[LITM84] Litman, D. J. and J. F. Allen (1984): A Plan Recognition Model for Clarification Subdialogues. In: Proc. of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics, Stanford Univ., Stanford, CA, 302–311. [CHIN*: 78]

[LITM85] Litman, D. J. (1985): Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogue. Ph.D. Thesis, Report TR70, Dept. of Computer Science, Univ. of Rochester, NY. [SPAR*: 362; MORI*: 370, 374f., 382]

[LITM86] Litman, D. J. (1986): Linguistic Coherence: A Plan-Based Alternative. In: Proc. of the 24th Annual Meeting of the Association for Computational Linguistics, New York, NY, 215–223. [CARB*: 135]

[LNR74] The LNR Research Group (1974): MEMOD: A Model for the Representation of Language and Semantic Structures in Human Long-Term Memory. Unpublished Memo, Center for Human Interaction Processing, Univ. of California at San Diego, CA.

[LOND82] London, B. and W. J. Clancey (1982): Plan Recognition Strategies in Student Modeling: Prediction and Description. In: Proc. AAAI-82, 335–338. [KASS*: 391, 398]

[MACL82] Maclaran, R. (1982): Comments on Joshi's Paper. In: N. V. Smith, ed.: Mutual Knowledge. London: Academic Press. [JAME*: 256]

[MACM83] Macmillan, S. A. (1983): User Models to Personalize an Intelligent Agent. Ph.D. Thesis, School of Education, Stanford Univ., Stanford, CA. [CHIN*: 84, 106]

[MAND88] Mandl, H. and A. Lesgold, eds. (1988): Learning Issues for Intelligent Tutoring Systems. New York: Springer.

[MANN84] Mann, W. C. (1984): Discourse Structure for Text Generation. Technical Report ISI/RR-84-127, Information Sciences Institute, Marina del Rey, CA. [PARI*: 206]

[MARB83] Marburger, H. and W. Wahlster (1983): Case Role Filling as a Side Effect of Visual Search. In: Proc. of the 1st Conference of the European Chapter of the Association for Computational Linguistics, Pisa, Italy, 188–195. [WAHL*: 23; MORI*: 371]

[MARB87] Marburger, H. (1987): A Strategy for Producing Cooperative NL Reactions in a Database Interface. In: P. Jorrand and V. Sgurev, eds.: Artificial Intelligence II: Methodology, Systems, Applications. Amsterdam etc.: North-Holland. [WAHL*: 26; CHIN*: 101; MORI*: 370]

[MARB88] Marburger, H. (1988): Generierung kooperativer natürlichsprachlicher Antworten in einem Dialogsystem mit Zugang zu relationalen Datenbanken. Ph. D. Thesis, Dept. of Computer Science, Univ. of Saarbrücken, Fed. Rep. of Germany. [WAHL*: 26; CHIN*: 101]

[MARH87] Mark, H. F., D. F. Othmer, C. G. Overberger and G. T. Seaborg, eds.; Executive Editor: M. Grayson; Associate Editor: D. Eckroth (1987): Encyclopedia of Chemical Technology. New York: John Wiley and Sons. [PARI*: 213]

[MARJ85] Martin, J. H. (1985): Knowledge Acquisition through Natural Language Dialogue. In: Proc. of the 2nd IEEE Conference on Artificial Intelligence Applications, Miami, FL, 582–586. [CHIN*: 88]

[MARP83] Martin, P., D. Appelt and F. Pereira (1983): Transportability and Generality in a Natural-Language Interface System. In: Proc. IJCAI-83, 573–581. [WAHL*: 30]

[MARS83] Martins, J. P. (1983): Reasoning in Multiple Belief Spaces. Ph.D. Thesis, Dept. of Computer Science, State Univ. of New York at Buffalo, Amherst, NY. [KOBS*: 56, 59]

[MASS88] Masson, M. E. J., W. C. Hill, J. Conner and R. Guindon (1988): Misconceived Misconceptions? In: Proc. of the CHI'88 Conference on Human Factors in Computing Systems, Washington, D.C., 151–156.

[MAST89] Mastaglio, T. (1989): A User Modelling Approach for a Computer-Based Critic. Dissertation Research Proposal, Dept. of Computer Science, University of Colorado, Boulder, CO.

[MATT86] Matthews, M. and G. Biswas (1986): USCSH: An Active Assistance Interface for UNIX. Technical Report USC-CS 86-003, Dept. of Computer Science, Univ. of South Carolina, Columbia, SC. [CHIN*: 78]

[MATZ82] Matz, M. (1982): Towards a Process Model for High School Algebra. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 397]

[MAYS80a] Mays, E. (1980): Correcting Misconceptions about Database Structure. In: Proc. of the 3rd CSCSI/SCEIO Conference, Victoria, B.C., 123–128. [MCCO*: 237]

[MAYS80b] Mays, E. (1980): Failures in Natural Language Systems: Applications to Data Base Query Systems. In: Proc. AAAI-80, 327–330. [WAHL*: 26; CHIN*: 101; CARB*: 146]

[MCCA80] McCauley, C., C. L. Stitt and M. Segal (1980): Stereotyping: From Prejudice to Prediction. Psychological Bulletin 87, 195–208. [RICH*: 35]

[MCCA87] McCalla, G. I. and J. E. Greer (1987): The Practical Use of Artificial Intelligence in Automated Tutoring: Current Status and Impediments to Progress. Research Report 87-2, Dept. of Computational Science, Univ. of Saskatchewan, Saskatoon, Canada.

[MCCL78] McCawley, J. D. (1978): Conversational Implicature and the Lexicon. In: P. Cole, ed.: Syntax and Semantics 9: Pragmatics. New York: Academic Press. [JAME*: 261]

[MCCO83] McCoy, K. F. (1983): Correcting Misconceptions: What to Say When the User is Mistaken. In: Proc. of the CHI'83 Conference on Human Factors in Computing Systems, Boston, MA, 197–201. [PARI*: 201]

[MCCO85a] McCoy, K. F. (1985): Correcting Object-Related Misconceptions. Ph.D. Thesis, MS-CIS-85-57, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA. [WAHL*: 27; CHIN*: 101; MCCO*: 245; FINI*: 416]

[MCCO85b] McCoy, K. F. (1985): The Role of Perspective in Responding to Property Misconceptions. In: Proc. IJCAI-85, 791–793. [WAHL*: 27]

[MCCO86] McCoy, K. F. (1986): The ROMPER System: Responding to Object-Related Misconceptions Using Perspective. In: Proc. of the 24th Annual Meeting of the Association for Computational Linguistics, New York, NY, 97–105. [PARI*: 201f.; COHE*: 325, 330]

[MCCO87] McCoy, K. F. (1987): Contextual Effects on Responses to Misconceptions. In: G. Kempen, ed.: Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics. Dordrecht/Boston: Kluwer. [MCCO*: 242]

[MCCO88] McCoy, K. F. (1988): Reasoning on a Dynamically Highlighted User Model to Respond to Misconceptions. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), 52–63. [PARI*: 201f.; MCCO*: 233; SPAR*: 362]

[MCCO*] McCoy, K. F.: Highlighting a User Model to Respond to Misconceptions. In this volume. [WAHL*: 7, 27; CHIN*: 101; QUIL*: 126; CARB*: 152, 161; PARI*: 201; JAME*: 260; MORI*: 375; FINI*: 416]

[MCDO80] McDonald, D. D. (1980): Natural Language Production as a Process of Decision Making under Constraint. Ph.D. Thesis, Dept. of Computer Science and Electrical Engineering, MIT, Cambridge, MA. [MCCO*: 236]

[MCKE82] McKeown, K. R. (1982): Generating Natural Language Text in Response to Questions about Database Structure. Ph.D. Thesis, Univ. of Pennsylvania, Philadelphia, PA. [MCCO*: 243]

[MCKE85a] McKeown, K. R. (1985): Discourse Strategies for Generating Natural-Language Text. Artificial Intelligence 27, 1–41. [CARB*: 139, 155; FINI*: 415]

[MCKE85b] McKeown, K. R. (1985): Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text. Cambridge: Cambridge Univ. Press. [PARI*: 202f., 206, 211, 223; JAME*: 279; COHE*: 325]

[MCKE85c] McKeown, K. R., M. Wish and K. Matthews (1985): Tailoring Explanations for the User. In: Proc. IJCAI-85, 794–798. [WAHL*: 12; PARI*: 201, 203; MCCO*: 251; FINI*: 415]

[MCKE87] McKeown, K. R. and C. L. Paris (1987): Functional Unification Grammar Revisited. In: Proc. of the 25th Annual Meeting of the Association for Computational Linguistics, Stanford Univ., Stanford, CA, 97–103. [PARI*: 223]

[MCLE82] McLeod, J. (1982): Institute of Child Guidance and Development at the Univ. of Saskatchewan. Journal of Learning Disabilities 15, 290–293. [COHE*: 314]

[MCLE85] McLeod, J. and M. Jones (1985): The Development of an Expert System for Computer-Guided Diagnosis of Children's Learning Problems: Some Emerging Problems. In: Proc. of the International Conference on the Computer as an Aid for Those with Special Needs, Sheffield, England, 68–78. [COHE*: 314]

[MCLE86] McLeod, J., M. Jones, G. Robertson and B. Wasson (1986): Computer-Guided Diagnosis: A Progress Report on a Canadian Project. Fifth Canadian Symposium on Instructional Technology, Ottawa, Canada, 101–105. [COHE*: 314]

[MILL82] Miller, M. L. (1982): A Structured Planning and Debugging Environment for Elementary Programming. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 388]

[MINS75] Minsky, M. (1975): A Framework for Representing Knowledge. In: P. Winston, ed.: The Psychology of Computer Vision. New York: McGraw-Hill. [RICH*: 37]

[MINT85] Minton, S. N., P. J. Hayes and J. E. Fain (1985): Controlling Search in Flexible Parsing. In: Proc. IJCAI-85, 785–787. [LEHM*: 165]

[MITC86] Mitchell, T. M., R. M. Keller and S. T. Kedar-Cabelli (1986): Explanation-Based Learning: A Unifying View. Machine Learning 1, 47–80. [LEHM*: 193]

[MOOR80] Moore, R. C. (1980): Reasoning about Knowledge and Action. Technical Note 191, AI Center, SRI International, Menlo Park, CA. [WAHL*: 16, 18; KOBS*: 62]

[MOOR82] Moore, R. C. and G. G. Hendrix (1982): Computational Models of Belief and the Semantics of Belief Sentences. In: S. Peters and E. Saarinen, eds.: Processes, Beliefs, and Questions. Dordrecht etc.: Reidel. [KOBS*: 55]

[MORI82] Morik, K. (1982): Überzeugungssysteme der Künstlichen Intelligenz: Validierung vor dem Hintergrund linguistischer Theorien über implizite Äußerungen. Tübingen: Niemeyer. [MORI*: 372, 383]

[MORI84] Morik, K. (1984): Partnermodellierung und Interessenprofile bei Dialogsystemen der Künstlichen Intelligenz. In: C.-R. Rollinger, ed.: Probleme des (Text-) Verstehens: Ansätze der Künstlichen Intelligenz. Tübingen: Niemeyer. [WAHL*: 19; MORI*: 365]

[MORI85a] Morik, K. (1985): User Modeling, Dialog Structure, and Dialog Strategy in HAM-ANS. In: Proc. of the 2nd Conference of the European Chapter of the Association for Computational Linguistics, Geneva, Switzerland, 268–273. [WAHL*: 21; KOBS*: 53, 59; MORI*: 373, 375f., 379]

[MORI85b] Morik, K. and C.-R. Rollinger (1985): The Real Estate Agent: Modeling Users by Uncertain Reasoning. AI Magazine 6, 44–52. [WAHL*: 13, 21; KOBS*: 53; CHIN*: 78, 83; CARB*: 161; FINI*: 414ff.]

[MORI88] Morik, K. (1988): Discourse Models, Dialog Memories, and User Models. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 95–97. [RICH*: 39; SPAR*: 362; MORI*: 366]

[MORI*] Morik, K.: User Models and Conversational Settings: Modeling the User's Wants. In this volume. [WAHL*: 19; RICH*: 36; KOBS*: 53, 59; CHIN*: 78; MCCO*: 251; JAME*: 256, 278, 294; SPAR*: 343, 360, 362; FINI*: 415]

[MOTR86] Motro, A. (1986): Query Generalization: A Method for Interpreting Null Answers. In: L. Kerschberg, ed.: Expert Database Systems. Menlo Park, CA: Benjamin/Cummings. [FINI*: 418]

[MURR87a] Murray, D. M. (1987): A Survey of User Cognitive Modelling. Report DITC 92/87, National Physics Laboratory, Teddington, England. [WAHL*: 9]

[MURR87b] Murray, D. M. (1987): Embedded User Models. In: H.-J. Bullinger and B. Shackel, eds.: Human-Computer Interaction: INTERACT'87. Amsterdam: North Holland. [WAHL*: 9]

[NADA83] Nadathur, G. and A. K. Joshi (1983): Mutual Beliefs in Conversational Systems: Their Role in Referring Expressions. In: Proc. IJCAI-83, 603–605. [KOBS*: 62]

[NEAL87] Neal, L. R. (1987): User Modeling for Syntax-Directed Editors. In: H.-J. Bullinger and B. Shackel, eds.: Human-Computer Interaction: INTERACT'87. Amsterdam: North Holland.

[NESS87] Nessen, E. (1987): SC-UM: User Modeling in the SINIX Consultant. Memo No. 18, Dept. of Computer Science, Univ. of Saarbrücken, Fed. Rep. of Germany. [WAHL*: 5; CHIN*: 92; QUIL*: 128]

[NEUM83] Neumann, B. and H.-J. Novak (1983): Event Models for Recognition and Natural Language Description of Events in Real-World Image Sequences. In: Proc. IJCAI-83, 724–726. [MORI*: 370]

[NEW67] The New Book of Knowledge: The Children's Encyclopedia. New York: Grolier Inc., 1967. [PARI*: 201, 205]

[NILS83] Nilsson, N. (1983): Artificial Intelligence Prepares for 2001. The AI Magazine 4(4), 7–14. [WAHL*: 31]

[NIKU85] Nisbett, R. E. and Z. Kunda (1985): Perception of Social Distributions. Journal of Personality and Social Psychology 48, 297–311. [JAME*: 294, 308]

[NIRO80] Nisbett, R. E. and L. Ross (1980): Human Inference: Strategies and Shortcomings of Social Judgment. Englewood Cliffs, NJ: Prentice-Hall. [JAME*: 285]

[NORM86a] Norman, D. A. (1986): Cognitive Engineering. In: D. A. Norman and S. W. Draper, eds.: User Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, NJ: Erlbaum. [JAME*: 255]

[NORM86b] Norman, D. A. and S. W. Draper, eds. (1986): User Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, NJ: Erlbaum. [WAHL*: 9]

[NOTH84] Nothdurft, W. (1984): äh folgendes Problem äh: Die interaktive Ausarbeitung 'des Problems' in Beratungsgeprächen. In: R. Wimmer and G. Zifonun,

eds.: Forschungsberichte des Instituts für Deutsche Sprache. Tübingen: Narr. [MORI*: 372]

[NOVA84] Novak, H.-J. and B. Neumann (1984): Szenenbeschreibung und Imagination in NAOS. In: C.-R. Rollinger, ed.: Probleme des (Text-)Verstehens. Tübingen: Niemeyer. [WAHL*: 24]

[OSHE82] O'Shea, T. (1982): A Self-Improving Quadratic Tutor. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 400]

[PARI84] Paris, C. L. (1984): Determining the Level of Expertise. In: Proc. of the 1st Annual Workshop on Theoretical Issues in Conceptual Information Processing, Atlanta, GA, 128–135. [PARI*: 212, 229]

[PARI85a] Paris, C. L. (1985): Description Strategies for Naive and Expert Users. In: Proc. of the 23rd Annual Meeting of the Association for Computational Linguistics, Chicago, IL, 238–246. [PARI*: 225; COHE*: 330]

[PARI85b] Paris, C. L. and T. Kwee (1985): Guide to the Unification Process and its Implementation: Progress Report on Extending the Grammar. Technical Report, Computer Science Dept., Columbia Univ., NY. [PARI*: 223]

[PARI87a] Paris, C. L. (1987): The Use of Explicit User Models in Text Generation: Tailoring to a User's Level of Expertise. Ph.D. Thesis, Dept. of Computer Science, Columbia Univ., NY. [PARI*: 211, 223f.]

[PARI87b] Paris, C. L. and K. R. McKeown (1987): Discourse Strategies for Describing Complex Physical Objects. In: G. Kempen, ed.: Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology and Linguistics. Dordrecht/Boston: Kluwer. [PARI*: 224]

[PARI88] Paris, C. L. (1988): Tailoring Object Descriptions to a User's Level of Expertise. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), 64–78. [FINI*: 415]

[PARI*] Paris, C. L.: The Use of Explicit User Models in a Generation System for Tailoring Answers to the User's Level of Expertise. In this volume. [CHIN*: 107; QUIL*: 130; CARB*: 161; JAME*: 256, 260]

[PART70] Partee, B. H. (1970): The Semantics of Belief-Sentences. In: J. Hintikka, J. Moravcsik and P. Suppes, eds.: Approaches to Natural Languages. Dordrecht etc.: Reidel. [KOBS*: 57]

[PERR78] Perrault, C. R., J. Allen and P. Cohen (1978): Speech Acts as a Basis for Understanding Dialogue Coherence. In: Proc. of Theoretical Issues in Natural Language Processing 2, Urbana-Champaign, IL, 125–132. [COHE*: 329]

[PERR80] Perrault, C. R. and J. F. Allen (1980): A Plan-Based Analysis of Indirect Speech Acts. American Journal of Computational Linguistics 6, 167–182. [WAHL*: 19; CARB*: 135]

[PERR81] Perrault, C. R. and P. R. Cohen (1981): It's for your Own Good: A Note on Inaccurate Reference. In: A. K. Joshi, B. Webber and I. A. Sag, eds.: Elements of Discourse Understanding. Cambridge: Cambridge Univ. Press. [KOBS*: 62]

[PERR88] Perrault, C. R. (1988): An Application of Default Logic to Speech Act Theory. In: P. Cohen, J. Morgan and M. Pollack, eds.: Plans and Intentions in Communication and Discourse. Cambridge, MA: MIT Press. [FINI*: 416]

[POLL82a] Pollack, M. E., J. Hirschberg and B. Webber (1982): User Participation in the Reasoning Process of Expert Systems. MS CIS-82-9, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA. [WAHL*: 5]

[POLL82b] Pollack, M. E., J. Hirschberg and B. Webber (1982): User Participation in the Reasoning Process of Expert Systems. In: Proc. AAAI-82, 358–361. [CARB*: 152]

[POLL83] Pollack, M. E. (1983): Generating Expert Answers through Goal Inference. Technical Note 316, SRI International, Stanford, CA. [MORI*: 372]

[POLL85] Pollack, M. E. (1985): Information Sought and Information Provided. In: Proc. of the CHI'85 Conference on Human Factors in Computing Systems, San Francisco, CA, 155–160. [FINI*: 418]

[POLL86a] Pollack, M. E. (1986): A Model of Plan Inference that Distinguishes between the Beliefs of Actors and Observers. In: Proc. of the 24th Annual Meeting of the Association for Computational Linguistics, New York, NY, 207–214. [WAHL*: 12; QUIL*: 127; CARB*: 136, 143, 153; SPAR*: 362]

[POLL86b] Pollack, M. E. (1986): Inferring Domain Plans in Question-Answering. Ph.D. Thesis, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA. [WAHL*: 12; QUIL*: 112, 127]

[POLL87] Pollack, M. E. (1987): Some Requirements for a Model of the Plan-Inference Process in Conversation. In: R. Reilly, ed.: Communication Failure in Dialogue. Amsterdam etc.: North Holland. [CARB*: 151ff.]

[POLS88] Polson, M. and J. Richardson, eds. (1988): Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum.

[POOL86] Poole, D. L., R. Aleliunas and R. Goebel (1986): Theorist: A Logical Reasoning System for Default and Diagnosis. Technical Report CS-86-06, Dept. of Computer Science, Univ. of Waterloo, Canada. [COHE*: 322]

[POWE79] Power, R. (1979): The Organisation of Purposeful Dialogues. Linguistics 17, 107–152. [WAHL*: 7; KOBS*: 54]

[PRAT87] Pratt, J. M. (1987): The Social Impact of User Models. In: H.-J. Bullinger and B. Shackel, eds.: Human-Computer Interaction: INTERACT'87. Amsterdam: North Holland.

[PSOT88] Psotka, J., D. Massey and S. Mutter, eds. (1988): Intelligent Tutoring Systems: Lessons Learned. Hillsdale, NJ: Lawrence Erlbaum.

[QUIL85] Quilici, A. (1985): Human Problem Understanding and Advice Giving: A Computer Model. Technical Report 85-69, Computer Science Dept., Univ. of California, Los Angeles, CA. [QUIL*: 113]

[QUIL86] Quilici, A., M. Dyer and M. Flowers (1986): AQUA: An Intelligent UNIX Advisor. In: Proc. of the 7th European Conference on Artificial Intelligence, Brighton, England, 33–38.

[QUIL88] Quilici, A., M. Dyer and M. Flowers (1988): Recognizing and Responding to Plan-Oriented Misconceptions. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), 38–51. [QUIL*: 108; PARI*: 201]

[QUIL89] Quilici, A. (1989): The Correction Machine: Formulating Explanations for User Misconceptions. Proc. IJCAI-89 (forthcoming).

[QUIL*] Quilici, A.: Detecting and Responding to Plan-Oriented Misconceptions. In this volume. [WAHL*: 7, 27; RICH*: 39; CHIN*: 101; CARB*: 152, 161; PARI*: 201; MCCO*: 238; JAME*: 260, 307]

[QUIN56] Quine, W. V. O. (1956): Quantifiers and Propositional Attitudes. Journal of Philosophy 53, 177–187. Reprinted in: Quine, W. V. O. (1966): The Ways of Paradox. New York: Random House. [KOBS*: 55]

[QUIN60] Quine, W. V. O. (1960): Word and Object. Cambridge, MA: MIT Press.

[QUIN86] Quinn, L. and D. M. Russell (1986): Intelligent Interfaces: User Models and Planners. In: Proc. of the CHI'86 Conference on Human Factors in Computing Systems, Boston, MA, 314–320. [WAHL*: 33]

[RAMS87] Ramshaw, L. A. (1987): Pragmatic Knowledge for Resolving Ill-Formedness. Technical Report, Dept. of Computer and Information Sciences, Univ. of Delaware, Newark, DE. [CARB*: 161]

[RAPA84] Rapaport, W. J. and S. C. Shapiro (1984): Quasi-Indexial Reference in Propositional Semantic Networks. In: Proc. of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics, Stanford Univ., Stanford, CA, 65–70. [KOBS*: 55]

[REIC78] Reichman, R. (1978): Conversational Coherency. Cognitive Science 2, 283–327. [JAME*: 279]

[REIS85] Reiser, B. J., J. R. Anderson and R. G. Farrell (1985): Dynamic Student Modelling in an Intelligent Tutor for LISP Programming. In: Proc. IJCAI-85, 8–14. [KOBS*: 56; CHIN*: 77, 100; KASS*: 388f., 393]

[REIT78] Reiter, R. (1978): On Closed World Data Bases. In: H. Gallaire and J. Minker, eds.: Logic and Data Bases. New York and London: Plenum Press. [CHIN*: 97; CARB*: 151; FINI*: 424]

[REIT80] Reiter, R. (1980): A Logic for Default Reasoning. Artificial Intelligence 13, 81–132. [RICH*: 41; FINI*: 416]

[RICH79a] Rich, E. (1979): Building and Exploiting User Models. Ph.D. Thesis, Computer Science Dept., Carnegie-Mellon Univ., Pittsburgh, PA. [WAHL*: 5, 11, 13f.; KOBS*: 53, 67; COHE*: 329; MORI*: 373–377]

[RICH79b] Rich, E. (1979): User Modelling Via Stereotypes. Cognitive Science 3, 329–354. [WAHL*: 5, 11, 13f.; RICH*: 36; KOBS*: 53, 67; CHIN*: 78f., 83; PARI*: 201, 229; SPAR*: 342, 362; FINI*: 414ff.]

[RICH82] Rich, E. (1982): Programs as Data for their Help Systems. In: Proc. of the National Computer Conference, Houston, TX, 481–485. [WAHL*: 32]

[RICH83] Rich, E. (1983): Users are Individuals: Individualizing User Models. In: International Journal of Man-Machine Studies 18, 199–214. [WAHL*: 5, 20; RICH*: 38; CHIN*: 78f., 89; SPAR*: 360; FINI*: 414]

[RICH84] Rich, E. (1984): Natural Language Interfaces. Computer 17, 39–47. [LEHM*: 173]

[RICH*] Rich, E.: Stereotypes and User Modeling. In this volume. [WAHL*: 13; CHIN*: 78f.; PARI*: 201; MCCO*: 235; JAME*: 256, 294; COHE*: 329; MORI*: 376; FINI*: 416, 422]

[RIVE89] Rivers, R. (1989): Embedded User Models: Where Next? Interacting with Computers 1, 14–30.

[ROBI80] Robinson, A. E., D. E. Appelt, B. J. Grosz, G. G. Hendrix and J. J. Robinson (1980): Interpreting Natural Language Utterances in Dialogs about Tasks. Technical Report TR210, SRI International, Menlo Park, CA. [CARB*: 135]

[ROBI81] Robinson, A. E. (1981): Determing Verb Phrase Referents in Dialogs. American Journal of Computational Linguistics 7, 1–18. [CARB*: 135]

[ROLL80] Rollinger, C. R. (1980): Readtime-Inferenzen für semantische Netze. In: C. R. Rollinger and H.-J. Schneider, eds.: Inferenzen in natürlichsprachlichen Systemen der Künstlichen Intelligenz. Berlin: Einhorn.

[ROSC77] Rosch, E. (1977): Human Categorization. In: N. Warren, ed.: Studies in Cross Cultural Psychology I. London: Academic Press. [CHIN*: 79, 105]

[ROSC78] Rosch, E. (1978): Principles of Categorization. In: E. Rosch and B. B. Lloyd, eds.: Cognition and Categorization. Hillsdale, NJ: Lawrence Erlbaum. [CHIN*: 79, 105]

[ROSC83] Rosch, E. (1983): Prototype Classification and Logical Classification: The Two Systems. In: E. Scholnick, ed.: New Trends in Cognitive Representation: Challenges to Piaget's Theory. Hillsdale, NJ: Erlbaum. [CHIN*: 79, 105]

[ROSE82] Rosenschein, J. S. (1982): Synchronization of Multi-Agent Plans. In: Proc. AAAI-82, 115–119. [WAHL*: 7; KOBS*: 54]

[ROSE85] Rosenschein, J. S. (1985): Rational Interaction: Cooperation among Intelligent Agents. Ph.D. Thesis, Stanford Univ., CA. [WAHL*: 7; KOBS*: 54]

[ROSN81] Rosner, M. (1981): Some Thoughts about Actions in Conversation. In: Proc. of the Workshop on Models of Dialogue: Theory and Application, Linköping, Sweden, 105–116. [WAHL*: 26, 32]

[ROSS85] Ross, P., J. Jones and M. Millington (1985): User Modeling in Command-Driven Systems. DAI Research Paper No. 264, Dept. of Artificial Intelligence, Univ. of Edinburgh, Scotland.

[ROWE84] Rowe, N. C. (1984): Modeling Degrees of Item Interest for a General Database Query System. International Journal of Man-Machine Studies 20, 421–443. [JAME*: 264]

[RUME73] Rumelhart, D. E. and A. A. Abrahamson (1973): A Model for Analogical Reasoning. Cognitive Psychology 5, 1–28. [MCCO*: 243]

[RUME86] Rumelhart, D. E., J. L. McClelland and the PDP Research Group (1986): Parallel Distributed Processing. Cambridge, MA: MIT Press. [RICH*: 50]

[SACE74] Sacerdoti, E. (1974): Planning in a Hierarchy of Abstraction Spaces. Artificial Intelligence 5, 115–135. [QUIL*: 111]

[SACE77] Sacerdoti, E. (1977): A Structure for Plans and Behavior. Amsterdam: North-Holland. [KASS*: 394]

[SAMP82] Sampson, G. (1982): The Economics of Conversation: Comments on Joshi's Paper. In: N. V. Smith, ed.: Mutual Knowledge. London: Academic Press. [JAME*: 259]

[SCHG79] Schank, G. (1979): Zum Ablaufmuster von Kurzberatungen: Beschreibung einer Gesprächsstruktur. In: J. Dittmann, ed.: Arbeiten zur Konversationsanalyse. Tübingen: Niemeyer [MORI*: 372]

[SCHI72] Schiffer, S. R. (1972): Meaning. Oxford: Clarendon Press. [KOBS*: 62]

[SCHM78] Schmidt, C. F., N. S. Sridharan and J. L. Goodson (1978): The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence. Artificial Intelligence 11, 45–83. [CARB*: 152f.]

[SCHR77] Schank, R. C. and R. Abelson (1977): Scripts, Plans, Goals, and Understanding. Hillsdale, NJ: Lawrence Erlbaum. [QUIL*: 111]

[SCHR82] Schank, R. C. (1982): Dynamic Memory. Cambridge, MA: Cambridge Univ. Press. [QUIL*: 125]

[SCHR86] Schank, R. C. (1986): Explanation Patterns: Understanding Mechanically and Creatively. Hillsdale, NJ: Lawrence Erlbaum. [QUIL*: 111]

[SCHU83a] Schuster, E. (1983): Custom-Made Responses: Maintaining and Updating the User Model. MS-CIS-83-13, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA.

[SCHU83b] Schuster, E. and T. W. Finin (1983): Understanding Misunderstandings. MS-CIS-83-12, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia, PA. [WAHL*: 7]

[SCHU85] Schuster, E. (1985): Grammars as User Models. In: Proc. IJCAI-85, 20–22. [LEHM*: 164]

[SCHU86a] Schuster, E. (1986): The Role of Native Grammars in Correcting Errors in Second Language Learning. Computational Intelligence 2, 93–98.

[SCHU86b] Schuster, E. and T. Finin (1986): VP2: The Role of User Modeling in Correcting Errors in Second Language Learning. In: A. G. Cohn and J. R. Thomas, eds.: Artificial Intelligence and its Applications. Chichester etc.: Wiley. [FINI*: 411, 418]

[SCHU88] Schuster, E. (1988): Establishing the Relationship between Discourse Models and User Models. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 82–85. [RICH*: 39; SPAR*: 362]

[SCHW79] Schwitalla, J. (1979): Nonresponsive Antworten. Deutsche Sprache 7, 193–211. [JAME*: 279]

[SEAR69] Searle, J. R. (1969): Speech Acts: An Essay in the Philosophy of Language. Cambridge, England: Cambridge Univ. Press. [KOBS*: 61; CHIN*: 85]

[SELF74] Self, J. A. (1974): Student Models in Computer-Aided Instruction. International Journal of Man-Machine Studies 6, 261–276. [KASS*: 387]

[SELF88] Self, J. A. (1988): Bypassing the Intractable Problem of Student Modelling. In: ITS-88: Intelligent Tutoring Systems, Montreal, Canada, 18–24.

[SHAF76] Shafer, G. (1976): A Mathematical Theory of Evidence. Princeton, NJ: Princeton Univ. Press. [RICH*: 47]

[SHAN79] Shanon, B. (1979): Where Questions. In: Proc. of the 17th Annual Meeting of the Association Computational Linguistics, La Jolla, CA, 73–75.

[SHEP26] Sheperd, H. R. (1926): The Fine Art of Writing. New York: Macmillan. [PARI*: 206]

[SHNE87] Shneiderman, B. (1987): Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA: Addison-Wesley. [JAME*: 257, 308]

[SHOR76] Shortliffe, E. H. (1976): Computer-Based Medical Consultation: MYCIN. New York: Elsevier. [RICH*: 47; COHE*: 329]

[SHRA82] Shrager, J. and T. Finin (1982): An Expert System that Volunteers Advice. In: Proc. AAAI-82, 339–340. [KASS*: 388; FINI*: 411]

[SIDN81a] Sidner, C. L. (1981): Focusing for Interpretation of Pronouns. American Journal of Computational Linguistics 7, 217–231. [CARB*: 139, 155]

[SIDN81b] Sidner, C. L. and D. J. Israel (1981): Recognizing Intended Meaning and Speakers' Plans. In: Proc. IJCAI-81, 203–208. [FINI*: 415]

[SIDN83a] Sidner, C. L. (1983): Focusing in the Comprehension of Definite Anaphora. In: M. Brady and R. Berwick, eds.: Computational Models of Discourse. Cambridge, MA: MIT Press. [MCCO*: 244]

[SIDN83b] Sidner, C. L. (1983): What the Speaker Means: The Recognition of Speaker's Plans in Discourse. Computers and Mathematics with Applications 9, 71–82. [WAHL*: 12; CARB*: 135, 156]

[SIDN85] Sidner, C. L. (1985): Plan Parsing for Intended Response Recognition of Speaker's Plans in Discourse. Computational Intelligence 1, 1–10. [CARB*: 135]

[SIKL78] Siklossy, L. (1978): Impertinent Question-Answering Systems: Justification and Theory. In: Proc. of the 1978 ACM National Conference, Washington, D.C., 39–44. [JAME*: 279]

[SIMM84] Simmons, R. F. (1984): From Menus to Intentions in Man-Machine Dialogue. Technical Report TR-84-33, AI-TR-84-04, Dept. of Computer Sciences, Univ. of Texas at Austin, TX.

[SLAT86] Slator, B. M., M. P. Anderson and W. Conley (1986): Pygmalion at the Interface. Communications of the ACM 29, 499–514. [LEHM*: 184]

[SLEE77] Sleeman, D. H. (1977): A System which Allows Students to Explore Algorithms. In: Proc. IJCAI-77, 780–786. [KASS*: 388]

[SLEE81] Sleeman, D. H. and M. J. Smith (1981): Modeling Student's Problem Solving. Artificial Intelligence 16, 171–187. [CHIN*: 77, 100; KASS*: 392, 396]

[SLEE82a] Sleeman, D. H. (1982): Assessing Aspects of Competence in Basic Algebra. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 396; FINI*: 414]

[SLEE82b] Sleeman, D. H. (1982): Inferring (Mal) Rules From Pupil's Protocols. In: Proc. of the 6th European Conference on Artificial Intelligence, Orsay, France, 160–164. [PARI*: 211; MCCO*: 237]

[SLEE82c] Sleeman, D. H. and J. S. Brown (1982): Intelligent Tutoring Systems. New York: Academic Press. [WAHL*: 7; KOBS*: 54; CHIN*: 77; COHE*: 329; SPAR*: 360; KASS*: 387]

[SLEE82d] Sleeman, D. H. and R. J. Hendley (1982): ACE: A System which Analyzes Complex Explanations. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [KASS*: 388]

[SLEE83] Sleeman, D. H. (1983): A Rule Directed Modeling System. In: R. Michalski, J. Carbonell and T. M. Mitchell, eds.: Machine Learning. Los Altos, CA: Tioga Press. [KASS*: 393, 396]

[SLEE84] Sleeman, D. H. (1984): An Attempt to Understand Students' Understanding of Basic Algebra. Cognitive Science 8, 387–412. [KASS*: 396f.]

[SLEE85] Sleeman, D. H. (1985): UMFE: A User Modeling Front End Subsystem. International Journal of Man-Machine Studies 23, 71–88. [WAHL*: 15; RICH*: 36; KOBS*: 53, 56f.; PARI*: 229; SPAR*: 361; KASS*: 408; FINI*: 416]

[SMIB82] Smith, B. (1982): Reflection and Semantics in a Procedural Language. Ph.D. Thesis, Technical Report MIT/LCS/TR-272, MIT, Cambridge, MA. [KOBS*: 62; FINI*: 413]

[SMID83] Smith, D. E. and M. R. Genesereth (1983): Finding All of the Solutions to a Problem: Serious Applications of Meta-Level Reasoning II. Memo HPP-83-21, Stanford Univ. Heuristic Programming Project, Stanford, CA. [CHIN*: 98]

[SMIN82] Smith, N. V. (1982): Mutual Knowledge. London etc.: Academic Press.

[SNYD79] Snyder, M. (1979): Self-Monitoring Processes. In: L. Berkowitz, ed.: Advances in Experimental Social Psychology 12. New York: Academic Press. [JAME*: 298]

[SOUL84] Soulhi, S. (1984): Representing Knowledge about Knowledge and Mutual Knowledge. In: Proc. of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics, Stanford Univ., Stanford, CA, 194–199. [WAHL*: 18]

[SOWA76] Sowa, J. F. (1976): Conceptual Graphs for a Database Interface. IBM Journal of Research and Development 20, 336–357. [CARB*: 146]

[SPAR84] Sparck Jones, K. (1984): User Models and Expert Systems. Technical Report No. 61, Computer Laboratory, Univ. of Cambridge, England. [WAHL*: 5, 20; KOBS*: 61; SPAR*: 344, 350, 352; MORI*: 366f., 374]

[SPAR86] Sparck Jones, K. (1986): Issues in User Modeling for Expert Systems. In: A. Cohn and J. Thomas, eds.: Artificial Intelligence and its Applications. Chichester etc.: Wiley. [RICH*: 39; SPAR*: 350]

[SPAR88] Sparck Jones, K. (1988): User Models, Discourse Models, and Some Others. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 98–100. [RICH*: 39; SPAR*: 362]

[SPAR*] Sparck Jones, K.: Realism about User Modeling. In this volume. [WAHL*: 20; KOBS*: 61; JAME*: 294; MORI*: 374]

[SPOH85] Spohrer, J. C., E. Soloway and E. Pope (1985): A Goal/Plan Analysis of Buggy Pascal Programs. Technical Report YALEU/CSD/RR 392, Dept. of Computer Science, Yale Univ., New Haven, CT. [KASS*: 403]

[SPOH89] Spohrer, J. C. and E. Soloway (1989): Simulating Student Programmers. Proc. IJCAI-89 (forthcoming).

[SPRE87] Sprenger, M. (1987): Konzeption des Partnermodells in WISBER. Technical Report No. 19, Project WISBER, Computer Science Dept., Univ. of Hamburg, Fed. Rep. of Germany. [WAHL*: 13]

[SPRE88] Sprenger, M. (1988): Interpretation von Modalverben zur Konstruktion von Partnermodelleinträgen. Memo No. 18, Project WISBER, Computer Science Dept., Univ. of Hamburg, Fed. Rep. of Germany. [WAHL*: 13]

[SRID85] Sridharan, N. S. (1985): User Modelling and Plan Recognition. In: Proc. IJCAI-85, 1301–1302.

[STEV79] Stevens, A., A. Collins and S. E. Goldin (1982): Misconceptions in Student's Understanding. International Journal of Man-Machine Studies 11, 145–156. [CHIN*: 77, 100; MCCO*: 237]

[STEV80] Stevens, A. and A. Collins (1980): Multiple Conceptual Models of a Complex System. In: P.-A. Federico, R. E. Snow and W. E. Montague, eds.: Aptitude, Learning, and Instruction. Hillsdale, NJ: Erlbaum. [MCCO*: 237]

[STEV82] Stevens, A., A. Collins and S. E. Goldin (1982): Misconceptions in Student's Understanding. In: D. H. Sleeman and J. S. Brown, eds.: Intelligent Tutoring Systems. New York: Academic Press. [QUIL*: 116, 125]

[STIC83] Stich, S. P. (1983): From Folk Psychology to Cognitive Science: The Case against Belief. Cambridge, MA: MIT Press. [KOBS*: 54f.]

[STRE86] Streitz, N. A. (1986): Mental Models and Methaphors: Implications for the Design of Adaptive User-System Interfaces. In: H. Mandl and A. Lesgold, eds.: Learning Issues for Intelligent Tutoring Systems. New York: Springer.

[SUTC87] Sutcliffe, A. G. and A. C. Old (1987): Do Users Know They Have User Models? Some Experiences in the Practise of User Modeling. In: H.-J. Bullinger and B. Shackel, eds.: Human-Computer Interaction: INTERACT'87. Amsterdam: North Holland.

[SWAR83] Swartout, W. R. (1983): XPLAIN: A System for Creating and Explaining Expert Consulting Programs. Artificial Intelligence 21, 285–325. [COHE*: 325, 329; FINI*: 415]

[TANG89] Tang, H., N. Major and R. Rivers (1989): From Users to Dialogues: Enabling Authors to Build an Adaptive, Intelligent System. In: Proc. of the 5th Con-

ference of British Computer Society Humen-Computer Interaction Specialist Group, Nottingham, England (forthcoming).

[TAUB88] Tauber, M. J. (1988): On Mental Models and the User Interface. In: T. R. G. Green, J.-M. Hoc, D. M. Murray and G. C. Van der Veer, eds.: Working with Computers: Theory versus Outcome. New York: Academic Press.

[TAYL78] Taylor, S. E., S. T. Fiske, N. L. Etcoff and A. J. Ruddermann (1978): Categorical and Contextual Bases of Person Memory and Stereotyping. Journal of Personality and Social Psychology 36, 778-793. [RICH*: 36]

[TENN78] Tennant, H. (1978): The Evaluation of Natural Language Question Answerers. Technical Report, Univ. of Illinois at Urbana-Champaign, IL. [PARI*: 203]

[TETL85] Tetlock, P. E. and A. S. R. Manstead (1985): Impression Management versus Intrapsychic Explanations in Social Psychology: A Useful Dichotomy? Psychological Review 92, 59–77. [JAME*: 298]

[THMA83] Thomas, J. C. (1983): Psychological Issues in the Design of Database Query Languages. In: M. E. Sime and M. J. Coombs, eds.: Designing for Human-Computer Communication. London: Academic Press. [JAME*: 257, 276]

[THOM80] Thompson, B. H. (1980): Linguistic Analysis of Natural Language Communication with Computers. In: Proc. IJCAI-80, 190–201. [CARB*: 145]

[TOU82] Tou, F., M. Williams, R. Fikes, A. Henderson and T. Malone (1982): Rabbit: An Intelligent Database Assistant. In: Proc. AAAI-82, 314–317. [MCCO*: 245]

[TOUR86] Touretzky, D. (1986): The Mathematics of Inheritance Systems. Palo Alto, CA: Morgan Kaufmann. [RICH*: 43]

[TREN86] Trenouth, J. (1986): A Review and Classification of User Modelling Systems and Methods. KBET Research Report, Computer Science Dept., Univ. of Exeter, England.

[TREN88] Trenouth, J. (1988): A Pragmatic Architecture for Embedded User Modelling: "U.M. in K.B.E.T." Research Report No. 156, Computer Science Dept., Univ. of Exeter, England.

[TUBM86] Tubman, J. (1986): An Expert System for Educational Diagnosis Using THEORIST. M. Math. Thesis, Technical Report CS-86-32, Dept. of Computer Science, Univ. of Waterloo, Canada. [COHE*: 314, 332]

[TVER77] Tversky, A. (1977): Features of Similarity. Psychological Review 84, 327–352. [MCCO*: 243]

[VANA87] Van Arragon, P. (1987): User Modeling Bibliography. Research Report CS-87-22, Dept. of Computer Science, Univ. of Waterloo, Canada.

[VANB86a] van Beek, P. (1986): A Model for User Specific Explanation from Expert Systems. M. Math. Thesis, Technical Report CS-86-42, Dept. of Computer Science, Univ. of Waterloo, Canada. [COHE*: 326; SPAR*: 361f.]

[VANB86b] van Beek, P. and R. Cohen (1986): Towards User Specific Explanation Systems. In: Proc. of the 6th Canadian Conference on Artificial Intelligence, Montreal, Canada, 194–198. [COHE*: 326, 332; SPAR*: 361f.]

[VANB87] van Beek, P. (1987): A Model for Generating Better Explanations. In: Proc. of the 25th Annual Meeting of the Association for Computational Linguistics, Stanford, CA, 215–220.

[VANL88] VanLehn, K. (1988): Student Modeling. In: M. C. Polson and J. J. Richardson, eds.: Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum.

[VAND81] van Dijk, T. A. (1981): Studies in the Pragmatics of Discourse. The Hague: Mouton. [JAME*: 285]

[VARD85] Vardi, M. Y. (1985): A Model-Theoretic Analysis of Monotonic Knowledge. In: Proc. IJCAI-85, 509–512. [WAHL*: 17]

[VERW62] Verwiebe, F. L., G. R. van Hooft and R. R. Suchy (1962): Physics: A Basic Science. Princeton, NJ: van Nostrand. [PARI*: 205]

[WAHL83] Wahlster, W., H. Marburger, A. Jameson and S. Busemann (1983): Over-Answering Yes-No-Questions: Extended Responses in a NL Interface to a Vision System. In: Proc. IJCAI-83, 643–646. [WAHL*: 19, 23; MORI*: 370, 375, 383]

[WAHL84] Wahlster, W. (1984): Cooperative Access Systems. Future Generation Computer Systems 1, 103–111. [WAHL*: 5]

[WAHL86a] Wahlster, W. (1986): Some Terminological Remarks on User Modeling. Paper presented at the International Workshop on User Modeling, Maria Laach, Fed. Rep. of Germany. [SPAR*: 360; MORI*: 366]

[WAHL86b] Wahlster, W. and A. Kobsa (1986): Dialog-Based User Models. In: Proc. of the IEEE-74 (Special Issue on Natural Language Processing), 948–960. [WAHL*: 4]

[WAHL88] Wahlster, W. (1988): Distinguishing User Models from Discourse Models. In: Computational Linguistics 14(3), Special Issue on User Modeling (A. Kobsa and W. Wahlster, eds.), Discussion Section, 101–103. [WAHL*: 6; RICH*: 39; SPAR*: 362]

[WAHL89] Wahlster, W. (1989): User and Discourse Models for Multimodal Communication. In: J. W. Sullivan and S. W. Tyler, eds.: Architectures for Intelligent Interfaces: Elements and Prototypes. Reading, MA: Addison-Wesley (forthcoming).

[WAHL*] Wahlster, W. and A. Kobsa: User Models in Dialog Systems. In this volume. [KOBS*: 52, 61, 67; CHIN*: 78; COHE*: 330; SPAR*: 360, 362; KASS*: 407]

[WALK87] Walker, A. (1987): Knowledge Systems and Prolog. Reading, MA: Addison Wesley. [FINI*: 428]

[WALL82] Wallis, J. W. and E. H. Shortliffe (1982): Explanatory Power for Medical Expert Systems: Studies in the Representation of Causal Relationships for Clinical Consultation. Technical Report STAN-CS-82-923, Heuristic Programming Project, Dept. of Medicine and Computer Science, Stanford Univ., Stanford, CA. [PARI*: 201f., 211; COHE*: 329; FINI*: 415]

[WALL84] Wallis, J. W. and E. H. Shortliffe (1984): Customized Explanations Using Causal Knowledge. In: B. G. Buchanan and E. H. Shortliffe, eds.: Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Reading, MA: Addison-Wesley. [KOBS*: 56]

[WALS66] Walster, E., E. Aronson and D. Abrams (1966): On Increasing the Persuasiveness of a Low Prestige Communicator. Journal of Experimental Social Psychology 2, 325–342. [JAME*: 292]

[WARD74] Ward, C. D. and E. McGinnies (1974): Persuasive Effects of Early and Late Mention of Credible and Noncredible Sources. Journal of Psychology 86, 17–23. [JAME*: 287]

[WASS83] Wasserman, K. and M. Lebowitz (1983): Representing Complex Physical Objects. Cognition and Brain Theory 6, 333–352. [PARI*: 222]

[WASS85] Wasserman, K. (1985): Unifying Representation and Generalization: Understanding Hierarchically Structured Objects. Ph.D. Thesis, Dept. of Computer Science, Columbia Univ., NY. [PARI*: 222]

[WEBB83] Webber, B. L. and E. Mays (1983): Varieties of User Misconceptions: Detection and Correction. In: Proc. IJCAI-83, 650–652. [WAHL*: 26; CHIN*: 101]

[WEBB84] Webber, B. L. and T. Finin (1984): In Response: Next Steps in Natural Language Interaction. In: W. Reitman, ed.: Artificial Intelligence Applications for Business. Norwood, NJ: Ablex Publ. Co. [FINI*: 411, 415, 418]

[WEBB86] Webber, B. L. (1986): Questions, Answers and Responses: Interacting with Knowledge Base Systems. In: M. Brodie and J. Mylopoulos, eds.: On Knowledge Base Management Systems. Berlin etc.: Springer. [CARB*: 158]

[WEIN84] Weiner, E. J. (1984): A Knowledge Representation Approach to Understanding Metaphors. Computational Linguistics 19, 1–14. [MCCO*: 243f., 248]

[WEIS73] Weissler, A. and P. Weissler (1973): A Woman's Guide to Fixing the Car. New York: Walker and Company. [PARI*: 201, 205]

[WENG87] Wenger, E. (1987): Artificial Intelligence and Tutoring Systems. Los Altos, CA: Morgan Kaufmann.

[WERN88] Werner, E. (1988): Toward a Theory of Communication and Cooperation for Multiagent Planning. In: Proc. of the 1988 Conference on Theoretical As-

pects of Reasoning about Knowledge, Monterey, CA (to appear). [WAHL*: 7; KOBS*: 54]

[WEYD79] Weyd, H. (1979): "Immerhin". In: H. Weyd, ed.: Die Partikeln der deutschen Sprache. Berlin: de Gruyter. [JAME*: 280]

[WILE83] Wilensky, R. (1983): Planning and Understanding. Reading, MA: Addison-Wesley. [QUIL*: 111]

[WILE84] Wilensky, R., Y. Arens and D. N. Chin (1984): Talking to UNIX in English: An Overview of UC. In: Communications of the ACM 27, 574–593. [WAHL*: 5, 19, 32; CHIN*: 74; QUIL*: 128]

[WILE86] Wilensky, R., J. Mayfield, A. Albert, D. N. Chin, C. Cox, M. Luria, J. Martin and D. Wu (1986): UC: A Progress Report. Technical Report UCB/CSD 87/303, Division of Computer Science, Univ. of California, Berkeley, CA. [WAHL*: 5, 19, 32; KOBS*: 60, 65; CHIN*: 74, 78, 82, 87f., 103; QUIL*: 128f.; SPAR*: 362; FINI*: 414f.]

[WILE87] Wilensky, R. (1987): Some Problems and Proposals for Knowledge Representation. Report No. UCB/CSD 87/351, Division of Computer Science, Univ. of California, Berkeley, CA. [CHIN*: 79, 105]

[WILK83] Wilks, Y. and J. Bien (1983): Beliefs, Points of View and Multiple Environments. Cognitive Science 7, 95–119. [WAHL*: 29, 32; SPAR*: 362]

[WILK87] Wilks, Y. and A. Ballim (1987): Multiple Agents and the Ascription of Belief. In: Proc. IJCAI-87, 119–124. [WAHL*: 32]

[WILL84] Williams, M. D. (1984): What Makes RABBIT Run. International Journal of Man-Machines Studies 21, 333–352. [RICH*: 36]

[WILS86] Wilson, P. T. and R. C. Anderson (1986): What They Don't Know Will Hurt Them: The Role of Prior Knowledge in Comprehension. In: J. Orasanu, ed.: Reading Comprehension: From Research to Practice. Hillsdale, NJ: Erlbaum. [PARI*: 201, 212]

[WINO77] Winograd, T. (1977): A Framework for Understanding Discourse. In: P. A. Carpenter and M. A. Just, eds.: Cognitive Processes in Comprehension. Hillsdale, NJ: Erlbaum. [JAME*: 261]

[WINO86] Winograd, T. and F. Flores (1986): Understanding Computers and Cognition: A New Foundation for Design. Norwood, NJ: Ablex. [JAME*: 256]

[WOOD73] Woods, W. (1973): An Experimental Parsing System for Transition Network Grammars. In: R. Rustin, ed.: Natural Language Processing. New York : Algorithmics Press. [PARI*: 223]

[WOOL83] Woolf, B. and D. McDonald (1983): Human-Computer Discourse in the Design of a Pascal Tutor. In: Proc. of the CHI'83 Conference on Human Factors in Computing Systems, Boston, MA, 230–234. [MCCO*: 237]

[WOOL84a] Woolf, B. (1984): Context Dependent Planning in a Machine Tutor. Ph.D. Thesis, Univ. of Massachusetts, Amherst, MA. [MCCO*: 237]

[WOOL84b] Woolf, B. and D. D. McDonald (1984): Building a Computer Tutor: Design Issues. Computer 17, 61–73. [KASS*: 388]

[WYER74] Wyer, R. S., Jr. (1974): Cognitive Organization and Change: An Information-Processing Approach. Hillsdale, NJ: Erlbaum. [JAME*: 269]

[YOUN88] Young, R. M. and A. MacLean (1988): Choosing between Methods: Analysing the User's Decision Space in Terms of Schemas and Linear Models. In: Proc. of the CHI'88 Conference on Human Factors in Computing Systems, Washington, D.C., 139–143.

[ZADE65] Zadeh, L. A. (1965): Fuzzy Sets. Information and Control 8, 338–353. [CHIN*: 83]

[ZADE82] Zadeh, L. A. (1982): A Computational Approach to Fuzzy Quantifiers in Natural Language. Memorandum UCB/ERL M82/36 (Revised), Division of Computer Science, Univ. of California, Berkeley, CA. [CHIN*: 83]