Prose Generation from Expert Systems

An Applied Computational Linguistics Approach

Perry L. Miller & Glenn D. Rennels

The PROSENET/TEXTNET approach is designed to facilitate the generation of polished prose by an expert system. The approach uses the augmented transition network (ATN) formalism to help structure prose generation at the phrase, sentence, and paragraph levels. The approach also uses expressive frames to help give the expert system builder considerable freedom to organize material flexibly at the paragraph level. The PROSENET /TEXTNET approach has been used in a number of prototype expert systems in medical domains, and has proved to be a convenient and powerful tool.

As expert systems are more frequently used in diverse areas of life, it becomes increasingly important that the interface between these systems and their users be as effective as possible. One component of this interface for many systems involves the generation of English prose to communicate the expert system's conclusions and recommendations.

A number of different approaches have been taken to the generation of natural language text by computer. As discussed in the next section, these approaches include the straightforward output of "canned" text, the automated translation of if-then inference rules, as well as systems that are more sophisticated from both a linguistic and semantic standpoint.

This article describes the PROSENET/TEXTNET approach, which uses a linguistic formalism, the augmented transition network (ATN), to structure the generation of prose whose expression can vary widely depending on its content. The prose is pieced together from prose fragments that can vary in length from single words to whole sentences or even paragraphs, depending of the complexity and variability of the material being discussed. The approach is designed to allow the expert system builder great flexibility in polishing and refining the exact prose that will be output in different circumstances.

The rationale for the approach can best be understood by considering the process of writing prose manually. Creating a polished paper usually involves multiple drafts, as the organization and prose expression is refined in an iterative, incremental fashion. Even when a satisfactory draft evolves, an author frequently makes minor changes on each reading of the paper, particularly if the author sets the manuscript aside between readings.

Compared to the manual process of writing a paper, the process of creating a computer program to generate prose can be even more frustrating and complex, particularly if the program must react to multiple conditions and still produce polished prose. Indeed, in a preliminary evaluation of the advice of the Roundsman system described later in this article, most of the corrections made by the domain expert involved rather subtle modifications of the system's prose expression (Rennels 1987).

The PROSENET/TEXTNET approach, designed to facilitate the incremental refinement of an expert system's prose output, involves three components: (1) sentence-level and phrase-level ATNs to structure sentence-level and phrase-level prose expression, (2) expressive frames to structure paragraph-level content, and (3) paragraph-level ATNs to structure paragraph-level prose expression.

This approach has already been used to implement several prototype expert systems in a variety of medical domains, including several systems developed at Yale to critique physician plans (Miller 1986), including ATTENDING, which critiques anesthetic management (Miller 1983a, 1983b, 1984); HT-ATTENDING, which critiques the management of essential hypertension (Miller and Black 1984);

VQ-ATTENDING, which critiques aspects of ventilator management (Miller 1985); and ICON, which critiques radiologic differential diagnosis (Miller et al. 1986). Another prototype, the Roundsman system (Rennels 1987; Rennels et al. 1987), was developed at Stanford to use structured representations of the clinical literature to discuss the management of breast cancer.

PROSENET was initially developed as part of the ATTENDING system. TEXTNET is an adapted version of PROSENET, implemented using object-oriented programming techniques for the Roundsman project.

This article first discusses other approaches that have been taken to generation of natural language text by computer. It then describes the PROSENET/TEXTNET approach, using examples of prose output from the ATTENDING, HT-ATTENDING, VQ-ATTENDING, and Roundsman systems.

Background

Several different approaches have been taken to computer-based generation of natural language. At one extreme are systems that generate *canned text,* that is, predefined sentences and phrases are output directly from a computer program in a straightforward fashion. Here the text-generating logic is embedded within the program itself, with no attempt to incorporate any linguistic or semantic formalism. This approach may well be satisfactory when the text expression does not need to be adapted its to a range of different conditions.

At the other extreme are research projects that explore basic natural language issues (for example, McDonald 1980; Appelt 1982; McKeown 1982; Swartout 1982). In this type of system, the process of language generation typically starts from some underlying semantic representation, and the construction of phrases and sentences is driven by semantic and linguistic knowledge. The goal of this work is to explore the many complex linguistic and semantic issues involved in the generation of natural language. This work is still very much in the basic research stage, and is therefore not yet a practical approach for research projects that focus on expert system design rather than on fundamental natural language issues.

Previous expert system research has addressed prose generation in a number of projects that have explored rule-based explanation. For example, if a system like Mycin (Shortliffe 1976) is asked to explain why it advocates a particular recommendation, the system can use the if-then rules in its knowledge base to generate an explanation. That is, if a rule has been invoked that says "If you observe A, B, and C, then you can conclude D," and the user asks why D was inferred, the system can explain by saying "I inferred D because I observed A, B, and C." Alternatively, if the user asks why the system is asking whether C is true, the system can reply, "I am trying to decide whether I can conclude D; I know that A and B are true, so if C is true, then I will indeed be able to conclude D." Since if-then rules can chain together in making inferences, these chains of rules can be used to generate increasingly detailed explanations if the user persists in asking "why ... why ... why."

The following example shows Mycin, which gives advice concerning the diagnosis and treatment of infectious disease, using an if-then rule to explain why it is asking whether an organism (ORGANISM-1) is a hospital-acquired infection. Mycin replies with the following explanation, using the if-then rule which triggered its question as a template (Scott et al. 1984, p. 347).

[1.0] This will aid in determining the category of ORGANISM-1.
It has already been established that
    [1.1] the gram stain of ORGANISM-1 is gramneg, and
    [1.2] the morphology of ORGANISM-1 is rod, and
    [1.3] the aerobicity of ORGANISM-1 is facultative
Therefore, if
    [1.4] the infection with ORGANISM-1 was not acquired while the patient was hospitalized
then
    there is strongly suggestive evidence (.8) that the category of ORGANISM-1 is enterobacteriaceae
ALSO: there is weakly suggestive evidence (.1) that the identity of ORGANISM-1 is pseudomonas-aeruginosa
[RULE037]

This type of rule-based explanation has the advantage that the expert system is able to give a very literal explanation of its reasoning because it is translating its internal logic into English prose. Explanation generated in this fashion may not always fully address the user's question, however, since the system gives only a literal translation of the rule, and is not able to step back and discuss in broader terms why the rule was included in the knowledge base, or what its full implications are. Also, from the standpoint of generating polished prose, this rule-based approach has significant limitations. Any prose that is generated by direct, literal translation of an if-then rule is very rigidly structured, and, as a result, may sound stilted and unnatural.

Ideally, one would like to closely tie the prose generated by an expert system to the system's internal analysis, but at the same time to allow the system designer flexibility to refine the prose itself so that it will flow smoothly at both the sentence and paragraph levels. The PROSENET /TEXTNET approach was developed to help provide this flexibility.

Examples

This section provides two examples of prose output from medical expert systems using the PROSENET/TEXTNET approach.

ATTENDING

ATTENDING (Miller 1983a, 1983b, 1984) is an expert system designed to critique anesthetic management. In managing a complicated patient, an anesthetist must frequently assess a variety of risks that are implied by the patient's medical problems. Indeed, there are frequently risk tradeoffs, where techniques good for one medical problem may be bad for another, and vice versa. ATTENDING is

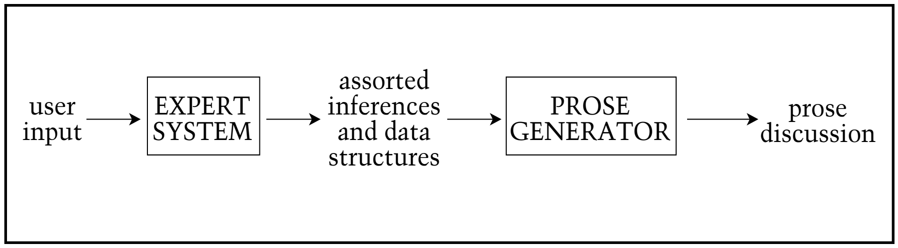


Figure 1. An Overview of an Expert System Interacting with the Prose Generator.

designed to help evaluate the implications of these risks and risk tradeoffs.

The example shown below is an excerpt from a critique discussing a physician's plan for a patient's anesthetic management. The critique produced varies significantly depending on the particular plan and the particular patient described. To make it easier for the reader to appreciate the prose generation process that underlies the critique, we have used square brackets to enclose the individual prose fragments from which the critique is composed.

> [In regard to] [induction] [,] [induction using thiopental] [could have] [two] [possible risks] [.] [First,] [in a patient with] [asthma] [,] [there is] [the] [conceivable] [risk of] [bronchospasm] [.] [Second,] [there is] [the] [possible] [risk of] [hypotension in the presence of hypovolemia] [.] [An alternative approach] [to] [induction using thiopental] [would be] [induction with ketamine] [.] [This has] [the advantage of] [helping avoid hypotention since ketamine is supportive of blood pressure] [,] [and] [of] [suppressing bronchospasm since ketamine is a bronchodilator] [.]
>
> [From the standpoint of] [maintenance] [,] [inhalational technique] [has] [the advantage of] [preventing bronchospasm by keeping the anesthetic level deep] [.] [On the other hand,] [it] [could have] [the] [risk of] [hypotension in the presence of hypovolemia] [.] [The choice of] [enflurane] [involves] [the] [conceivable] [risk of] [bronchospasm resulting from airway irritability] [.] [An alternative] [would be] [halothane] [.] [This] [has] [the advantage of] [helping prevent bronchospasm since halothane is a bronchodilator] [.] [An alternative approach] [to] [inhalational technique] [would be] [balanced anesthesia] [.] [This] [has] [the] [risk of] [possible] [bronchospasm due to light anesthesia] [.] [It does] [, however,] [have] [the advantage of] [maintaining blood pressure] [.]

Roundsman

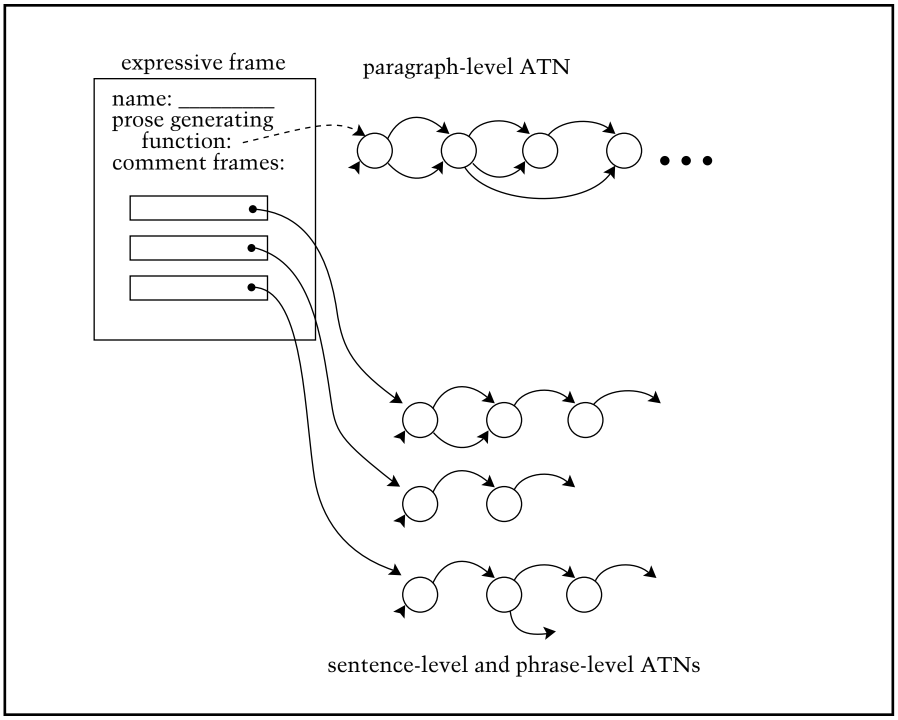
Roundsman (Rennels 1987; Rennels et al. 1987) is an expert system developed as a Ph.D. project to demonstrate the process of reasoning from the clinical literature. Roundsman has a library of 25 articles about the management of breast cancer. In the context of a particular patient and management plan, Roundsman selects the most relevant articles and then discusses the implications of each article in detail, including how well it matches the clinical case described.

The following two paragraphs show Roundsman discussing how well a particular article ("fisher85a") matches a proposed plan. Here again, the prose fragments from which the discussion is composed are enclosed in square brackets.

> [fisher85a] [was a randomized, controlled trial] [conducted at] [multiple NSABP centers][.] [Subjects were assigned to] [verified wide excision with axillary dissection and adjuvant radiotherapy] [(] [N=] [373] [)] [or the alternative therapy of] [verified wide excision with axillary dissection] [(] [N=] [358] [)] [.] [For patients who underwent] [the first protocol] [the] [proportion free of ipsilateral breast recurrence at five years] [was equal to] [0.9] [,] [overall survival at five years] [turned out to be] [0.91] [and] [recurrence-free survival at five years] [was] [0.81] [.] [Under] [the second protocol] [the] [proportion free of ipsilateral breast recurrence at five years] [turned out to be] [0.77] [,] [overall survival at five years] [was] [0.9] [and] [recurrence-free survival at five years] [was equal to] [0.68] [.]
>
> [Are these results relevant to your patient] [?] [It is encouraging that] [first] [,] [the adjuvant modality you propose was specified for this study as well] [(] [chemotherapy given if axillary nodes are path. positive] [)] [.] [Second] [,] [this study population is quite similar to your patient] [(] [the women in this group had T sizes ranging up to T2a but excision margins were verified free of tumor] [)] [.] [We suspect it makes little difference that] [the intervention was somewhat nonstandard] [(] [they did not radiate supraclavicular nodes] [)] [.] [More troublesome is that] [the study population was probably in a better prognostic stratum than your patient] [(] [this study stratum was defined by negative axillary node histology; about 40% of clinical stage I patients like yours will have positive histology] [)] [.]

Overview: The PROSENET /TEXTNET Approach

Figure 1 outlines in broad overview how the PROSENET/TEXTNET approach is used to generate prose from an expert system. First, the expert system itself performs its analysis, determining the content of the material to be output to the user. Once this content has been determined, the prose generator is activated. The prose generator contains all the prose-generating logic needed to express the content in English prose. One major advantage of the approach is that the logic that generates the prose is completely separated from (although closely tied to) the internal logic of the expert system itself. This separation makes it easier for the system builder to refine incrementally the various prose fragments and the prose-generating logic without any modification of the expert system



Fgure 2. A Schematic Overview of the Three Components of a Prose Generator Constructed Using the PROSENET/TEXTNET Approach.

itself.

The internal structure of a prose generator built using the PROSENET/TEXTNET approach is shown in figure 2. The prose generator consists of three parts.

1. Phrase-level and sentence-level ATNs. These ATNs adapt the expression of the prose output to the nuances of the material being discussed.

2. Paragraph-level expressive frames. These are used by the system to gain a global picture of all the comments it is to make about a given topic.

3. Paragraph-level ATNs. These ATNs are employed by the paragraph-level expressive frames. They may, for example, be used to insert connective words and phrases to help the prose flow smoothly.

Depending on the domain and on the character of the discussion to be generated, different PROSENET/TEXTNET components may be most useful in a particular expert system. For example, ATTENDING makes extensive use of phrase-level and sentence-level ATNs, allowing the system a very fine grain in adapting the phrasing of sentences to the material described. Roundsman also uses primarily the flexibility afforded by phrase-level and sentence-level ATNs. In contrast, HT-ATTENDING makes only moderate use of the expressive variability afforded by phrase-level and sentence-level ATNs. It uses expressive frames extensively, but uses paragraph-level ATNs quite sparsely. VQ-ATTENDING makes quite extensive use of all three components.

The sections that follow discuss each of these three components in turn.

ATNs: Structuring Sentence-Level and Phrase-Level Prose

Figure 3 shows several example ATNs, which are simplified versions of ATNs used by ATTENDING to generate parts of the prose critique on anesthetic management shown earlier. Each ATN consists of states (circles) connected by arcs (arrows). Associated with each arc is (1) an action, which might be a prose fragment to be output, a prose generating function, or some other action as described below; and (2) a test that determines when the arc may be used.

Whenever an ATN is activated, a path is traced through the network, starting at the ATN's initial state, following arcs from state to state until a pop arc is traversed. A pop arc marks the end of the path in the ATN. Whenever a push arc is traversed, processing in the current ATN is temporarily suspended while a lower ATN subnetwork is activated.

As each arc in the ATN is traversed, the arc's prose-generating action is performed, resulting in output of a sequence of prose fragments. The exact sequence of arcs is controlled by the tests associated with each arc. These tests examine various inferences and conclusions made by the expert system. In this way, the expert system's analysis directly controls the exact prose produced.

For example, using the bottom ATN shown in figure 3, the following prose fragments might be produced:

at least the theoretical risk of _____

at least the theoretical risk that this might cause ______

the conceivable risk of _____

the risk of ______

the risk that this could cause _____

The blanks are filled in by a prose-generating function, DESC_RISK, with a description of the risk being discussed, for example, bronchospasm.

As can be seen, the ATN formalism gives the system designer a great deal of power to adapt the prose expression to the nuances of the material being discussed. Figure 4 shows the different actions that can be associated with ATN arcs, in addition to ATN primitives such as push, pop, and jump.

1. If a prose fragment is associated with an arc, that prose is output whenever the arc is traversed.

2. If a prose-generating function is associated with an arc, that function is called whenever the arc is traversed. For example, the function might inspect data structures constructed by the expert system and extract prose to be generated.

3. An *option arc* contains a list of arcs, one of which is randomly chosen whenever the option arc itself is traversed. This allows the expert system to vary its expression, just as human authors avoid using identical words and phrasing too frequently.

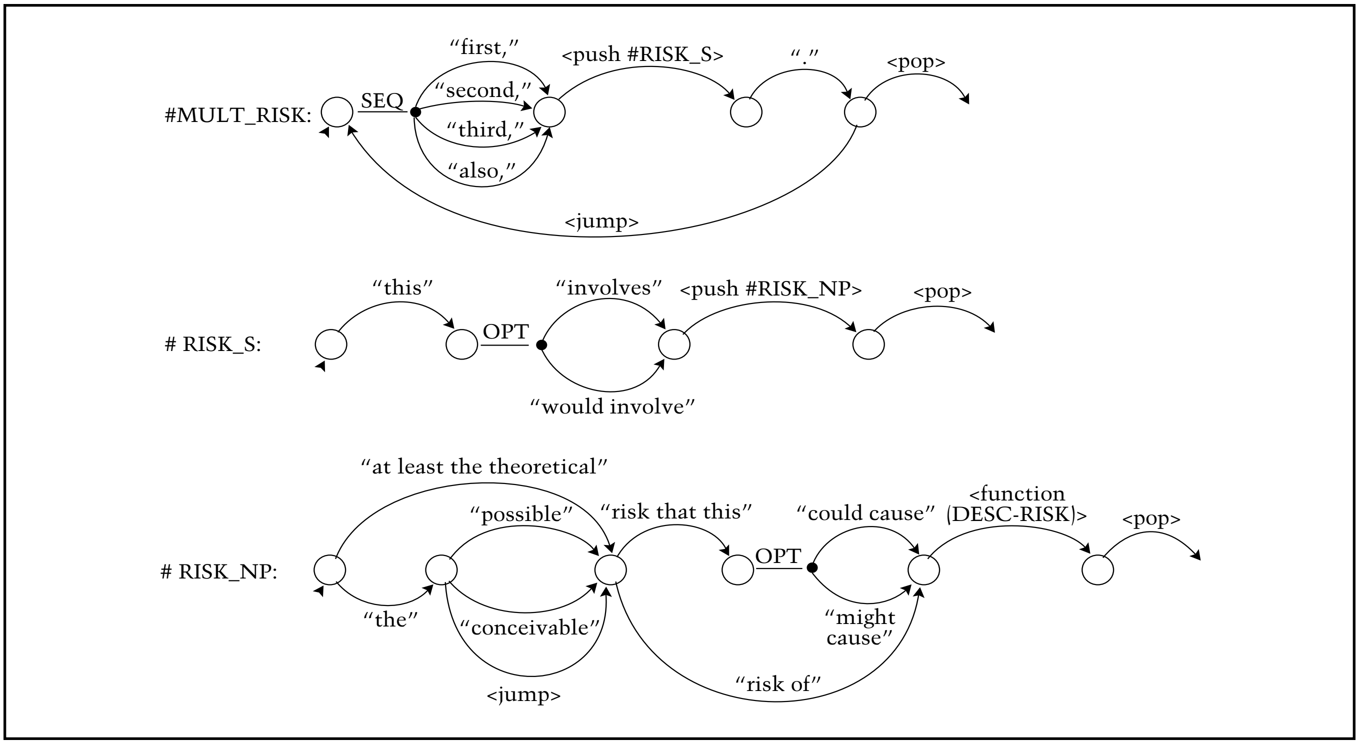


Figure 3. Example of Sentence-Level and Phrase-Level ATNs.

4. A *sequence arc* is similar to an option arc, except that its list of arcs is chosen sequentially (not randomly) when the sequence arc is traversed multiple times. An endsequence arc reinitializes the sequence arc.

Expressive Frames: Structuring Paragraph-Level Content

Expressive frames are used to structure paragraph-level content, and to coordinate the processing of the sentence-level and phrase-level ATNs discussed in the previous section. Typically, expressive frames are organized around a given topic, for example, a particular medical test or treatment. The expressive frame contains all the different comments that might be made about that topic in all possible circumstances.

There might be 20 different comments that could conceivably be made about a given topic in different circumstances. In a particular discussion, however, only a handful of the 20 comments might apply. Depending on the exact subset of comments that applied to a particular case, the system designer might like to structure the discussion of the topic differently.

In other words, it is frequently useful for a system to obtain a global picture of the material to be discussed about a given topic, so the system can decide how best to express that material: the exact sequence in which the various points should be made and the overall context in which they should be expressed. Expressive frames provide this capability.

Figure 5 shows the structure of an expressive frame. Each expressive frame has (1) a name, (2) a prose-generating function, and (3) a set of comments, represented as comment frames. Each comment frame, in turn, contains a variety of information.

1. Each comment frame contains a test that indicates when that comment is to be included in the prose output. This test looks at inferences and data constructed by the expert system.
2. Each comment frame contains a pointer to the comment's ATN.
3. Each comment frame may also contain other information that is used to organize the comments into a coherent paragraph. For example, this information might indicate the sequence in which comments are to be generated, or might describe various features of the content of each comment.

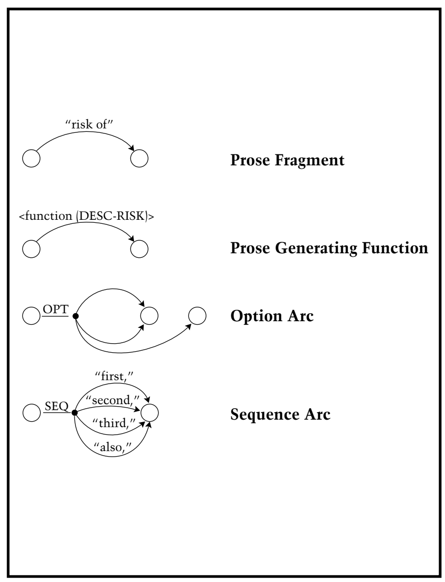


Figure 4. Examples of the Different Actions that Can Be Associated with ATN Arcs.

When an expressive frame is activated, the test associated with each comment frame is evaluated to determine which comments are to be included in the prose output. The list name: FRAME1

prose generating function: FRAME1-DISS

comment frames:

{test: test1; comment: atn1; ... <other information>}

{test: test2; comment: atn2; ... <other information>}

•
•
•

Figure 5. A Schematic Diagram of an Expressive Frame.

of chosen comment frames is then turned over to the expressive frame's prose-generating function, written by the expert system builder.

In a simple scenario, the expressive frame's prose-generating function might merely sort the chosen comments according to a sequence indicator, and then activate each comment's ATN in that sequence. Alternatively, the prose-generating function might perform a more complex analysis to group and sequence the chosen comments, and might employ a paragraph-level ATN to produce additional prose context for the comments. The use of such a paragraph-level ATN is discussed in the next section.

The fundamental advantage of using expressive frames is that they allow the system to obtain a global picture of all the comments it is about to make about a given topic. This global perspective makes it possible to adapt the order and context of the discussion to the material being discussed in a flexible, convenient way.

Paragraph-Level ATNs: Structuring Paragraph-Level Prose

Once an expressive frame has been activated, and has created a list of comments to be output, it may be possible to generate just those comments in sequence to obtain a readable, polished prose discussion. For instance, the example shown below is an excerpt from a critique produced by HT-ATTENDING (Miller and Black 1984) discussing the management of essential hypertension. In this excerpt, each sentence is a single prose fragment and is stored in a separate comment frame. These various comments are generated in a predetermined order with no need for a paragraph-level ATN to insert connective words to help the thread of thought flow smoothly.

> [A diuretic is always included in the antihypertensive regimen when the therapy progresses beyond step one.] [A thiazide diuretic or similar acting agent is the drug of choice for most patients.] [However, if this patient has significant renal insufficiency (glomerular filtration rate less than 50 ml/min), then a loop diuretic like furosemide, bumetanide, or ethacrynic acid would generally be considered most appropriate.] [Loop diuretics are not as effective antihypertensive agents in patients with normal renal function.]

In other circumstances, however, it may be very helpful to sequence the various comments differently depending on the content of the material. Furthermore, it may be useful to insert connective words or phrases to help the prose flow smoothly. In other situations, it may be useful to structure more profoundly the prose context in which the comments are presented.

Paragraph-level ATNs can be used to insert connective words and phrases around comments, or to allow more drastic structuring of the paragraph-level prose context. Figure 6 gives an example of a paragraph-level ATN used by the VQ-ATTENDING system (Miller 1985).

VQ-ATTENDING is designed to critique aspects of the ventilatory management of a patient receiving mechanical respiratory support. An interesting feature of VQ-ATTENDING is its goal-directed design. VQ-ATTENDING first infers those treatment goals that it considers to apply to a particular patient's management. It then discusses those goals explicitly in its prose critique. It also uses the goals internally to drive its critiquing analysis of the user physician's plan.

The paragraph-level ATN shown in figure 6 is used to structure a paragraph discussing the oxygenation goals that VQ-ATTENDING considers to apply to the patient described. The descriptions of the goals themselves are fairly short phrases, each contained in separate comment frames. Instead of just listing these goals, the paragraph-level ATN allows these goals to be discussed in a prose context that seems natural to the user physician.

The following two paragraphs were generated using the ATN shown in figure 6. Each paragraph discusses a different patient. The prose generated by the paragraph-level ATN is shown in bold type. The descriptions of the goals themselves (from comment frames) are in light type.

> **[In regard to] [oxygenation,] [there are several goals for this patient's management.] [One goal is]** [to maintain an adequate paO2] **[.] [A second goal is]** [to reduce the risks associated with high PEEP] **[.] [A third goal is]** [to reduce the level of oxygenation support] **[.]**
>
> **[In regard to] [oxygenation,] [the primary goal in this patient's management is]** [to acheive an adequate paO2] **[.] [Secondary goals are]** [to avoid the risk of oxygen toxicity] **[and]** [to avoid the risks associated with high PEEP] **[.] [The urgency of the primary goal, however, may override the secondary considerations.]**

In generating these paragraphs, VQ-

ATTENDING first inspects the various oxygenation goal comment frames that are to be included in the discussion. This inspection is performed by the prose-generating function associated with the oxygenation goal expressive frame. Then, using the paragraph-level ATN, VQ-ATTENDING adapts the overall prose context to the semantic content of the comment-frames. In this case, a different context is used depending on whether or not one oxygenation goal has overriding clinical priority.

Discussion

It is important to emphasize that nuances of prose discussion can be extremely important. This is certainly true in medicine, but is equally true in many other domains. In an evaluation of the advice of the Roundsman system, subtle corrections in the system's prose expression were viewed by the domain expert as crucial.

A physician's trust in an expert system's advice will depend in part on the proper use of domain terms. A physician would be unlikely to trust anyone, human or machine, whose clumsy use of medical vocabulary suggested a lack of grounding in the nuances of the domain. Also, since a medical expert system's advice is often designed to impact patient care, with the potential for harm (and for legal responsibility), subtle nuances may be important in preventing an incorrect interpretation and corresponding incorrect action.

When writing prose, every author is familiar with the frustrations of achieving a satisfactory organization of the material, and of incrementally refining the prose expression until it too is satisfactory. As expert systems become increasingly sophisticated, and as they develop increasingly complex insights to share with their users, it is inevitable that the problem of producing polished prose will need to be confronted as an integral problem of expert system design.

It is important to give the expert system designer a set of tools with which to handcraft the system's prose generator, just as the designer must handcraft any paper he or she writes. The PROSENET/TEXTNET approach is a first step in providing practical tools for this purpose.

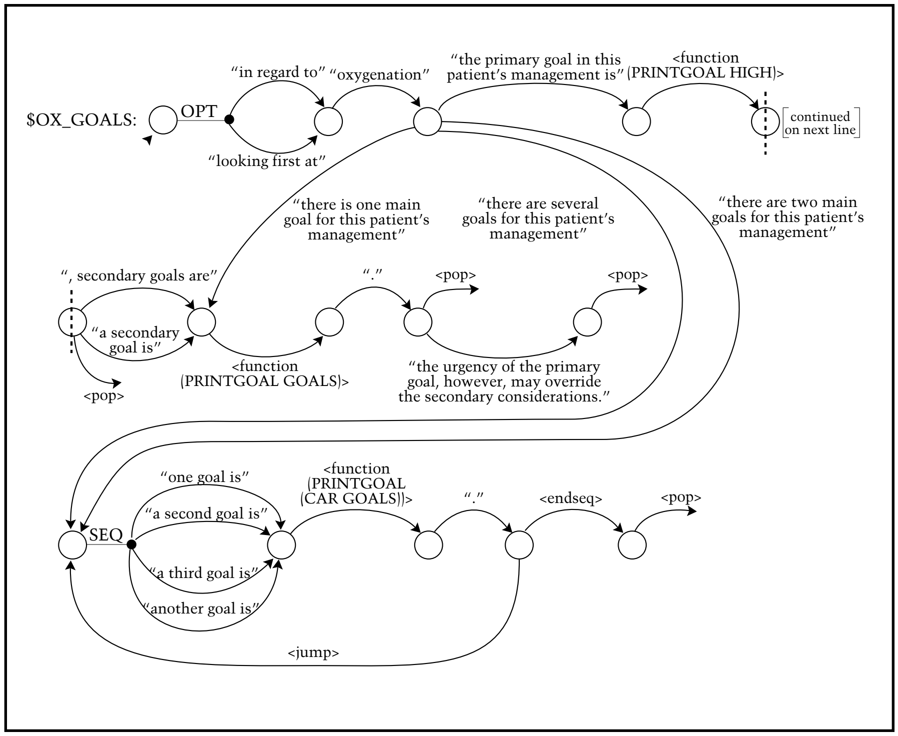


Figure 6. An Example of a Paragraph-Level ATN.

As discussed in the previous sections, the approach gives the expert system designer considerable flexibility at three levels: (1) it allows phrase and sentence level expression to reflect the nuances of the material being discussed; (2) it gives the system a global overview of the content of all the material to be output concerning a given subject, so that the material can be grouped and sequenced appropriately; and (3) it allows paragraph-level expression to adapt to the content being discussed.

It is useful to compare the PROSENET/TEXTNET approach to the type of rule-based explanation discussed previously. When an explanation is generated directly from rules, the content of the explanation is restricted to that of the rules themselves. There will, however, always be nuances possible in the choice of wording that cannot be derived directly from those rules. The PROSENET/TEXTNET approach allows these nuances to be captured in a prose generator, which is separated from the system's knowledge base, yet still controlled by the system's analysis.

Indeed, the separation of the logic of the prose generator from that of the expert system is one of the key advantages of this approach. Once the expert system is performing satisfactorily, there is still a great deal of work remaining to construct a prose generator capable of producing polished prose, just as there is a great deal of work required to convert an outline to a polished document.

The PROSENET/TEXTNET approach allows the system designer to massage and refine the prose-generating logic incrementally until its output is satisfactory for the different circumstances that might need to be described. As discussed earlier, this approach has already been applied sucessfully in a variety of medical domains. The PROSENET/TEXTNET approach is a first step in providing expert system designers with prose-generating tools that are simultaneously powerful, convenient, and practical.

Note on Implementing PROSENET/TEXTNET

A good introductory description of the augmented transition network formalism is found in Winston and Horn (1984). Detailed descriptions of the implementation of both PROSENET and TEXTNET are available. The appendices of Miller (1986) show Lisp code for PROSENET, written in a simple Lisp subset. Chapter 7 of Rennels (1987) describes in detail the implementation of TEXTNET, which was designed to take full advantage of object-oriented programming techniques. Implementing PROSENET or TEXTNET is straightforward. The entire PROSENET program, for example, is only 6 or 7 pages of Lisp code. Once PROSENET is implemented, the process of developing polished prose output is performed by writing and iteratively massaging PROSENET ATNs and expressive frames, as described previously. The PROSENET program itself need not be tailored for a particular domain.

Acknowledgments

This work was supported in part by NIH Grants R01 LM04336 and LM07033 from the National Library of Medicine.

References

Appelt, D. 1982. Planning Natural Language Utterances to Satisfy Multiple Goals. Ph.D. diss., Dept. of Computer Science, Stanford Univ.

McDonald, D. D. 1980. Natural Language Production as a Process of Decision Making under Constraint. Ph.D. diss., Dept. of Electrical Engineering, Massachusetts Inst. of Technology.

McKeown, K. R. 1982. Generating Natural Language Text in Response to Questions about the Data Base Structure. Ph.D. diss., Moore School of Electrical Engineering, Univ. of Pennsylvania.

Miller, P. L. 1986. *Expert Critiquing Systems: Practice-Based Medical Consultation by Computer.* New York: Springer-Verlag.

Miller, P. L. 1985. Goal-Directed Critiquing by Computer: Ventilator Management. *Computers and Biomedical Research* 18: 422-438.

Miller, P. L. 1984. *A Critiquing Approach to Expert Computer Advice: ATTENDING.* London/Boston: Pitman.

Miller, P. L. 1983a. Critiquing Anesthetic Management: The "ATTENDING" Computer System. *Anesthesiology* 58: 362-369.

Miller, P. L. 1983b. ATTENDING: Critiquing a Physician's Management Plan. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5: 449-461.

Miller, P. L., and Black, H. R. 1984. Medical Plan-Analysis by Computer: Critiquing the Pharmacologic Management of Essential Hypertension. *Computers and Biomedical Research* 17: 38-54.

Miller, P. L.; Shaw, C.; Rose, J. R.; and Swett, H. A. 1986. Critiquing the Process of Radiologic Differential Diagnosis. *Computer Methods and Programs in Biomedicine* 22: 12-25.

Rennels, G. D. 1987. *A Computational Model of Reasoning from the Clinical Literature.* New York: Springer-Verlag.

Rennels, G. D.; Shortliffe, E. H.; Stockdale, F. E.; and Miller, P. L. 1987. A Computational Model of Reasoning from the Clinical Literature. *Computer Methods and Programs in Biomedicine* 24: 139-149.

Scott, A. C.; Clancey, W. J.; Davis, R.; and Shortliffe, E. H. 1984. Methods for Generating Explanations. In *Rule-Based Expert Systems,* eds. B. G. Buchanan and E. H. Shortliffe, 338-362. Reading, Mass.: Addison-Wesley.

Shortliffe, E. H. 1976. *Computer-Based Medical Consultations: MYCIN.* New York: American Elsevier.

Swartout, W. 1982. GIST English Genera-