Section 1: Use of SUMEX-AIM to study prose comprehension and the interaction of semantic knowledge sources.

I have now for some time worked on problems of comprehension and memory for text. The beginnings of this work were reported in a chapter in "Models of human memory" (Norman, 1970). Much of the early work was concerned with "The representation of meaning in memory", the title of a book which appeared in 1974. Recently, a psychological process model of text comprehension and production was published (a reprint is enclosed as Appendix X). For several years now, this work has been done in cooperation with Teun van Dijk, a linguist at the University of Amsterdam. We regard our processing model as a good start, but in order to go further we need to develop new theoretical techniques - hence our desire to gain access to SUMEX-AIM.

My work has been supported by a grant, MH-15872 from the National Institute of Mental Health, which is now in its 11th year. Recently, two more applied projects were added to this, a study of readability supported by the National Institute of Education for 3 years, and a project concerned with information acquisition from large, complex redundant, and frequently irrelevant and unreliable textual sources (a 4-year grant from the Office of Naval Research, co-investigator with Lyle Bourne). Both of these projects provide good testing grounds for various features of the model; they complement rather than compete with the basic program. The two recent "applied" projects speak to the need for a better understanding of the psychological processes in text comprehension.

At present, our model is still rather primitive and only partly formalized, but I think it is a nontrivial step forward. Instead of the usual intuitive, atheoretical approach that has characterized research in the psychology of language comprehension so far, it becomes possible to ask more specific questions, which we hope will lead to accelerated progress in this field.

Our problem arises from the very fact that we were able to build this processing model of comprehension. It is a rather complex model, and we have found that computer simulation of the model is necessary to accurately derive its predictions. Although it is possible to do a certain amount of hand simulation, we now have a LISP program that simulates the construction of coherence graphs by the model. This program was written at Yale last summer by one of my students, Ely Kozminsky, but the inadequacies of our local facilities have made it very difficult to get this program to work here. This program actually simulates only a single aspect of the model - the construction of the coherence graphs for different parameter value combinations. Its operation is crucial for us now, because without it we cannot test the empirical adequacy of our model. We currently have recall protocols from 600 subjects on 20 different texts, but we are unable to simulate the model's analysis of these texts, thereby evaluating the model, without access to a suitable LISP facility.

This is, however, only a short-term problem. Even if we could run the coherence graph program, our long-term theoretical goals would be unattainable. The set of protocols we have collected, large as it is, was designed to test only one particular feature of the model. Clearly, we want eventually to go beyond that; this means we must undertake much more ambitious modelling than we have attempted so far. Access to some of the sophisticated artificial intelligence languages available on SUMEX-LISP and AGE in particular -- would greatly facilitate this task.

AGE could be a very useful program for us since, formally, our model has certain similarities with existing speech understanding programs like HEARSAY II. HEARSAY starts from some physical description of a speech stimulus and works this through several interacting levels to a conceptual representation. In our case, we start where HEARSAY leaves off, with a propositional level that, through a coherence graph generator is mapped into what we call the text base. This in turn is converted by macro-operators into the macrostructure (gist) of the text, in conjunction with a control schema, dependent upon the subject's goals and strategies. (See Kintsch and van Dijk, 1978, for a more complete description of these components.) HEARSAY-like techniques appear promising to investigate whether this complex system actually runs, and whether its implications are indeed compatible with the behavior of human subjects. My research emphasis always has been and will remain on a combined theoretical and experimental approach, but in order to do the theoretical portion of the project I need to borrow methods and techniques from artificial intelligence. I now have a collaborator, Dr. James Miller, who has experience with this kind of work, and is a reasonably experienced LISP programmer. Dr. Miller is a recent Ph.D. from UCLA where he has had some experience with the SUMEX version of LISP. With his help, I think we can do what we need to do.

The nature of the proposed theoretical work is briefly outlined in Appendix B, which was prepared by Dr. Miller. As is quite obvious, our ideas for the construction of a HEARSAY type processing model are still highly tentative. In fact, without actually doing some of the work, it is hard to see how we can advance them much further. However, I want to point out that in principle our previous theoretical work on comprehension appears to be highly compatible with the structure we suggest here. We would like to try to develop this work along the lines suggested partly because it appears interesting and promising, but also because the approach we have used up to now may no longer be adequate; we need some more powerful method to describe the interactions in this system.

We do not promise that we can develop a full-fledged simulation of text comprehension in the next few years. But we do think that we can get a partial model of text comprehension in which several important components are fully worked out, as described in the enclosed paper. Other components, at least for the foreseeable future, must be dealt with informally (e.g., the model starts and ends with semantic representations rather than text proper; certain information presumably contained in the model's semantic memory may initially be supplied by the user.) Future developments may remedy this situation, but, even with all its limitations, the model promises to be quite useful. I will not say anything about the intrinsic interest of such a model and its implications for our understanding of understanding, but the model would be very helpful to many other researchers that have to deal with prose comprehension or production. A number of divergent areas exist -- cognitive psychology, education, and the behavioral effects of drug abuse for instance -- where there are many researchers who need and want to deal with prose comprehension and memory. Unfortunately, these researchers often neglect this area or contact it in a very superficial way, because of the absence of a productive theoretical framework. If we could develop a program that at least partially simulated the comprehension process, it would not be hard to find potential users. Currently, a group of psychologists in Pittsburgh under the direction of Dr. J. Voss has used the quantitative techniques in Appendix A to analyze some of their text memory experiments with good results. However, they were unable to use the full power of the model because we could not furnish them with a working program to derive some of the more complex predictions! Similarly, several educational researchers have used the present model to investigate practical problems of interest; a working model would greatly facilitate their efforts. As a final, almost randomly chosenexample, consider the small but active group of researchers interested in cognitive effects of drug abuse. While these researchers would like to move away from paired-associate experiments, they require a more fully developed cognitive model to support the much more ecologically valid prose experiments that they would like to do. Such experiments are of course meaningful only when one can interprete their results in terms of some reasonably comprehensive theoretical system. Without modern AI techniques, a model such as ours is just too complex to deal with.

In summary, I think we can develop a partial model of text comprehension by marrying our present empirical, piecemeal approach to certain AI techniques that are either already in existence or being developed (primarily, AGE 0). I also suggest that such a development might obtain considerable theoretical and practical significance.

Section 2: The HEARSAY implementation of the prose model. The purpose of this section is to show how the components of the prose processing model described by Kintsch (1974; Kintsch and van Dijk, 1978) might be implemented in a HEARSAY-like control structure. The work on this model to date has been oriented toward the development and evaluation of individual sections of a global model of human prose processing, and this work, as well as other research in cognitive psychology and artificial intelligence, has described a number of significant and necessary components for a successful system. Our current goal in this research is to represent these individual components in a common formalism, and to describe their interaction in the processing of a segment of prose. We believe that the HEARSAY formalism of multiple independent, but cooperating knowledge sources may be a useful system in which to describe the proposed interactions.

The levels of representation of this system are diagrammed in Figure 1; they may be described as follows:

Strategies: A text may be read for a number of different reasons, with correspondingly different types of information being extracted from that text. Sherlock Holmes stories may be read in an attempt to solve the crime, or to gain an understanding of the social structure of 19th century Engla1d. Which of these interpretations the reader chooses will be dependent upon the goals the reader has, and the strategies required by these goals. If the goal of the reader is to solve a crime, he might choose strategies such as "collect and interrelate pieces of evidence" and "predict the actions of the criminals and the police." Alternatively, reading the story for information on the structure of the character's society would require such goals as "observe interpersonal relations" and "predict behavior of members of social class x."

Schema: This level describes the structure of a text, including such information as the purposes of the possible subdivision of the text, and the information typically contained in each subdivision. As such, this schematic level is somewhat more specific than other knowledge structures referred to as schemata (Bobrow and Norman, 1976).

Word and world knowledge: This level corresponds to what has generally been called semantic memory. It contains a reader's vocabulary, the interrelated definitions of these words, and frame-like clusters of semantic information.

Episodic memory: This memory structure contains a record of propositions derived from a text. Although it is very close to a simple list of propositions from the text, it is presumed to be interconnected with semantic memory, allowing access to world knowledge from an episodic trace, and vice versa.

Propositions: Propositions are the basic conceptual units of the prose system. They are derived from the surface form of the text, and represent the information contained in the text. For instance, a simple sentence such as "John threw the ball" would be represented by the proposition (THROW, JOHN, BALL): the verb of the sentence serves as the primary relation of the proposition, and the agent and object of the sentence are the arguments of this relation. A more complete description of the propositional system can be found in Kintsch (1974).

Text base: A simple string of propositions that might be found in the analysis of a prose passage is converted by the prose system into a hierarchically structured text base. This structure shows how a set of text propositions are interrelated, primarily through the repetition of proposition arguments or relations. The text base is primarily made up of propositions explicitly found in the text itself, although it may also contain inferences necessary to maintain the coherence of the text base's hierarchical structure. Although the entire text base is retained in episodic memory, only a limited portion can remain active during processing. Those propositions that remain active following the processing of one sentence can be used to integrate propositions from following sentences into the text base.

Macropropositions: A second way the prose system deals with its limited storage and processing capabilities is by developing macropropositions from various subsets of a text's propositions. These macropropositions may be derived by techniques such as generalization, deletion of unnecessary propositions, construction, and integration, (see van Dijk, 1977). These macropropositions allow the system to maintain the most important information about a text in a limited number of "concentrated" propositions.

Macrostructure: Macropropositions are built into a hierarchical structure similar to the text base's construction from text propositions. This structure becomes a representation of the meaning of the text, with the different levels of the hierarchy describing the text at different levels of specificity.

Text: This level is the basic prose confronted by the system.

Corresponding to the nature of the HEARSAY formalism, and as illustrated in Figure 1, these representational levels are interconnected to enable their specific information to take part in decisions and inferences at different levels. These interconnections are as follows:

(a) A reader's strategies can lead to the selection of a story schema appropriate for the desired strategy.

(b) The selected schema provides information useful for the construction and organization of the macrostructure. It can help identify individual propositions in the textbase as being relevant or irrelevant, and can keep the number of propositions under consideration within the processing capacity of the system.

(c) World knowledge can provide information useful for the generation of inferences at the level of propositions and macropropositions.

(d) Episodic memory can retrieve previously encountered propositions if the text base is having difficulty maintaining coherence available from repeated proposition relations or arguments.

(e) The macrostructure can provide information to be included in episodic memory, and can help select a schema appropriate for the current macrostructure.

(f) Macropropositions are entered into the macrostructure at appropriate points, and provide information useful for the generation of inferences that will maintain the coherence of the macrostructure.

(g) Components of the text base are entered into episodic memory as they are encountered, and use macro-operators (van Dijk, 1978) to convert propositions in the text base into macropropositions.

(h) Individual propositions are entered into the text base, and provide information for inferences that will maintain the coherence of the text base.

(i) A system capable of translating natural language into propositions is presumed to exist in the human prose processing system; the development of such a parser is not an immediate goal of the present research. Hand coded propositions will be entered into the system during at least the initial stages of the research.

A significant task of this research will of course be the specification of the actions of these knowledge sources to actually generate the understanding of prose, van Dijk.(1977) has considered one of these stages in some depth -- the development of macropropositions from the contents of the text base -- and it would be appropriate to show how his discussion can be implemented as productions appropriate for HEARSAY-like knowledge sources.

One of van Dijk's macrorules described how a set of propositions may be generalized into a higher order macroproposition. In this way, the propositions inherent in the sentences, "John moved the chair", "John moved the table", and "John moved the chest" can be generalized into "John moved the furniture." This could be implemented in a knowledge-based system by assuming (a) the propositional representation of the above sentences as (MOVE, JOHN, CHAIR), (MOVE, JOHN, TABLE), and (MOVE, JOHN, CHEST), (b) an associative knowledge structure capable of identifying CHAIR, TABLE, and CHEST as various types of furniture, and (c) a production corresponding to:

(1) IF: a set of propositions have common relations or arguments, and the corresponding non-matching relations or arguments are instances of a super-ordinate category,

THEN: replace the set of propositions with a macro-proposition consisting of the shared components and the super-ordinate category.

This production would lead to the generalization of the above propositions to the macroproposition (MOVE, JOHN, FURNITURE).

More complex sets of text propositions would require more complex productions that would likely require greater use of the memory network and world knowledge. For instance, the above production could not generalize the propositions (CLEAN, FATHER, KITCHEN), (TYPE, MOTHER, BOOK), and (PAINT, CHILDREN, DOGHOUSE) into (WORK, FAMILY, $), (i.e., "The family is working;" the $ holds the place of concepts that could not be generalized). This generalization would require a set of productions similar to the following:

(2) IF: A common argument or relation is needed to unite a set of propositions,

THEN: activate memory around the components of the propositions and try to locate a common concept suitable for generalization.

(3) IF: A common concept for the corresponding relations or arguments of a set of propositions can be found,

THEN: replace the set of propositions with a new proposition containing the discovered concept, and generalizations of the other relations or arguments as possible.

It is assumed that these productions, working with the memory network, would locate WORK as a superordinate for CLEAN, TYPE, and PAINT, and FAMILY for FATHER, MOTHER, and CHILDREN, allowing the described generalization.

It is important to note, however, that this generalization process is not strictly data-driven, but may require the observance of subject strategies and the schema guiding the organization of the story information. For instance, suppose the set of propositions above was replaced by: (CLEAN, FATHER, KITCHEN), (TYPE, MOTHER, BOOK), and (PLAY, CHILDREN, BASEBALL). The application of productions 2 and 3 would lead to two options for a generalized macroproposition. All three propositions could be condensed to (DO, FAMILY, $); alternatively, the (PLAY, CHILDREN, BASEBALL) proposition could be left alone and the two remaining propositions could be generalized to (WORK, PARENTS, $). Both of these alternatives have advantages; the selection of the most suitable generalization would be dependent upon the demands of the currently adopted strategy and schema.

A second rule of van Dijk's describes how a number of text propositions can be integrated into one macroproposition. This rule could be written in production form as:

(4) IF: there is a set of propositions such that one of them instantiates a frame for which the remaining propositions are normal conditions, components, or consequents (i.e., capable of filling slots in the frame),

THEN: replace the set of propositions with the one that instantiates the frame),

This production could then integrate the propositions (GO, JOHN, PARIS), (TAKE, JOHN, CAB, STATION), (BUY, JOHN, TICKET), and (TAKE, JOHN, TRAIN, PARIS) into the macroproposition (GO, JOHN, PARIS).

A complementary rule describes how macropropositions might be constructed from a set of text propositions:

(5) IF: there is a set of propositions that are normal conditions, components, or consequents of a frame,

THEN: replace the set of propositions by a new proposition that instantiates that frame while retaining the specific information in the old propositions.

Four points should be noted in conclusion. First, the general rules described by van Dijk will likely require a number of specific productions for the complete instantiation of a rule. Two subsets of productions were noted for the simple cases of generalization above.

Second, the use of one macrorule should not rule out the application of another macrorule to the same set of propositions, or the outcome of the first rule's application. For instance, a construction macrorule such as the following might be useful for the processing of the example of the working parents and playing children:

(6) IF: the relations of two propositions are member of the same semantic field,

THEN: construct a macroproposition that contrasts the two propositions.

This rule would result in:

PROP1: (WORK, PARENTS, $)

PROP2: (PLAY, CHILDREN, BASEBALL)

PROP3: (BUT, PROP1, PROP2)

Third, an eventual stage of the model might be concerned with the development of new productions, and the addition of these productions to existing knowledge sources. For instance, production 1 in the generalization discussion can be viewed as a special case of productions 2 and 3: the semantic relation shared by the relation (MOVE) and the first argument (JOHN) is simply the identity relation. Substitution of this "discovered" relation by production 3 would lead to the same macroproposition as did the less general production 1. The existence of production 1 is certainly desirable, since it prevents the unnecessary memory access in productions 2 and 3. It might then be possible for the prose system itself to generate production 1 by supervising its application of productions 2 and 3 in the (MOVE, JOHN, FURNITURE) example, and invoking a rule similar to: .

(7) IF: a rule accesses the memory network, and the information used to direct the memory search is identical to the information returned from the memory network,

THEN: the memory access in this rule may be unnecessary in some cases: inspect the executed rule and the patterns that triggered it, and attempt to create a special case of the rule.

This production would note that some of the information sent to the memory network for the location of a common concept (MOVE and JOHN) was the same as the information returned, and would attempt to synthesize a rule like production 1 that performs a simple pattern matching and substitution in cases such as these. Such rule discovery would become an important part of the model's capabilities leading to the investigation of many topics relevant to the development of reading performance, and would correspond to recent work in artificial intelligence that has found a HEARSAY-like structure to be extremely convenient for the implementation of such discovery techniques (cf. Lenat's AM system, 1977). In any case, the application of a HEARSAY structure to this domain of research appears to be not only feasible, but conducive to the development of both psychological theory and artificial intelligence techniques.

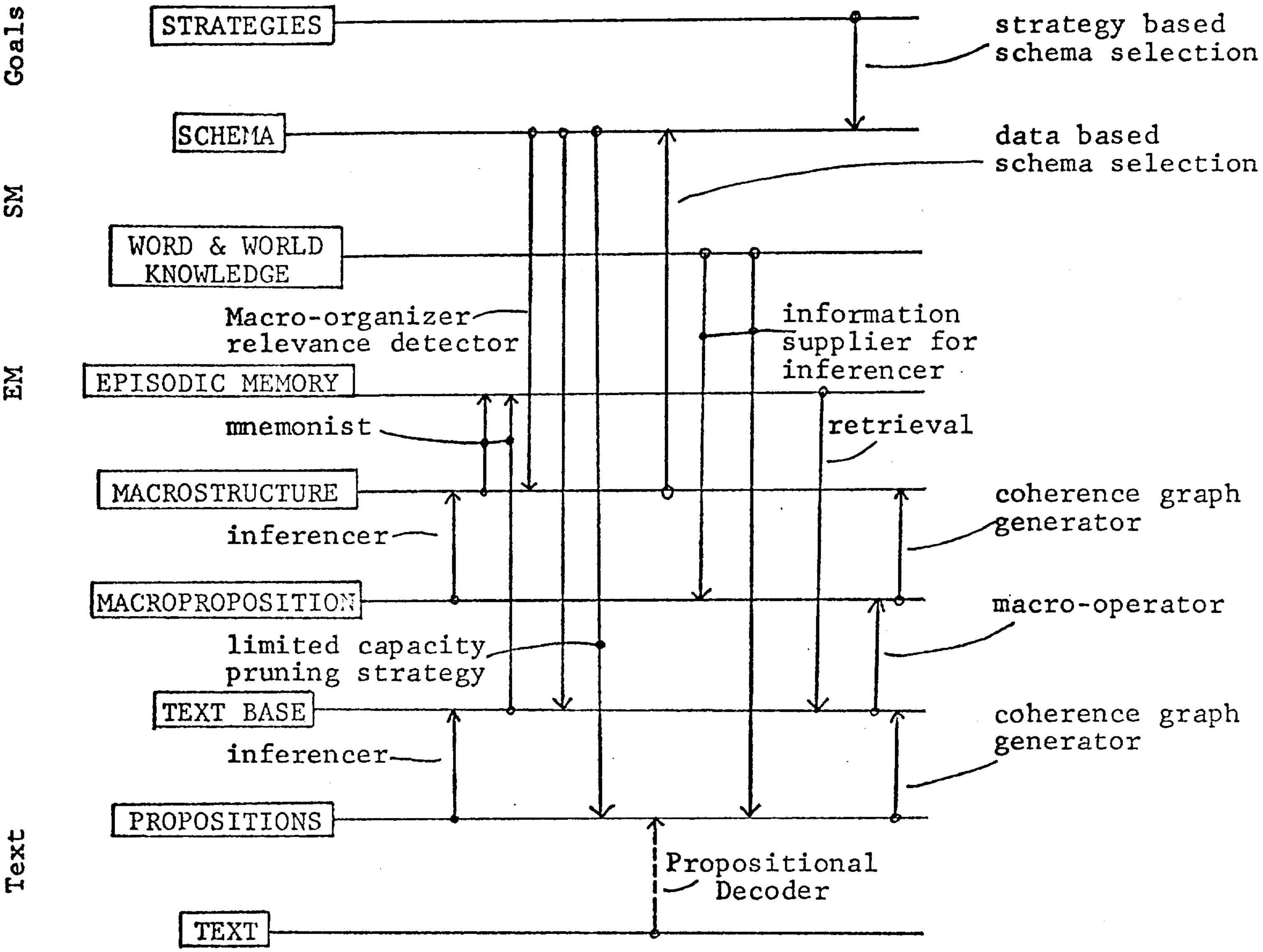
Goals
SM
EM
Text
STRATEGIES
strategy based schema selection
SCHEMA
data based schema selection
WORD & WORLD KNOWLEDGE
Macro-organizer relevance detector
information supplier for inferencer
EPISODIC MEMORY
mnemonist
retrieval
MACROSTRUCTURE
coherence graph generator
inferencer
MACROPROPOSITION
macro-operator
limited capacity pruning strategy
TEXT BASE
coherence graph generator
inferencer
PROPOSITIONS
Propositional Decoder
TEXT

FIGURE 1

Section 3. Use of SUMEX-AIM to Study Planning in Software Design

In the following section, we will describe the background, current status, and proposed uses of SUMEX-AIM by the second of the two research projects that are part of the Colorado application. The second research program is a study of the processes involved in planning in problem solving domains that require the integration of a large amount of information in order to synthesize a plan. This research is supported by the Personnel and Training Research Program of the Office of Naval Research. The project has been underway for about nine months. We are proposing to use software tools being developed at SUMEX-AIM to construct models for the planning data that we have obtained in experiments that we describe below. In particular, we would like to use the AGE-0 or AGE-1 systems being developed by Nii and Aiello (1978).

We start by describing the task we are using in our research and our initial theoretical ideas. We will then go on to give a brief description of the results of our initial experiments. We will argue that a HEARSAY-II like model is an appropriate theory for the planning data that we have collected. We have been very strongly influenced in our current thinking about planning by the work of Fredrick and Barbara Hayes-Roth at RAND Corporation (Hayes-Roth and Hayes-Roth, 1978).

Our research uses the task of software design to study planning. The focus of our activities in the last several months has been the selection of design problems and the recording of thinking out loud protocols from expert subjects. We have selected relatively elementary problems; all of our designers can easily understand the objective of the system to be developed. One problem is to construct a page-keyed index for a text book; the other is to write a simple appointment book that will make use of a computer and time-sharing terminal. None of these problems involve techniques that are so exotic that a competent computer scientist would not be able to construct a passable solution. We would guess that a skilled individual could write a running program to solve one of these problems in from a few days to a couple of weeks.

The initial theoretical framework motivating our research was derived from a planning model proposed by Sacerdoti (1975), NOAH. NOAH is an integrated problem solving system that solves planning problems by an iterative process. The major theoretical construct incorporated into NOAH is a procedural net. This term refers both to a data structure and to the process of constructing that data structure. A procedural net is constructed iteratively, beginning with a top node that is essentially a description of an intention to solve the problem. This node is then expanded into a schematic description of the solution plan. This abstract description is then examined and evaluated for completeness and consistency by a set of "critics". The elements of the plan may be rearranged by the critics. The next level of the plan is then derived from the current level by expanding each node into a more detailed set of operations. Once this expansion has taken place, the critics then reorder the new plan eliminating any inconsistencies, conflicts, or duplicate operations. The process of "expand-criticize" continues iteratively until a plan has been generated whose elements describe the operations necessary to actually solve the problem. This completed structure is called a procedural net.

We would like to point out that there are really three separable components to this model. The first is the completed plan - the procedural net. This is a hierarchical structure with the top level being a highly abstract characterization of the solution and the bottom level being a complete detailed description of the actual sequence of actions that will solve the problem. The second assumption is that this hierarchical structure is constructed in a top-down, breadth-first fashion, using the processes that were described in the previous paragraph. The third assumption concerns the organization of knowledge incorporated into the completed solution plan. In Sacerdoti's (1975) system, each node describing a subgoal at a given level of abstraction contains all of the information necessary to construct a solution plan at the next level of detail. In other words, each node contains a more detailed plan for accomplishing its goal. These three components can be described as the structure of the completed plan, the dynamics of the process that generated the plan, and the organization of the constituent knowledge that was used to synthesize the plan.

The analysis of results from several experiments and the reading of the literature on planning and problem solving has led us to several conclusions concerning the three elements of our original theoretical framework. The first is that a completed plan, in our case a software design, does have the structure of a procedural net. Much of the theoretical work on planning in the robotics and psychological literatures argues that plans have this hierarchical nature. The second conjecture, that plans are generated in a strict top-down, breadth-first manner, has not fared so well. In our initial experiment with expert subjects, one of the three subjects gave us a perfect top-down expansion of his design. The protocol given by the second expert could be characterized as mostly top-down, but there were some interesting exceptions. The third expert identified the critical element of the problem, a structure that was a fairly low level of the procedural net, and proceeded to expand that part of the design first. Our protocols indicate that expert subjects are clearly aware of the fact that various design techniques result in expansion of the completed design structure in different ways. Two of our subjects explicitly stated their design strategy and then generated a protocol consistent with that strategy. The third element of Sacerdoti's NOAH system, assumptions about the organization of knowledge, was not incorporated in any strong way into our theory. However, this left us without any kind of adequate description of how knowledge is included in the plan representation.

Our initial characterization of the organization of the information that was incorporated into the plan was very crude. We proposed that this knowledge could be partitioned into two broad domains. The first domain involved the knowledge that the subject used to understand the description of the task. Thus if the task were to design a program to do theoretical calculations in physics, understanding the task would presumably require detailed knowledge of the relevant physics. The second knowledge domain we assumed to encompass a subject's knowledge of computer science, design techniques, etc. This partition of knowledge types is so crude that it gave us no adequate way to develop processes that would lead to the particular synthesis seen in a completed design. Neither did these hypothesized knowledge structures adequately describe the kinds of knowledge that were utilized by expert subjects in selecting design strategies. Careful study of our protocols indicated that diverse kinds of knowledge underlie the behavior of expert designers. The inadequacy of our initial classification was further made clear to us by careful study of Hayes-Roth and Hayes-Roth (1978). In this paper, the Hayes-Roths define a model for planning that incorporates many of the notions of the HEARSAY-II model for speech understanding. This model assumes that a plan is synthesized from a large number of diverse types of knowledge, but that these kinds of knowledge can be organized into various subcategories.

As a theoretical exercise, we have taken one of our protocols and attempted to identify in it the specialists, or knowledge sources, that we think would be required to explain the kind of behavior that is recorded in the protocol. Our preliminary attempts at this effort have identified for us, four classes of knowledge structures that seem to be involved in constructing a complete software design, or plan. These are shown in Tables 1 to 4.

The first group of knowledge structures seem to be involved in understanding the problem to be solved. (See Table 1). These knowledge structures are used to define the given elements of the problem, the goal that is to be achieved, and the environment in which the solution is to be implemented. In a complete theory of planning that included a language understanding system, these knowledge structures would obviously be very closely associated with the knowledge structures that control the processes of understanding.

We have labeled the second group of knowledge structures, shown in Table 2, as the pre-planning group. There are two general kinds of knowledge structures in this category. The first, when invoked, establish policies and priorities (e.g., the plan should be implementable in the least possible time, or "hard" subproblems should be deferred as long as possible). A second class of knowledge structures in this group attempts to identify subproblems or elements of the task whose solutions are already known to the designer. For example, in the page-keyed indexing system, expert subjects immediately recognize that a pattern match process of some variety will be required. Knowledge schemata that are relevant to pattern matching are then invoked.

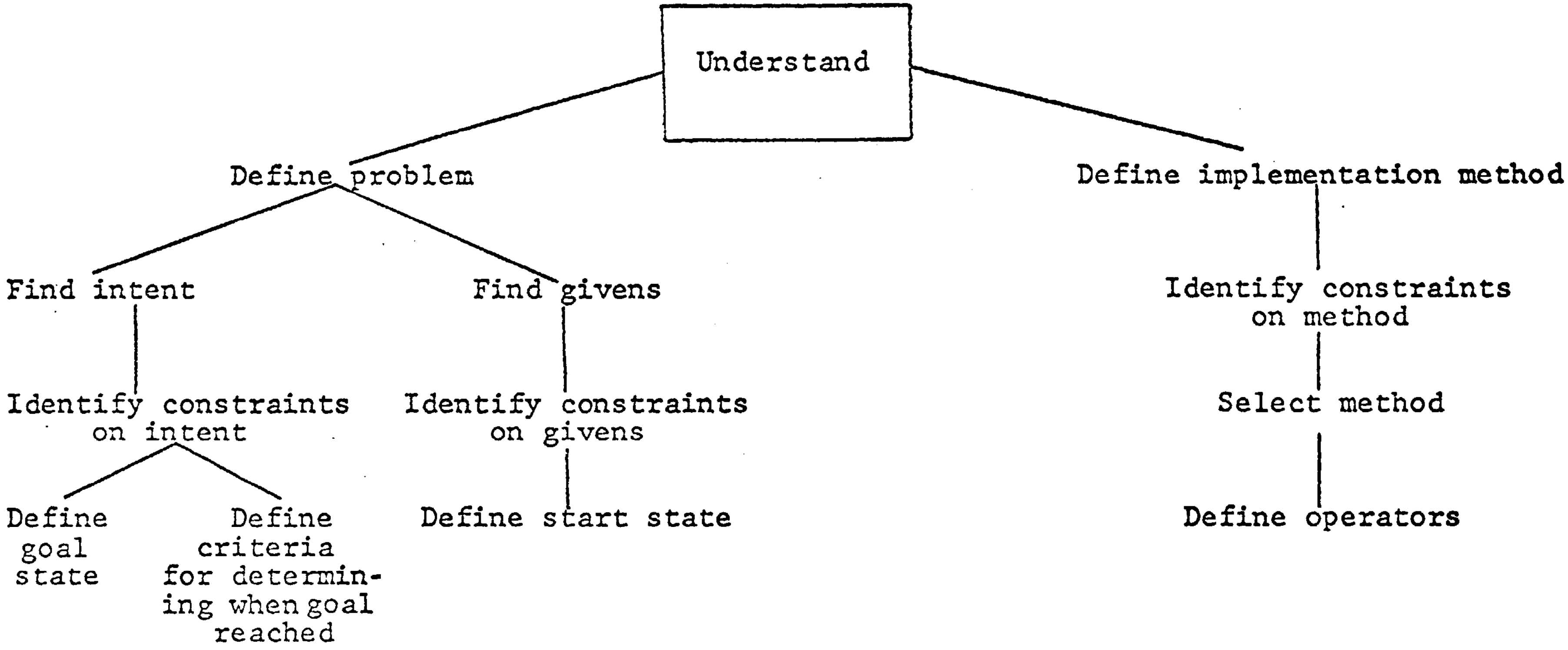
Understand
Define problem
Define implementation method
Find intent
Find givens
Identify constraints on method
Identify constraints on intent
Identify constraints on givens
Select method
Define goal state
Define criteria for determining when goal reached
Define start state
Define operators

TABLE 1

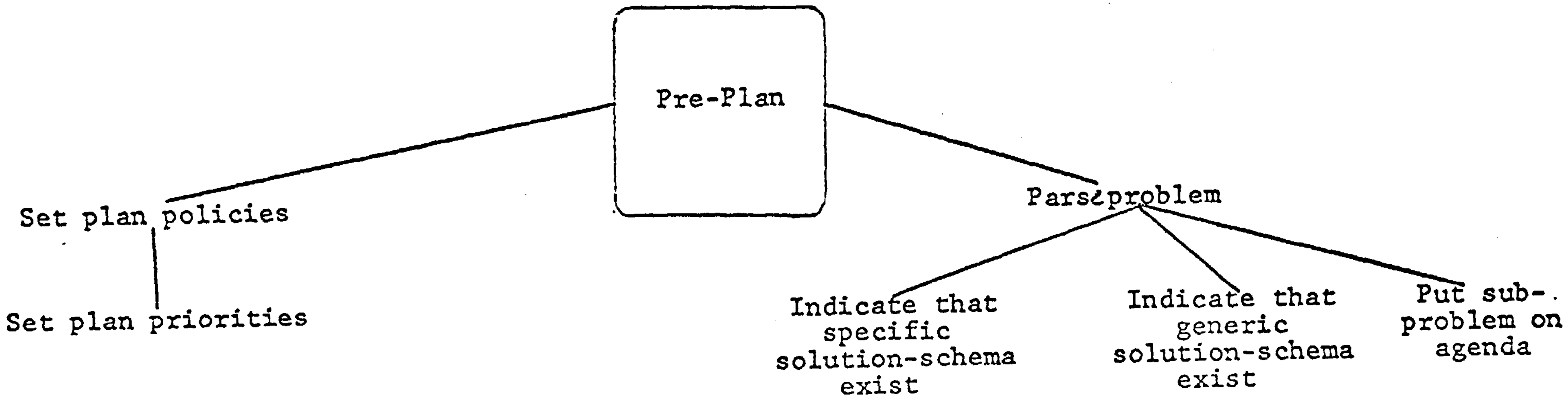
Pre-Plan
Set plan policies
Set plan priorities
Parseproblem
Indicate that specific solution-schema exist
Indicate that generic solution-schema exist
Put sub-problem on agenda

TABLE 2

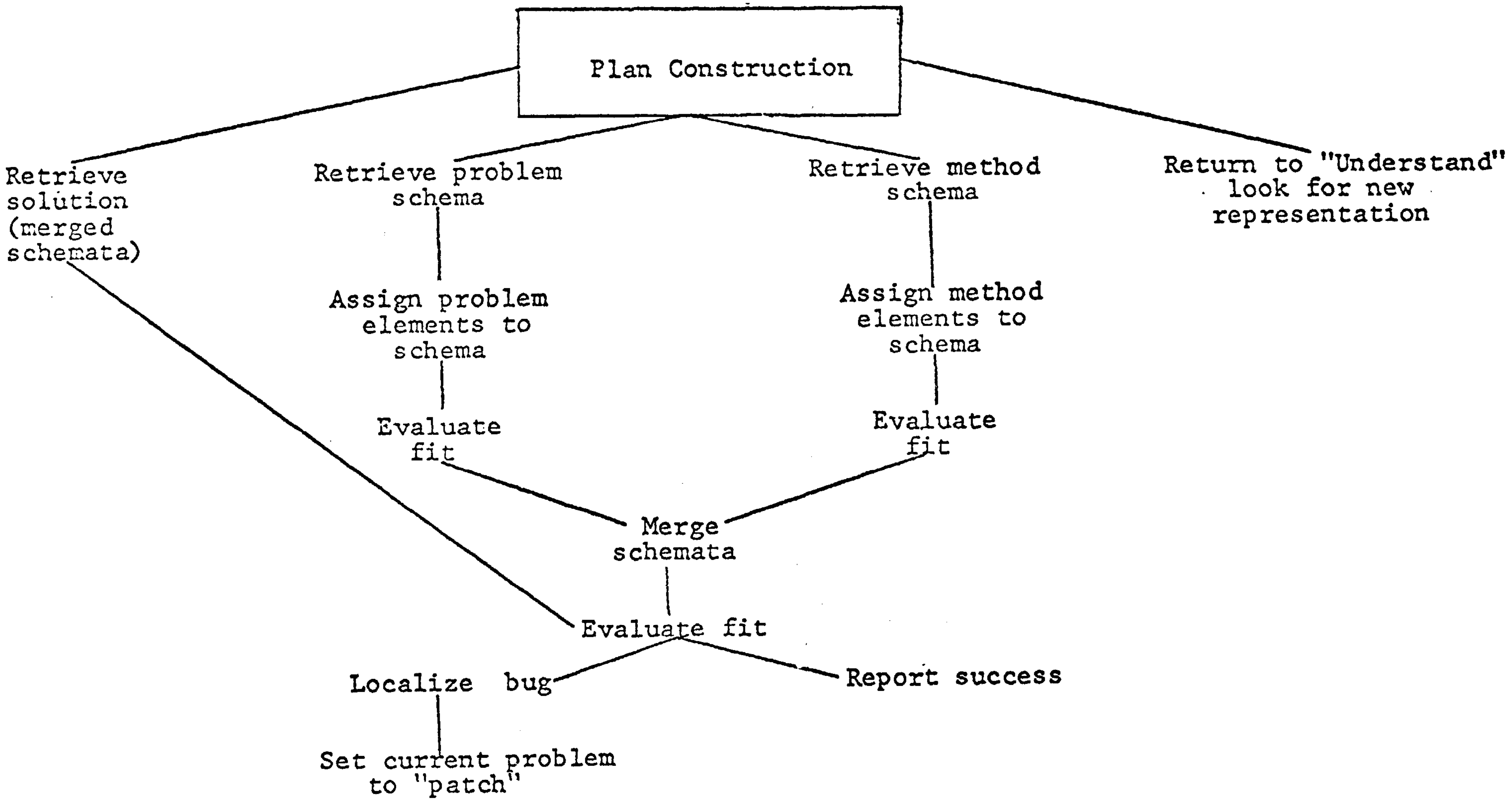
Plan Construction
Retrieve solution (merged schemata)
Retrieve problem schema
Retrieve method schema
Return to "Understand" look for new representation
Assign problem elements to schema
Assign method elements to schema
Evaluate fit
Evaluate fit
Merge schemata
Evaluate fit
Localize bug
Report success
Set current problem to "patch"

TABLE 3

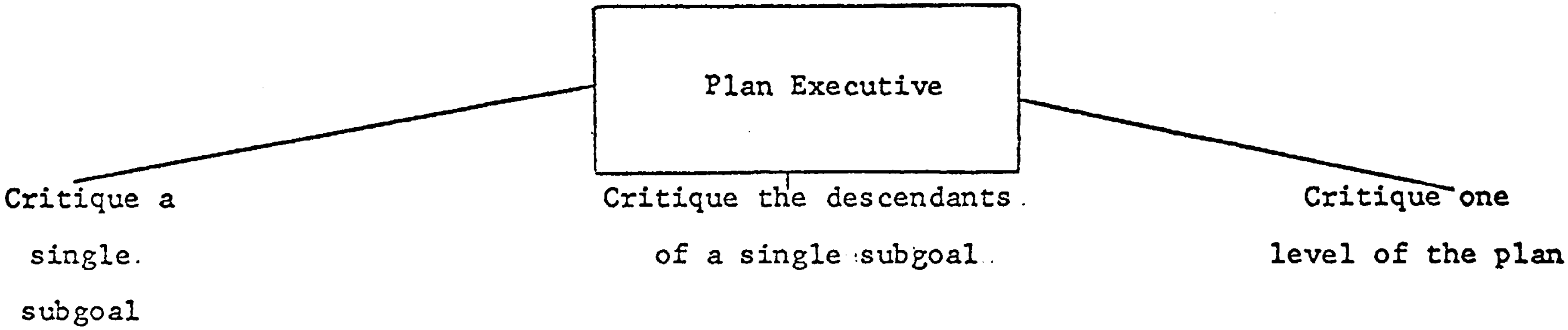
Plan Executive
Critique a single subgoal
Critique the descendants of a single subgoal
Critique one level of the plan

TABLE 4

The plan construction knowledge structures, shown in Table 3, control the actual synthesis of the developing plan. One of the major functions of these knowledge structures is to retrieve schemata that are applicable to the solution of subproblems that have been identified in the plan and apply these schemata, modifying them if necessary. These knowledge sources control a process that is very analogous to Sussman's (1977) Problem Solving By Debugging Almost Right Plans.

The final set of knowledge structures that we have identified are the plan executive, shown in Table 4. These knowledge structures are used to simulate the execution of the plan, i.e. to critique and modify parts of the developing plan. They may be applied to a single subgoal, the descendents of a particular subgoal, or an entire level of the plan.

We do not see the above mentioned knowledge structures as a complete specification of the knowledge necessary to solve a software design problem. They were constructed by the fairly careful perusal of a single design protocol, extracting the major knowledge components that the subject appeared to use. They have not been validated in any way. They are undoubtedly incomplete, and more than likely partially incorrect. What we hope to have demonstrated by the above description is that the task domain yields nicely to an analysis of this kind.

In summary, there are two aspects of our data that have led us to adopt a HEARSAY-II like framework to characterize the organization of the knowledge that is incorporated into the completed plan and the dynamics of the actual synthesis process. The first is that a careful reading of the protocols indicates to us that subjects manage to assemble fairly modular pieces of knowledge into a completed plan. Moreover, we are impressed by the diversity of these knowledge structures. Second, it is very clear from our data that expert designers make sophisticated strategic and resource allocation decisions that influence their planning behavior. One of our experts explicitly mentioned the fact that he could generate a plan top-down and breadth-first, but various criteria for the adequacy of the completed plan and other resource allocation decisions dictated that some quite different planning method be used. Our current theoretical framework has no way of dealing with expert subjects' ability to make such resource allocation decisions and then act on them. On the other hand, Hayes-Roth and Lesser (1977) show that HEARSAY-II can be made to use a large number of different strategies by well motivated modifications of the executive processes of the system. Hayes-Roth and Hayes-Roth (1978) make the identical point about their planning model.

We have found it relatively easy and very instructive to examine the protocol and generate lists of hypothetical knowledge sources. However, it soon becomes apparent that attempting to work with a HEARSAY-II like model at a qualitative level is simply not adequate. It is very hard to determine whether the interactions of the various knowledge sources that are postulated lead to the kind of performance, the planning behavior, that one is attempting to model. The only conceivable way of demonstrating the adequacy of such theoretical ideas is to incorporate these conjectures into a HEARSAY-II like system and demonstrate that knowledge sources can be designed that capture the theoretical insights that we have obtained from the protocols. Thus, fruitful continuation of this line of theoretical work on our part requires that we actually construct running simulations of our models incorporating these ideas. We do not have the personnel resources, nor access to the necessary software tools to construct such systems de novo. We currently have access to a Control Data 6400 system that supports an early version of the University of Texas LISP system. However, even if we had a system supporting modern dialects of LISP, the task of developing a knowledge-based system from the very beginning would be beyond our current capabilities.

Examination of the AGE-0 manual (Nii and Aiello, 1978) has encouraged us to believe that our theoretical framework meshes well with the AGE superstructure. We believe that access to this set of modelling tools would make possible the development of simulation models incorporating our theoretical ideas without unduly taxing our resources.

The aspect of AGE that is most appealing to us is the hierarchical structure of both the knowledge sources and the developing solution (the hypothesis). We feel that the knowledge structures we outlined above would map nicely into AGE-type knowledge sources, although we are well aware that they would have to be drastically modified and greatly expanded. The structure of the developing solution (what is called in AGE the hypothesis structure, and in other HEARSAY-like systems the blackboard) that we envision would consist of three distinct, but communicating, hierarchical structures. We will call these structures "planes", after Hayes-Roth and Hayes-Roth (1978). However, the particular planes we envision are somewhat different than those used by Hayes-Roth and Hayes-Roth. The first plane is the plan plane; this is where the actual solution to the problem is built up, level by level. The second plane is the plan abstractions plane; information relevant to the solution, but not part of the actual plan; would be included here. Examples would be policy decisions (e.g., "the human interface aspect is the most critical"), observations about techniques to use (e.g., "this might work very well as a linked list"), or potential problems (e.g., "what will happen if the term file overflows?"). The third plane is the problem description plane; this represents the problem solver's understanding of the problem. Initially it would contain a representation of the problem text, i.e., the output of some text comprehension process. It could be augmented at later times by new information about the problem; for example, if midway through designing a page-keyed index system, the person realizes that a hyphen actually serves two functions - to divide words at line boundaries and as a character in words that are always hyphenated - this new piece of information would be added to the problem description plane.

An example will make clearer how different facets of "the same" piece of information are divided across the different planes. In the page-keyed index problem, the subject is told that "the page number appears after a block of text". That information would be deposited on the problem description plane. On the plan abstractions plane might appear the datum "the page number is going to be problemmatic, because it will not yet be available when a particular occurrence of a term is found"; while the plan plane might contain several items related to the resolution of this problem.

Each of the planes has a hierarchical structure, with equivalent levels on all three planes. We have not as yet further refined what those levels are, but they vary on an abstract - detailed dimension. We also realize that AGE does not explicitly support the concept of planes, but we suspect that the additional bookkeeping necessary to implement this structure will not be very difficult.

The knowledge structures we have postulated separate very nicely according to the planes upon which they deposit information. The set of structures we have called understanding adds to the hypothesis on the problem description plane; the plan construction knowledge structures place information on the plan plane; and the pre-planning knowledge structures contribute to the plan abstractions plane. The executive knowledge structures, as currently conceived, add to both the plan and plan abstraction planes, but we expect that as they are expanded and modified, these structures will be part of the higher-order knowledge sources that control the order in which knowledge sources are invoked, i.e., the kernel, as it is termed in AGE.

We see the direction of hypothesis propogation in our model as being top-down and bottom-up within planes, and "sideways", or within-level, across planes. At the moment we do not anticipate the need for knowledge sources whose inputs and outputs cross both plane and level boundaries; however, we realize that the model will have to be fleshed out in much more detail before we can assert this claim in any strong way.

There are some aspects of our theory that remain to be integrated into the AGE framework. Two of them deserve mention. First, we are uncertain about the control processes that will order the activation of the various knowledge sources. This is due in part to our lack of familiarity with AGE's control structure as based on the limited information on this topic in the AGE-Ø manual. Moreover, the notion of control structures in our theory is currently being refined and expanded. We have determined that designers use many different kinds of control structures to solve solftware design problems; one of the major goals of this work will be to elaborate the possible control structures and the circumstances under which each is used. Second, we intend to include in our model the notion of resource limits, especially memory limits. It is well understood that human beings are not perfect processors of information. We feel strongly that any theory of human behavior must not contain processes that are inconsistent with those limits. An important focus of our work will be an attempt to integrate concepts such as short-term memory limits into a HEARSAY-like model.

While we realize that it will take a large effort on our part to be able to do useful work with AGE, we feel that without this or some similar tool, the modelling task we have set ourselves would be nearly impossible. We expect that it will take several months to familiarize ourselves with AGE and with INTERLISP, as currently only one of us has any familiarity with LISP. It will probably take us one year to become familiar with the modelling tools and to develop an initial model. We intend to take a second year to refine that model and to compare it to data. In fact, we would expect to develop many different models over the second year, as we explore the effects of different processes and knowledge structures on planning behavior. We would hope that the modelling enterprise would be fruitful enough that it would continue over several additional years, but that will depend critically on the outcome of these initial modelling efforts.

Section 4 : Hardware and software requirements for the Colorado SUMEX project

We currently have access to two computer facilities that fulfill various aspects of our research. The experimental direction of our work requires data collection and analysis; the on-line facilities of the Computer Laboratory for Instruction in Psychological Research (CLIPR) and the extensive statistical programs available on the university CDC 6400 are quite suitable for this. However, the desire to formalize and implement our theoretical work in artificial intelligence-like knowledge-based systems requires access to efficient artificial intelligence programming systems. We are specifically seeking access to the UCI-LISP and INTERLISP systems maintained on SUMEX, and the AGE system for implementing HEARSAY-like systems, which is under development by Feigenbaum, Nii, and Aiello. As noted in other sections of this proposal, the similarity in our theoretical orientation to HEARSAY structures makes access to AGE highly desirable. Correspondingly, access to SUMEX is needed since (a) AGE, written in INTERLISP, could not run on either of the computers currently available to us, and (b) although some members of our research group are experienced with LISP, they do not have the experience to construct a complex system like AGE from scratch.

We anticipate that the entire Colorado project will require between 30 and 60 hours of connect time per week, divided among the four to six members of the project. Of this time, some of the first thirty hours and any of the second thirty would be during non-peak hours and weekends. We estimate our disk space requirements at 500 pages for the entire project. Since part of this project is concerned with the analysis of prose, it may be desirable to main some of the experimental texts (in the form of proposition lists as described in Section 2 and the enclosed reprint (Kintsch and van Dijk, 1978) offline on DEC tapes (presuming that these tapes could be mounted by a SUMEX operator on request). Finally, we plan to access SUMEX by either the TYMNET or the ARPANET; we welcome your comments on which network would be most appropriate in view of the various agencies that fund the different aspects of our project.