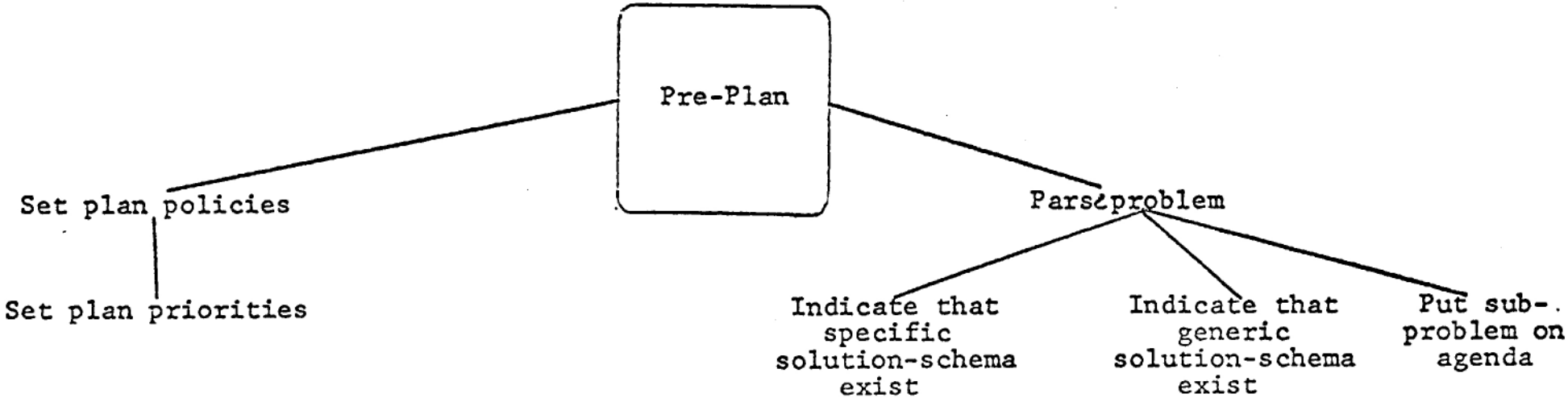
Pre-Plan
Set plan policies
Set plan priorities
Parse problem
Indicate that specific solution-schema exist
Indicate that generic solution-schema exist
Put sub-problem on agenda

TABLE 2

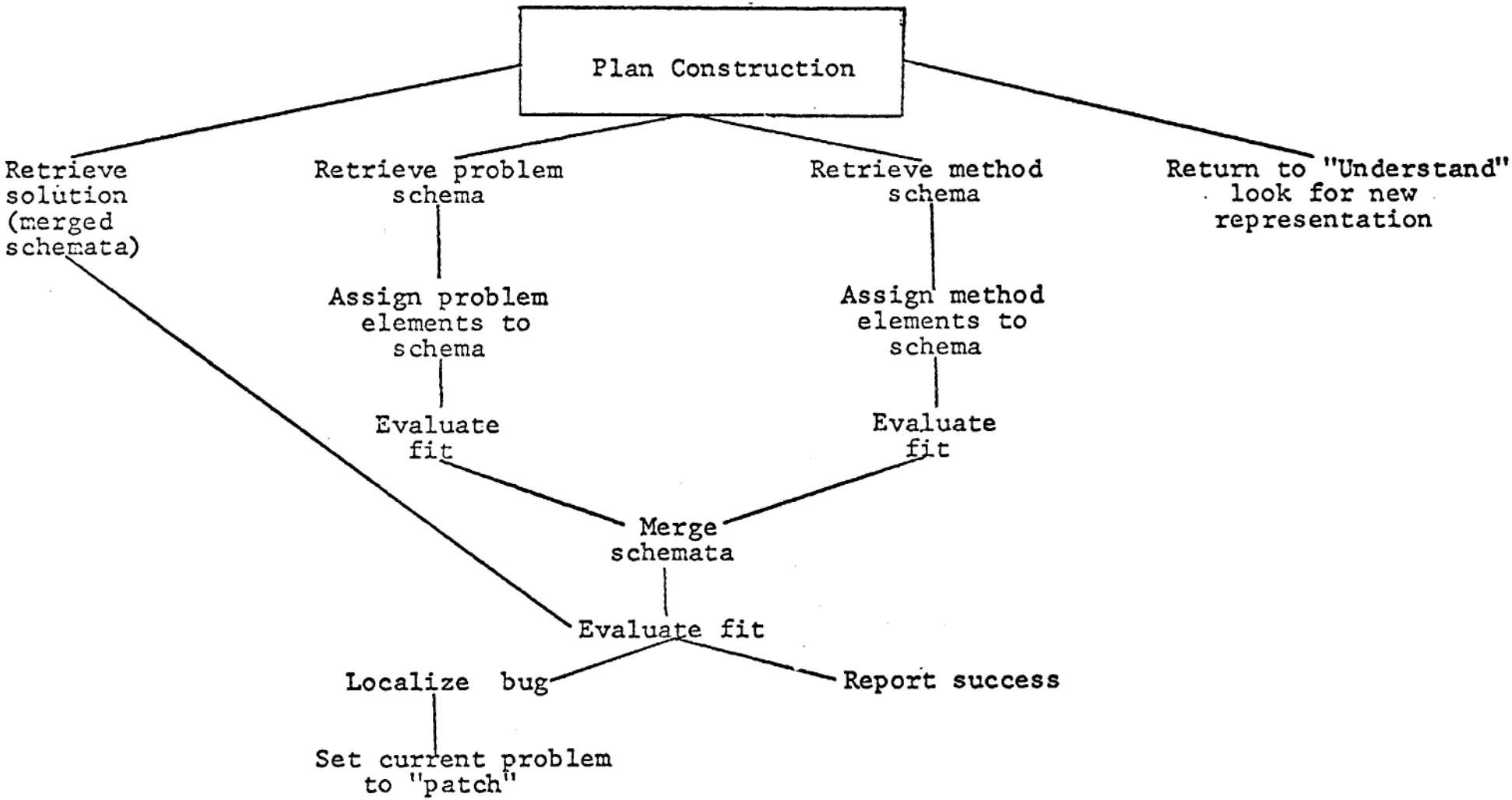
Plan Construction
Retrieve solution (merged schemata)
Retrieve problem schema
Retrieve method schema
Return to "Understand" look for new representation
Assign problem elements to schema
Assign method elements to schema
Evaluate fit
Evaluate fit
Merge schemata
Evaluate fit
Localize bug
Report success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

TABLE 3

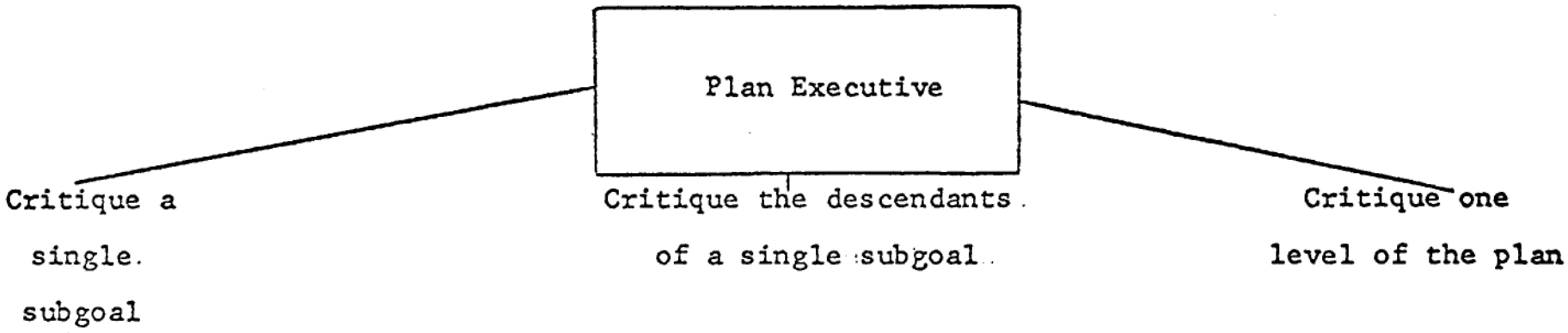
Plan Executive
Critique a single subgoal
Critique the descendants of a single subgoal
Critique one level of the plan

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.