

**Control for Intelligent Tutoring Systems:  
A Comparison of Blackboard Architectures and  
Discourse Management Networks**

William R. Murray

September 1988

**AD-A201 534**

**DTIC  
SELECTED  
OCT 27 1988  
S E D**

This report was prepared under the Navy Manpower, Personnel and Training R&D Program of the Office of the Chief of Naval Research Under Contract Number N00014-86-C-0487.

Reproduction in whole or part is permitted for any purpose of the United States Government. Approved for public release, distribution unlimited.

**Central Engineering Laboratories  
Santa Clara California**

**88 1027 030**

**Control for Intelligent Tutoring Systems:  
A Comparison of Blackboard Architectures and  
Discourse Management Networks**

**William R. Murray**

**September 1988**

**This report was prepared under the Navy Manpower, Personnel and Training R&D Program of the Office of the Chief of Naval Research Under Contract Number N00014-86-C-0487.**

**Reproduction in whole or part is permitted for any purpose of the United States Government. Approved for public release, distribution unlimited.**

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED 1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY 2b DECLASSIFICATION/DOWNGRADING SCHEDULE 3 DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

4 PERFORMING ORGANIZATION REPORT NUMBER(S) R-6267 5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION CENTRAL ENGINEERING LABORATORIES 6b OFFICE SYMBOL (if applicable) 7a NAME OF MONITORING ORGANIZATION OFFICE OF NAVAL RESEARCH - COGNITIVE SCIENCE PROGRAM

6c ADDRESS (City, State, and ZIP Code) FMC CORPORATION 1205 COLEMAN AVENUE SANTA CLARA, CA 95050 7b ADDRESS (City, State, and ZIP Code) OFFICE OF NAVAL RESEARCH (Code 1142PT) 800 NORTH QUINCY STREET ARLINGTON, VA 22217-5000

8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (if applicable) 222 9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-C-0487

8c ADDRESS (City, State, and ZIP Code) 800 NORTH QUINCY STREET ARLINGTON, VA 22217-5000 10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62233N PROJECT NO. RM33M20 TASK NO. WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification) CONTROL FOR INTELLIGENT TUTORING SYSTEMS: A COMPARISON OF BLACKBOARD ARCHITECTURES AND DISCOURSE MANAGEMENT NETWORKS

12 PERSONAL AUTHOR(S) WILLIAM R. MURRAY

13a TYPE OF REPORT TECHNICAL 13b TIME COVERED FROM TO 14 DATE OF REPORT (Year, Month, Day) 1988, SEPTEMBER, 16 15 PAGE COUNT 26

16 SUPPLEMENTARY NOTATION Supported by the Office of the Chief of Naval Research Manpower, Personnel, and Training R&D Program.

17 COSATI CODES FIELD GROUP SUB-GROUP 18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number) INTELLIGENT TUTORING SYSTEMS, INSTRUCTIONAL PLANNING, DYNAMIC PLANNING, BLACKBOARD ARCHITECTURE, DISCOURSE MANAGEMENT NETWORK, TRAINING DEVICES.

19 ABSTRACT (Continue on reverse if necessary and identify by block number) An intelligent tutoring system must have some control mechanism for deciding what instructional action to perform next. Traditional CAI systems have inflexible control mechanisms that procedurally encode tutorial strategies and prevent them from being readily extended to handle new domains or tutorial strategies. Two alternative architectures proposed recently to address these problems are discourse management networks and blackboards. This paper compares these two architectures and argues that an intelligent tutoring system controlled by a blackboard architecture can deliver significantly more effective and flexible instruction than one controlled by a discourse management network.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED UNLIMITED SAME AS RPT DTIC USERS 21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL DR. SUSAN CHIPMAN 22b TELEPHONE (Include Area Code) (202) 696-4318 22c. OFFICE SYMBOL ONR 1142CS

The key components of each architecture are compared and related to the functionality that results. The key components of the blackboard architecture are a global database, independent knowledge sources, and agenda control. In contrast, discourse management networks are much simpler, resembling finite state machines or augmented transition networks. Discourse management networks are characterized by explicit tutorial states and a fixed control mechanism that handles transitions between states. They are well-suited for supporting opportunistic tutoring strategies, but they must be coupled with other control mechanisms to support lesson planning, discourse that requires dialog planning, and mixed-initiative interaction where student actions can invalidate assumptions underlying the current discourse plan. Blackboard architectures provide more flexibility in handling discourse situations and for the integration of lesson planning and discourse planning. Furthermore, blackboard architectures more easily support multiple tutorial strategies and the development of customized, globally coherent curriculum plans and lesson plans. Both architectures explicitly represent tutorial strategies apart from domain knowledge to simplify the construction of tutors for new domains. However, the blackboard architecture has the additional advantage that it facilitates the separation of knowledge about planning from both domain knowledge and tutorial strategies.

These claims are supported by considering an architectural formalization of each architecture and the way in which each architecture's key components facilitate or hamper the dynamic planning of instruction. The relationship between this ability to plan instruction dynamically and to generate effective and flexible instruction is also supported, justifying the utility of this kind of comparison. To ground these discussions, two examples of each architecture are provided: GUIDON and MENO-TUTOR exemplify discourse planners implemented as discourse management networks while IDE-INTERPRETER and the BLACKBOARD INSTRUCTIONAL PLANNER exemplify planners implemented in the blackboard architecture. Application of each architecture to control an intelligent tutoring system to teach LISP is also considered to further illustrate differences in capability that result from choice of architecture.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



# Control for Intelligent Tutoring Systems: A Comparison of Blackboard Architectures and Discourse Management Networks

William R. Murray

Artificial Intelligence Center  
FMC Corporation  
1205 Coleman Avenue, Box 580  
Santa Clara, CA. 95052

## Abstract

An intelligent tutoring system must have some control mechanism for deciding what instructional action to perform next. Traditional CAI systems have inflexible control mechanisms that procedurally encode tutorial strategies and prevent them from being readily extended to handle new domains or tutorial strategies. Two alternative architectures proposed recently to address these problems are discourse management networks and blackboards. This paper compares these two architectures and argues that an intelligent tutoring system controlled by a blackboard architecture can deliver significantly more effective and flexible instruction than one controlled by a discourse management network.

The key components of each architecture are compared and related to the functionality that results. The key components of the blackboard architecture are a global database, independent knowledge sources, and agenda control. In contrast, discourse management networks are much simpler, resembling finite state machines or augmented transition networks. Discourse management networks are characterized by explicit tutorial states and a fixed control mechanism that handles transitions between states. They are well-suited for supporting opportunistic tutoring strategies, but they must be coupled with other control mechanisms to support lesson planning, discourse that requires dialog planning, and mixed-initiative interaction where student actions can invalidate assumptions underlying the current discourse plan. Blackboard architectures provide more flexibility in handling discourse situations and for the integration of lesson planning and discourse planning. Furthermore, blackboard architectures more easily support multiple tutorial strategies and the development of customized, globally coherent curriculum plans and lesson plans. Both architectures explicitly represent tutorial strategies apart from domain knowledge to simplify the construction of tutors for new domains. However, the blackboard architecture has the additional advantage that it facilitates the separation of knowledge about planning from both domain knowledge and tutorial strategies.

These claims are supported by considering an architectural formalization of each architecture and the way in which each architecture's key components facilitate or hamper the dynamic planning of instruction. The relationship between this ability to plan instruction dynamically and to generate effective and flexible instruction is also supported, justifying the utility of this kind of comparison. To ground these discussions, two examples of each architecture are provided: GUIDON and MENO-

TUTOR exemplify discourse planners implemented as discourse management networks while IDE-INTERPRETER and the BLACKBOARD INSTRUCTIONAL PLANNER exemplify planners implemented in the blackboard architecture. Application of each architecture to control an intelligent tutoring system to teach LISP is also considered to further illustrate differences in capability that result from choice of architecture.

## Table of Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Importance of Dynamic Instructional Planning in Intelligent Tutoring Systems	1
1.2. Types of Instructional Planning	3
<b>2. Discourse Management Networks</b>	<b>4</b>
2.1. Architectural Formalization	4
2.2. Two Exemplars	5
2.3. Support for Dynamic Instructional Planning	8
<b>3. The Blackboard Architecture</b>	<b>9</b>
3.1. Architectural Formalization	10
3.2. Two Exemplars	10
3.3. Support for Dynamic Instructional Planning	12
<b>4. Scenario: Controlling a Programming Language Tutor</b>	<b>14</b>
4.1. Scenario 1: Dynamic Instructional Planning in the Blackboard Architecture	14
4.2. Scenario 2: Opportunistic Tutoring with a Discourse Management Network	15
<b>5. Summary</b>	<b>17</b>

**List of Figures**

<b>Figure 2-1:</b>	<b>The discourse management network architecture</b>	<b>5</b>
<b>Figure 2-2:</b>	<b>MENO-TUTOR's Discourse Management Network</b>	<b>6</b>
<b>Figure 2-3:</b>	<b>Student program in MENO-TUTOR dialog</b>	<b>7</b>
<b>Figure 2-4:</b>	<b>GUIDON's Discourse Management Network</b>	<b>8</b>
<b>Figure 3-1:</b>	<b>Blackboard Interpretation Cycle</b>	<b>10</b>
<b>Figure 3-2:</b>	<b>Domain and Control Blackboards of the BLACKBOARD INSTRUCTIONAL PLANNER</b>	<b>11</b>
<b>Figure 3-3:</b>	<b>Categories of instructional actions in the BLACKBOARD INSTRUCTIONAL PLANNER</b>	<b>11</b>

## 1. Introduction

The research discussed in this paper addresses the problem of control for intelligent tutoring systems. At any point during instruction a tutoring system must select from many possible instructional actions. For example, topics can be introduced, reviewed, motivated, summarized, explained in depth, related to earlier topics, etc. Similarly, student knowledge can be assessed by true/false or multiple choice tests, direct questioning, self-assessment, and many other means. The problem of control is to select the most effective action, given the current lesson objectives, student model, tutorial strategy, subject matter, and resources available.

An important decision in designing an intelligent tutoring system is the choice of architecture for this control mechanism. Traditional CAI systems have inflexible control mechanisms that procedurally encode tutorial strategies, and which prevent them from being readily extended to handle new domains or tutorial strategies. Two alternatives proposed recently to address these problems are discourse management networks and blackboard architectures. This paper compares the advantages and disadvantages of using these two architectures for control in intelligent tutoring systems. Based on this comparison, we argue that blackboards address important limitations of discourse management networks that limit their ability to deliver customized, globally coherent, planned instruction, and to support more flexible mixed-initiative instruction.

The framework for comparison is the ability of each architecture to support the dynamic planning of instruction. *Dynamic instructional planning* is a planner-based approach to control for intelligent tutoring systems in which appropriate plan generation<sup>1</sup> and revision occur *during* the tutorial session and in response to the changing tutorial situation. A dynamic instructional planner reasons about alternative lesson objectives, and alternative instructional plans to realize the tutorial strategies it selects. We argue that such planning is necessary to build effective, robust, and flexible tutoring systems and that blackboard architectures provide better support for this planning than discourse management networks. Before comparing the two architectures, we first justify the relevance and importance of dynamic instructional planning to intelligent tutoring systems.

### 1.1. Importance of Dynamic Instructional Planning in Intelligent Tutoring Systems

Why should tutoring systems incorporate dynamic instructional planning? Two points must be supported here: first, that a tutoring system should have a plan; and second, that it should be able to plan. It should have a plan to properly manage its time and generate globally coherent instruction. Proper time management prevents spending too much time on relatively unimportant topics or packing too many topics or exercises into one lesson. Global coherence means that topics are logically connected to support instructional objectives and that topics are sequenced and presented in a manner sensitive to the student's perceived knowledge, the tutor's instructional objectives, the student's interest

---

<sup>1</sup>For the purposes of this paper we allow plan selection to be considered a trivial form of plan generation, otherwise discourse management networks would not support dynamic instructional planning merely by their inability to assemble a plan during instruction.

and motivation, and the time available for lessons. Transitions between topics are smooth and the tutor is consistent; that is, it does not contradict previous or future instruction. One sign of incoherent instruction is recurring student confusion about the tutor and its intentions, resulting in distractions from the subject matter being taught.

Another advantage of having a plan is the ability to introduce material in a layered fashion, first providing an overview of the material and lesson plan, and then introducing successive layers of detail that build on and refine earlier material. A plan also allows the tutor to recognize opportunities to motivate and lead into future instruction in response to unexpected student questions, or to defer answering the questions since they will be addressed later in the lesson. The tutor can use a plan to select a sequence of problem-solving cases that cover a set of topics, and to focus its conversation on the central topics for each case, deferring or abbreviating discussion of other topics best addressed in subsequent cases. Finally, a plan assists in explaining and motivating current instruction based on its relationship to future instruction.

One alternative to dynamic instructional planning is for the tutor simply to interpret a highly detailed plan provided by a human curriculum author; this approach is used in traditional computer-assisted instruction (CAI). Although *theoretically* the plan could provide for any eventuality that might arise, *practically* the combinatorics of different student responses and tutorial states require that student initiative be curtailed, fine-grained student modeling avoided, and a significant amount of time spent preparing lesson plans. One goal of intelligent tutoring systems is to explicitly represent the tutorial strategies that are encoded procedurally in such a system, allowing for more rapid development of tutors for new instructional domains, and experimentation with alternate strategies.<sup>2</sup> By providing intelligent tutoring systems with a planning capability, their reliance on human-generated plans is reduced while their ability to generate the high-quality expository instruction of well-crafted CAI systems is increased.

A tutor that can plan is also better able to handle a mixed-initiative dialog than a tutor that cannot plan. Replanning can be used to handle topic transitions brought about by student questions and to revise lesson plans to omit or cover topics to satisfy student requests. Again, the combinatorics of the different possible student requests, questions, and tutorial states argue for representing the knowledge for handling student initiative in planning knowledge rather than as different plan contingencies in a pre-stored plan. This planning knowledge allows the tutor to plan topic transitions, to replan in response to the amount of time remaining in the lesson, and to decide how to return to the current topic or whether to abandon it altogether.

Planning knowledge is used not only to customize the lesson plan in response to student requests during the lesson, but also before the lesson begins. A tutor that can plan can generate an initial lesson

---

<sup>2</sup>Intelligent tutoring systems also differ from CAI systems by their emphasis on sophisticated student modeling, the tailoring of instruction to the student model, and the ability to solve problems, model expertise, and answer questions about the domain they are teaching.

plan customized to the student's interests, assessed capabilities, and background. The lesson plan can be revised as the tutor's student model changes or in response to student requests. Instead of the tutor assuming a fixed set of instructional objectives for all students, the student can request instruction on particular areas of a subject. As before, it is impractical to anticipate the number of different student interests, objectives, and backgrounds in teaching complex subjects. It is more economical to represent lesson planning knowledge than to provide lesson plans for all the different possible combinations. The representation is more economical in the sense that the same knowledge is represented explicitly in a concise form and not replicated implicitly in the branches of a single highly conditionalized lesson plan, or as a very large number of lesson plans unique to different situations.

These arguments for the tutor being able to plan rest on a subtle point that is worth reemphasizing. Simple plan-following tutors (CAI) can always replicate the behavior of a dynamic instructional planner for any particular situation. However, the dynamic instructional planner is better able to handle the combinatorial explosion of different tutorial situations when mixed-initiative instruction is allowed, and where lesson plans are tailored to different student models and to varying amounts of time for each lesson. The CAI system limits mixed-initiative dialog and lesson customization, and omits or simplifies the student model to reduce the number of different tutorial situations. The dynamic instructional planner imposes fewer limitations since it can apply its planning knowledge to the different tutorial situations that arise.

The economic representation of planning knowledge in a dynamic instructional planner also contributes to its greater ease of use than that of a CAI system. The CAI system procedurally encodes planning knowledge and implicit assumptions about the tutorial state at each branch of its pre-stored instructional plan. The planning knowledge in a dynamic instructional planner makes fewer assumptions about the tutorial situation and can thus be more readily applied in the construction of new tutoring systems.

## 1.2. Types of Instructional Planning

We may distinguish three levels of instructional planning:

- *Curriculum planning* - planning an extended sequence of lessons for a subject.
- *Lesson planning* - determining the subject matter to present in a single lesson, and the order of presentation.
- *Discourse planning* - planning communicative actions between the tutor and student within a lesson.

These levels cannot in practice be so cleanly separated and frequently discourse planning and lesson planning are intertwined, as are lesson planning and curriculum planning. Typically, a human instructor's lesson plan will include not only a sequence of topics but also some common discourse procedures such as collecting homework or having students solve and explain problems on the board. This three-level distinction is useful here since discourse management networks support primarily

discourse planning<sup>3</sup> while the blackboard architecture supports all three kinds.

The remainder of this paper compares discourse management networks and blackboards with respect to their support for dynamic instructional planning. Section 2 formalizes the discourse management network architecture, provides two examples, and then discusses this architecture's support for dynamic instructional planning. Similarly, Section 3 formalizes the blackboard architecture, provides two examples, and then discusses support for dynamic instructional planning. Next, a specific application, control of a LISP tutor, is considered in Section 4 to illustrate the advantages of the blackboard architecture. The last section summarizes these arguments.

## 2. Discourse Management Networks

Discourse management is the selection by the tutor of the next discourse action or intended sequence of actions. In this architecture, the tutor does not reason about the results of discourse actions or future tutorial situations; the only planning performed is plan selection or skeletal planning. However, planning is not a prerequisite for flexible behavior [10] and tutors built with this architecture allow sophisticated control of tutorial dialogs.

A discourse management network (DMN) is a kind of procedural network similar to an augmented transition network. Nodes represent tutorial procedures or actions corresponding to tutorial states. Arcs represent state transitions. The tutor is in one state at any time. The state determines what actions are performed. After performing the actions or procedures, control follows one of the arcs leaving from the state. The choice of arc, if there is more than one, depends on the predicates on the arcs.

### 2.1. Architectural Formalization

Figure 2-1 represents the key features of the discourse management network architecture. Circles represent tutorial states named  $S_0 \dots S_n$  and  $S_{\text{default}}$ . Arrows represent possible transitions to other states, not all of which are shown. Assume the tutor is originally in state  $S_0$ . Then  $A_0$  represents the actions that will be performed in state  $S_0$ . The heavy line represents a default transition to the state  $S_{\text{default}}$ . Other lines from  $S_0$  represent other possible transitions that can be taken. Each predicate  $P_1 \dots P_n$  is considered in turn and the first true predicate causes control to transfer to that state. If no predicate  $P_1 \dots P_n$  is true then the default transition is taken. The discourse management network has no memory other than the current state and the registers  $R_1 \dots R_m$ . These registers can be set to scalar or symbolic values by any of the actions and accessed in any of the predicates. Actions can only set registers or perform communicative actions (e.g., give a test). Actions cannot change control in any way other than that described above. For example, an action cannot change the current state or the arcs emanating from a state.

---

<sup>3</sup>Planning in the sense of choosing a course of action, not in the sense of reasoning about the results of actions and projecting future discourse situations.

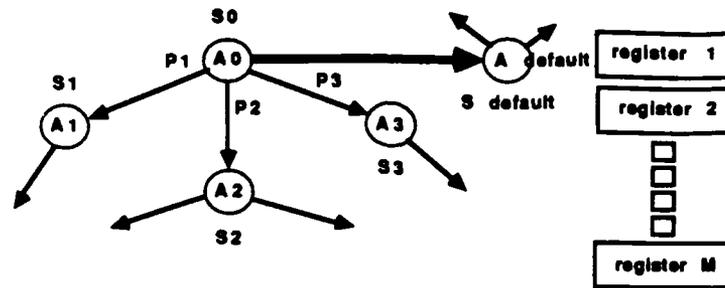


Figure 2-1: The discourse management network architecture

As formalized here, discourse management networks are a kind of finite state machine with registers used to reduce the number of states. Usually discourse management networks are coupled with other control mechanisms (e.g., an agenda) and an external memory to provide a topic selection mechanism. Two examples of discourse management networks are discussed below, along with extensions they make to the abstract model above.

## 2.2. Two Exemplars

MENO-TUTOR [12] uses a discourse management network to control a Socratic question and answer dialog with a student that alternately probes the student's knowledge and refines his knowledge by teaching new information or correcting misconceptions. Figure 2-2 [13] shows the state transition diagram of MENO-TUTOR. Numbered states indicate a possible sequence of states that might occur in a dialog. It differs from the abstract model above in its use of hierarchical abstraction: tutorial states at the bottom of the diagram are refinements of the tutorial states above them. In this diagram only possible transitions from more abstract to less abstract states are shown; default transitions are not shown. The actions of a particular state result in the generation of a single utterance by a surface natural language generator.

MENO-TUTOR also differs from the idealized model in its use of meta-rules to represent state transitions other than to default states. Rather than explicitly represent these arcs and their predicates, the meta-rules operate as demons that can override default transitions. This movement to a new state from a specified prior state happens only when a meta-rule's trigger condition is true. Since the meta-rule can be replaced by arcs to the new state with predicates using the meta-rule's trigger condition, the meta-rules provide only a more compact representation but no added functionality compared to the abstract model above.

MENO-TUTOR was reverse-engineered to produce Socratic question and answer dialogs similar to those obtained from human protocols. One such dialog that MENO-TUTOR could produce, if coupled to a surface language generator, is shown below (these examples are from [13]). The Pascal program being referred to is shown in Figure 2-3.

Tutor: Do you know that GRADES, in line 9, is a control variable for the WHILE loop?

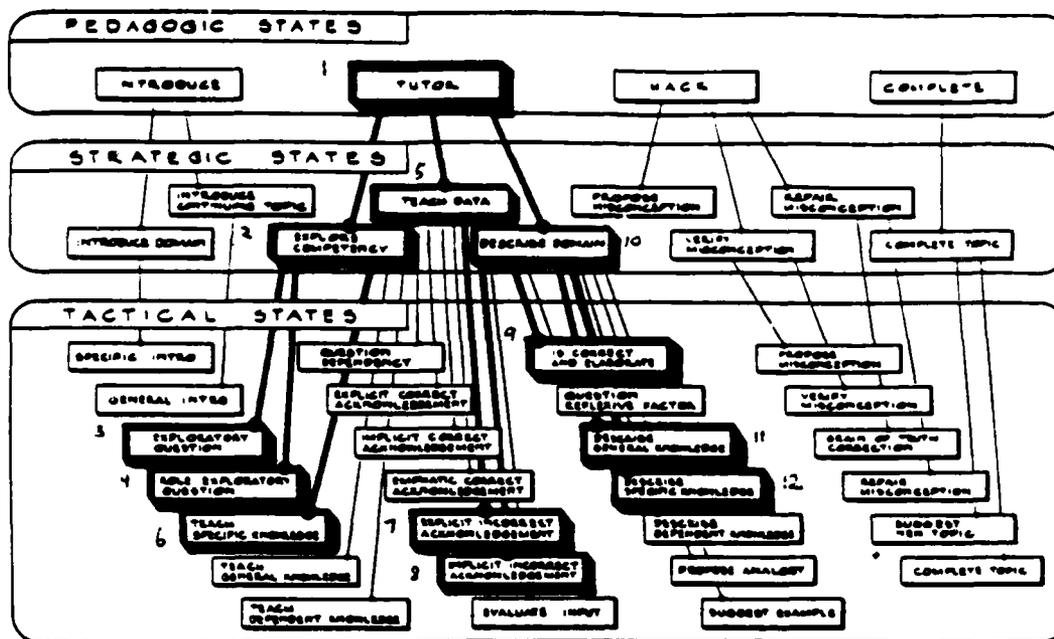


Figure 2-2: MENO-TUTOR's Discourse Management Network

Student: Yes

Tutor: Does that mean that GRADES should be

- Initiated before the loop?
- Added into SUM within the loop?
- Modified within the loop?
- Both (a) and (c)

Student: C

Tutor: You seem to understand about the control variable of the WHILE loop. Now let's talk about some other command statements.

What do you think READ (GRADES) in line 8 does with the value of GRADES?

- Reads all values of GRADES at one time.
- Reads in a single value of GRADES.
- Assigns several new input values to GRADES.

Student: A

Tutor: Then do you think that  $GRADES := GRADES + 1$  updates the next of those values?

Student: Yes

Tutor: There is your misconception. Adding one to a variable is a common way to step through a range of values. But it cannot apply here because your READ instruction in line 8 retrieves only a single value. You should replace line 13 with a second call to READ (GRADES).

```

1 PROGRAM LESSON1 (INPUT, OUTPUT);
2 VAR
3   SUM, GRADES, STUDENTS: INTEGER;
4   MEDIAN: REAL;
5 BEGIN
6   SUM:=0;
7   STUDENTS:=0;
8   READ (GRADES);
9   WHILE GRADES>0 DO
10    BEGIN
11     SUM:=SUM + GRADES;
12     STUDENTS:=STUDENTS+1;
13     GRADES:=GRADES+1;
           should be READ(GRADES);
14    END;
15   MEDIAN:=SUM/STUDENTS;
16   WRITELN
17   ('THE MEDIAN GRADE IS', MEDIAN:8:3)
18 END

```

Figure 2-3: Student program in MENO-TUTOR dialog

GUIDON [2] extends the utility of the discourse management network formalism by introducing quite sophisticated tutorial actions. Instead of simple actions, such as suggesting a new topic, a packet of tutorial rules (t-rules) can be invoked. Such a collection of t-rules is called a discourse procedure, and t-rules themselves can call other discourse procedures. The result is that most of the selection of discourse actions occurs as a result of the operation of the t-rules within a state, and the discourse management structure is more of a convenient means of organizing discourse procedures, rather than the primary control resulting in the selection of discourse actions.

The domain of discourse for which GUIDON was developed is a MYCIN consultation. A medical student using GUIDON attempts to learn or refine his understanding of the knowledge represented by MYCIN's diagnostic rules. Discourse topics include MYCIN rules, goals, clinical data, and student hypotheses. Discourse procedures are organized around these topics as shown in Figure 2-4 ([3], page 59). Selection of the next discourse procedure is also performed within each state, thus the predicates on the arcs are implicit in the t-rules that determine the next discourse procedure to select.

The point here is not that GUIDON's enhancements alter the DMN architecture or that GUIDON does not handle discourse in a sophisticated way (since it does), but that GUIDON's power comes from its use of these discourse procedures and t-rules and *not* from the underlying discourse network architecture. The discourse management network architecture - the states, arcs, and predicates - is not where most of the decision-making is performed regarding what discourse action to take, rather this is performed by the numerous t-rules in each state.

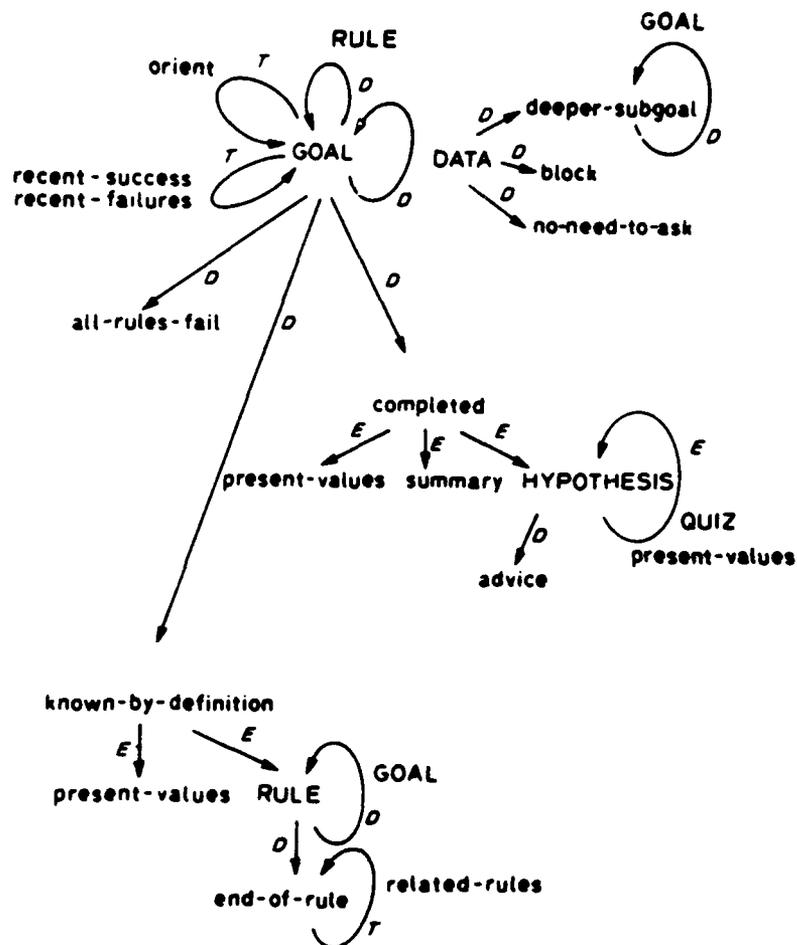


Figure 2-4: GUIDON's Discourse Management Network

### 2.3. Support for Dynamic Instructional Planning

The main feature that discourse management networks lack to support dynamic instructional planning is an explicit plan representation that can be generated and interpreted *during* instruction. The lack of such a representation hampers:

- *Incremental planning* - since there is no explicit representation of partially specified subplans or operators, or representation of constraints on those parts of the plan that have not been elaborated to the level of primitives.
- *Plan repair or improvement by revising existing plans* - since there is no plan data structure that can be modified and then subsequently interpreted.
- *Gracefully suspending, resuming, or abandoning plans* - since suspension and resumption of *instructional* plans requires more than placing a plan on a stack and then popping it off later. Instead a coherent line of discourse must be maintained in suspending or abandoning a plan and moving to an entirely new or resumed plan. Topic transitions may need to be planned or an explanation generated for why the current line of instruction is now changing.

Sophisticated dialog management in GUIDON and MENO-TUTOR is still maintained, essentially by dynamically selecting from pre-stored discourse plans.<sup>4</sup> This dynamic selection allows reactive discourse that responds to student questions and requests, but still does not provide the flexibility and global coherence that dynamic instructional planning could provide. Two examples from GUIDON will illustrate limitations that can be addressed by dynamic instructional planning. First, note that GUIDON's discussions can be verbose and lacking global direction since GUIDON does no planning before the dialog begins. Instead, GUIDON considers only one level of MYCIN's AND-OR goal tree at any time when controlling its dialog. Clancey [3] discusses two deficiencies that became apparent when GUIDON was applied to the nonmedical domain of SACON (structural analysis):

First, better overviews are required. A SACON dialogue is tedious because GUIDON methodically guides the dialogue in a depth-first way, only motivating the discussion one step at a time. To convey a global sense of purpose, the program must reason about the relation between subgoals (e.g., distinguish between definitions and hierarchical subtype) and summarize subtrees.

Second, the constraint on time for the session prevents treating each "context" (e.g., culture, loading, organism) with equal emphasis; some contexts must be omitted or mentioned in passing. While GUIDON has a number of methods for reducing the number of rules that are mentioned and has methods for mentioning them economically, it never looks down into the solution, beyond one level of the tree, to decide what topics to focus on. ([3], pages 230 - 231)

GUIDON is also limited in its ability to tolerate student interruption since it lacks dynamic instructional planning. Although students can interrupt, students cannot receive new case data during the interruption since the current (suspended) discourse procedure or tutorial rule may need to be abandoned or modified to account for the student's new knowledge. GUIDON cannot do this and so attempts to avoid situations where this problem might arise:

Exchange of initiative in the tutorial involves complex interactions with the knowledge models and discourse procedures, given that student interruptions can disrupt a plan at any time. We minimize these interactions in GUIDON by taking the initiative for just short sequences; long lectures or explanations are more prone to student interruptions that overturn the teacher's plans. When it makes a difference, as during hypothesis evaluation, the program preserves the current dialogue context, and hence the logic behind its initiative, by not giving new case data during a student interruption. In a more flexible tutor, modeling and tutoring would occur in tandem. Specifically, in presentations of any considerable length, like hypothesis evaluation, it is desirable for the tutor to periodically stop, listen, and reevaluate its present course. ([3], pages 89 - 90)

It is precisely this ability to replan dynamically that would allow dynamic instructional planning to provide greater flexibility in handling student interrupts than that provided by GUIDON.

### 3. The Blackboard Architecture

This section considers the blackboard architecture: its architectural formalization, two examples of planners implemented in that architecture, and then its support for dynamic instructional planning.

---

<sup>4</sup>These pre-stored discourse plans are represented in GUIDON as discourse procedures. In MENO-TUTOR, they are sequences of states connected by default transitions.

### 3.1. Architectural Formalization

The distinguishing features of a blackboard architecture are:

1. *Hierarchically structured global database* - Solution elements to the problem being solved are posted here. The hierarchical structure of the blackboard facilitates problem abstraction and the efficient triggering of knowledge sources.
2. *Independent knowledge sources* - These are rules triggered by changes to the blackboard whose actions contribute to an evolving solution by adding or modifying solution elements on the blackboard. Knowledge sources communicate only by adding or changing the contents of the blackboard.
3. *Agenda control* - An agenda of possible actions to perform is maintained. Activation records created by the triggering of knowledge sources are added to the agenda. A scheduler selects the next action to execute from the agenda.

Figure 3-1 depicts the standard blackboard interpretation cycle. Knowledge sources are triggered by changes to the blackboard. This causes new activation records to be added to the agenda, indicating possible actions corresponding to the knowledge sources. The scheduler selects the next action to perform. Its execution results in additional changes to the blackboard, possibly triggering new knowledge sources.

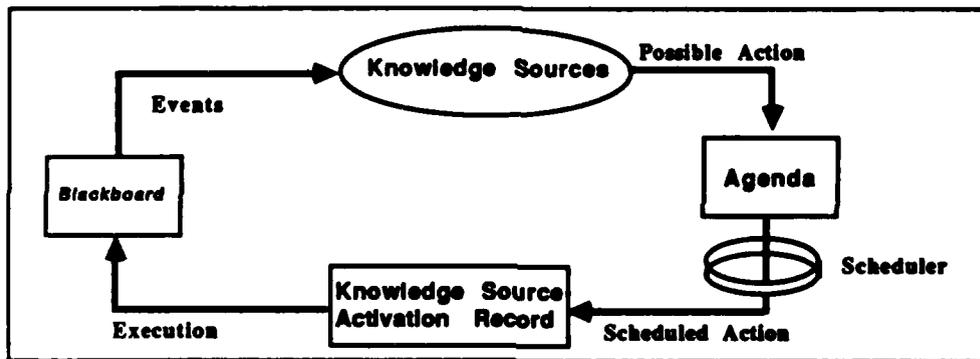


Figure 3-1: Blackboard Interpretation Cycle

### 3.2. Two Exemplars

The BLACKBOARD INSTRUCTIONAL PLANNER [6] (abbreviated here as BB-IP) best illustrates the direct application of a blackboard architecture to construct a dynamic instructional planner. Conceptually, the problem being solved - a lesson plan<sup>5</sup> being incrementally constructed - is represented on a *domain blackboard*. A second blackboard, called the *control blackboard*, represents decisions about how to construct and modify this plan. The lesson plan itself represents decisions about what instructional actions to take now and in the future. These blackboards are shown in Figure 3-2. There are three levels in the domain blackboard. The highest level represents a partial ordering of lesson objectives, concepts, or skills to teach. The second level represents tutorial discourse plans. Each plan consists of one or more steps that are in turn represented on the third level. Each step

<sup>5</sup>Part of the lesson plan is a current discourse plan.

influences the scheduler to select instructional actions from one particular category of instructional actions. For example, an instructional plan could first bias the scheduler to select instructional actions to motivate a topic, then actions to provide an overview of the topic, then actions to present the topic in detail, and finally actions to assess the student's understanding of the actions. There are several types of actions to present or assess a topic, including presentation of graphics or specific kinds of tests such as a true-false test. The categories of instructional actions are shown in Figure 3-3. Plan steps can direct the scheduler to favor actions from any one of these categories. The next step in a plan is selected once the current step's goal is satisfied. The goal might be that all such actions on the agenda have been executed or that the tutor's belief that the student is likely to know a lesson objective has exceeded some threshold. The control blackboard sets up default preferences to favor particular kinds of tutorial discourse plans (e.g., those that always assess a student's knowledge of a topic before discussing it). These initial preferences are altered during the tutorial session by the action of knowledge sources that monitor time remaining, the efficiency of instructional plans, student questions, and student requests.

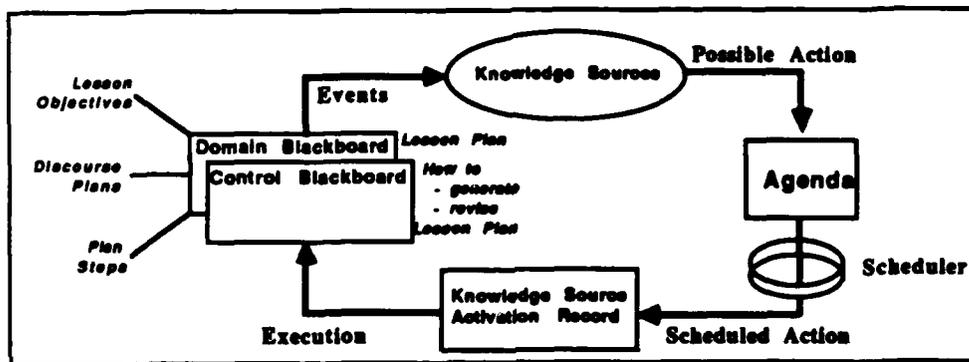


Figure 3-2: Domain and Control Blackboards of the BLACKBOARD INSTRUCTIONAL PLANNER

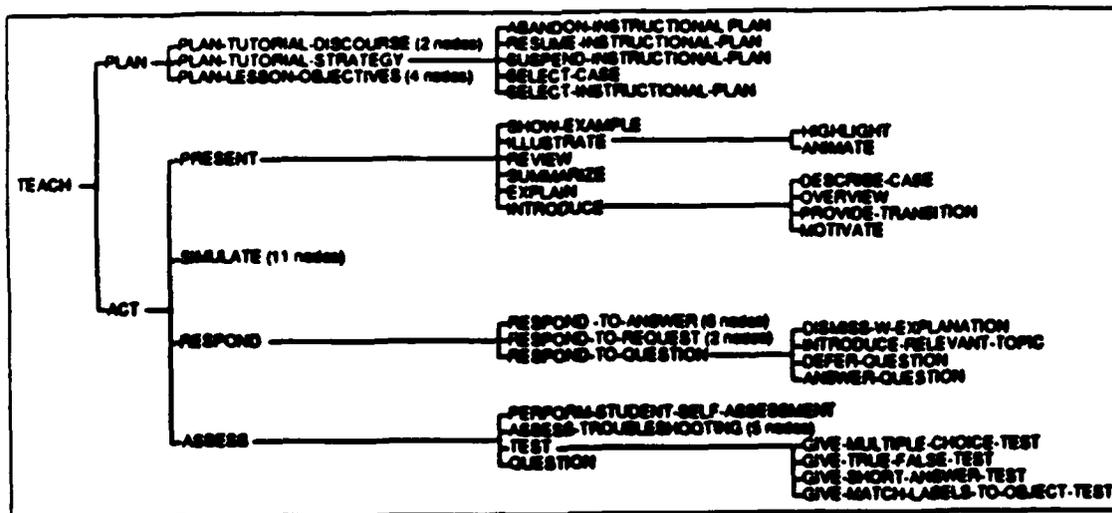


Figure 3-3: Categories of instructional actions in the BLACKBOARD INSTRUCTIONAL PLANNER

IDE-INTERPRETER [8] is a second example of an instructional planner implemented in a blackboard architecture. The planner is not built over a general-purpose blackboard system, as BB-IP is

built over BB1, but the planner's architecture contains the three key components of the blackboard architecture. There is an explicit representation of the lesson plan at various levels of abstraction using a tree data structure (first key component). Changes to the tree trigger rules that correspond to the knowledge sources of the blackboard architecture (second key component). The actions of these knowledge sources are to further refine parts of the lesson plan. Each rule is encapsulated in a record as a possible task to be performed on an agenda (third key component - agenda control). A scheduler, *consisting of a collection of heuristics*, selects the next goal expansion task. Terminal nodes in the tree correspond to instructional units that are directly executable self-contained instructional procedures. Typically these are associated with the presentation or assessment of a particular topic. Thus this planner emphasizes lesson planning over discourse planning since most specific discourse actions are procedurally encoded in the instructional units and not reasoned about by IDE-INTERPRETER. BB-IP differs in its equal emphasis on both lesson and discourse planning, as illustrated by its reasoning about more fine-grained discourse actions along with actions that refine or sequence lesson objectives.

### 3.3. Support for Dynamic Instructional Planning

Each of the three key architectural features of the blackboard architecture supports dynamic instructional planning. The hierarchically structured global database supports an explicit hierarchically structured plan representation. This in turn supports the following functionality that facilitates dynamic instructional planning:

- *Planning at multiple levels of abstraction* - this plan abstraction reduces the search space for an effective instructional plan and facilitates the separation of those aspects of the instructional plan related to curriculum planning, lesson planning, and discourse planning.
- *Incremental planning* - the explicit plan representation allows the representation of those parts of the plan that have not yet been refined and any constraints on partially refined plans. This allows the planner to avoid premature commitment while retaining constraints that ensure plan consistency.
- *Plan repair and plan improvement* - the current plan need not be discarded entirely if some part of it fails, instead it can be patched to retain the remainder of the plan. If the plan is not failing but an unexpected opportunity to improve it arises - perhaps a student question leading into planned instruction - then the plan can also be revised accordingly. Without the ability to modify only parts of the plan either the entire plan must be discarded and another selected, or the curriculum designer must attempt to anticipate all possible plan failures or opportunities for improvement - and again the combinatorics argue against the practicality of this approach.
- *Plan justifications* - justifications for plan steps can be recorded and used to explain to the student, or to an instructional designer debugging a tutorial strategy, the reasons for the current instructional actions. These plan step justifications can also be used in replanning. Although plan justifications can be used for explanation in systems that only select but do not assemble, patch, or revise plans, such as GUIDON, those systems cannot use the justifications to replan.

The second key component of the blackboard architecture is the use of independent knowledge sources. This architectural feature supports dynamic planning by separating diverse kinds of knowledge

while allowing their cooperation during the problem solving process, in this case the construction of an instructional plan. The following kinds of knowledge are easily separated with this scheme:

- *Possible instructional actions* - new instructional actions, such as giving a new kind of test or explanation, can be added (or actions can be deleted) without requiring modification of existing tutorial strategies.
- *Tutorial strategies* - new tutorial strategies can be added without modification of the representation of instructional actions.
- *Planning strategies* - different approaches to planning can be added without affecting the library of tutorial strategies or instructional actions.

Although systems based on discourse management networks separate out the first two kinds of knowledge, and thus improve on CAI where both kinds of knowledge are procedurally encoded, they do not separate out the third kind of knowledge. The second kind of knowledge can in turn be separated into different methods for curriculum planning, lesson planning (topic/skill selection and sequencing), strategic discourse planning (for preplanned presentation and assesment of topics), and reactive discourse planning (for handling student questions and requests).

Another disadvantage of DMNs is the lack of independence of knowledge sources when used in that architecture. With the extensions that Clancey has made to the basic DMN architecture, the t-rules and discourse procedures act as knowledge sources that select discourse actions, select future DMN states, and save values in registers. Such knowledge sources are not independent since they must communicate through registers, and there must be protocols for which knowledge sources set which registers and which read them. For example, some of GUIDON's t-rules set registers indicating the value of discussing particular MYCIN domain rules. Other t-rules responsible for presenting the rules must access these values. Clancey notes that:

Interactions of this kind are rare in GUIDON, but they illustrate the general problem of control when the decision about what to do is separated from its execution. The program must maintain a record of possible and/or intended actions, so that after all possibilities have been weighed, the program can return to the best plan and follow it. ... For the moment, we must hand-craft interactions between t-rule packets. ([3], page 80)

The blackboard architecture facilitates solutions to these problems: the agenda maintains a record of possible actions, the global database records an explicit plan representation of intended actions, and knowledge sources are independent and communicate entirely through the blackboard.

The third key component of the blackboard architecture, agenda control, supports dynamic instructional planning by allowing the best instructional action (either an interactive action involving the student or further planning by the tutor) to be selected at any time with respect to possibly conflicting multiple goals. Agenda control supports global planning by allowing multiple goals to be considered simultaneously. It also allows actions to be favored that achieve more than one goal at a time. Although similar results can be obtained in theory by procedurally encoding heuristics that take into account all current goals and available actions, in practice this flexibility is difficult to obtain for complex domains. This difficulty arises from the unmanageable number of different possible sets of goals and actions that must be anticipated.

## 4. Scenario: Controlling a Programming Language Tutor

To illustrate the additional support that the blackboard architecture provides for dynamic instructional planning relative to discourse management networks, we will consider a hypothetical scenario. This scenario provides examples of how dynamic instructional planning in a blackboard architecture provides more flexible and effective instruction than a system based on discourse management networks, that cannot so plan. We consider the operation of the tutor first when controlled by a dynamic instructional planner implemented in a blackboard architecture and then by a discourse management network. To consider a familiar domain, the scenario assumes that the tutor is teaching LISP in a self-paced course. The student is called S and the tutor T below.

### 4.1. Scenario 1: Dynamic Instructional Planning in the Blackboard Architecture

T greets S and explains that the first order of business is to draw up a curriculum plan for the term based on the student's interests and the requirements set by the computer science department. T provides S with a brief questionnaire about his background and interests. S knows Pascal and is interested in learning LISP for use in an AI course he is taking concurrently. He is planning on doing a term project in that class in LISP, and is less interested in the theoretical underpinnings of LISP (e.g., the lambda calculus) than in using LISP for AI programming applications. T draws up a lesson plan given the inferred cognitive stereotype of the student (novice with only one Pascal course), the number of lessons expected in the course (20 lessons, each lasting about an hour to an hour and a half), and the student's interests. A different curriculum plan is drawn up for each student. Another student in a programming languages course might receive a much smaller number of lessons providing only a brief introduction to LISP and concentrating on those unique features that distinguish it from other languages.

A typical lesson between T and S might appear as follows. T briefly reviews material covered in the last lesson and then explains today's lesson plan, which covers mapping functions and iteration. T proceeds to explain the concepts involved, relating them to analogous concepts in Pascal. T points out both the differences and similarities between iteration in Pascal and in LISP. T proceeds to explain DO loops, but is interrupted by a request by S to explore. (The exploration mode of the tutor allows S to perform experiments of his own in the LISP environment.) T suspends its discourse plan to elaborate further on DO and lets S interact with the interpreter and editor. T monitors S during this time. S writes a simple DO to print out integers and their squares. Next S writes another DO but this one loops forever since the exit test never becomes true. T's program analyzer, which has been checking S's DO loops for well-formedness, detects this error. T breaks the infinite loop and explains the problem. The student edits his DO, retries it, is satisfied, tries some other examples, and finally exits the exploratory mode. T has also monitored the time involved and would have reacquired the initiative if too much time had been spent exploring or too little progress appeared to result from the exploration. When T reacquires the initiative, it is satisfied that S has learned those aspects of DO that it was about to explain. Both the further explanation of DO and an assessment of the student's ability to use DO are now obviated by the student's demonstrated capability during the exploratory session. T replans to omit these parts of the lesson and then proceeds to discuss mapping constructs.

We pick up the scenario a few lessons later. Before moving onto more advanced topics, such as object-oriented programming in LISP, T switches to a case-method style of instruction for this particular lesson. Each programming exercise tests some of the skills that S should have acquired and retained. S has only minor problems with the first two exercises but his solution of the third exercise has multiple bugs. (T uses a program analyzer such as TALUS [5] that can detect multiple bugs in student programs; a PROUST-like [4] analyzer is used when T teaches Pascal.) T plans which bugs to address and in what order. It switches to a Socratic question and answer discourse strategy to track down and remediate likely misconceptions about iteration indicated by the most important program bugs. T focuses on those bugs related to iteration, and quickly explains and repairs the more simple remaining bugs for the student. (T could let S repair the bugs but decides that the time is better spent addressing underlying misconceptions.) T switches to expository instruction to review those aspects of iteration the student appeared to have trouble with and then resumes case-method instruction to assess the student's capabilities in other areas covered.

At the end of this assessment and review of basic skills, T explains the student's progress so far and what remains to be learned. The remaining curriculum is reviewed and the student is reminded that it can be revised (within limits) to accommodate his specific interests. The student expresses an interest in covering one of the optional topics (pattern matching in LISP) since he has decided to build a theorem prover for his term project in the AI class he is taking concurrently. Since T cannot delete or compress any of the remaining lessons because they cover material mandated by the computer science department it asks S if it is OK to add a short lesson on this. S agrees, so T inserts into the curriculum plan a lesson on using LISP to write simple pattern matchers. S is now satisfied with the overall curriculum plan, requesting only that more time be spent on working through examples in the future.

Still later in the lesson, T is explaining object-oriented programming by focussing on an application that is used as a running example throughout the discussion. Much less expository instruction is given since more time is spent motivating concepts through examples, and giving the student exercises to solve with the tutor's assistance. At times, S switches to exploratory mode, examines the case library, and requests that T solve one of the cases.<sup>6</sup> When this happens, T tutors opportunistically - it explains new material that arises in the case or material that it believes the student is weak on. It also replans to omit any similar cases that it had intended to cover, unless it appears necessary based on its assessment of the student's understanding of the tutor's problem solving.

#### 4.2. Scenario 2: Opportunistic Tutoring with a Discourse Management Network

Now we reconsider the same scenario assuming that *only* a discourse management network controls the tutor. In this case no individualized lesson planning can be done without recourse to mechanisms outside of the discourse management network such as an agenda or auxiliary planning mechanism. Alternatively the DMN could be extended so that some states correspond to lesson

---

<sup>6</sup>The available library of cases that the student can select from changes to include only those that the student should also be able to solve at that time.

planning decisions that result in the selection of a curriculum plan from a plan library. But then states now represent both lesson planning actions and discourse management actions, undercutting the perspicuity of the DMN. Even with this extension the curriculum planning is not nearly so customized as the blackboard tutor provides. S cannot interact with the DMN tutor T' to formulate a customized curriculum that achieves his objectives while satisfying department requirements. T' is also less capable of customizing individual lesson plans than T. Although the student can change the current topic with T', he cannot interact with T' to draw up an intended sequence of topics or cases for a particular lesson.

Let us assume that the DMN tutor is used either as a problem solving monitor that tutors opportunistically (like GUIDON) or as a Socratic question and answer tutor (like MENO-TUTOR). In either case the tutor is not the primary source of instruction [11]; instead it assumes that the student is somewhat familiar with the material already but his knowledge can be refined and is likely to have misconceptions. Thus the tutor must be used with some auxiliary form of instruction such as a workbook (like Anderson's LISP tutor [7]) or classroom instruction.

The DMN tutor is best suited to handle one particular class of tutorial strategy, either Socratic question and answer or another opportunistic approach.<sup>7</sup> Its ability to handle expository instruction is limited and its inability to plan dialog hampers it once it leaves the stereotypical discourse situations for which DMNs are best suited. When S gives the DMN tutor a program with multiple bugs it cannot plan which bugs to address, which to ignore, and what order (and how) to discuss the bugs since these are not local decisions (e.g., one bug may mask other more serious bugs or two similar bugs could be discussed together as resulting from the same misconception). The DMN tutor also cannot plan a series of cases along with the topics to discuss for each case to cover a particular set of topics. The ability to defer discussing one topic now since a subsequent case will better address the same topic requires foresight that the DMN does not have.

The DMN tutor does not allow the student as much freedom to express initiative as the blackboard-based tutor. The student cannot interrupt the tutor's presentation with requests that may change assumptions underlying the instructional plan currently being followed. Thus in the earlier examples where the student interrupted the tutor and entered an exploratory mode, or requested a change in tutorial strategy, these degrees of flexibility are not readily supported in the DMN tutor. As before, any specific example in the scenario could be handled by a special-purpose DMN but in general the combinatorics prevent a single DMN from handling the possibility of these kinds of interruptions at any time, for any tutorial situation. The blackboard-based tutor is better equipped for these interruptions since it has an explicit modifiable plan representation and can apply knowledge about replanning at the time the interruption occurs.

This discussion is not meant to imply that S cannot benefit from the DMN controlled tutor T', only

---

<sup>7</sup>Even though GUIDON can exhibit many different *discourse* strategies for presenting MYCIN rules or assessing the student's knowledge of these, its overall tutorial strategy is opportunistic.

that T' is quite limited compared to the blackboard-based tutor T. Most previous work in intelligent tutoring systems adopts an opportunistic tutoring framework similar to that of T', which can be well supported by discourse management networks. Many successful tutors have been built in this framework (e.g., GREATERP [7], WEST [1], WUSOR [9], and GUIDON [2])<sup>8</sup> while tutors such as T are still only hypothetical. The computational overhead in the blackboard architecture is also undeniably greater than that of the simpler discourse management network. However, the point of these scenarios is to illustrate that we can still build much more powerful tutors, encompassing both opportunistic and highly customized expository instruction, and that research in planners and architectures to support them, such as blackboards, is justifiable.

## 5. Summary

This paper compares the blackboard architecture to discourse management networks and concludes that the blackboard architecture better supports dynamic instructional planning. This paper also argues that dynamic instructional planning allows more flexible and effective instruction than tutors that have no plan or only a pre-stored plan. To support these arguments the key architectural features of each architecture were compared and discussed in terms of the support or limitations they provide for dynamic instructional planning. A hypothetical scenario illustrating control by the two architectures also points out differences in capabilities.

Since discourse management networks are not sufficient in themselves to support the full range of behavior desirable of intelligent tutoring systems, especially the ability to customize lesson plans, to generate globally coherent instruction, to select the best action based on multiple conflicting goals, and to allow student initiative that may alter a tutor's current plan, research in blackboard-based dynamic instructional planners is justified. It is also clear that such blackboard-based planners are not strictly necessary to build opportunistic tutors that do not perform lesson or dialog planning and which limit student initiative. In these cases, DMNs are sufficient and sophisticated tutors, such as GUIDON, can and have been built with that architecture.

This paper has not addressed the planning knowledge or the kind of planning required to realize a dynamic instructional planner such as the one described in the scenario above. The planning required is an instance of a larger class of problems characterized by an environment that is

- *Incomplete* - since the tutor does not know the student's knowledge completely at any point,
  - *Uncertain* - since what it does know it does not know with complete certainty,
  - *Dynamic* - since the student's knowledge changes during and between tutorial sessions,
  - *Multi-agent* - since the tutor and student cooperate to facilitate the student's learning,
- and where

---

<sup>8</sup>These tutors all adopt an opportunistic tutoring strategy, even though WEST and WUSOR act as coaches for a student who is playing a game while GUIDON and GREATERP assist students in problem solving for particular cases.

- *Results of actions are uncertain* - since the tutor cannot predict with certainty the results of its actions.

To specify a blackboard architecture as preferable to one based on discourse management networks to handle these kind of planning difficulties does not provide an ITS designer with a solution to the control problem, although it is a step in this direction. Any kind of planner can be implemented in the blackboard architecture. Exactly what planning techniques are best in this domain, how to implement a practical dynamic instructional planner in the blackboard architecture, and the knowledge required for such planners are some of the difficult issues that remain to be addressed in future research.

#### Acknowledgements

I would like to thank Perry Thorndyke and Lee Brownston for reviewing drafts of this paper. This research has been funded under contract N00014-86-C-0487, by the Office of Chief of Naval Research, Office of Naval Research, Naval Training Systems Center, and the Air Force Human Resources Laboratory.

## References

- [1] R. R. Burton, and J. S. Brown.  
An Investigation of Computer Coaching for Informal Learning Activities.  
*International Journal of Man-Machine Studies* (11):5 - 24, 1979.
- [2] W. Clancey.  
Tutoring Rules for guiding a case method dialogue.  
*International Journal of Man-Machine Studies* (11):25 - 49, 1979.
- [3] W. Clancey.  
*Knowledge-based Tutoring*.  
The MIT Press, 1987.
- [4] W. L. Johnson and E. Soloway.  
Intention-based Diagnosis of Programming Errors.  
In *Proceedings of the National Conference on Artificial Intelligence*, pages 162 - 168. American Association for Artificial Intelligence, August, 1984.
- [5] W. R. Murray.  
Automatic Program Debugging for Intelligent Tutoring Systems.  
*Computational Intelligence* 3(1):1 - 16, 1987.
- [6] W. R. Murray.  
*Dynamic Instructional Planning in the BBI Blackboard Architecture*.  
Technical Report R-6168, FMC Corporation, August, 1988.
- [7] B. Reiser, J. Anderson, and R. Farrell.  
Dynamic Student Modelling in an Intelligent Tutor for Lisp Programming.  
In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 8 - 14. 1985.
- [8] D. M. Russell.  
The Instructional Design Environment: The Interpreter.  
*Intelligent Tutoring Systems: Lessons Learned*.  
Lawrence Erlbaum Associates, Inc., 1987.
- [9] J. C. Stansfield, B. Carr, and I. P. Goldstein.  
*Wumpus Advisor I: a First Implementation of a Program that Tutors Logical and Probabilistic Reasoning Skills*.  
Technical Report AI Lab Memo 381, Massachusetts Institute of Technology, 1976.
- [10] S. Vere.  
Planning.  
*Encyclopedia of Artificial Intelligence*.  
John Wiley and Sons, Inc., 1987.
- [11] E. Wenger.  
*Artificial Intelligence and Tutoring Systems*.  
Morgan Kaufmann, 1987.
- [12] B. P. Woolf, and D. D. McDonald.  
Building a Computer Tutor: Design Issues.  
*IEEE Computer* 17(9):61 - 73, 1984.

- [13] B. P. Woolf.  
Representing Complex Knowledge in an Intelligent Machine Tutor.  
*Computational Intelligence* 3(1):45 - 55, 1987.

ONR Distribution

Personnel Analysis Division,  
AF/MPXA  
5C360, The Pentagon  
Washington, DC 20330

Air Force Human  
Resources Lab  
AFHRL/MPD  
Brooks, AFB, TX 78235

AFOSR,  
Life Sciences Directorate  
Bolling Air Force Base  
Washington, DC 20332

Technical Director, ARI  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Technical Director,  
Army Human Engineering Lab  
ATTN: SLCHE-D  
Aberdeen Proving Ground  
MD 21005-5001

Dr. Beth Adelson  
Department of Computer Science  
Tufts University  
Medford, MA 02155

Dr. Robert Ahlers  
Code N711  
Human Factors Laboratory  
Naval Training Systems Center  
Orlando, FL 32813

Technical Director  
Air Force Human Resources Lab.  
Brooks AFB, TX 78236-5601

Dr. John R. Anderson  
Department of Psychology  
Carnegie-Mellon University  
Schenley Park  
Pittsburgh, PA 15213

Dr. Stephen J. Andriole, Chairman  
Department of Information Systems  
and Systems Engineering  
George Mason University  
4400 University Drive  
Fairfax, VA 22030

Dr. Patricia Baggett  
School of Education  
610 E. University, Rm 1302D  
University of Michigan  
Ann Arbor, MI 48109-1259

Dr. Eva L. Baker  
UCLA Center for the Study  
of Evaluation  
145 Moore Hall  
University of California  
Los Angeles, CA 90024

Dr. Meryl S. Baker  
Navy Personnel R&D Center  
San Diego, CA 92152-6800

Prof. Dott. Bruno G. Bara  
Unita di ricerca di  
Intelligenza Artificiale  
Universita di Milano  
20122 Milano - via F. Sforza 23  
ITALY

Dr. Gautam Biswas  
Department of Computer Science  
Box 1688, Station B  
Vanderbilt University  
Nashville, TN 37235

Dr. John Black  
Teachers College, Box 8  
Columbia University  
525 West 120th Street  
New York, NY 10027

Dr. Deborah A. Boehm-Davis  
Department of Psychology  
George Mason University  
4400 University Drive  
Fairfax, VA 22030

Dr. Jeff Bonar  
Learning R&D Center  
University of Pittsburgh  
Pittsburgh, PA 15260

Dr. David Bowers  
Rensis Likert Associates  
3001 S. State Street  
Ann Arbor, MI 48104-7352

ONR Distribution

Dr. Robert Breaux  
Code 7B  
Naval Training Systems Center  
Orlando, FL 32813-7100

Dr. John S. Brown  
XEROX Palo Alto Research  
Center  
3333 Coyote Road  
Palo Alto, CA 94304

Dr. John T. Bruer  
James S. McDonnell Foundation  
1034 So. Brentwood Blvd., Ste. 1610  
St. Louis, MO 63117

Dr. Bruce Buchanan  
Computer Science Department  
Stanford University  
Stanford, CA 94305

Lt. Col. Hugh Burns  
AFHRL/IDI  
Brooks AFB, TX 78235

Assistant for Long Range Rqmts.  
CNO Executive Panel (Op-00K)  
4401 Ford Avenue  
Alexandria, VA 22302-0268

Dr. Joanne Capper, Director  
Center for Research into Practice  
1718 Connecticut Ave., N.W.  
Washington, DC 20009

Dr. Pat Carpenter  
Carnegie-Mellon University  
Department of Psychology  
Pittsburgh, PA 15213

Dr. John M. Carroll  
IBM Watson Research Center  
User Interface Institute  
P.O. Box 704  
Yorktown Heights, NY 10598

Director, Manpower Program  
Center for Naval Analyses  
4401 Ford Avenue  
P.O. Box 16268  
Alexandria, VA 22302-0268

Center for Personnel  
Security Research  
Suite E, Bldg. 455  
99 Pacific Street  
Monterey, CA 93940-2481

Professor Chu Tien-Chen  
Mathematics Department  
National Taiwan University  
Taipei, TAIWAN

Dr. Michélene Chi  
Learning R & D Center  
University of Pittsburgh  
3939 O'Hara Street  
Pittsburgh, PA 15260

Dr. Raymond E. Christal  
UES LAMP Science Advisor  
AFHRL/MOEL  
Brooks AFB, TX 78235

Dr. William Clancey  
Institute for Research  
on Learning  
3333 Coyote Hill Road  
Palo Alto, CA 94304

Dr. Allan M. Collins  
Bolt Beranek & Newman, Inc.  
10 Moulton Street  
Cambridge, MA 02238

Dr. Stanley Collyer  
Office of Naval Technology  
Code 222  
800 N. Quincy Street  
Arlington, VA 22217-5000

Dr. Albert T. Corbett  
Department of Psychology  
Carnegie-Mellon University  
Pittsburgh, PA 15213

CAPT P. Michael Curran  
Chief of Naval Operations  
OP-939  
Pentagon  
Washington, DC 20350-2000

ONR Distribution

Brian Dallman  
Training Technology Branch  
3400 TCHTW/TTGXC  
Lowry AFB, CO 80230-5000

Dr. Robert B. Davis  
Curriculum Laboratory  
(Education)  
University of Illinois  
Urbana, IL 61801

Chief, Survey & Market  
Analysis Division  
Defense Manpower Data Center  
1600 Wilson Blvd., #400  
Arlington, VA 22209-2593

Defense Technical Info. Ctr.  
Attn: TC  
Cameron Station, Bldg. 5  
Alexandria, VA 22314  
(12 copies)

Deputy Dir. Military Personnel  
Policy Division  
*Office of the DCNO (MPT)*  
(Op-13B)  
Department of the Navy  
Washington, DC 20370-2000

Deputy Director Total Force  
Training & Education Division  
*Office of the DCNO (MPT)*  
(Op-11B)  
Department of the Navy  
Washington, DC 20370-2000

Head, Leadership Branch  
Naval Military Personnel Command  
Attn: LCDR E. Marits, NMPC-621  
Department of the Navy  
Washington, DC 20370-5620

Head, Military Compensation  
Policy Branch  
*Office of the DCNO (MPT)*  
(Op-134)  
Department of the Navy  
Washington, DC 20370-2000

Dr. Andrea di Sessa  
University of California  
School of Education  
Tolman Hall  
Berkeley, CA 94720

ERIC Facility-Acquisitions  
4350 East-West Hwy, Suite 1100  
Bethesda, MD 20814-4475

Dr. Martha Evans  
Dept. of Computer Science  
Illinois Institute of Technology  
Chicago, IL 60616

Dr. Marshall J. Farr, Consultant  
Cognitive & Instructional Sciences  
2520 North Vernon Street  
Arlington, VA 22207

Dr. Paul Felovich  
Southern Illinois Univ.  
School of Medicine  
P.O. Box 3926  
Springfield, IL 62708

Dr. Gerhard Fischer  
University of Colorado  
Department of Computer Science  
Boulder, CO 80309

Dr. Kenneth D. Forbus  
University of Illinois  
Department of Computer Science  
1304 West Springfield Avenue  
Urbana, IL 61801

Dr. Barbara A. Fox  
University of Colorado  
Department of Linguistics  
Boulder, CO 80309

Dr. John R. Frederiksen  
BBN Laboratories  
10 Moulton Street  
Cambridge, MA 02238

Dr. Michael Friendly  
Psychology Department  
York University  
Toronto ONT  
CANADA M3J 1P3

ONR Distribution

Julie A. Gadsden  
Information Technology  
Applications Division  
Admiralty Research Establishment  
Portsmouth, Portsmouth PO6 4AA  
UNITED KINGDOM

Eric Gaussens  
Research & Development Dept.  
Framentec S.A.  
Tour Fiat Cedex 16  
Paris la Defense  
F. 92084  
France

Dr. Dedre Gentner  
University of Illinois  
Department of Psychology  
603 E. Daniel St.  
Champaign, IL 61820

Dr. Meg Gerrard  
Psychology Department  
Iowa State University  
Ames, IA 50010

Dr. Robert Glaser  
Learning Research &  
Development Center  
University of Pittsburgh  
3939 O'Hara Street  
Pittsburgh, PA 15260

Dr. Sam Glucksberg  
Department of Psychology  
Princeton University  
Princeton, NJ 08540

Prof. Clark Glymour  
Dept. of Philosophy  
Carnegie-Mellon University  
Pittsburgh, PA 15213

Dr. Sherrie Gott  
AFHRL/MOMJ  
Brooks AFB, TX 78235-5601

Dr. T. Govindaraj  
Georgia Institute of  
Technology  
School of Industrial  
and Systems Engineering  
Atlanta, GA 30332-0205

Dr. Jordan Grafman  
Neuropsychology Section  
Medical Neurology  
Branch-NINCDS  
Building 10, Room 5C416  
Bethesda, MD 20892

Dr. James G. Greeno  
School of Education  
Stanford University  
Room 311  
Stanford, CA 94305

Dr. Dik Gregory  
Admiralty Research  
Establishment/AXB  
Queens Road  
Teddington  
Middlesex, ENGLAND TW110LN

Dr. Henry M. Halff  
Halff Resources, Inc.  
4918 33rd Road, North  
Arlington, VA 22207

Dr. Bruce W. Hamill  
Research Center  
The Johns Hopkins University  
Applied Physics Laboratory  
Johns Hopkins Road  
Laurel, MD 20707

Dr. Barbara Hayes-Roth  
Knowledge Systems Laboratory  
Stanford University  
701 Welch Road  
Palo Alto, CA 94304

Dr. James Hollan  
NPRDC, UCSD  
Code 501  
San Diego, CA 92152

Dr. Keith Holyoak  
Department of Psychology  
University of California  
Los Angeles, CA 90024

Ms. Julia S. Hough  
Lawrence Erlbaum Associates  
110 W. Harvey Street  
Philadelphia, PA 19144

ONR Distribution

Dr. Ed Hutchins  
Intelligent Systems Group  
Institute for  
Cognitive Science (C-015)  
UCSD  
La Jolla, CA 92093

Dr. Janet Jackson  
Rijksuniversiteit Groningen  
Biologisch Centrum, Vleugel D  
Kerklaan 30, 9751 NN Haren  
The NETHERLANDS

Dr. Claude Janvier  
Universite' du Quebec a Montreal  
P.O. Box 8888, succ: A"  
Montreal, Quebec H3C 3P8  
CANADA

Dr. Robin Jeffries  
Hewlett-Packard Laboratories, 3L  
P.O. Box 10490  
Palo Alto, CA 94303-0971

Prof. David W. Johnson  
Cooperative Learning Center  
University of Minnesota  
150 Pillsbury Dr., S.E.  
Minneapolis, MN 55455

Dr. Marcel Just  
Carnegie-Mellon University  
Department of Psychology  
Schenley Park  
Pittsburgh, PA 15213

Dr. Daniel Kahneman  
Department of Psychology  
University of California  
Berkeley, CA 94720

Dr. Milton S. Katz  
European Science Coordination  
Office  
U.S. Army Research Institute  
Box 65  
FPO New York 09510-1500

Dr. Wendy Kellogg  
IBM T. J. Watson Research Ctr.  
P.O. Box 704  
Yorktown Heights, NY 10598

Dr. Jeffery L. Kennington  
School of Engineering &  
Applied Sciences  
Southern Methodist University  
Dallas, TX 75275

Dr. David Kieras  
Technical Communication Program  
TIDAL Bldg., 2360 Bonisteel Blvd.  
University of Michigan  
Ann Arbor, MI 48109-2108

Dr. Janet L. Kolodner  
Georgia Institute of Technology  
School of Information &  
Computer Science  
Atlanta, GA 30332

Dr. Kenneth Kotovsky  
Community College of  
Allegheny County  
808 Ridge Avenue  
Pittsburgh, PA 15212

Dr. Benjamin Kuipers  
University of Texas at Austin  
Department of Computer Sciences  
Taylor Hall 2.124  
Austin, Texas 78712

Dr. Jill Larkin  
Carnegie-Mellon University  
Department of Psychology  
Pittsburgh, PA 15213

Commander J.M. LaRocco  
Naval School of Health Sciences  
National Naval Medical Center  
Bldg. 141  
Washington, DC 20814-5033

Dr. Robert W. Lawler  
Matthews 118  
Purdue University  
West Lafayette, IN 47907

Dr. Alan M. Lesgold  
Learning R&D Center  
University of Pittsburgh  
Pittsburgh, PA 15260

ONR Distribution

Dr. Jim Levin  
Department of  
Educational Psychology  
210 Education Building  
1310 South Sixth Street  
Champaign, IL 61820-6990

Dr. John Levine  
Learning R&D Center  
University of Pittsburgh  
Pittsburgh, PA 15260

Dr. Clayton Lewis  
University of Colorado  
Department of Computer Science  
Campus Box 430  
Boulder, CO 80309

Science and Technology Division  
Library of Congress  
Washington, DC 20540

Vern M. Malec  
NPRDC, Code 52  
San Diego, CA 92152-6800

Dr. Jane Mallin  
Mail Code EF5  
NASA Johnson Space Center  
Houston, TX 77058

Dr. William L. Maloy  
Naval Education and Training  
Program Support Activity  
Code 047  
Building 2435  
Pensacola, FL 32509-5000

Dr. Elaine Marsh  
Naval Center for Applied Research  
in Artificial Intelligence  
Naval Research Laboratory  
Code 5510  
Washington, DC 20375-5000

Dr. Sandra P. Marshall  
Dept. of Psychology  
San Diego State University  
San Diego, CA 92182

Dr. Manton M. Matthews  
Department of Computer Science  
University of South Carolina  
Columbia, SC 29208

Dr. Joseph C. McLachlan  
Code 52  
Navy Personnel R&D Center  
San Diego, CA 92152-6800

Dr. James McMichael  
Technical Director  
Navy Personnel R&D Center  
San Diego, CA 92152-6800

Dr. Barbara Means  
SRI International  
333 Ravenswood Avenue  
Menlo Park, CA 94025

Prof. George A. Miller  
Dept. of Psychology  
Princeton University  
Princeton, NJ 08544

Dr. James R. Miller  
MCC  
3500 W. Balcones Center Dr.  
Austin, TX 78759

Mark L. Miller  
Advanced Technology Group  
Apple Computer Inc.  
20525 Mariani Ave.  
MS 22C  
Cupertino, CA 95014

Dr. William Montague  
NPRDC Code 13  
San Diego, CA 92152-6800

Dr. Judy Moracco  
Code CEL-MP53  
Washington Navy Yard  
Bldg. 200  
Washington, DC 20374

Dr. Randy Mumaw  
Training Research Division  
HumRRO  
1100 S. Washington  
Alexandria, VA 22314

## ONR Distribution

Dr. Allen Munro  
Behavioral Technology  
Laboratories - USC  
1845 S. Elena Ave., 4th Floor  
Redondo Beach, CA 90277

Deputy Technical Director  
NPRDC Code 01A  
San Diego, CA 92152-6800

Director, Human Factors  
& Organizational Systems Lab,  
NPRDC (Code 07)  
San Diego, CA 92152-6800

Director, Manpower and Personnel  
Laboratory,  
NPRDC (Code 06)  
San Diego, CA 92152-6800

Director, Training Laboratory,  
NPRDC (Code 05)  
San Diego, CA 92152-6800

Head, Fleet Liaison Dept.  
NPRDC (Code 31)  
San Diego, CA 92152-6800

Head, Human Factors Dept.  
NPRDC (Code 41)  
San Diego, CA 92152-6800

Head, Manpower Systems Dept.  
NPRDC (Code 61)  
San Diego, CA 92152-6800

Head, Personnel Systems Dept.  
NPRDC (Code 62)  
San Diego, CA 92152-6800

Head, Testing Systems Dept.  
NPRDC (Code 63)  
San Diego, CA 92152-6800

Head, Training Systems Dept.  
NPRDC (Code 52)  
San Diego, CA 92152-6800

Head, Training Tech. Dept.  
NPRDC (Code 51)  
San Diego, CA 92152-6800

Library  
NPRDC  
Code P201L  
San Diego, CA 92152-6800

Spec. Asst. for Research, Exper-  
imental & Academic Programs,  
NTTC (Code 016)  
NAS Memphis (75)  
Millington, TN 38054

Assistant Chief of Staff  
for Research, Development,  
Test, and Evaluation  
Naval Education and  
Training Command (N-5)  
NAS Pensacola, FL 32508

Director, Instructional Dvlpmt.  
and Educational Pgm. Support Dept.  
Naval Education & Training Pgm.  
Management Support Activity  
(NETPMSA)  
Pensacola, FL 32509

Technical Director  
Naval Health Research Center  
P.O. Box 85122  
San Diego, CA 92138-9174

Director, Recreational Svcs. Dept.  
Naval Military Personnel Command  
(N-651C)  
1300 Wilson Blvd., Room 932  
Arlington, VA 22209

Chairman, Department of  
Administrative Sciences  
Code 54  
Naval Postgraduate School  
Monterey, CA 93943-5100

Chairman, Department of  
Operations Research  
Code 55  
Naval Postgraduate School  
Monterey, CA 93943-5100

ONR Distribution

Commanding Officer,  
Naval Research Laboratory  
Code 2627  
Washington, DC 20390

Deputy Director Manpower,  
Personnel and Training Div.  
Naval Sea Systems Command  
Attn: Code CEL-MP63  
Washington, DC 20362

Head, Human Factors Laboratory  
Naval Training Systems Ctr.  
Code 71  
Orlando, FL 32813-7100

Library  
Naval Training Systems Center  
Orlando, FL 32813

Library  
Naval War College  
Newport, RI 02940

Commanding Officer  
Navy Personnel R&D Ctr.  
San Diego, CA 92152-6800

Technical Director  
Navy Personnel R&D Center  
San Diego, CA 92152-6800

Director, Research & Analysis Div.  
Navy Recruiting Command (Code 223)  
4015 Wilson Blvd., Room 215  
Arlington, VA 22203-1991

Dr. Donald A. Norman (C-015)  
Institute for Cognitive Science  
University of California  
La Jolla, CA 92093

Special Assistant for Marine  
Corps Matters,  
ONR Code 00MC  
800 N. Quincy St.  
Arlington, VA 22217-5000

Chairman, MPT R&D Committee  
Office of the Chief of  
Naval Research  
Code 222  
Arlington, VA 22217-5000

Director, Biological/Human  
Factors Division (Code 125)  
Office of the Chief of  
Naval Research  
Arlington, VA 22217-5000

Director, Navy Family Support Pgm  
Office of the DCNO (MPT) (Op-156)  
1300 Wilson Blvd., Room 828  
Arlington, VA 22209

Office of Naval Research,  
Code 1142  
800 N. Quincy St.  
Arlington, VA 22217-5000

Office of Naval Research,  
Code 1142BI  
800 N. Quincy Street  
Arlington, VA 22217-5000

Office of Naval Research,  
Code 1142CS  
800 N. Quincy Street  
Arlington, VA 22217-5000  
(6 Copies)

Office of Naval Research,  
Code 1142PS  
800 N. Quincy Street  
Arlington, VA 22217-5000

Biological Intelligence  
Code 1142BI  
Office of Naval Research  
Arlington, VA 22217-5000

Cognitive Science  
Code 1142CS  
Office of Naval Research  
Arlington, VA 22217-5000

Director, Cognitive &  
Neural Sciences (Code 1142)  
Office of Naval Research  
Arlington, VA 22217-5000

Director, Life Sciences  
Code 114  
Office of Naval Research  
Arlington, VA 22217-5000

ONR Distribution

Director Research Programs  
Office of Naval Research  
Code 11  
Arlington, VA 22217-5000

Mathematics  
Code 1111MA  
Office of Naval Research  
Arlington, VA 22217-5000

Psychologist  
Office of Naval Research  
Branch Office, London  
Box 39  
FPO New York, NY 09510

Psychologist  
Office of Naval Research  
Detachment  
1030 E. Green Street  
Pasadena, CA 91106-2485

Dr. Harold F. O'Neil, Jr.  
School of Education - WPH 801  
Department of Educational  
Psychology & Technology  
University of Southern California  
Los Angeles, CA 90089-0031

Dr. Judith Orasanu  
Basic Research Office  
Army Research Institute  
5001 Eisenhower Avenue  
Alexandria, VA 22333

Military Assistant for Training and  
Personnel Technology,  
OUSD (R & E)  
Room 3D129, The Pentagon  
Washington, DC 20301-3080

Assistant for Manpower and Training  
Office of the CNO (Op-987H)  
5D772, The Pentagon  
Washington, DC 20350

Head, Manpower, Personnel,  
and Training Branch  
Office of the CNO (Op-813)  
4A478, The Pentagon  
Washington, DC 20350-1000

Dr. Douglas Pearce  
1133 Sheppard W  
Box 2000  
Downsview, Ontario  
CANADA M3M 3B9

Office of the Deputy Assistant  
Secretary of the Navy  
Manpower & Reserve Affairs  
5D800, The Pentagon  
Washington, DC 20350-1000

Dr. Tjeerd Plomp  
Twente University of Technology  
Department of Education  
P.O. Box 217  
7500 AE ENSCHEDE  
THE NETHERLANDS

Dr. Martha Polson  
Department of Psychology  
University of Colorado  
Boulder, CO 80309-0345

Dr. Steven E. Poltrock  
MCC  
3500 West Balcones Center Dr.  
Austin, TX 78759-6509

Dr. Harry E. Pople  
University of Pittsburgh  
Decision Systems Laboratory  
1360 Scaife Hall  
Pittsburgh, PA 15261

Dr. Joseph Psotka  
ATTN: PER-IC  
Army Research Institute  
5001 Eisenhower Ave.  
Alexandria, VA 22333-5600

Mr. Paul S. Rau  
Code U-33  
Naval Surface Weapons Center  
White Oak Laboratory  
Silver Spring, MD 20903

Dr. Steve Reder  
Northwest Regional  
Educational Laboratory  
400 Lindsay Bldg.  
710 S.W. Second Ave.  
Portland, OR 97204

ONR Distribution

Dr. James A. Reggia  
University of Maryland  
School of Medicine  
Department of Neurology  
22 South Greene Street  
Baltimore, MD 21201

Dr. J. Wesley Regan  
AFHRL/IDI  
Brooks AFB, TX 78235

Dr. Fred Relf  
Physics Department  
University of California  
Berkeley, CA 94720

Dr. Brian Reiser  
Department of Psychology  
Green Hall  
Princeton University  
Princeton, NJ 08540

Dr. Gilbert Ricard  
Mail Stop K02-14  
Grumman Aircraft Systems  
Bethpage, NY 11787

Dr. J. Jeffrey Richardson  
Center for Applied AI  
College of Business  
University of Colorado  
Boulder, CO 80309-0419

Dr. Linda G. Roberts  
Science, Education, and  
Transportation Program  
Office of Technology Assessment  
Congress of the United States  
Washington, DC 20510

Dr. William B. Rouse  
Search Technology, Inc.  
4725 Peachtree Corners Circle  
Suite 200  
Norcross, GA 30092

Dr. Eduardo Salas  
Human Factors Division  
Code 712  
Naval Training Systems Ctr.  
Orlando, FL 32813-7100

Dr. Walter Schneider  
Learning R&D Center  
University of Pittsburgh  
3939 O'Hara Street  
Pittsburgh, PA 15260

Dr. Janet W. Schofield  
816 LRDC Building  
University of Pittsburgh  
3939 O'Hara Street  
Pittsburgh, PA 15260

Dr. Miriam Schustack  
Code 52  
Navy Personnel R & D Center  
San Diego, CA 92152-6800

Dr. Judith W. Segal  
OERI  
555 New Jersey Ave., NW  
Washington, DC 20208

Dr. Colleen M. Seifert  
Institute for Cognitive Science  
Mail Code C-015  
University of California, San Diego  
La Jolla, CA 92093

Dr. Ben Shneiderman  
Dept. of Computer Science  
University of Maryland  
College Park, MD 20742

Dr. Randall Shumaker  
Naval Research Laboratory  
Code 5510  
4555 Overlook Avenue, S.W.  
Washington, DC 20375-5000

Dr. Edward E. Smith  
Department of Psychology  
University of Michigan  
330 Packard Road  
Ann Arbor, MI 48103

Program Director  
Manpower Research &  
Advisory Services  
Smithsonian Institution  
801 N. Pitt St., Suite 120  
Alexandria, VA 22314-1713

ONR Distribution

Dr. Al Smode  
Naval Training Systems Ctr.  
Code 71  
Orlando, FL 32813-7100

Dr. Elliot Soloway  
Yale University  
Computer Science Department  
P.O. Box 2158  
New Haven, CT 06520

Dr. Richard C. Sorensen  
Navy Personnel R&D Center  
San Diego, CA 92152-6800

Dr. Kathryn T. Spoehr  
Brown University  
Department of Psychology  
Providence, RI 02912

Dr. Kurt Steuck  
AFHRL/MOA  
Brooks AFB  
San Antonio, TX 78235-5601

Dr. Albert Stevens  
Bolt Beranek & Newman, Inc.  
10 Moulton St.  
Cambridge, MA 02238

Dr. John Tangney  
AFOSR/NL, Bldg. 410  
Bolling AFB, DC 20332-6448

Dr. Kikumi Tatsuoka  
CERL  
252 Engineering Research  
Laboratory  
103 S. Mathews Avenue  
Urbana, IL 61801

Dr. Perry W. Thorndyke  
FMC Corporation  
Central Engineering Labs  
1205 Coleman Avenue, Box 580  
Santa Clara, CA 95052

Dr. Martin A. Tolcott  
3001 Veazey Terr., N.W.  
Apt. 1617  
Washington, DC 20008

Dr. Douglas Towne  
Behavioral Technology Labs  
University of Southern California  
1845 S. Elena Ave.  
Redondo Beach, CA 90277

Headquarters, U. S. Marine Corps  
Code MPI-20  
Washington, DC 20380

Dr. Jerry Vogt  
Navy Personnel R&D Center  
Code 51  
San Diego, CA 92152-6800

Dr. Ralph Wachter  
JHU-APL  
Johns Hopkins Road  
Laurel, MD 20707

Dr. Keith T. Wescourt  
FMC Corporation  
Central Engineering Labs  
1205 Coleman Ave., Box 580  
Santa Clara, CA 95052

Dr. Barbara White  
BBN Laboratories  
10 Moulton Street  
Cambridge, MA 02238

David C. Wilkins  
212 Digital Computer Lab  
1304 W. Springfield Ave.  
Urbana, IL 61801

Dr. Kent E. Williams  
Inst. for Simulation and Training  
University of Central Florida  
P.O. Box 25000  
Orlando, FL 32816-0544

Dr. Robert A. Wisher  
U.S. Army Institute for the  
Behavioral and Social Sciences  
5001 Eisenhower Avenue  
Alexandria, VA 22333-5800

Dr. Martin F. Wiskoff  
Defense Manpower Data Center  
550 Camino El Estero  
Suite 200  
Monterey, CA 93943-3231

**ONR Distribution**

**Dr. Wallace Wulfeck, III  
Navy Personnel R&D Center  
Code 51  
San Diego, CA 92152-6800**

**Dr. Masoud Yazdani  
Dept. of Computer Science  
University of Exeter  
Prince of Wales Road  
Exeter EX44PT  
ENGLAND**

**Dr. Joseph L. Young  
National Science Foundation  
Room 320  
1800 G Street, N.W.  
Washington, DC 20550**