DTIC FILE COPY

FMC

Report No. R-6168 (4)

AD-A199 132

Dynamic Instructional Planning in the BB1 Blackboard Architecture

William R. Murray

August 1988

DTIC
ELECTE
SEP 0 6 1988
S D
D

Central Engineering Laboratories
Santa Clara California

88 9 6 105

ADA199132

SECURITY CLASSIFICATION OF THIS PAGE

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):

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

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
800 NORTH QUINCY STREET
ARLINGTON, VA 22217-5000

8a NAME OF FUNDING / SPONSORING ORGANIZATION:

8b OFFICE SYMBOL (If applicable): N00014

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.	PROJECT NO	TASK NO.	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification): DYNAMIC INSTRUCTIONAL PLANNING IN THE BB1 BLACKBOARD ARCHITECTURE

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, AUGUST

15 PAGE COUNT: 19

16 SUPPLEMENTARY NOTATION:

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

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

An intelligent tutoring system that delivers effective instruction must select pedagogical actions appropriate to its tutorial situation. The approach taken in this research is to view this control problem as a dynamic planning problem. Dynamic instructional planning is the ability to generate, monitor, and revise instructional plans during the course of instruction. Planning and execution of instructional actions must be interleaved because the tutor operates in a changing environment with incomplete information.

An appropriate architecture is the BB1 Blackboard Architecture, which supports the building of knowledge-based planners that represent and reason about their own actions. Dynamic instructional planning uses these capabilities to apply pedagogical knowledge to reason about instructional actions that a tutor can perform.

We have built the Blackboard Instructional Planner in BB1 to teach troubleshooting a complex physical device by first imparting a mental model of the device and its operation. The planner

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):

22c OFFICE SYMBOL:

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

SECURITY CLASSIFICATION OF THIS PAGE

generates instructional plans from skeletal plans, executes them, and monitors their effectiveness. Instructional plans are modified and particular instructional actions selected in response to changes in the student model, changes in resources available, requests and questions of the student, and to properties of the subject matter currently being presented.

DTIC
COPY
INSPECTED
1

A-1

Dynamic Instructional Planning in the BB1 Blackboard Architecture

William R. Murray

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

Abstract

An intelligent tutoring system that delivers effective instruction must select pedagogical actions appropriate to its tutorial situation. The approach taken in this research is to view this control problem as a dynamic planning problem. Dynamic instructional planning is the ability to generate, monitor, and revise instructional plans during the course of instruction. Planning and execution of instructional actions must be interleaved because the tutor operates in a changing environment with incomplete information.

An appropriate architecture is the BB1 Blackboard Architecture, which supports the building of knowledge-based planners that represent and reason about their own actions. Dynamic instructional planning uses these capabilities to apply pedagogical knowledge to reason about instructional actions that a tutor can perform.

We have built the Blackboard Instructional Planner in BB1 to teach troubleshooting a complex physical device by first imparting a mental model of the device and its operation. The planner generates instructional plans from skeletal plans, executes them, and monitors their effectiveness. Instructional plans are modified and particular instructional actions selected in response to changes in the student model, changes in resources available, requests and questions of the student, and to properties of the subject matter currently being presented.

1. The Problem - Control and Planning in Intelligent Tutoring Systems

This research addresses the problem of control for intelligent tutoring systems. At any point 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.

This choice of actions is difficult since the tutorial situation is inherently *dynamic*. The student model changes as the student learns new information and forgets or confuses previously learned information. The lesson objectives change as the tutor refines initial lesson objectives, responds to student requests to cover or omit topics, and reacts to the changing amount of time left in the lesson. The current subject matter changes as the tutor moves from topic to topic. The tutor may select different tutorial strategies at different times, according to properties of the subject matter, the student model, and resources available. For example, a STEAMER-like [9] device simulation may be used for new material amenable to graphic presentation while only brief textual summaries are provided for previously seen material. The major resource that changes throughout a lesson is time, although others, such as the availability of appropriate videodisk frames, may also change.

The approach taken in this research is to view the problem of control as a planning problem. This approach encompasses the spectrum of tutoring systems from simple plan-following CAI tutors that interpret a curriculum plan to tutors that dynamically generate and interpret plans for instruction such as described in this paper. Even tutors emphasizing opportunistic intervention (e.g., WEST [3]) follow plans that determine when intervention is appropriate and how it should be performed (e.g., GUIDON [4]) [15].

We define an instructional plan as a sequence of steps where each step specifies a category of appropriate instructional actions. The categories can be broad (e.g., to perform actions to assess the student's knowledge of a topic), or they can specify only one kind of action (e.g., to give a true/false test). Instructional actions include not only interactions with the student but also decisions of the tutor such as the selection of a task to assign or a sequencing of topics to be discussed.

Similar instructional plans share a common tutorial strategy. A tutorial strategy is defined to be a class or category of instructional plans, that all teach by the same method or style of instruction. Examples of tutorial strategies include expository, case-method, exploratory[1], or Socratic question and answer. One instructional plan for the expository tutorial strategy might motivate a concept by presenting examples. Another plan for the same strategy might present the concept as analogous to a previously learned concept, pointing out similarities and differences.

The process of producing these plans is instructional planning. In particular, dynamic instructional planning is a planner-based approach to control for intelligent tutoring systems where appropriate plan generation 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.

This paper presents the Blackboard Instructional Planner, a dynamic instructional planner implemented in the BB1 Blackboard Architecture [7] [1]. Section 2 first motivates this research by considering the rationale for dynamic instructional planning compared to the simple plan-following approach of CAI systems. Section 3 presents four examples of dynamic instructional planning currently handled by this planner. Sections 4 and 5 discuss the BB1 Blackboard Architecture and the design of the Blackboard Instructional Planner within this architecture. Section 6 discusses the planner's knowledge representation for instructional actions and plans, and how it supports dynamic instructional planning. Section 7 contrasts the work presented here with related work. Finally, Section 8 summarizes the research contributions of this work.

2. Motivation for Dynamic Instructional Planning

Why should tutoring systems incorporate dynamic instructional planning? Two points must be supported here: first that a tutoring system should have a plan, and secondly 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 and motivation, and the time available for lessons. Transitions between topics are

---

[1]The tutor provides a microworld for the student to explore, perhaps providing guidance or structuring the world to encourage learning a particular skill or concept.

smooth and the tutor is consistent, i.e., it does not contradict previous or future instruction. One sign of incoherent instruction is recurring student confusion about the tutor and its intentions that distracts 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 using unexpected student questions, or to defer the questions since they will be addressed later in the lesson. 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 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 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.[2] 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 contigencies in a prestored plan. This planning knowledge allows the tutor to plan topic transitions, to replan the amount of time remaining in the lesson, and to decide how to return to the current topic or whether to abandon it altogether.

---

[2]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.

Planning knowledge is used not only to customize the lesson plan during the lesson in response to requests, but also initially. A tutor that can plan can generate an initial lesson plan customized to the student's interests, assessed capabilities, and background. The lesson plan can be revised as the 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 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 ease of use compared to a CAI system. The CAI system procedurally encodes planning knowledge and implicit assumptions about the tutorial state at each branch of its prestored 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.

3. Four Examples of Dynamic Instructional Planning

This section presents four examples of dynamic instructional planning. Four examples are shown since for any one example other planner architectures or approaches to control could be extended to handle that one example for the particular situation depicted. The key point is that the Blackboard Instructional Planner produces the instructional planning behavior described as a natural consequence of its architecture for all four examples and for all similar situations.

The goal of the example tutor built in this architecture is to teach the troubleshooting of the lower hoist assembly, a complex hydraulic-mechanical-electrical system in the Mark-45 naval gun turret. The lower hoist assembly conveys ammunition from a ship's magazine to a mechanical storage drum where shells are stored prior to firing. The lower hoist consists of the three interacting assemblies shown in Figure 3-1. The hoist drive assembly lifts the shells between pawls attached to a chain. The latch valve assembly locks the chain in place between lifting operations. The control valve assembly coordinates the operation of the latch valve assembly and the hoist drive assembly. The tutor first teaches the student what the structure of the lower hoist is by explaining the role of the three assemblies and how the components of each assembly contributes to its function. Both normal and faulted operation is discussed. Then the tutor presents troubleshooting cases to be solved with decreasing assistance from the tutor. These examples assume that a STEAMER-like device simulation of the lower hoist is available.

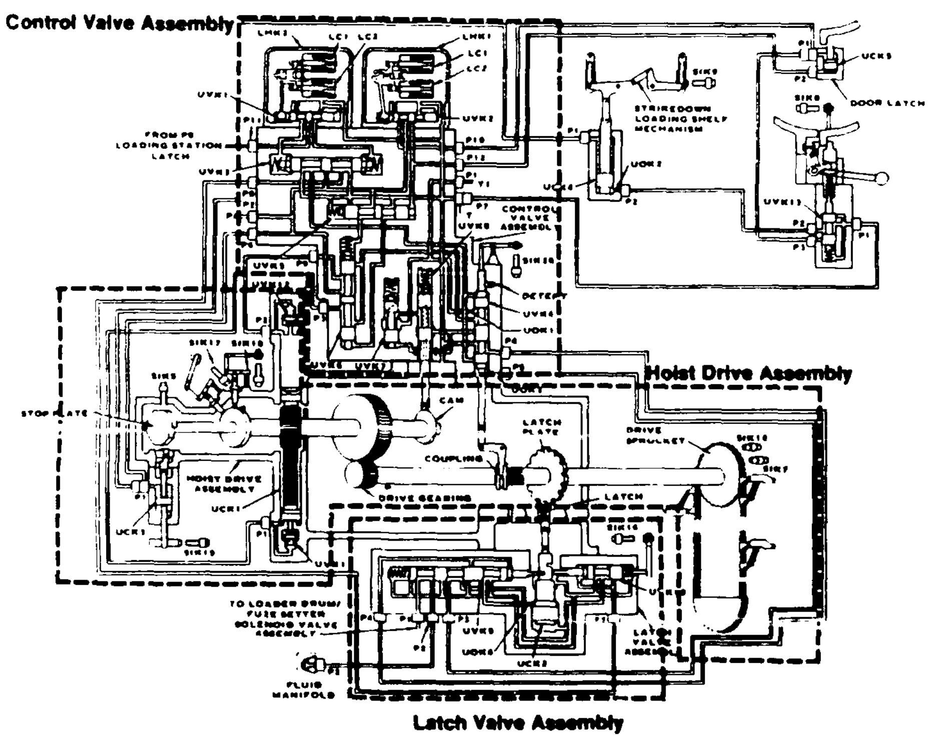
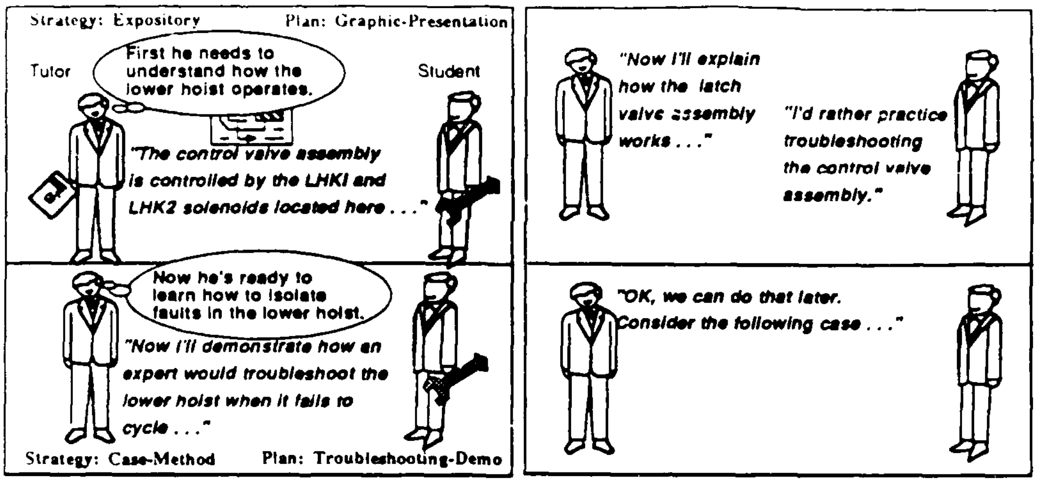


Figure 3-1: Schematic of the Lower Hoist Assembly

In the first example, the tutor selects tutorial strategies appropriate to the subject matter being presented and the student model. As shown in the upper half of Figure 3-2(a), the tutor selects an expository tutorial strategy to present the control valve assembly of the lower hoist. Since the control valve assembly is amenable to graphic presentation, the GRAPHIC-PRESENTATION instructional plan is selected. Later in the tutorial session, when the tutor believes the student has acquired a mental model of the lower hoist and its operation, the tutor selects a different tutorial strategy and instructional plan. As shown in the lower half of the figure, the tutor selects a case-method strategy and the TROUBLESHOOTING-DEMONSTRATION instructional plan. In this plan the tutor demonstrates how an expert would diagnose and repair a fault in the lower hoist. The key point in this example is that at different times in the lesson the tutor selects different tutorial strategies and instructional plans appropriate to the subject matter being taught and the student's knowledge of the subject.



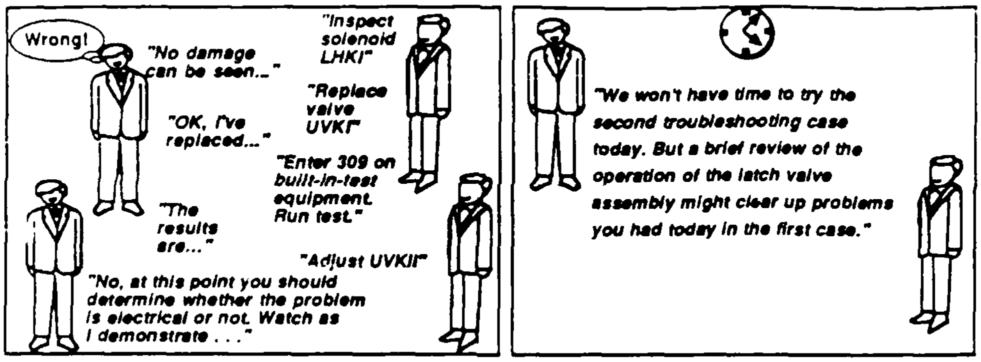
(a) Dynamic Selection of Appropriate Tutorial Strategies and Instructional Plans

(b) Dynamic Replanning to Handle a Student Request

Figure 3-2

In the second example shown in Figure 3-2(b), the tutor has just started a discussion of the latch valve assembly. This is the first step in an instructional plan using an expository tutorial strategy. During the introduction the student interrupts with a request to practice troubleshooting. The tutor must decide if the request should be granted in light of its current lesson objectives. In this example the tutor decides that the student's request can be accomodated. The current plan for the expository tutorial strategy is abandoned and an instructional plan for the case-method tutorial strategy is selected. The key point in this example is that the tutor could not anticipate the student's request, but could replan to handle the request once it decided that the request was appropriate.

In the third example, shown in Figure 3-3(a), the tutor prompts the student for troubleshooting actions to diagnose a fault in the lower hoist. The tutor accepts each action, updates the device simulation, and describes any other results of the action (e.g., measurements). The tutor does not comment on the efficacy of the student's actions, although its domain expert determines that each action is inappropriate. The assessment of each student action contributes to a global assessment of the effectiveness of the current instructional plan. When the tutor's assessment of the current plan's effectiveness falls below a certain threshold, the tutor replaces the plan with another that is more likely to meet the student's needs. In this case, the tutor elects to demonstrate how an expert would take over at this point. The key point in this example is that the tutor monitored the effectiveness of the instructional plan it selected and replanned to select another instructional plan that provides more assistance when it appeared this was appropriate.



(a) Monitoring a Tutorial Strategy and Abandoning an Ineffective Strategy

(b) Replanning to Cope with Changing Resources

Figure 3-3

In the final example shown in Figure 3-3(b) the tutor must cope with changing resources. At the beginning of the lesson the tutor had planned on covering two particular troubleshooting scenarios with the case-method tutorial strategy. However, after completing the first case the tutor does not believe there is enough time for the second case. The tutor must replan to make the best use of the remaining available time. The tutor switches to an expository strategy to review the latch valve assembly since the student had difficulties understanding its operation in the first case. The key point in this example is that the tutor replanned to cope with changing resources.

These four examples demonstrate some of the dynamic instructional planning capabilities of the Blackboard Instructional Planner. It dynamically selects tutorial strategies and instructional plans appropriate to the tutorial situation and replans as the tutorial situation changes. Replanning can be initiated by changes in the subject matter being presented, by detecting that an instructional plan is ineffective, and by time running out. In this process instructional plans can be suspended, resumed, or abandoned. All four examples are currently handled with the planner simulating the student interface, domain expert, graphic device simulation, and execution of instructional actions.

These examples also illustrate the dynamic nature of the tutoring process and the need for dynamic instructional planning. In the next section, we will consider how the BB1 Blackboard Architecture is an appropriate architecture for a dynamic instructional planner.

4. The BB1 Blackboard Architecture for Dynamic Instructional Planning

Blackboard architectures in general support knowledge-based problem solving by means of a global database to record an evolving problem solution, and independent knowledge sources that contribute to the evolving solution. An agenda records the knowledge sources that can be executed. Each record of a knowledge source and its triggering context on the agenda is called a knowledge source activation record or KSAR. A scheduler selects the next knowledge source to execute.

BB1 differs from earlier blackboard architectures (e.g., Hearsay-II [5]) in its approach to scheduling. Rather than using a fixed algorithmic scheduler, another blackboard called the control blackboard records the heuristics that form the scheduling function. By adding to and altering the records on this blackboard, BB1 can vary the scheduler, and thus the choice of knowledge sources executed by the blackboard system. Essentially, the blackboard paradigm is applied recursively in BB1 to solve *its* control problem - deciding which KSAR to execute next.

Figure 4-1 shows the execution cycle of the BB1 Blackboard Architecture with the blackboards and knowledge sources of the Blackboard Instructional Planner. Changes to the domain blackboards or control blackboard cause events. These events trigger domain and control knowledge sources. Domain knowledge sources represent instructional actions that can be performed at a moderately high level of granularity (e.g., MOTIVATE LOWER-HOIST). Control knowledge sources represent actions that affect instructional plans (e.g.,SUSPEND CURRENT INSTRUCTIONAL-PLAN).[3] Triggered knowledge sources of either

[3]Actions are represented in an English-like language described in Section 6.

kind are stored as KSARs on the agenda until they are either selected for execution by the scheduler or obviated.

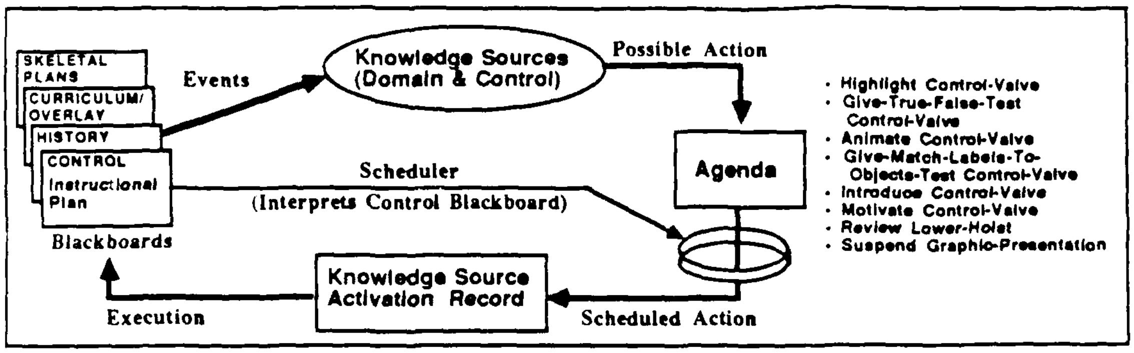


Figure 4-1: The BB1 Blackboard Execution Cycle

The scheduler interprets the control plan blackboard to determine a rank ordering of the KSARs on the agenda. Records on the control plan indicate preferences. For example, one record might indicate a preference for presentation actions or assessment actions. Another record ranks KSARs generated from control knowledge sources over those generated from domain knowledge sources. Together the records on the control plan blackboard form the pieces of a heuristic evaluation function that are weighted then summed together to prioritize KSARs.

This software architecture supports the knowledge-based problem solving required by dynamic instructional planning. Determining appropriate lesson objectives and instructional actions is an instructional problem that can be heuristically solved by the application of pedagogical knowledge. Furthermore, the explicit representation of the instructional plan on the control blackboard allows the planner to modify plans prior to execution, and to suspend, resume, or abandon plans once their execution has begun. Finally, by separating domain and control knowledge, tutorial strategies and instructional plans can be represented independently of instructional actions.

5. The Blackboard Instructional Planner

The Blackboard Instructional Planner further refines the architecture shown in Figure 4-1 by its representation of the instructional plan on the control blackboard. Consider the GRAPHIC-PRESENTATION instructional plan used in the upper half of Figure 3-2(a). Its representation on the control blackboard and its influence on the scheduling of instructional actions are shown in Figure 5-1 below. Each step in the plan specifies a category of instructional actions to favor.[4] More than one instructional action can be taken for each plan step, as explained below.

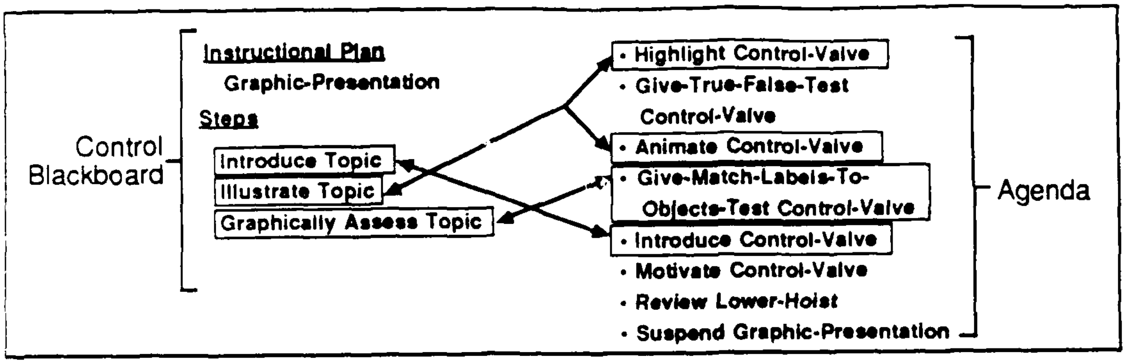


Figure 5-1: Representation and Interpretation of an Instructional Plan

The first step in the instructional plan favors actions on the agenda that match INTRODUCE TOPIC. Only the action INTRODUCE CONTROL-VALVE matches. After performing that one action for the first plan step, the next plan step, ILLUSTRATE TOPIC, becomes the current plan step. Since both HIGHLIGHT and ANIMATE are ILLUSTRATE actions (see Figure 6-1), they are both favored equally over the other actions on the agenda. Both actions are performed before moving on to the third plan step, GRAPHICALLY ASSESS TOPIC. The ASSESS category of instructional actions matches both the GIVE-TRUE-FALSE-TEST and GIVE-MATCH-LABELS-TO-OBJECTS-TEST instructional actions. The modifier GRAPHICALLY biases the scheduler to prefer assessment actions using the device simulation over those that do not, so the GIVE-MATCH-LABELS-TO-OBJECTS-TEST action is performed next. This completes the execution of the GRAPHIC-PRESENTATION instructional plan.

[4]These categories are shown in Figure 6-1 and explained in Section 6.

6. A Language for Instructional Actions and Plans

Actions that the intelligent tutoring system can perform are represented as verbs. Figure 6-1 shows a verb taxonomy developed for the class of instructional problems discussed in this paper. (Some subtrees in the figure are omitted to enhance readibility.) Associated with each verb are templates for its use. For example the GIVE-TRUE-FALSE-TEST action has this template:

GIVE-TRUE-FALSE-TEST FOR <TOPIC>

There is a similar taxonomy for nouns with categories such as TOPIC, CASE, FAULT, TUTORIAL-STRATEGY, and INSTRUCTIONAL-PLAN. Nouns are used to instantiate verb templates.

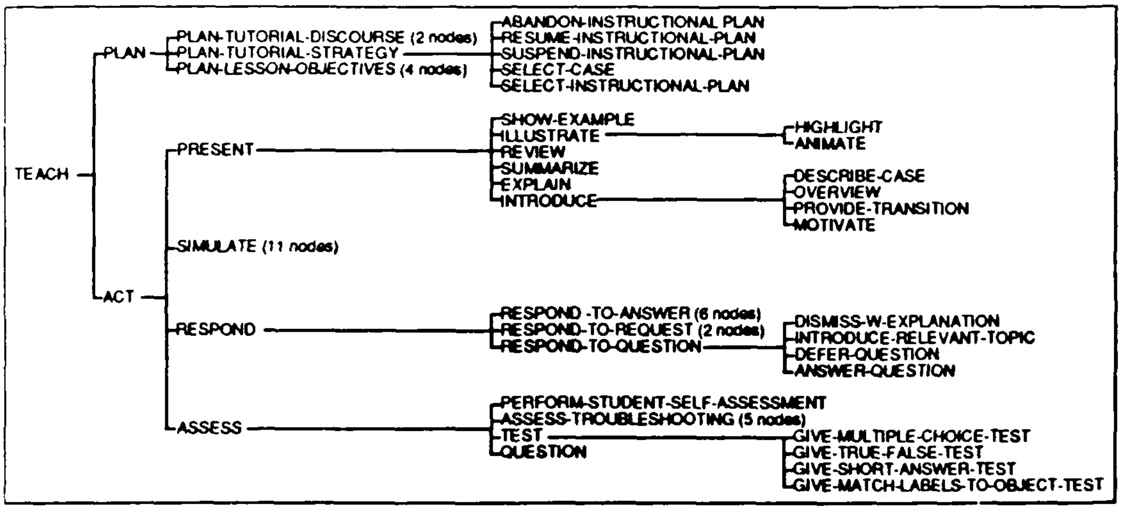


Figure 6-1: A Taxonomy of Verbs Representing Instructional Actions

Knowledge source actions are represented by sentences that consist of only terminal nouns and verbs, e.g., HIGHLIGHT CONTROL-VALVE. Steps in instructional plans need only partially specify actions to perform, so they can use nonterminal nouns and verbs, e.g., PRESENT TOPIC. For these two sentences the knowledge source action would match the instructional plan step since HIGHLIGHT is a PRESENT action and CONTROL-VALVE is a TOPIC. The language also allows modifiers for both verbs and nouns to occur in plan steps. These allow finer distinctions as shown earlier in Section 5.

The language is not domain specific, rather it is intended for any instructional problem where the tutor first teaches a mental model of the device using a device simulation, and then teaches troubleshooting using the mental model and device simulation. On the other hand, the current language is not expressive enough to represent coaching tutorial strategies for an exploratory learning environment nor is it fine grained enough to represent Socratic question and answer strategies.

This language supports dynamic instructional planning by allowing action sequences to be only partially specified and by allowing context-sensitive preferences to be expressed. By only partially specifying action sequences, i.e. by specifying categories of actions such as RESPOND-TO-QUESTION the planner can defer committing to particular actions until the tutorial situation, which is changing, is known. Different actions are available in different tutorial situations. For example, the action INTRODUCE-RELEVANT-TOPIC is available only if the student's question addresses a topic that the tutor intends to cover later in the lesson. Thus in one tutorial situation the plan step RESPOND-TO-QUESTION QUESTION ABOUT REQUESTED TOPIC may match INTRODUCE-RELEVANT-TOPIC LATCH-VALVE-ASSEMBLY but in another situation that action may not be available and the action chosen might be ANSWER-QUESTION. Modifiers can also be context-sensitive, supporting the choice of actions appropriate to the changing tutorial situation. The modifier UNSELECTED, for example, in the plan step SELECT-CASE UNSELECTED CASE causes the action SELECT-CASE CASE2 to be preferred over SELECT-CASE CASE1 if CASE1 has already been discussed.

7. Related Work

This section first considers approaches to control that are not planner-based and then considers other dynamic instructional planners that do not use the blackboard architecture. Approaches that are not planner-based include the use of rules, networks, and algorithms. A rule-based approach to control for Socratic question and answer is exemplified by the WHY [14] system. In the GUIDON system [4] tutorial rules are metarules that allow opportunistic tutoring during case-method instruction. In MENO-TUTOR, skeletal plans for instruction are expressed as default state transitions through an ATN-like discourse management network. The network is augmented with metarules that allow opportunistic transitions from the execution of one skeletal plan to the execution of another.

All the systems above are not dynamic instructional planners even though they perform sophisticated control for intelligent tutoring systems. They lack an explicit plan representation generated and interpreted *during* the tutorial session and modified when replanning occurs. This lack may limit their ability to utilize non-opportunistic tutorial strategies such as expository presentation of material under resource constraints. At present, all of these systems concentrate on opportunistic tutorial strategies such as Socratic question and answer or case-method. In general, these rule-based approaches to control and simpler algorithmic approaches (e.g., BIP [2]) appear sufficient for tutors relying on a fixed set of opportunistic tutorial strategies and instructional plans, and where the replanning required is limited and can be anticipated. However, algorithmic approaches and rule-based approaches that do not separate domain and control knowledge are difficult to extend to cover new domains, strategies, and plans.

A planner-based approach to instruction based on the application of classical (i.e. STRIPS-based [6]) planning techniques to intelligent tutoring systems is presented by Peachey and McCalla [12]. Instructional actions have a coarse granularity (e.g., TEACH-X) and are represented with ADD and DELETE lists in standard STRIPS notation. Plans tend to be unnecessarily elaborated prior to instruction and the representation of actions makes strong assumptions (e.g., predictability) not appropriate to instructional actions. Replanning occurs to patch plan failures, but not to take advantage of new opportunities for plan improvement. However, this research is a significant contribution in its early application of planning to intelligent tutoring systems, and the attention given to the problem of replanning.

IDE-Interpreter [13] also adopts a planner-based approach to instruction to interpret an instructional design developed in IDE, the Instructional Design Environment. An explicit plan representation is incrementally refined by rules. Goals for plan refinement are maintained on an agenda; goal selection is performed heuristically. The plan is represented as a tree that is refined top-down. Portions are elaborated only as necessary for instruction to begin. The use of rules, explicit plan representation, and an agenda provides an architecture similar to the blackboard system. However, the Blackboard Instructional Planner provides even more flexibility by allowing the planner to revise its heuristics so it can plan its own planning and replanning behavior. The blackboard architecture also allows instructional planning to combine opportunistic, goal-driven, and hierarchical planning [10]. At present these capabilities have not yet been exploited and the two planners have similar capabilities, albeit different architectures.

The research presented here continues previous research by MacMillan and Sleeman [11]. Their work focussed on architectural considerations in designing a dynamic instructional planner, and an approach to dynamic instructional planning where instructional plans are represented hierarchically on a domain blackboard. No language framework, such as the one discussed in Section 6, is used to express instructional actions or plans. Plans on the control blackboard control the incremental assembly, execution, and revision of these instructional plans. In contrast, the approach presented here relies on skeletal plans for instruction, represented on the control blackboard, and using the language framework discussed in Section 6.

8. Summary and Research Contributions

The first key point in this paper is that dynamic instructional planning is an effective approach to control for intelligent tutoring systems. This planning approach allows the selection of instructional actions appropriate to the tutor's lesson objectives, student model, tutorial strategy, and taking into account student questions and requests.

Secondly, plans for a dynamic instructional planner can be represented in a perspicuous English-like formalism. This declarative representation for plans can be used to express skeletal plans whose steps only partially specify instructional actions, deferring complete specification until the exact tutorial situation is known. The language used in the Blackboard Instructional Planner is appropriate for a class of instructional problems and is still evolving.

The language and planner together provide a medium to develop and test instructional theories. Tutorial strategies and instructional plans can be prototyped and refined in the Blackboard Instructional Planner. Later, these strategies and plans can be ported to simpler rule-based or algorithmic architectures if desired.

Finally, the major contribution of this research is the design and implementation of a dynamic instructional planner in the BB1 Blackboard Architecture that relies on a language framework for representing instructional plans and actions.

9. Acknowledgements

Perry Thorndyke first proposed the blackboard architecture for dynamic instructional planning. Stuart Macmillan and Derek Sleeman later proposed a blackboard architecture for dynamic instructional planning with the ability to improve its own planning behavior. The work presented here builds on their research and that of Barbara Hayes-Roth on BB1 and the language framework ACCORD [8]. I would like to thank N.S. Sridharan, Perry Thorndyke, Barbara Hayes-Roth, Stuart Macmillan, Marshall Harris, Lee Brownston and Jens Christensen for reviewing drafts of this paper. The Office of Naval Research funded this research under contract N00014-86-C-0487, serial number SIIP-1287-BM-041.

References

[1] Hayes-Roth, B.
A Blackboard Architecture for Control.
Artificial Intelligence 26(3):251 - 321, 1985.

[2] Barr, A., Beard, M., and Atkinson, R.
The Computer as a Tutorial Laboratory.
International Journal of Man-Machine Studies (8):567 - 596, 1976.

[3] Burton, R.R., and Brown, J.S.
An Investigation of Computer Coaching for Informal Learning Activities.
International Journal of Man-Machine Studies (11):5 - 24, 1979.

[4] Clancey, W.
Tutoring Rules for guiding a case method dialogue.
International Journal of Man-Machine Studies (11):25 - 49, 1979.

[5] Erman, L.D., Hayes-Roth, F., Lesser, V.R. and Reddy, D.R.
The Hearsay-II Speech-understanding System: Integrating Knowledge to Resolve Uncertainty.
Computing Surveys (12):213 - 253, 1980.

[6] Fikes, R.E., and Nilsson, N.J.
STRIPS: a New Approach to the Application of Theorem Proving to Problem Solving.
Artificial Intelligence 2:189-208, 1971.

[7] Hayes-Roth, B. and Hewett, M.
BB1: An Implementation of the Blackboard Control Architecture.
Blackboard Systems.
Addison-Wesley, 1987.

[8] Hayes-Roth, B., Garvey, A., Johnson, M.V., and Hewett, M.
A Modular and Layered Environment for Reasoning about Action.
Technical Report KSL 86-38, Stanford University, April, 1987.

[9] Hollan, J.D., Hutchins, E.L., and Weitzman, L.
STEAMER: an interactive inspectable simulation-based training system.
AI Magazine 5(2):15 - 27, 1984.

[10] Johnson, M.V., and Hayes-Roth, B.
Integrating Diverse Reasoning Methods in the BB1 Blackboard Control Architecture.
In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 30 - 35. Morgan Kaufmann Publishers, Inc., Los Altos, CA., July, 1987.

[11] Macmillan, S.A., and Sleeman, D.H.
An Architecture for a Self-improving Instructional Planner for Intelligent Tutoring Systems.
Computational Intelligence 3(1):17 - 27, 1987.

[12] Peachey, D.R., and McCalla, G.I.
Using Planning Techniques in Intelligent Tutoring Systems.
International Journal of Man-Machine Studies 24:77 - 98, 1986.

[13] Russell, D. M.
The Instructional Design Environment: The Interpreter.
Intelligent Tutoring Systems: Lessons Learned.
Lawrence Erlbaum Associates, Inc., 1987.

[14] Stevens, A.L., and Collins, A.
The Goal Structure of a Socratic Tutor.
In *Proceedings of the National ACM Conference*, pages 256 - 263. Association for Computing Machinery, 1977.

[15] Wenger, E.
Artificial Intelligence and Tutoring Systems.
Morgan Kaufmann, 1987.