Computers and Electronics in Agriculture, 5 (1991) 265–283
Elsevier Science Publishers B.V., Amsterdam

A computer-based tutorial structure for teaching and applying a complex process

Daniel L. Schmoldt[a,1] and William G. Bradshaw[b,2]
[a]*Pacific Southwest Research Station, USDA Forest Service, 4955 Canyon Crest Dr., Riverside, CA 92507-6099, U.S.A.*
[b]*Fiscal and Public Safety, USDA Forest Service, 201 14th Street, S.W. Washington, DC 20250, U.S.A.*

(Accepted 2 August 1990)

ABSTRACT

Schmoldt, D.L. and Bradshaw, W.G., 1991. A computer-based tutorial structure for teaching and applying a complex process. *Comput. Electron. Agric.*, 5: 265–283.

Economic accountability concerns for wildfire prevention planning have led to the development of an ignition management approach to fire problems. The Fire Loss Prevention Planning Process (FLPPP) systematizes fire problem analyses and concomitantly establishes a means for evaluating prescribed prevention programs. However, new users of the FLPPP have experienced difficulty with both understanding and implementation. Consequently, an on-line tutorial is being developed for delivery of the rationale and methods of the FLPPP. This format takes the form of a knowledge-based intelligent tutoring system. The flexible design of the system permits several types of knowledge to be easily included. Mixed initiative interaction between user-student and system affords the student substantial control in exploring areas for which he or she may require a tutor. A close look at tutoring systems is recommended as an alternative to hard-copy documents for teaching individuals in an unfamiliar domain.

INTRODUCTION

Planning for wildfire protection is a matter of deciding on the distribution of personnel and equipment resources before ignitions occur, based on historic fire patterns and impacts. The decisions include choices on how much emphasis should be placed on ignition prevention, fuels management, or initial attack forces. Recent changes in fire protection policy of the USDA Forest Service have stressed the point that any fire management option applied should be based on its economic efficiency; that is, total protection costs plus

[1]Present address: Forest Products Laboratory. One Gifford Pinchot Drive, Madison, WI 53705. U.S.A.
[2]Formerly with the Pacific Southwest Research Station, Riverside, CA 92507-6099, U.S.A.

0168-1699/91/$03.50

the level of damages incurred under that protection program (losses) should be the lowest of all available options (Gale, 1977).

This principle has been slow in coming to planning for wildfire *prevention* programs, in part because a systematic approach to program planning has been lacking. Not only do prevention costs require careful accounting, but benefits are difficult to tally because the aim of prevention work is to minimize wildfire occurrence. That is without some a priori organizational planning, failure cannot be defined as the occurrence of some undesirable event (a wildfire) and success is not necessarily guaranteed given the absence of that event (success is dependent on factors such as climate and not just prevention efforts).

Analysis tools are becoming available, however, to help fire managers in both long term and short term planning. Among the tools applicable to prevention planners is a new approach to planning called the Fire Loss Prevention Planning Process (FLPPP), developed in 1985. The process provides methods for analyzing the fire problems an administrative unit may face. Its basic tenet is to target prevention programs toward those ignitions with the greatest potential to become large or damaging wildfires or both.

A major step towards developing evaluation methods for prevention programs is producing a planning system that sets measurable goals; this is accomplished in the FLPPP. The next step is to develop documentation and training to introduce potential users to the planning process which, as is often the case, differs from what users have done traditionally. Preliminary information indicates that publications are an inefficient means of introducing personnel to the planning system and guiding its application (R. Johnson, personal communication, 1987). Even as users apply the process correctly, they have questions about what they are doing. An interactive presentation, similar to the dialogue of advisor and advice seeker, may be more beneficial.

As with many other management tools, the FLPPP is amenable to computer-based implementation. However, in its present form, it has only been used operationally on the San Bernardino National Forest in California, U.S.A. The FLPPP resides in the minds of a few fire prevention personnel and, indirectly in the prevention programs that have been implemented on the forest. The research described in this study examines what it takes to produce a knowledge-based intelligent tutoring system to assist people in learning and applying a systematic planning tool such as the FLPPP. Development of the system outlined in this paper includes both encoding FLPPP knowledge and providing the algorithms necessary to use the knowledge in a tutoring environment. While the system described here is analogous to a training manual and, hence, is basically an on-line tutorial, it differs from a document tutorial by virtue of its interactive nature and dynamic knowledge. For these reasons, we feel it is more correctly described in an anthropomorphic sense as *tutor* rather than tutorial.

The intended users of this tutorial package are prevention personnel at the field level in fire protection organizations (local, state, and federal). The intent of the FLPPP itself is to develop and administer a prevention program, within appropriated budgets, based on a systematic analysis of fire problems. And, because a function of the tutor is the transfer of analysis and planning skills, implementation of the tutor should not be difficult. That there is a market for the tutorial system is already known. People who have tried implementing the FLPPP have experienced difficulty and self doubt, as to whether what they were doing was 'right' (R. Johnson, personal communication, 1988). An interactive training/transfer tool would be welcomed.

The study reported here, known as Tutoring Wildfire Prevention Planning, is the second of two research stages whose long-term goal is development of a computer-based process for training personnel in prevention planning. Stage 1, Ignition Pattern Diagnosis, consisted of eliciting the prevention planning knowledge and skills of personnel familiar with application of the FLPPP. Knowledge of the planning process was (1) gathered from several prevention personnel involved in developing the process and (2) systematized through identification of the logical structure of the planning process.

Stage 2 codifies that knowledge into a knowledge-based expert system *tutor* for application of the planning system. This computer-advisor approach has previously demonstrated effectiveness at capturing expertise for training others and transferring new technologies (Schmoldt and Martin, 1989).

Stage 2 also targets the construction of a tutorial program to guide prevention planners through (and, in some cases, introduce them to) the systematic fire problem analysis and program planning components of the FLPPP. Building the tutor from information gained in the Ignition Pattern Diagnosis study offers further opportunity to study intuitive problem solving skills. Some prevention planning details overlooked in the previous study may require some additional elucidation as this subsequent work proceeds. Opportunities will also arise, as the tutor is constructed, to test various mathematical modelling processes (e.g., fire behavior modelling) to see how well they might fit into the FLPPP analyses, relieving the user of some analytical burdens.

This paper focuses on the structure of FLPPP knowledge, its representation, and its use within a tutoring environment.

KNOWLEDGE OF WILDFIRE PREVENTION PLANNING

The planning process for wildfire loss prevention consists of the sequence of steps depicted in Fig. 1. The four major steps of the FLPPP appear on the left-hand side of the figure and must be performed in that order. The primary advantage of this planning approach is its systematic analysis of fire problems. Subsequent steps rely on inventories and maps produced in the analysis phase.

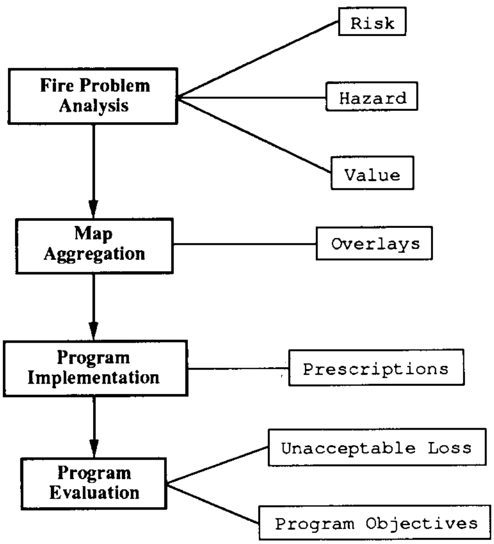

Fig. 1. Boldface components represent stages of the Fire Loss Prevention Planning Process and the sequence in which they must be performed. Attached components on the right-side side of the illustration constitute information sources used or documents produced in each stage.

Analysis of wildfire ignition problems, the first step in the FLPPP, involves the examination of three independent aspects of the wildfire situation: risk, hazard, and values at risk. *Risk analysis* examines human activities that have the potential to cause ignitions. *Hazard analysis* estimates resistance to control should a wildfire occur. Ratings of resistance to control on contiguous areas are identified on a hazard map. *Value analysis* focuses on monetary and non-monetary value changes resulting from fire. Each analysis is performed independently, e.g., analysis of the potential for ignition is not affected by the fire behavior analysis. Because of this independence, no particular order need be followed although each step must be included for a comprehensive analysis.

Map aggregation integrates the information produced in the analysis step to produce relatively autonomous management compartments. Documentation from each of the fire problem analyses are synthesized into a comprehensive description for each compartment. This summary becomes a companion document to the aggregation map.

Subsequent to the acquisition and aggregation of fire problem information, case-specific decisions create prescriptions to address the specific problems identified. Because each organization and their administrative units have differing resources and policies for prevention work, program implementation details need to be made at that level. The FLPPP analysis makes suggestions for prescriptions and provides a framework for their accounting.

As a fire season progresses, it is possible to compare fire losses with program prescriptions. Where programs are not meeting prevention needs, shortfalls can be analyzed and corrective measures taken. This aspect of the process is made more explicit below.

This synopsis has provided a cursory description of the FLPPP. An example is presented below which contains more detail and will fill in some of the gaps in the brief outline above.

A planning example

The knowledge elicitation methods of Stage 1 not only identified the basic structure of the FLPPP (Fig. 1) but also brought out the intuitive components of each analysis step. Generically, fire problem analysis steps are these: (1) inventory the constituent factors of the analysis, i.e., risk, hazard, and values at risk; (2) map the location of the factors; (3) set criteria for rating the relative contributions of the factors; and (4) delineate and rate areas of homogeneous risk or hazard or value.

Risk analysis, for example, begins with a ranked listing of wildland uses that may produce ignitions (based on current and historical use patterns). Such a listing might include the following: fishing, residential development, power lines, dispersed camping, firewood cutting, roadways, and developed recreation areas. The relative contribution of each factor to total risk depends on levels of activity and probability of producing firebrands (ignitions). Inventoried factors must then be mapped as in Fig. 2. Areas of homogeneous risk can then be delineated on an overlay of the inventory map based on the prior inventory and mapping, a ranking of risks, their level of activity, and their respective likelihood for ignitions. Areas with elevated risk, labelled for example as moderate or high risk, are located using the above inventory information; the remaining areas become low risk by default. In practice, prevention personnel may use any number of levels that seems useful and permits interpretation.

Hazard analysis, in contrast to risk, inventories factors related to wildfire behavior (vegetative fuel type, topographic features). Hazard rating criteria are more quantitative and can be established by using calculator-based models to estimate fire behavior parameters. These parameters may group certain fuel-slope mixtures according to resistance to control by suppression forces. Using an overlay with the inventory map, areas of homogeneous hazard may be identified and delineated.

The third overlay to be produced contains values at risk. In some ways, the assessment of values may be the most difficult analysis step because of the diversity of inputs which must be considered. Timber, social, ecological, archaeological, watershed, and wildlife are some of the component values that may be modified by wildfire. Often, this assessment will require consultation with various subject matter experts trained in these areas.

After all analyses have been completed, the three overlays are aggregated into one comprehensive map of the entire administrative area (Fig. 3). The

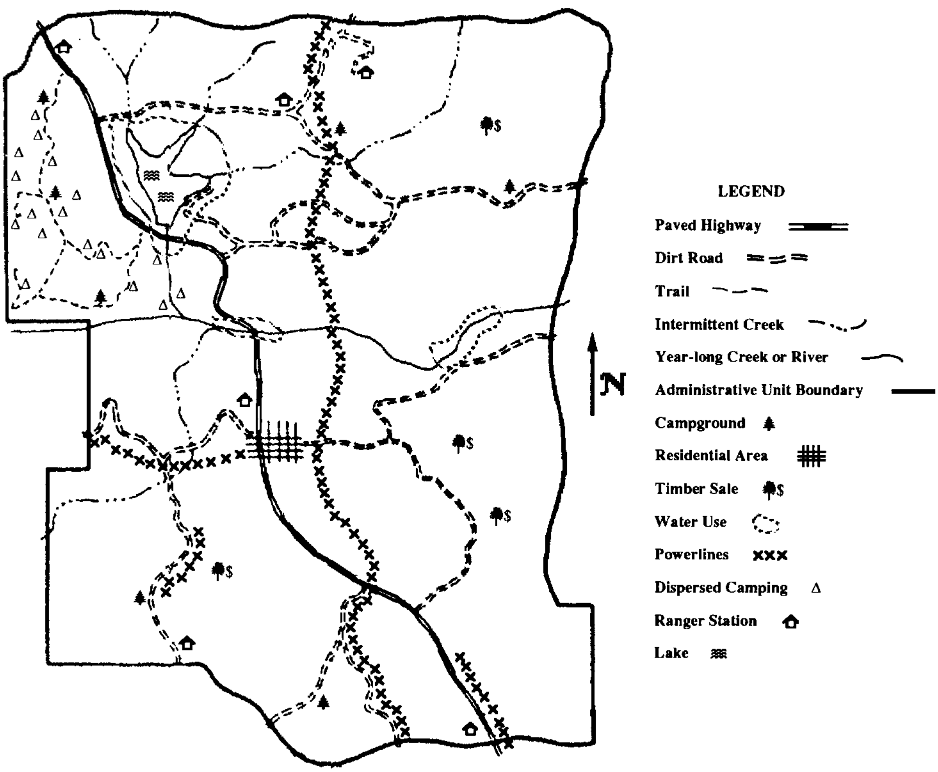

Fig. 2. This example administrative unit contains an inventory of risks similar to what a prevention officer might construct.

final fire prevention compartments are initially bounded by the risk analysis areas because ignition potential is the major concern of prevention programs. The planner rates each compartment for risk, hazard, and value based on the analysis overlays. The example of Fig. 3 only used three possible rating scores: low, moderate, and high.

In some cases, preestablished boundaries set by risk areas may need to be further dissected if there are major variations in hazard or value rating. The planner then has a final inventory that summarizes and documents, in text form, specific people activities, fuel conditions, and values being protected in each compartment.

Aggregate map and fire analysis documentation form the data base for subsequent implementation planning and follow-up evaluation. Currently, creation of the maps remains a manual task; however, it is not difficult to envision future utilization of geographic information systems for storage, visualization, and manipulation of this spatial information.

From the combined inventory and aggregate map, specific action items are

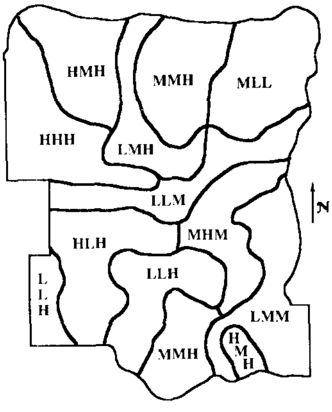

Fig. 3. An aggregation of the risk, hazard, and value overlays results in management areas with a rating for each constituent analysis. High, moderate, and low ratings are identified by H, M, and L, respectively. This mapped summary and its accompanying document serve as the basis for prescription writing designed to target these problems.

identified for each management unit. Prescriptions are written incorporating each of the prevention action items, which cover duties to be performed. These prescriptions may vary for ranges of fire conditions used in fire planning.

Evaluation of program effectiveness entails examining prescriptions in relation to the extent and location of fire losses. Because the goal of prevention is a non-event (no fire), successful prevention efforts can only be assessed in terms of the quantity and severity of events that do occur, i.e., unacceptable losses. When unacceptable fire losses have not occurred, then, assuming average fire conditions, it may be possible to conclude that prevention efforts have been successful.

Faced with losses considered unacceptable by land managers, however, a careful examination of those specific, damaging fire problems should identify the most prominent causes. Prevention personnel may refer to the program implementation document to verify completion of all prescribed tasks; failure to achieve specified tasks indicates a performance problem. Unacceptable losses occurring after tasks have been performed indicates that prescriptions may not be effective and need to be reconsidered. In the course of rewriting prescriptions, it may be necessary to reanalyze the risk, hazard, or value assessments associated with particular areas of losses.

A brief discussion of the different types of knowledge represented in the tutor follows to provide the reader with some feel for the organization and interaction of prevention planning knowledge.

System knowledge

Four forms of knowledge in the planning process were elicited from prevention personnel in Stage 1: structural (conceptual model), declarative, analyt-

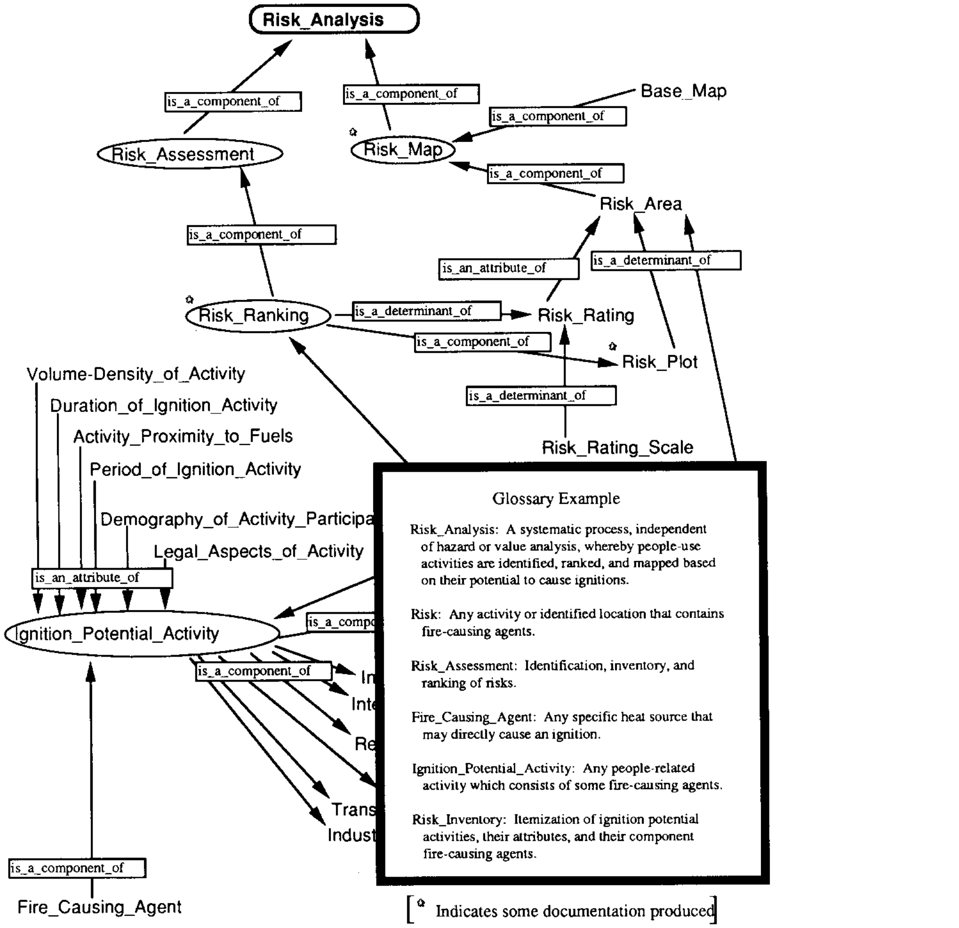

Fig. 4. In discussions with personnel familiar with the Fire Loss Prevention Planning Process, knowledge was elicited similar to that in this concept diagram for risk analysis. Concepts, tasks, and their relationships (arrows) are included. A portion of the diagram is obscured by the insert which contains several example definitions from the glossary for this diagram.

ical, and procedural. Methods used to collect and organize this knowledge have been detailed in Schmoldt and Bradshaw (1988). An example of *structural* and *declarative* knowledge is presented in Fig. 4. Concepts and tasks central to prevention planning are labelled and their inter-relationships illustrated by arrows. The textual insert depicts a simplified and abbreviated glossary which may contain declarative knowledge in the form of definitions, justifications, and examples of concepts.

During the course of using the FLPPP, various tasks need to be performed by the user, e.g., creating an inventory of risks for the planning area. Much of this script-like knowledge (Schmoldt, 1989) was recorded on audio tapes and is being formalized into coaching instructions. Analyzing hazard involves estimating fire behavior under certain weather and fuel conditions; this problem has been addressed mathematically (Andrews, 1986) and can be included in the tutor as a hidden analytical method with which the user need not be directly involved. Incorporation of different knowledge types will provide the user-student with complementary aspects of complete prevention planning knowledge.

However, certain gaps still remain in the domain knowledge that has been extracted to date. These gaps occur where the experts have some heuristic or intuitive knowledge about a particular concept, which may depend on the value of other concepts, such as inventorying factors or setting rating criteria in each analysis. This subjective knowledge is much more subtle and nebulous than structural and declarative expertise. If the tutor contained only the knowledge described above and provided a worksheet type of environment for developing prevention plans, it would be a useful tool for prevention personnel. But, inclusion of the subjective judgment of individuals especially knowledgeable in the FLPPP would produce a more complete specification of FLPPP knowledge and possibly eliminate some of the user-student's efforts where shortcuts or alternatives may be available. Plans are for the completed tutorial system to incorporate this aspect of the experts' knowledge.

INTELLIGENT TUTORING SYSTEMS

Computer-aided instruction began in the 1950's with behavioral psychology's 'linear program' view of instruction (Yazdani, 1986). Here, direction toward some desired behavior was intimated, and reinforcing stimuli result from a successful response. Since that time, many other approaches have been used, most recently, knowledge-based systems.

Besides being effective for problem-solving, knowledge systems contain some desirable properties for instructional uses. Expert judgement and intuition can be represented in the system, and conveyed to the student through tutored examples (example-driven problem solving) or declarative explanation (line of reasoning). The student can work many examples at varying complexity levels without an instructor's attention and possible exhaustion. Clancey (1985) transformed the diagnostic expert system, MYCIN (Davis et al., 1977), into a tutor for medical students. Other tutors have been created for teaching various subjects: computer programming (Anderson and Reiser, 1985; Johnson and Soloway, 1985), natural language (Imlah and Du Boulay, 1985), geometry (Anderson et al., 1985), physics (White and Frederiksen, 1986), and a pulp and paper process (Woolf et al., 1986). Merging computer-aided instruction and knowledge systems has resulted in intelligent tutoring systems (ITS).

Many of these ITS's have some common components, which are presented

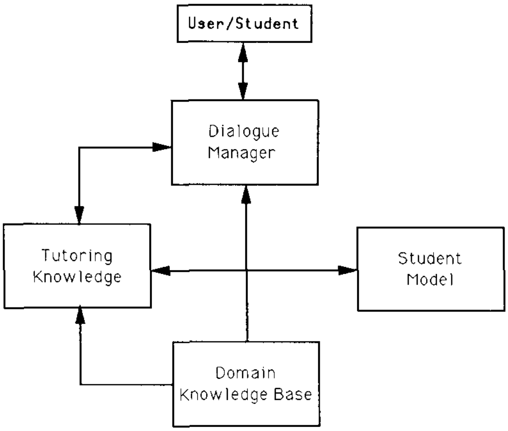

Fig. 5. A component diagram of the Fire Loss Prevention Planning Process tutor depicts the modules and the flow of information between them. The box labeled, user/student, represents a component that is not part of the tutor.

in Fig. 5. Knowledge of the actual domain being taught must be represented (in our case, wildfire prevention planning), sometimes in multiple forms at several levels of understanding (White and Frederiksen, 1986). In some formalisms, this knowledge is referred to as the 'ideal student' model (Yazdani, 1986) or the domain model. Strategies for tutoring the student in the particular domain must also be defined. These strategies are contained in the Tutoring Knowledge and may include immediate or delayed error feedback, maintaining an actual, student history/model (Student model), and specifying the intensity of tutor and student interaction. To provide a comprehensive learning environment, the student must have access to large amounts of explanation, both for correcting erroneous behavior and promoting desired behavior. Exact implementation of these tutorial features will depend on the teaching task and the types of students anticipated.

The ITS approach considered here differs from those described above by being a non-singular use system. Most ITS's contain a method of simulating or hypothesizing specific learning situations (examples). The FLPPP tutor will operate with examples provided by the user, which, presumably, will reflect actual prevention planning needs. In this way, the created knowledge-based system will help solve actual ignition management problems and, at the same time, will tutor fire prevention personnel in the use of the planning process.

Development of a tutor is, essentially, producing an interactive training/transfer mechanism. The flexible, step-by-step format of the ITS approach will provide for simultaneous learning of the planning process as well as for management of actual ignition problems. Learning is facilitated by the transparent nature of knowledge-based systems. The reasoning steps are open to examination by the user at any stage of the problem-solving process. Why particular information is requested of the user and how specific decisions are made can be explained at the user's request. This open, problem-solving architecture differs drastically from quantitative models which lack accountability and often obscure logical structure.

KNOWLEDGE REPRESENTATION AND CONTROL

Following distillation and representation of ignition management knowledge accomplished in the "Ignition Pattern Diagnosis" study, a tutoring system is now being constructed. To implement this tutorial, it becomes convenient to separate the tutor into modules. These are designated in Figure 5 as the domain knowledge base, student model, tutoring knowledge, and dialogue manager. This modularity permits incremental development and modification. A fully-functional tutor, which includes only a subset of the FLPPP knowledge (e.g., the risk analysis knowledge), may be prototyped and tested before the complete system is implemented. Consequently, problems in system design can be identified and corrected before there is system-wide dependence on faulty components.

Components of the ITS in Fig. 5 possess no particular one-to-one relationship to the components of a general knowledge-based system (KBS). A knowledge base, a working memory, or an inference engine may be present within several of the ITS components. For example, the dialogue manager serves as a natural language front-end; it contains production-rule-like templates as well as methods for matching stored templates with input and producing responses (see Dialogue Manager, below). Also, the prevention planning knowledge module contains static FLPPP knowledge extracted from experts, if-then rules extracted from experts, and dynamic information specific to a current problem which the user must provide. The knowledge system paradigm tends to become somewhat diffused within the functional nature of ITS structures.

Our tutorial design adheres to the idea of arbitrary-depth tutoring. The user-student determines, by questions and responses, what topics will be investigated and how intensely they will be explored. So, the tutor operates at whatever depth of understanding the user selects – directly, by questions posed to the tutor or, indirectly, by responses to tutor question. The domain knowledge base is organized around generic planning process ideas and tasks and can be used as a template to guide the user in personal prevention plan creation. Rule-based knowledge may be used to assist the user in completing particular aspects of fire-problem analysis and prevention plan development. All components of the generic process and the component interrelationships are available for inspection by the student via explanations and definitions. Providing the user student with extensive control over the tutoring session should increase acceptance of the system and ultimately promote the use of FLPPP ideas.

Domain knowledge base

A frame-based knowledge representation (Minsky, 1975) is the basic implementation structure used in the tutor. A frame can be viewed as a box containing many separate compartments. Each compartment holds some relevant information (usually attributes and their values) about the object being represented. Since their introduction, frames have been used in many expert systems (Barr and Feigenbaum, 1981) and in many shells (Harmon et al., 1988) for organizing knowledge into discrete units. As noted previously, our knowledge base contains several types of knowledge: structural, declarative, analytical, and procedural (some remains to be acquired from the experts). Each concept or task in the FLPPP is represented as a frame of knowledge; procedural rule sets and individual rules are also stored as frames. Contemporary, fire-management examples of frame-based knowledge representation appear in Fischer (1987) and Reinhardt (1987). An example frame for the concept 'ignition potential activity' is illustrated here:

FRAME: ignition-potential-activity
print-name "Ignition Potential Activity"
author-date "D. Schmoldt 3-15-88"
type concept
definition "Any people-related activity..."
justification "Specific people-related activities..."
relationships ((contains fire-causing-agents)...)
examples (water-use-campgrounds...).

This frame contains a number of attributes of 'ignition potential activity' (e.g., type, definition, and relationships) and values of those attributes. Declarative knowledge is stored as text between double quotation marks, as in the definition or justification attributes. Structural knowledge can be stored as a list of relationships and their related concepts or related tasks, e.g., *ignition-potential-activity* 'contains' *fire-causing-agents*. Also, specific examples of ideas to be learned are often extremely important in teaching – more-detailed examples of concepts may be included in a list of 'examples' attribute values, which may name other concept frames, e.g., *water-use-campground. Water-use-campgrounds* would then be a subclass of the *ignition-potential-activity* class and might eventually be linked to specific water-campground instances provided by the user. From these examples, it is apparent that fairly elaborate and interrelated concepts may be created using classes, subclasses, and instances of classes and subclasses. Other frames, which represent tasks (sequences of steps to be performed) in the planning process, might have an additional attribute in the place of an 'examples' attribute; this additional attribute might contain executable code (a *method*) to guide the user in completing some particular task.

As a particular user develops a prevention program, instance-specific information will be created by the user and stored as instances of more general, class and subclass frames. The construction of these specific instances can be performed by the program and, therefore, become entirely transparent to the user. The class-instance hierarchy trait of the frame representation lends itself well to the model of generic-to-specific objects of the FLPPP. This relationship coincides with the general notion posited by the frame knowledge representation for storing categories of knowledge and typical expectations of our experiences with those objects (Barr and Feigenbaum, 1981).

Where human expert judgment is useful at particular decision points in the planning process, it would also be desirable to attach if-then decision rules to still other attributes of a frame. These rules could be invoked to make decisions which the inexperienced user may be unable to do, e.g., infer the value of certain concepts or attributes from the values of others. An experienced prevention officer would probably not need this subjective guidance, except perhaps to ensure that his or her assignments are consistent across various instances. More likely, less experienced prevention personnel and foresters untrained in prevention should find it helpful as they develop specific plans.

A useful variant of the frame idea called a *script* was developed by Schank and Abelson (1977). Scripts are identical in structure to frames, but contain scenarios (expectations) of events and actions regarding some situations. While frames are concept oriented, scripts are event oriented. This capability makes them advantageous for storing and referencing planning process tasks which are often sequences of events. These scripts may also include attached procedures (methods) and rules much like frames.

Generalized frames (scripts and frames) allow the knowledge engineer (i.e. the constructor of the knowledge-based system) to collect, store, and catalogue knowledge regarding an individual object in a single, data structure locality, and yet, relate it to many other objects in the knowledge base.

Student model

The tutor contains two representations of a student's understanding at any point during a tutoring session. An *actual* student model records what the user-student has apparently learned. In contrast, a *speculative* student model stores a list of those concepts and tasks that can be assumed to be understood by the student from apparent understanding of related concepts and tasks. For example, if the user professes an understanding of the concept *ignition-potential-activity*, then some cognizance of the concept *fire-causing-agent* can be assumed also (Fig. 4). This student model dichotomy allows the tutor to maintain and use an awareness of what the user indicates knowing (professes) and what the user may be assumed to know.

As the planning process tutor is engaged by a user-student, a student model is created and represents the level of understanding and proficiency exhibited by the current user. FLPPP concepts that have been understood and planning tasks that have been accomplished by the user are recorded in the student model. As new aspects of the domain are learned by the user, the actual student model (represented as a list) is augmented with appropriate concepts or tasks. Then the speculative student model can be dynamically constructed from the actual student model list elements along with the tutoring productions described below (see Tutoring Strategy Model). Levels of detail provided in explanations are determined by the student model (i.e., what the user has accomplished and understood up to the current point in the session) and the learning needs of the user (i.e., what depth of planning process detail the user selects to investigate).

Benefits of this user-ability framework should include minimal hand-holding for proficient users and production of sufficient planning process detail for inexperienced users. As the student progresses in understanding ignition management planning, less-detailed and more concept-oriented explanations become appropriate. Users may even exhibit variable ability across components of the planning process; such non-uniformities can be accommodated by this structure.

Dialogue manager

A dialogue manager component permits a mixed-initiative interaction between student and tutor. Rather than allow the program to guide the tutoring process exclusively, a more student-oriented approach gives the user some control over his or her speed of comprehension and depth of understanding. At any time when the student is prompted for input, he or she may initiate a dialogue session by asking a question. When no more questions are offered by the user, the suspended tutoring process may be resumed. Most of the dialogue between user and tutor will translate into queries of the knowledge base; hence, the direct connection of dialogue manager and knowledge base depicted in Fig. 5.

Dialogue implies some form of natural language interface, much like ordinary conversation. Because the goal of this ITS is the transfer of the FLPPP and not natural language processing as a research topic, a simple but effective template-matching parser/responder was selected. This parser matches inputs from the user to query–response templates in a dictionary. This approach was incorporated in the early language understanding programs of Raphael (1968) and Bobrow (1968). This approach operates well in situations where the topic of discussion is narrow and many sentence structures and contents can be anticipated by the dictionary designer (Barr and Feigenbaum, 1981). These requirements appear to be satisfied due to the narrow focus of the FLPPP application.

The dialogue manager contains numerous query-response templates similar to the one presented here:

((WHAT (BEING-VERB-P X) THE (? Y) OF (* Z))
(REPLY 'ATTRIBUTE Y Z)).

When the user asks a question, the dialogue manager attempts to match the input question with the query part of each template, the first template to be tried in this case, (WHAT (BEING-VERB-P X) THE (? Y) OF (* Z)). Particular words (e.g., WHAT) must match exactly in their exact position with words in the input. The predicate (BEING-VERB-P) matches any verb of being of the input, e.g., 'is' or 'are', and assigns the word matched from the input to the variable X. Terms (? Y) and (* Z) are variable matches. The '?' is a wildcard, which matches any single word of the input and assigns that word to Y, and the '*' matches any number of words of the input and assigns that string of words to Z. The first query template that matches the input results in the activation of the associated response, in this case, (REPLY 'ATTRIBUTE Y Z). For, example, the input 'WHAT IS THE DEFINITION OF IGNITION POTENTIAL ACTIVITY' would match the above query-response template. So, the reply would list the attribute named in Y (DEFINITION) of the concept named in Z (IGNITION POTENTIAL ACTIVITY). For simple dialogue within a restricted topic, a relatively small number of templates should provide complete access to declarative, procedural, and structural knowledge in the knowledge base.

Tutoring knowledge

A tutoring strategy model operates on top of the domain knowledge to direct the tutoring process and to coordinate the activities of the other components. The essential structure driving the operation of this model is a set of pre-condition productions. The production technique has been applied to representing tutoring strategies (O'Shea, 1982; Heines and O'Shea, 1985; Darwyn and McCalla, 1986). In our implementation each production consists of pre-conditions and one learned effect. For example, in the production

(risk-analysis risk-assessment risk-mapping),

'risk-analysis' will be effected following the accomplishment of pre-conditions 'risk-assessment' and 'risk-mapping'. Similarly, there are productions which effect each of these pre-conditions in turn. The result is a step-wise refinement transition toward more detailed concepts and tasks. Productions vary from high level effects, such as 'risk-analysis', to more detailed effects, like 'ignition-potential-activity-description'. When the level of detail reaches the capabilities of the current user-student, then his or her concept understanding and task performance will accumulate learned concepts in the student model which will eventually lead to the accomplishment of higher-level concepts. Decomposition of higher levels tasks and concepts into lower-level, more atomic ones is similar to the *procedural net* representation of planning (Sacerdoti, 1977). In fact, the hierarchical and associative links between our frame representation of concepts and tasks also permits access to this type of decomposition. However, in our implementation the user determines the level of detail which the system unravels, and specific planning details must be supplied by the user because of location- and organization-specific implementation considerations.

Actual tutoring in this study involves two aspects of instruction; these have been labeled *exposition* and *coaching*. Planning process concepts that must be understood by the user require expository information. Use of declarative and structural knowledge via exposition attempts to convey the terminology and ideas contained in the FLPPP. Definitions, justifications, examples, and relationships are designed to produce a fundamental understanding of these concepts. In addition to exposition, performing planning process tasks may also require coaching. Coaching entails guiding the user through necessary steps of task accomplishment. For example, the user may know what a 'risk inventory' is, but may not possess the skills necessary to create one. Precondition productions and the decompositional tasks of a procedural net allow coaching activities to be accomplished quite easily.

These tutoring methods are invoked only if the user acknowledges the need for assistance on a specific topic. Again, this design stems from the authors' insistence on a tutor governed and guided by user abilities. Most eventual users of this tutor are largely concerned with receiving some assistance in developing wildfire prevention plans, as opposed to delving into the intricacies of the FLPPP itself. Consequently, the tutor must have the flexibility to accommodate both types of users.

DISCUSSION AND CONCLUSIONS

Knowledge-based system development can proceed from the 'pre-arranged' structure of available software packages (shells) or by starting from scratch with programming languages. The development environment provided by shells varies greatly between products (Olson and Hanson, 1988). There exists a general tendency for more expensive shells to contain more features and greater flexibility (Harmon et al., 1988). This enables the more expensive tools to be adaptable for a wide variety of application problems. Less expensive shells, on the other hand, are most useful for prototyping and developing small and non-complex applications (Schmoldt, 1987).

Programming languages generally cost less than shells and do not vary greatly in price from one language to another. However, one of the major detractions for system development using languages results from the need to construct ab initio the entire system. Knowledge representation methods, execution control strategies, and inference mechanisms must all be written in the selected language. This requires a substantial effort in addition to collecting and organizing the application-specific knowledge of the knowledge base. Despite any disadvantages of using languages, we felt our application would be difficult to develop and deliver with a commercial shell.

The Macintosh[1] II computer has been acclaimed (Shafer, 1988) as one of the best platforms for expert system development and delivery. However, in our circle of potential end-users, the Macintosh is not ubiquitous as a microcomputer – IBM-PC and compatible machines are still the platform of choice. So, Macintosh cannot presently be viewed as a possible delivery vehicle for software. We did consider the Macintosh II as a desirable development machine and, consequently, sought software that would take advantage of its processing capability and still allow for porting of final products to other machines (both PC and Data General). The above factors taken together indicated that Common Lisp would be a reasonable choice because of its flexibility for development and its portability to other machines.

Because the input/output activities constitute the only machine-dependent features of a program, these have been confined to a separate module of code. The current interface design emphasizes simplicity with line-oriented I/O, so, it should run on any machine to which the source code is transported. Specific machine-dependent versions could later be written to incorporate more sophisticated or user-friendly I/O, i.e. mouse input, windows, and menus. The remainder of the tutor operates oblivious to these methods as long as input strings are received and output strings are printed.

Many of the algorithms and data structures have been designed and implemented. Embellishment and fine tuning of Lisp code may be necessary as more FLPPP knowledge becomes incorporated and as more extensive system functions materialize.

The research reported here was conducted to study the suitability of computer-based aids to the training in and application of complex planning processes – aids which could improve the efficiency of an organization. Using the FLPPP tutor should make it easier for planners to update their data bases and resultant prevention plans. Pencil and paper use of the FLPPP is strenuous enough to discourage frequent updating. Risks, hazards, and values could be stored into the georeferenced data base of a geographic information system. Heuristic decisions that are now contained in the FLPPP tutor could be imbedded into the map production and overlay phases of plan development.

---

[1]Trademark of Apple Computer Inc. Trade names are mentioned solely for information. No endorsement by the U.S. Department of Agriculture is implied.

Eventual combining of the FLPPP tutor with a geographic information system could produce automated systems able to relieve planners of the many spatial data representation and analytical requirements of a pencil and paper application. Incorporation of analysis and planning capabilities into a knowledge-based system is not few, for example Gossym-Comax (Lemmon, 1986). What this approach does address, however, is the addition of tutoring knowledge to guide a user in learning concepts, rather than exclusively solving problems.

A point that should not be overlooked is the importance of fully eliciting and specifying knowledge structure before beginning to represent that knowledge. The Stage 1 research explicated the FLPPP structure far more completely than what the investigators, or individual experts, brought to the study.

The tutor structure described here possesses sufficient generality to be useful as a framework for other knowledge transfer applications. It is not difficult to envision, as computers become more infused into our workplaces and lives, that they may assume more of the role that has been traditionally held by hard-copy compendiums, i.e., books. Conversely, books may experience a gradual transformation into more of an interactive medium. Tutoring systems are one possible step in that direction. Upon completion of this study, we hope to compare the characteristics of this tutorial format with more traditional document manuals.

REFERENCES

Anderson, J.R. and Reiser, B.J., 1985. The LISP tutor. Byte, 10(4).

Anderson, J.R., Boyle, C. and Yost, G., 1985. The geometry tutor. In: Proc. Int. Joint Conf. Artificial Intelligence, Los Angeles, CA.

Andrews, P.L., 1986. BEHAVE: Fire behaviour prediction and fuel modeling system – BURN subsystem. USDA For. Serv. Gen. Tech. Rep. INT-194, 130 pp.

Barr, A. and Feigenbaum, E.A., 1981. The Handbook of Artificial Intelligence, Vol. 1. William Kaufmann, Los Altos, CA, 409 pp.

Bobrow, D.G., 1968. Natural language input for a computer problem-solving system. In: M. Minsky (Editor), Semantic Information Processing. MIT Press, Cambridge, MA, pp. 146–226.

Clancey, W.J., 1985. Use of MYCIN's rules for tutoring. In: B. Buchanan and E. Shortliffe (Editors), Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, Reading, MA.

Darwyn, R.P. and McCalla, G.I., 1986. Using planning techniques in intelligent tutoring systems. Int. J. Man–Mach. Stud., 24: 77–98.

Davis, R., Buchanan, B.G. and Shortliffe, E.H., 1977. Production rules as a representation for a knowledge-based consultation program. Artif. Intell., 8: 15–45.

Fischer, W.C., 1987. The fire effects information system. In: J.B. Davis and R.E. Martin (Editors), Proc. Symp. Wildland Fire 2000, 27–30 April 1987, South Lake Tahoe, CA. USDA Gen. Tech. Rep. PSW-101, pp. 128–132.

Gale, R.D., 1977. Evaluation of fire management activities on the national forests. Policy Analysis Staff Report, USDA Forest Service, Washington, DC, 127 pp.
Harmon, P., Maus, R. and Morrissey, W., 1988. Expert systems: tools and applications. Wiley. New York, 289 pp.
Heines, J.M. and O'Shea, T., 1985. The design of a rule-based CAI tutorial. Int. J. Man–Mach. Stud., 23: 1–25.
Imlah, W. and Du Boulay, B., 1985. Robust natural language parsing in computer assisted language instruction. Systems, 13(2).
Johnson, W.L. and Soloway, E., 1985. PROUST. Byte, 10(4).
Lemmon, H., 1986. Comax: an expert system for cotton crop management. Science, 233: 29–33.
Minsky, M., 1975. A framework for representing knowledge. In: P. Winston (Editor), The Psychology of Computer Vision. McGraw-Hill, New York, pp. 211–277.
Olson, R.L. and Hanson, J.D., 1988. AI tools and techniques: a biological perspective. AI Appl. Nat. Resour. Manage., 2(1): 31–38.
O'Shea, T., 1982. A self-improving quadratic tutor. In: D.S. Sleeman and J.S. Brown (Editors), Intelligent Tutoring Systems. Academic Press, Orlando, FL, pp. 309–336.
Raphael, B., 1968. SIR: A computer program for semantic information retrieval. In: M. Minsky (Editor), Semantic Information Processing. MIT Press, Cambridge, MA, pp. 33–145.
Reinhardt, E., 1987. An expert system for designing fire prescriptions. In: J.B. Davis and R.E. Martin (Editors), Proc. Symp. Wildland Fire 2000, 27–30 April, 1987, South Lake Tahoe, CA. USDA Gen. Tech. Rep. PSW-101, pp. 223–225.
Sacerdoti, E.D., 1977. A Structure for Plans and Behavior. Elsevier, Amsterdam, 126 pp.
Schank, R.C. and Abelson, R.P., 1977. Scripts, Plans, Goals and Understanding. Lawrence Erlbaum, Hillsdale, NJ.
Schmoldt, D.L., 1987. INSIGHT2+: An inexpensive tool for small applications. AI Appl. Nat. Resour. Manage., 1(2): 65–68.
Schmoldt, D.L., 1989. Organizing knowledge for tutoring fire loss prevention. Environ. Manage., 13(5): 573–582.
Schmoldt, D.L. and Bradshaw, W.G., 1988. A cumulative Delphi approach to knowledge acquisition. In: Y. Wilks (Editor), Proc. 3rd Annual Rocky Mountain Conf. Artificial Intelligence, 13–15 June 1988, Denver, CO, pp. 149–162.
Schmoldt, D.L. and Martin, G.L., 1989. Construction and evaluation of an expert system for pest diagnosis of red pine in Wisconsin. For. Sci., 35: 364–387.
Shafer, D., 1988. Special report: AI on the Macintosh. Intell. Syst. Anal., 143 pp.
White, B.Y. and Frederiksen, J.R., 1986. Intelligent tutoring systems based upon qualitative model evolution. In: Proc. 5th Nat. Conf. Artificial Intelligence, Philadelphia, PA, pp. 313–319.
Woolf, B., Blegen, D., Jansen, J.H. and Verloop, A., 1986. Teaching a complex industrial process. In: Proc. 5th Nat. Conf. Artificial Intelligence, Philadelphia, PA, pp. 722–728.
Yazdani, M., 1986. Intelligent tutoring systems: an overview. Expert Syst., 3(3): 154–163.