Reactive Instructional Planning to Support Interacting Teaching Strategies

JULITA VASSILEVA
Federal Armed Forces University - Munich
85577 Neubiberg, Germany
E-mail: jiv@informatik.unibw-muenchen.de

Abstract
We propose an architecture for reactive planning of contents in instruction. It is based on a framework for reactive planning which integrates opportunistic reactions with plan-based (plan-repairs and complete replanning). It is suggested how this framework can implement two radically different teaching styles, tutoring and coaching, as well as their interaction within one system. One additional advantage is the possibility to manage the way of system's reacting by means of different pedagogical rules. At this point the system has been implemented and experimented in the domain of integration of elementary functions.

1 Introduction

The evolution of content planning can be traced back to the pioneering approach of Peachey & McCalla (1986) which uses "classical" (Wilkins, 1988) planning techniques based on explicit representation of the target knowledge concept structure (curriculum) at different levels of granularity, in a STRIPS-like fashion (Fikes & Nilsson, 1971). Wasson (1990) develops further this planning paradigm into a least-commitment non-linear hierarchical planner (Chapman, 1987). In the context of planning instruction this means that the plan does not consist of a particular sequence of goals to be achieved, but rather of a set of plan fragments at a given level of granularity. The order of executing these plan fragments is decided at execution time. In this way unexpected instructional opportunities, like blocked learning paths, missing prerequisite knowledge or serendipitous gains in knowledge can be taken into account.

However, these are far from covering all unexpected or unforeseen situations at planning. The reason is that the description of the world in which any planner works is by necessity very limited (otherwise, the computational complexity of planning becomes intractable). The content planning approaches described above deliberately limit their world description to a concept structure representation (describing the legal moves in the world), a teaching goal expressed in terms of concepts to be learned (the desired status of the world) and a student model describing the current status of the world (concepts that have been "visited"). In this way they focus on the instructional content what has to be taught and can recognise only events which prevent or make easier the achievement of the goal. Usually the main teaching goals in developing a curriculum concern the content to be communicated to the student. However, in the actual teaching process based on one curriculum the teaching goals are far more complex and usually contain several dimensions. In didactic literature a distinction is made among three types of instructional goals: (cognitive, affective and psychomotoric) and four levels of instructional goals (knowledge, application, transfer and analysis / synthesis). In practice very often these goals are pursued simultaneously, though some of them are given more attention then others, which remain in the background and come to play only when a good opportunity arises. An addition to the enumerated above types of goals (a classification from didactic point of view) one can distinguish between cognitive and meta-cognitive goals. Teachers usually pursue at the same time cognitive (i.e. related to specific contents/subject that has to be taught) and meta-cognitive goals (general knowledge about how to learn, to solve problems, to organize the activities). It happens often that they interrupt working for a specific cognitive goal to stress on a meta-cognitive issue, and even change the contents taught at the moment, for example, to show how the same or an similar method is applied in a different domain or problem.

However, if we would like to consider at the stage of planning instruction all possible levels and types of simultaneous teaching goals, as well as the potential opportunities for fulfilling them, planning of the teaching session will become so complex that it will be virtually impossible. Human teachers also don't seem to plan their lessons considering explicitly all the possible levels of goals and background goals. They are very likely to do exactly what the classical content planner does: using their knowledge about the domain concepts and the logical dependencies among them, they plan a sequence of concepts to be learned by the student in order to understand a given goal-concept. In this way they ignore their background goals in order to make planning possible. However, unlike a computer-based instructional planner, which during the execution of the plan is able only to recognise blocked paths (when re-planning in necessary) or possible short-cuts (for achieving the same goal-concept), human-teachers are able to recognise arising opportunities to achieve their background goals and to use them.

A classical planner can't do this since it works only in its world description, which is limited to the necessary minimum. If we increase the world description, planning becomes intractable. The only solution is to plan in the limited description of the world and representation of goals, but provide the system with the ability to "sense" the changes in the environment in order to recognise opportunities for achieving a given set of background goals and furnish it with a set of means to achieve these goals (local opportunistic reactions, changing the style of interaction, or a replanning for the somewhat changed goal).

We propose a framework and architecture for reactive planning, integrating the concept of planning ahead with that of reaction to the environment. This framework differs from Wasson's planning paradigm in that an a-priori ordered plan is created. This allows a global evaluation of the plan and an informed selection according to optimality criteria. During the execution of this plan, the system tries to keep to the plan as long as possible, but it is able to react adequately to unforeseen situations at plan time. This is done by not only modifying the plan (avoiding blocked learning paths and making use of shortcuts), but also by reacting locally to arising situations and by opportunistically-triggered replanning. The ability to react without a plan is a highly desirable feature in the uncertain and dynamic instructional environments.

An architecture implementing the proposed reactive instructional planning framework has been developed. It is suggested that reactive planning can be used as an instrument for implementing interacting teaching strategies.

2 Planning

Classical Vs. Reactive Approaches

Planning is a problem solving technique that creates a sequence of actions (i.e. a plan) to achieve a goal and attempts to forecast the effects of executing the plan (Wilkins, 1988). The classical planning problem assumes a state-based definition of the world (the planner's application domain) by means of set of operators and their preconditions and effects characterized as state predicates. A planning problem consists of this domain description plus an initial and final (goal) state. The main assumption in classical planning is that the domain description doesn't change while the planning is being carried out. This important limitation leads to a distinction between plan time and execution time. However, this assumption is far too restrictive and is not true in realistic environments, where a lot of unexpected situations can arise that can block the execution of the plan. If one increases the world description so as to account for all thinkable factors that might interfere in plan execution, planning will become too complex for many applications.

<u>Classical planning approaches</u> try to avoid this problem by generating a set of possible plans consisting of unordered sub-goals and broken into fragments. Then the engagement with one particular plan-fragment is postponed until execution time, when the planner checks the state of the world to select to most appropriate plan fragment, or to re-plan according to the changes in the world. This approach, however, has inherent shortcomings - still all conditions at which different plan-fragments are appropriate have to be known and considered at planning time. However, a situation might be characterised by a complex combination of dynamic or continuous factors (like time) and it would be practically impossible to include them in the world description. Or a situation might occur so rarely that it is unreasonable expensive to consider it at planning time.

An alternative approach provides the so called <u>reactive planner</u>. A system is called reactive, if it can react in an acceptable amount of time to any changes that occur in the world while the system is running (Wilkins, 1988). A reactive planning system can react to events which have not been foreseen at planning stage for different reasons (e.g. because they were not known or because it would have been too expensive to consider them at planning stage). The most important characteristics of a reactive planner are that it:

1) senses the environment all the time and recognises situations which are not foreseen in advance, and

2) has a set of reactions for these situations which are not included in the world description to be considered normally at planning (or re-planning) time.

A classical planner creates a plan containing plan fragments to achieve certain subgoals (the subgoals are generated at planning time) and starts executing the plan until it reaches the next subgoal ("checkpoint"). Then it checks whether all the preconditions for carrying along the plan are still present. In case the preconditions are not in hand, it can re-plan, i.e. generate a new set of subgoals and partial plans for them. In case all preconditions are fulfilled (nothing in the environment has changed dramatically), it chooses an appropriate plan-fragment from the generated in advance fragments for reaching the next sub-goal. However, if an event occurs during the execution of the plan-fragment which prevents reaching the next "checkpoint" (pre-planned subgoal), the planner will block since it is not able to sense and to reason during the execution of a plan-fragment, but only at the "checkpoints" (i.e. when achieving a pre-defined sub-goal). In addition, its sensors are only able to consider events, contained in the world description, i.e. plan-constraints and factors concerning the optimality of the plan-fragment.

A reactive planner senses for changes in the environment all the time. A "checkpoint" in a reactive planner is not pre-defined, but dynamically occurring during plan execution as a result of arising changes in the environment. The planner is equipped with knowledge of how to recognise situations during the plan-execution, that need special opportunistic treatment. The sensor might look for parameters included in the world's description (and therefore considered at planning time), but also for a set of additional parameters which are not considered at planning. In addition to these environment-defined "checkpoints", a reactive planner might have also pre-defined "checkpoints" like a classical incremental planner. Or it might not have such "checkpoints" and just follow a pre-defined in advance plan without selecting a most optimal part fragment for every subgoal. The important feature for reactivity is that the planner, via continuous sensing allows the dynamic characteristics of the environment to define the points when a change in the plan execution may happen and these points are not defined in advance at the point of the generation of the plan.

The reaction to such a situation might be like in a classical planner - selecting a new optimal partial plan and re-planning. However, in a classical planner replanning takes place only when the current plan is no longer possible, while in a reactive planer, replanning can occur every time if the heuristics for reacting recommend this. In addition, a reactive planner might change the goal of the overall plan, or execute a pre-defined opportunistic reaction managed by some heuristic rules.

In this way, a reactive planner can

• react to situations which a classical planner will ignore (since they were not defined as "checkpoints" at planning time) and

• react in many different ways (including a local immediate reaction to the situation, or a re-planning for a different overall goal), while a classical planner is restricted to choosing a more appropriate plan-fragment (if the restrictions at planning time have not been violated) or re-planning (for the same goal taking into account the new state of the world).

Reactive planning is increasingly becoming an active area in AI research. There are a variety of systems (Hayes-Roth, 1993), (Epstein, 1995), (Schoppers, 1987), that have tightly integrated planning with some combination of execution, sensing and replanning. There is work on reactive planning (Firby, 1987), (Korf, 1990), (Brooks, 1991), (Beetz & McDermott, 1992), (Hanks & Weld, 1992), (Etzioni et. al, 1992), which emphasises the ability to react on unexpected situations rather than assume that a plan will usually work. This view is appropriate for some domains, like robot planning, but not in domains such as instruction where the cost of execution will usually be much higher than the cost of reasoning about actions. In partial-order planning framework, Ambros-Ingerson, (1987) developed an integrated planning, execution and monitoring system called IPEM and introduced the idea of run-time variables for sensing. Olawsky and Gini (1990) focused on the trade-offs and strategies on choosing when to sense and when to plan. Knoblock (1995) has developed a planning system which works in the domain of information gathering from distributed databases which features a tight integration of planning and execution, by executing the actions simultaneously with the planning, replanning and sensing.

There are a number of suggestions (Spector & Hendler, 1990), but no definitive answers yet, on an appropriate way to view long-term planning so as to integrate it with reaction. The area of ITS can be used as a domain for developing and testing ideas, architectures and applications because it has all the features of a domain, where reactive planning is necessary (Lyons et. al, 1991):

1) the agent (the ITS) can never be certain of the effects of its actions (whether the student really possesses the knowledge suggested by the student model);

2) the agent cannot make the assumption that the world remains static and unchanging while carrying out the plan;

3) the agent cannot assume that it knows everything about the world.

Thus, implementing a reactive planning framework in an instructional planner of an ITS will bring valuable experience and results for the field of AI planning on the problem of how to integrate planning with reaction.

3 Other Limitations of the Classical Paradigm

We shall discuss the features of the classical planning paradigm proposed by Wasson (1990) with respect to their advantages and disadvantages for instructional planning.

The main characteristics of a classical planner are:

• hierarchical (on different levels of concept generality),

• non-linear (the plan is a non-ordered set of goals),

• least-commitment (incremental creation of plan fragments, no complete plan).

The first feature reflects the possibility to use the hierarchical abstractions of domain concepts in order to organise planning better. This is a crucial feature for every planner that works in a complex domain.

The second feature provides a high run-time flexibility in adapting to the individual learning path of the student. Recent studies (Minton, Bresina & Drummond, 1994), however, conclude that the only significant difference between partial order and total order planning is *planning* efficiency (i.e. partially ordered planners save resources from creating global plans that will be never executed and from searching of all possible plan orderings). However, if the planning space is hierarchically organized, the eventual planning inefficiency of a total order planner is not a big problem, since it can be compensated by planning in smaller spaces. *Execution* flexibility can also be achieved with a total order planner and a post-processing step that removes unnecessary orderings from the total ordered plan (Veloso, Perez & Carbonell, 1990).

From a pedagogical viewpoint the partial order, least-commitment planning of instruction is also not necessarily optimal because of the following reasons. First, the dynamic process of incremental generation of local plans is hard to imagine and understand for a human. It is not possible to evaluate such a plan in advance. Second, the ordering of sibling-subgoals in the plan is be very important when there are goal-dependencies. This is often the case in teaching. Learning one concept can make it easier or harder for the student to understand another concept even thought there is no explicit strong relation of precedence between the two concepts. It would be an advantage, if the planner allowed informed ordering of goals that could be assigned by use of additional pedagogical knowledge source, or in an interactive regime during planning by the teacher or by the student himself.

One can advocate a more rigorous way of planning, by generating an optimal plan and ordering the subgoals in advance and trying to keep to it as long as possible. However, one has to take into account the dynamic nature of teaching. This can be done by accepting a reactive planning paradigm and providing the possibility for immediate reaction or replanning.

4 Requirements to an ITS based on Reactive Planning Framework

The main difficulty with building reactive planning systems is the enormous computational complexity, both at design time (the system should have an appropriate action for every possible situation in order to behave deterministic*) and at run time (the sensing system must be able to identify all relevant aspects of the world to identify situations). Solving the first problem in the context of instructional planning requires the definition of all important types of situations for different levels of teaching goals and corresponding reactions to address these goals (the pedagogical component should be equipped with knowledge of what to do). Solving the second problem means that the student model should be able to diagnose all important aspects of student behaviour and personality to identify these situations. The ITS should have also other "sensors" which would allow the system identify changes in the environment (e.g. time), arising teaching opportunities (e.g. situations that have happened before, possibilities to achieve background teaching goals etc.).

Student Model

The advantages of accepting a reactive planning paradigm depend critically on the system's sensing capability. Apart from the reactive planner's sensing system able to detect changes / situations in the environment, discussed in the previous section, an important part of the sensing system will be incorporated in the student model.

There have been many recent advances in the field of student modelling for ITS (McCalla et al., 1992), (Greer & McCalla, 1993), (Vassileva, 1992b), (Self, 1994). The main problems are to define

---

* This can be considered as a problem equivalent to modeling context in AI or to representing common-sense knowledge, which is a main stream in AI research (see the recent work of Doug Lenat).

those features of the student which are relevant to the instructional process and to model them and only them since "*there is no sense of diagnosing what you can't treat*" (Self, 1988). The most relevant information about the student in a instructional situation concerns her knowledge. Different approaches exist for representing student's knowledge. Some use stereotypes such as "Weak", "Average", "Good" others represent it as an overlay over the expert knowledge of the system, eventually, containing "bugs", i.e. transformations of the correct knowledge which lead to certain errors made often by students. More recently numeric approaches have been proposed, like Bayesian Belief Networks, (Villano, 1992), (as well as logic (predicate calculus)-based representations of the student's knowledge in terms of beliefs. Non-monotonic reasoning as well as the application of non-standard logics have been proposed in order to model the learning process of the student and to predict her behavior/knowledge (Self, 1991). Comparatively less interest has been paid on modeling the student's personal characteristics and on searching methods to use them in the teaching process.

In order to be able to diagnose situations when reaction is needed, an ITS needs to have a very sensible "sensing" system. That means that is has to be able to register all changes in the environment, in the student's behavior, to be able to recall similar situations from the history of the tutorial session or to "keep in mind" a set of background teaching goals and to recognise the arising opportunities to fulfil them. Of central importance for the success of a reactive planning framework therefore will be the student model. We can outline some directions in student modeling research which might be useful for developing of a useful "sensor" for the student's behavior:

- **Different levels of granularity for knowledge representation**. It seems promising that a model based on different levels of granularity will be useful for knowledge representation and diagnosis of the student's plans (Greer & McCalla, 1989). The different levels of granularity (abstraction or aggregation of knowledge elements) will help in identifying different levels of teaching goals and contexts arising to fulfil them. This will be very useful for deciding whether to react locally to the situation or to replan.
- **Belief revision systems** for representation of the student's knowledge will allow a better maintenance of the consistency of the student model. It will be interesting to investigate how the dynamic changes in the student model, managed using a belief revision system (for example, the evolutionary / revolutionary approach (Huang et. al, 1991-b), or using limited reasoning (Fagin & Halpern, 1987), non-monotonic (Etherington, 1987) or meta-reasoning approaches (Cialdea et. al, 1990), can be related to reactive instructional planning.
- **Focus of attention.** The concept of "focus of attention" (Huang et. al, 1991-b) developed for maintaining the consistency of the student model and easing the diagnosis could be used for establishing priority-schemes for reactions and/or planning.
- **Stereotypes of student personality types.** The student model needs also to represent a minimal set of parameters corresponding to the student's personality and psychological features, like motivation, intelligence, concentration and other factors according to the Aptitude Treatment Theory of individual instruction (Cronbach, 1967; Glaser, 1972). Stereotypes have proven to be a useful and practically feasible way of modeling user's general features like interests or psychological type

(Rich, 1986; Kay, 1994) which is certainly worth looking into in the context of modeling the personality characteristics of the student in an ITS. So far, student modeling in the field of ITS has only been focusing on modeling student knowledge.

So far even though there has been a lot of theoretical advances in the field of student modeling along the directions enumerated above, they have not been applied effectively in ITS, since there has been no equivalent development in the study of instructional planning and teaching strategies that could make use of them.

Pedagogical Component

The pedagogical component has to contain the knowledge of how to react to arising situations and opportunities to treat simultaneously background goals. It is possible to define some general tendencies in didactic literature about learning methods, sequencing of phases and activities to address different goals. However, unfortunately, so far it was only possible to capture a very limited set of factors (situations or individual features of the student) that influence the choice of when and how to address a specific teaching goal level. For example, according to the Aptitude - Treatment Theory of Instruction (Cronbach, 1967), (Einsiedler, 1976) one tries to isolate single factors and to prove their importance for the learning process as well as their mutual dependencies. However, the results of these studies are mostly a set of recommendations rather than proved results that could be formulated as rules. The reason is that this research has been intended to bring results about the existing way of instruction, which is as a rule carried out in a class (group) environment and not in individual "one-to-one" instruction. It is impossible to collect a group of students with homogeneous individual parameters. ITS, as an individual instructional tool, providing "one-to-one" instruction, offer for these studies an excellent opportunity to try variations of single factors on instruction (student characteristics, characteristics of the environment and pedagogical methods, strategies) and tactics), as well as their combinations and to empirically prove their influence on learning.

A framework for reactive instructional planning can provide an experimental tool with the needed flexibility to allow teachers and psychologists analyse, simulate and test the factors, situations and pedagogical decisions that are taken by human teacher "on the flow" during a dynamic instructional session.

Developing a Reactive Planning Framework for Instructional Planning

We have proposed a framework and architecture for reactive planning (Vassileva, 1990; 1992-a), (Vassileva et al., 1991). It implements some of the ideas discussed above and demonstrates one possible way of integration of planning ahead with reaction to the environment. This framework differs from Wasson's planning paradigm in that an a-priori ordered plan is created. This allows a global evaluation of the plan and an informed selection according to optimality criteria. During the execution of this plan, the system tries to keep to the plan as long as possible, but it is able to react adequately to unforeseen situations at plan time. This is done by not only modifying the plan (avoiding blocked learning paths and making use of shortcuts) as in Wasson's classical planning approach, but also by reacting locally to arising situations and by opportunistically-triggered replanning. The ability to react without a plan is a highly desirable feature in the uncertain and dynamic instructional environments. It is suggested that this reactive planning framework can be used to implement interacting teaching strategies.

5 TOBIE: an Architecture for Reactive Content Planning

In order to implement realistic instructional planning, we need to have the possibility to represent and teach a larger diapason of knowledge of a given domain, e.g. curricular, conceptual and problem solving knowledge. TOBIE (Teaching Operators-Based Instructional Environment) provides an architecture for representing different levels of knowledge organisation at the same time and also a uniform and modular language for domain knowledge representation (see Figure 1).

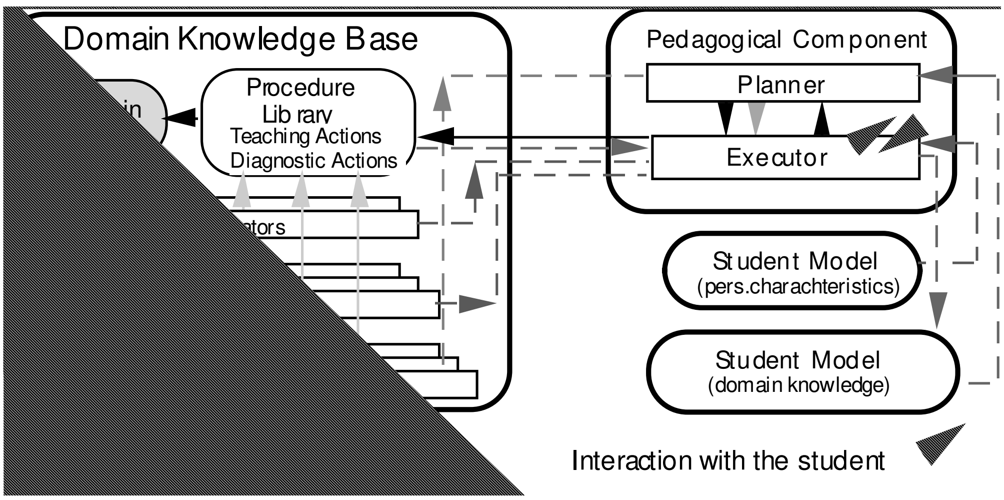

Figure 1. TOBIE Architecture

TOBIE (Vassileva, 1990, 1991) is an ITS-shell architecture based on content planning. It consists of Pedagogical Component (Planner and Executor), a Student Model, and a Domain Knowledge Base. The domain knowledge base contains the elementary objects of the domain corresponding to units of subject material that could be taught to the student. It can be considered as a directed AND/OR graph. Directed AND/OR graphs provide a mighty representation language in which curricular (concept) structures, goal (task) decompositions and problem-solving spaces can be expressed. For example, Figure 2 shows how AND/OR graphs represent domain knowledge on three different levels (curricular, problem-solving and performing single steps). AND/OR graphs can be represented implicitly by sets of production rules, encoded in TOBIE by means of Teaching Operators (TOs).

The TOs are STRIPS-like operators (Fikes & Nilsson, 1971) which consist of 6 parts (see Table 1): a name, a list of preconditions (concepts in the student model) under which the operator can be used, a list of expected effects (list of logical expressions to be added to the student model), an action which is a pointer to a teaching procedure in the Library of Procedures. These procedure present to the student teaching material in a specific way, e.g. present, explain, focus, remind, solve a problem, provide an exercise or test. The teaching procedure can also contain a recursive call of the system on a different level of organisation of domain knowledge, for example to show how to solve step by step a given type of problems (as mentioned before, in TOBIE it is possible to represent different levels of knowledge organisation). The TOs contain also a part called "Diagnosis". It contains a pointer to a Diagnostic Action — a procedure stored in the library of Teaching Actions which evaluates the success of the student and eventually analyses the student's answer and finds misconceptions. It adds to the student model either the effect-node(s) of the TO or the node corresponding to the diagnosed misconception. The teaching actions which request solving a problem of a give type are associated with the same diagnostic procedure. The last part of a TO is called "Type". It contains a list of parameters which describe the teaching action from a pedagogical point (e.g. if it is an explanation, example, exercise), the difficulty (if it is an exercise or test), and the type of media (text, graphics, picture with motion or sound).

Table 1. An example of a TO and a teaching action procedure (see the concept structure in Figure 2)

NAME	CONDITIONS	EFFECTS	ACTION	DIAGNOSIS	TYPE
teach_partfra	rat_t5, tr_partfra	part_fra	teach(partfra)	test(partfra)	linear

PROCEDURES LIBRARY:

teach (name):

pres_name,	/* presents a text explaining how to solve integrals by method "name"
examp_name,	/* presents an example of a solved problem with method "name"
demo(name),	/* calls recursively planning in the problem-solving space with background goal "name" and default teaching action "demo" (tutoring strategy).
exerc(name)	/* calls recursively planning in the problem-solving space with background goal "name" and default teaching action "exercise" (coaching strategy)
test(name):	/* calls recursively planning in the problem-solving space with background goal "name" and default teaching action "test" (like coaching but no feedback).

The model of the student's domain knowledge represents the current state of the student's knowledge in terms of the elementary objects that are believed to be learned (correct concepts and misconceptions). It is updated in two ways which are not discussed here, since it is of no direct link with the goals of this paper. The model of the student's personal characteristics represents certain preferences of the student to different types of teaching actions, psychological and motivational parameters, like field-dependence, concentration, confidence, persistence.

The Pedagogical Component contains two sub components - a Planner which dynamically generates plans in the knowledge structure to meet certain teaching or goals and an Executor which carries out the plan, re-invokes the Planner or reacts locally to arising opportunities. In the next section we shall discuss the Pedagogical Component and the way reactive planning is of carried out.

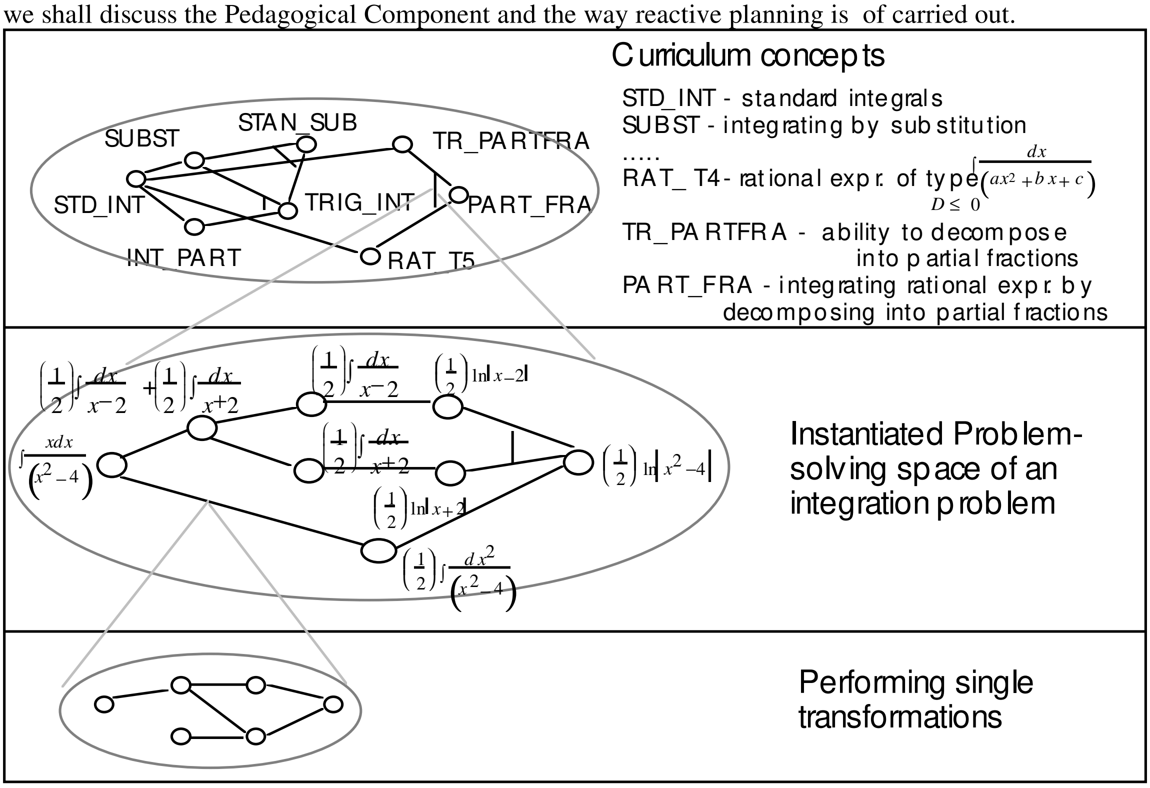

Figure 2. Representing different levels of knowledge organization

6 Reactive Planning in TOBIE

The structure of the Pedagogical Component is shown in Figure 3.

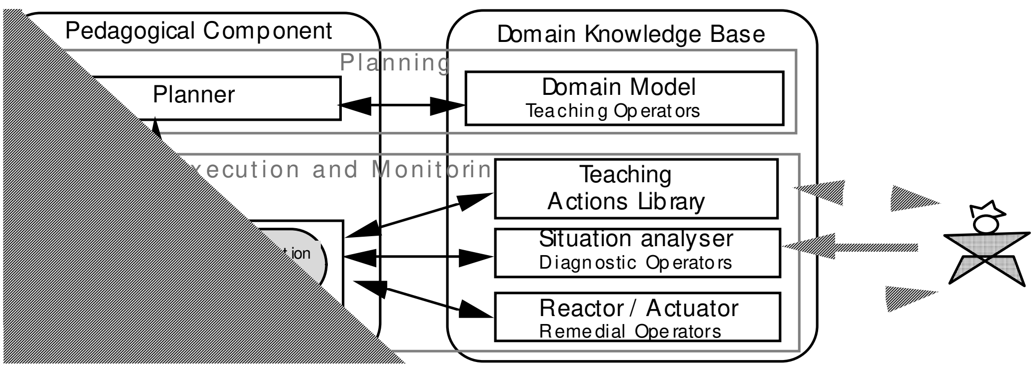

Figure 3. Reactive Planning in TOBIE

Planning

The Planner is activated within the domain concept structure at one given level of organization. The planning algorithm is a modification of the AO* (Nilsson, 1980). The optimisation function *h* can be selected so that different criteria for optimality can be implemented (e.g. the shortest, the plan avoiding certain concept, plan with a certain topology-type etc.). The solution graph of an AND/OR graph imposes only a partial ordering on the solution steps. If there are no subgoal interactions, the order of applying the operators is not important. In principle, goal dependencies can:

1) make the further execution of a plan impossible since there are TOs which influence negatively on the student's knowledge, i.e. which could delete concepts from the Student Model, or

2) make the plan no longer optimal because of unexpected acquiring of goals.

The first type of goal interaction is dangerous. In our architecture, however, only the Remedial Operators can delete concepts (corresponding to misconceptions) from the student model and they are not considered at planning stage. Misconceptions and their remediation are never planned in our system, but treated opportunistically. However, assuming that there is no explicit goal interaction, the plan can be partially reordered interactively (by the teacher) or automatically according to certain pedagogical criteria (for example, more concrete concepts before abstract ones, simple concepts before more difficult ones etc.) and considering the pedagogical type of the TOs. If it happens that the student acquires unexpectedly concepts, no plan-shortcut is made automatically. This is a situation when the system decides what to do according to the reaction rules. So, unlike Wasson's planner our system doesn't automatically take advantage of learning opportunities as they arise on the flow. A shortcut will be made if this corresponds to a background goal, or if the rules state that this have to be done.

Situations

After an initial plan is generated, it is passed to the Executor. It executes the TOs by invoking the procedure assigned by the "teaching action" part of the operator from the Procedure Library (see Table 1). This procedure can consist of other procedures which present explanation of the concept, give examples, start an exercise. The main teaching action procedure of a TO contains always a procedure which tests whether the student has acquired knowledge on the concept. The diagnosed knowledge on the concept or misconception is included in the student model which provides a condition for the next Teaching or Remedial Operator to be applied.

In case that a misconception has been diagnosed and entered the student model, or an undiagnosable error has been made (or a call for help from the student), a not planned situation arises. Another type of unexpected situation can occur when the teaching action of a TO invokes instructional planning at another level of knowledge organization. This is not considered at planning time, since the actions of the TOs are stored separately in a library and TOs are selected only because of their conditions, effects and pedagogical type. In principle, the calls to another level could have been considered at planning stage, but this would make planning much more complex and very often the resulting plan will not be feasible because of external factors, like time. Since switching to a different level means instruction in a comparatively independent part of the material, it won't be pedagogically justified to interrupt the process in the middle because of external factors. That is why the decision whether to permit a switch and how to do it has to be taken at plan-execution time, when the need arises (and not long in advance).

The so far discussed situations could, in principle, be recognised and treated by a classical planner too. Another type of situations can arise from a combination of external (environment) factors which can not be predicted at planning stage. For example, a violation of time-restriction, evidence that the student is no longer concentrated, opportunity to fulfil a teaching goal that has been staying in the background (for example, on a different level of organization of material), evidence from history that the student has had difficulties with a concept before etc. This type of situation is recognised by the so-called "Diagnostic operators". They are rules encoding combinations factors (variables) with different values describing the current context. We distinguish among five types of factors: parameters of the environment (time, resources); history (how long did it take to study concepts with similar difficulty, did he ever learn the concept, did he ever show success on problems involving knowledge on this concept etc.); background teaching goals; the model of the student's domain knowledge and the model of the student's personal characteristics.

A search for matching diagnostic operators is done at every tact of executing the plan, i.e. after the execution of a elementary procedure included in the action-procedure of a TO (see Table 1.). The various situations matched by the diagnostic operators can't be treated in an equal way. That is why every diagnostic operator assigns also a specific reaction.

Reactions

Our system provides four principle type of reactions: ignoring the situation, an opportunistic reaction without changing the plan; local plan repair; global re-planning. They are shown in Figure 4.

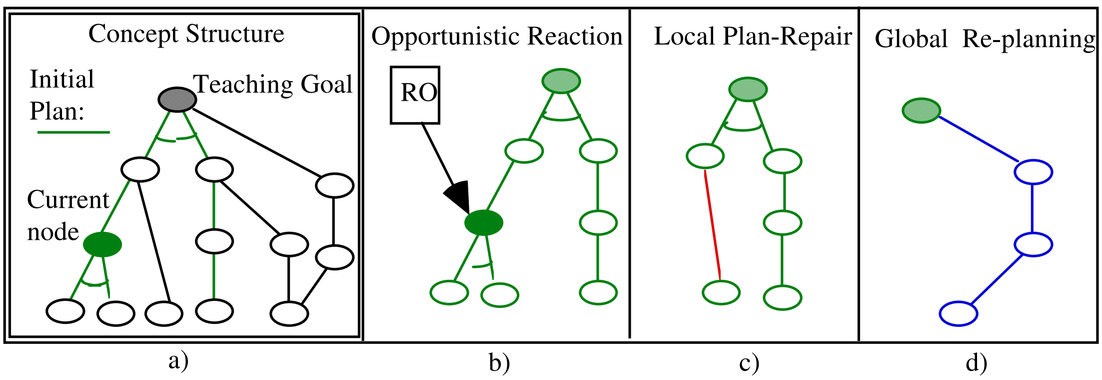

Figure 4: Reactions.

Figure 4, a) shows an AND/OR graph, representing a domain structure at some level of organisation, for example, any of the levels shown in Figure 2. The initial plan for achieving the teaching goal and the current node (concept or problem solving state) where an unexpected situation has arisen are presented. The first type of reaction (ignoring the situation) allows the system to follow a plan rigidly. An opportunistic local reaction (Figure 4, b) provides an immediate feedback while keeping the initial plan. A Remedial Operator (RO) will be executed when the student model contains the misconception matching the operator's preconditions. Special ROs are provided for unidentified errors (general hint, humorous remark, encouragement, etc.). Local Plan Repair (Figure 4, c) means that only the part of the plan related to the current node will be changed. In this way the system tries to find an alternative way to teach a difficult concept without changing the overall plan. A global replanning means finding an alternative plan for the main teaching goal (see Figure 4, d).

A summary of the possible reactions to the four situations is given in Table 2.

Matching Situations with Reactions

A set of rules is responsible to select a reaction to situations that occur during plan-execution. These rules we call "reaction rules" or "pedagogical rules", since we believe this is more a pedagogical decision. The conditions of these rules are based on the same groups of factors that are matched by the diagnostic operators, but they have one additional factor - the type of situation. That is why we provide a tool for creating pedagogical rules and diagnostic operators that define new possible situations as combinations of factors. Four approaches are possible:

- to define at hoc the reactions to the possible situations (the current solution);

- to interview teachers with the goal to extract rules and implement them using the rule-editing tool or to ask them to implement the rules directly themselves (however, this requires that the teachers are able to articulate the factors influencing their decisions, which is not often the case);

- to observe what human teachers do in real situations and try to extract some knowledge out of their behavior, i.e. analysing protocols of teaching sessions, and applying machine learning techniques in order to define cases and, eventually, to generate decision trees;

- to try to define some guidelines from existing didactic theories, or to try to model within this framework the teaching strategies of existing ITS, like Wasson (1990), COCA (Major, 1993).

Table 2. Situations and Reactions

Situation \ Reaction	Local Opport. Reaction	Local Plan- Repair	Global Replanning
unidentified error, misconceotion	execute remedial operator	make a plan for the current parent goal and replace the corres-ponding part of the old plan	start the planner anew
combination of factors	execute remedial operator		
call to another level	keep old plan in stack and revise it after returning	revise old plan from stack using "shortcuts" from obtained goals at the other level	after coming back, re-plan to make use of the changed environment

7 Implementing Two Teaching Styles And Their Combinations

The TOBIE architecture and the reactive planning framework described above allow implementing two radically opposite teaching styles and their interaction in one system.

Tutoring is a teaching style aimed at communicating new material, presentation-oriented, straightforward, the initiative is in the tutor, the student is guided and prompted to reply, solve a problem, etc.

Coaching is a teaching style aimed at developing the skills of applying existing knowledge to new situations, to re-organize knowledge structures and develop meta-cognitive skills. The initiative is in the hands of the student, the coach can comment and interfere to give advice, or provide help when requested or when this is necessary to overcome misconceptions and difficulties.

Both strategies can be modelled as a planning process. The only difference is who takes the active role. In the first case, it is the system who creates a plan for teaching a given concept and leads the student step by step to achieving the goal. In all previous sections we have been focusing implicitly on a tutoring style, that is why now we shall only concentrate on the implementation of a coaching style and how one system can interactively use both styles.

A typical case when coaching style is used is to support the student in exercises for solving problems (represented as a problem state space, a goal and a initial state). The system can generate a solution by creating a plan which leads from the initial to the goal state (if several plans are possible, it selects an optimal one), but instead of executing it, it only observes the student's actions and tries to match them with the system's plan. In case that the student makes the same steps and goes along the system's plan, no reaction from the system is needed. In case of a difference, however, a situation occurs which requires a reaction. The possible reactions are defined in exactly the same way as discussed in the previous section. They are: ignoring the difference (keeping silent); local reaction aimed at bringing the student back on the system's plan (like in the model-tracing style of the Lisp-tutor); local plan-repair (trying to find a way to accommodate the student's solution within the system's plan) and complete replanning (trying to find another plan for solving the problem that fits with the student's solution).

Other situations that need reaction can be defined in analogy with those described in the previous section: a combination of factors (e.g. environment, the history, the student model, the structure of the problem solving space, opportunities to discuss background teaching goals, personal characteristics of the student). For example, if the time is nearly over, and the plan which the student is following is too long or involves complicated (expensive) steps, this creates an opportunity to interfere.

The pedagogical rules for selecting a reaction for the specific situation involve the factors mentioned above and depend on the type of situation. For example, if there is enough time and the student is confident, to choose a re-planning reaction to accommodate his way of solving instead of bringing him back with a remedial operator to the corresponding state in system's plan.

Here we have to state that there is no difference in the planning and executing mechanisms used to implement both teaching styles. The difference in the interaction and initiative is completely due to the different teaching actions, the procedures that carry out the dialogue with the student and the presentation of material.

A combination of the two strategies will be shown with example. Let's suppose that the system is teaching in the domain of symbolic integration and has planned instruction according to the curriculum from Figure 2. The current goal is to teach the concept (method in our case) PART_FRA - "integration by decomposing into partial fractions". The action of the TO for this goal is a sequence of five procedures (see Table 1), the first two presenting a textual explanation and example, the third one demonstrating a step-by-step solution of an example problem, the fourth one - coaching the student's solving of another problem and the fifth one - testing how the student copes with a problem alone. The demo-procedure invokes planning on a different level of knowledge organization, where the TOs represent admissible transformations between different types of expressions (states) and heuristics for selection of transformations for different states (Slagle, 1963). A problem solving space is generated (instantiated for the example problem) by executing TOs which perform transformations over the instantiated integrand-types (the problem-states). There are two main types of teaching-action procedures at this level, which correspond to the two teaching styles. A procedure for tutoring style ("demo") performs the transformation encoded with the TO and displays the result. By executing the actions of the TOs from the plan, the solution of the problem is shown step by step.

The exercise ("exerc") procedure invokes the planner again on the problem-solving level for a different problem. This time a coaching style procedure is activated. Such a procedure visualises the initial state, asks the student to select a transformation (a TO), performs it (only if it is applicable; if not - a situation arises) and builds increasingly another problem-solving space the student model.

Let's suppose that the student has to solve the problem: $\int \frac{x}{x^2-4}dx$. The student chooses to apply a standard transformation $xdx \Rightarrow \left(\frac{1}{2}\right)dx^2$, resulting in $\left(\frac{1}{2}\right)\int \frac{dx^2}{x^2-4}$ and then selects a substitution to transform it to $\left(\frac{1}{2}\right)\ln\left|x^2-4\right|$ which is the right solution. However, the plan of the system was to teach him solve with the method of partial fractions (background goal inherited from the upper level), so the system's solution would be to transform the initial integrand into $\frac{A}{x-2}+\frac{B}{x+2}$, then to find out the coefficients A, B by solving a system of linear equations and finally to obtain $\left(\frac{1}{2}\right)\int\frac{dx}{x-2}+\left(\frac{1}{2}\right)\int\frac{dx}{x+2}$ which transforms into a sum of standard integrals $\left(\frac{1}{2}\right)\ln|x-2|+\left(\frac{1}{2}\right)\ln|x+2|$, equivalent to $\left(\frac{1}{2}\right)\ln\left|x^2-4\right|$. Because of the background teaching goal, a reaction will be given at the situation which arises at the first step (where the student's solution differs from the system's). The reaction will be a remedial action to bring him to the system's plan. If there is not a specific background goal or if the pedagogical rules do not assign so high priority to background goals, the system would let the student proceed in his own way. As mentioned before, the pedagogical rules can define a completely different behavior of the system.

8 Conclusions and Further Work

In this paper we propose an architecture for reactive planning of contents in instruction. In contrast with classical non-linear, hierarchical, least commitment planning, our approach allows global evaluation of the plan and selection of optimal one, coping with interactive goals and hierarchical planning on different levels of organising the material. The architecture is based on a framework for reactive planning integrating opportunistic reactions with plan-based (plan-repairs and complete replanning). It is suggested how this framework can implement two radically different teaching styles, teaching and coaching, as well as their interaction within one system. One additional advantage is the possibility to manage the way of the system's reacting by means of different pedagogical rules. At this point the system has been implemented in the domain of integration of elementary functions at three levels or domain knowledge organisation: curricular - methods for integration, solving integration problems and performing single transformations. A tool for editing diagnostic operators and pedagogical rules has been developed and three rule-sets have been created ad-hoc. Our current work is aimed at identifying such rules by interviewing teachers. A tool for induction of pedagogical rules from protocols is being implemented now, applying machine learning techniques. We intend to compare the rules derived from protocols with those which follow from some general frameworks and theories known in didactic.

Acknowledgements:

This work has been supported by project I-406 of the Bulgarian Ministry of Science and Higher Education.

References:

Ambros-Ingerson J. (1987) IPEM: Integrated Planning, Execution and Monitoring. PhD thesis, Department of Computer Science, University of Essex.

Beetz M. & McDermott D. (1992) Declarative Goals in Reactive Plans, in Artificial Intelligence Planning Systems, Proceedings of the 1st International conference AIPS'92, College Park, MD: 3-12.

Brooks, R.A. (1991) Intelligence without representation, Artificial Intelligence, 47(1-3), 139-160.

Cialdea, M., Micareli, A., Nardi, D, Spohrer, J. and Aiello, L. (1990) Meta-level reasoning for diagnosis in ITS, Technical Report DIS, University of Rome "La Sapienza".

Cronbach, L.J. (1967) How can Instruction be Adapted to Individual Differences? in R.M. Gagne (Ed.) Learning and Individual Differences, Merrill: Columbus, Ohio, 353-379.

Einsiedler, W. (1976) Lehrstrategien und Lehrerfolg: eine Untersuchung zur lehrziel- und schülerorientierten Unterrichtsforschung. Beltz: Weinheim.

Epstein, S. (1995) On Heuristic Reasoning, Reactivity and Search, Proceedings IJCAI'95, Montreal, 20-25 August 1995, 454-461.

Etherington D.W. (1987) Formalizing nonmonotonic reasoning systems, Artif. Intel., 31, 41-85.

Etzioni O., Hanks, S., Weld, D., Draper, D., Lesh, N., Williamson, M. (1992) An approach to planning with incomplete information, in Proc. 3rd International Conference on Principles of Knowledge Representation and Reasoning, Cambridge MA, 115-125.

Fagin R., Halpern J.Y (1987) Belief, awareness and limited reasoning, Artif.Intel, 34, 39-76.

Fikes, R., Nilsson, N. (1971) STRIPS: a new approach to the application of theorem proving to problem solving.. *Artificial Intelligence*, 2(3/4), 189-208.

Firby, R.-J. (1987) An investigation into reactive planning in complex domains, in Proceedings of the 6th National Conference on Artificial Intelligence, Seattle, WA, 202-206.

Glaser, R. (1972) Individuals and Learning: The New Aptitudes, Educational Researcher, 1(6), 5-13.

Greer, J.& McCalla, G. (1989) A Computational Framework for Granularity and its Application to Educational Diagnosis, Proceedings IJCAI-89, Detroit, 477-482.

Greer, J. & McCalla, G. (1993) Student Modeling: the Key to Individualized Knowledge-Based Instruction, NATO ASI Series; Springer Verlag: Berlin.

Hanks S. & Weld D. (1992) The systematic Plan adapter: A formal foundation of case-based planning. Technical Report 92-09-04, Department of Computer Science and Engineering, University of Washington, Seattle.

Hayes-Roth, B. (1993) Opportunistic Control of Action in Intelligent Agents, IEEE Transactions on Systems, Man , and Cybernetics, vol.23., no 6, 1575-1585.

Huang X., McCalla, G., Greer, J., Neufeld, E. (1991-a) Revising Deductive Knowledge and Stereotypical Knowledge in a Student Model, User Modelling and User Adapted Interaction, 1, 1, 87-115.

Huang, X., McCalla, G. I., and Neufeld, E. (1991-b) "Using Attention in Belief Revision", Proc. 9th Conference of the American Association for Artificial Intelligence, Anaheim, California, July 1991, pp. 275-280.

Kay, J. (1994) Lies, Damned Lies and Stereotypes, Proceedings of the 4. International Conference on User Modeling, UM'94, Cape Cod, MA.

Kay, J. (1995) Special Issue on Student Modeling, J. Kay, (Guest Editor) User Modeling and User Adapted Interaction, **4** (3), 149-196.

Knoblock (1995) Planning, Executing, Sensing, and Replanning for Information Gathering, in Proceedings IJCAI-95, 1686-1693.

Korf R.E. (1990) Real Time Heuristic Search, Artificial Intelligence, 42 (2-3): 189-211.

Lyons D., Hendriks A., Mehta S. (1991) Achieving Robustness by Casting Planning as Adaptation of a Reactive System, IEEE International Conference on Robotics and Automation, April, 1991, IEEE, New York.

McCalla G., Greer J., Coulman R. (1992) Enhancing the Robustness of Model-Based Recognition, in Proceedings of the 3rd International Workshop on User Modelling, Dagstuhl, Germany, August 1992, 240-248.

Minton, S., Bresina J., Drummond, M. (1994) Total Order and Partial Order Planning: A Comparative Analysis, *Journal of Artificial Intelligence Research*, 2, 227-262.

Nilsson, N. (1980) *Principles of Artificial Intelligence*, Tioga Publ.: Palo Alto.

Olawsky D. & Gini M. (1990) Deferred Planning and Sensor Use, in Proc. of the Workshop on Innovative Approaches to Planning, Scheduling and Control, San Diego, CA, 166-174.

Peachey D., McCalla, G. (1986) Using Planning Techniques in Intelligent Tutoring Systems, Int. J. Man-Machine Stud., 24, 77-98.

Rich, E. (1986) Users are individuals: Individualizing User Models, Int. J. Man-Machine Studies, 18: 199-214.

Schoppers M. (1987) Universal Plans for Reactive Robots in Unpredictable Environments, Proceedings IJCAI-87, Morgan Kaufman, San Mateo CA, 1039-1046.

Self, J. (1991) Formal Approaches to Student Modeling, in Greer & McCalla (Eds) Student Modeling: the Key to Individualized Knowledge-Based Instruction, NATO ASI Series; Springer Verlag: Berlin.

Self, J. (1994) The Role of Student Models in Learning Environments, *IEICE Trans. Inf. & Syst.*, E77-D, no.1. 3-8.

Slagle, J. (1963) A heuristic Program that Solves Symbolic Integration Problems in Freshman's Calculus, *Journal of the ACM*, 10, 507-520.

Spector L., Hendler, J. (1990) An Abstraction -Partitioned Model for Reactive Planning" in Y. Wilks and P. McKevitt eds. 4-th Rocky Mountain Conference on AI, Computing Research Laboratory, New Mexico State University, Las Cruces, N.M., June 1990.

Vassileva J. (1990) An Architecture and Methodology for Creating a Domain-Independent Plan-based Intelligent Tutoring System, *Educational & Training Technologies International*, 27,4, 386-397.

Vassileva J. (1992a) Pedagogical Decisions within an ITS-shell. Computers & Education, 18, 39-43.

Vassileva J. (1992b) A Three-Dimensional Perspective on the Current Trends in Student Modelling, Proceedings of EAST-WEST Conference on Emerging Technologies in Education, Moscow, 315-320.

Vassileva J., Radev R., Dimchev B., Madjarova J. (1991) TOBIE: Experimental ICAI-Software in Mathematics, in Proceedings of the International Conference on Computer Assisted Learning in Science and Engineering CALISCE'91, Lausanne, 9-11 September 1991, 145-150.

Veloso, M., Perez, M. & Carbonell, J. (1990) Nonlinear planning with parallel resource allocation. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control.*

Wasson B. (1990) Determining the Focus of Instruction: Content Planning for Intelligent Tutoring Systems, Doctoral Thesis, Department of Computational Science, University of Saskatchewan.

Wilkins, D. (1988) *Practical Planning: Extending the Classical AI Planning Paradigm*, Morgan-Kaufmann: San Mateo.