Instructional Planning Approaches: from Tutoring towards Free Learning[1]

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

BARBARA WASSON
Department of Information Science, University of Bergen, N-5020 Bergen, Norway
E-Mail: barbara.wasson@ifi.uib.no

Abstract: This paper provides a review of existing instructional planning approaches, classified in two groups: delivery and content planning (Wasson, 1990). It is suggested that instructional planning doesn't have to be associated with a particular teaching style or locus of control. Two examples are used to illustrate how instructional planning can implement different instructional strategies, some of which allow the student a high degree of freedom and initiative. Finally, the paper suggests new applications of instructional planning in educational collaborative problem solving- and simulation-systems.

1 Introduction

The debate about the locus of control (LOC) between the Computer System / Designer (SYS) and the Learner (L) has been active since the dawn of the development of Computer Based Learning Systems. LOC is defined as "the relative balance of initiative between any two parties" (McAleese, 1990). So far in practice both extremes - "LOC in SYS" and "LOC in L" have been realised in computer systems. The first extreme can be seen in numerous examples of CAI and curriculum-based intelligent tutoring systems; the other one in microworlds, open learning environments, and educational hypermedia systems. Arguments against both extremes can be found. Reiber (1994) would characterize this as a debate between instructivists and constructivists, respectively. However, these arguments seem to be on different wavelengths: they assume different roles of computers in education. While the "LOC in SYS" camp advocates the role of the computer is to teach the student some knowledge i.e. a role taken by human teachers, the "LOC in L" camp accepts computers only as a teaching medium which is completely neutral with respect to teaching, (like an OH projector, a laboratory tool, an encyclopaedia or an enhanced telephone). Since both a teacher and a teaching medium are necessary components of the learning environment, the arguments favoring one over the other, are unacceptable.

AI-ED research has taken a course between the two extremes. Usually, in Intelligent Tutoring Systems (ITS) the LOC lays closer to the SYS (since the system selects the problems which the student is going to work on, and interferes with instructional interactions when certain events occur). Yet there are coaches and systems supporting negotiation of teaching goals, and collaboration among groups of students, where the LOC is moved far in the direction of the learner.

Instructional Planning is irrelevant only if we take the "LOC in L" position, i.e. if we assume that the computer is nothing more than a passive medium. In this case planning has to be made by a human teacher during the organisation of the learning session, by selecting and providing the appropriate environment for every single learning step (or small discovery) that the learners are supposed to make. This automatically excludes the possibility of learning alone with the computer, which significantly shrinks the impact of computers in education. We hope this is not the role that constructivism defines for the computer in learning. Even if the learning environment is only a passive "electronic Play-Doh", material allowing the learner to build her own mental models of the Reality, someone has to take the decision to provide this material with specific properties (color, consistency, hardness etc.) or to select another material (e.g. LEGO building tiles) which is more appropriate for modeling a certain object or phenomenon. Therefore, in this case the planning of instruction again becomes a major issue. As Pontecorvo explains, there are some constructivists, e.g. Bruner (1966), Simons (1993), who would accept this argument: "We look at instruction as systematic and planned

[1] Published in Proceedings of Euro-AIED'96, Lisbon, 30.9.- 2.10. 1996, pp.1-8.

activity which is aimed at learning and includes teaching. There is no opposition between the view of constructivist learner and the view of a planning teacher (Pontecorvo, 1993, p. 139)".

2 Approaches for Instructional Planning

Pedagogical decision making is concerned with both the content (what goals to focus on) and the delivery (how to achieve the goals) of instruction (Wasson, 1990; Dijkstra et al., 1992). Content planning entails generating, ordering and selecting which content goals should be the focus of an instructional interaction given the current state of the student, and monitoring the execution of the content plan in order to determine when to re-plan, or generate a new plan (Wasson, 1990). Delivery planning is concerned with choosing the actual activities and instructional interactions that help the learner achieve the goals (e.g., Mohan, et al., 1992; Merrill, Li and Jones, 1992). Delivery planning is what many authors call "teaching strategies" in ITS research.

2.1 Delivery Planning Approaches

The majority of instructional planners developed to date have focused on aspects of delivery. The systems essentially start with an already chosen content and concentrate on determining how to deliver it — translating the already chosen content into a pedagogically effective form for presentation to the student. "Delivery planning" should decide how to sequence explanations, tests, presentation, problems, exploration, and how to manage the initiative in a tutorial dialogue. Most of the early approaches have been influenced by ideas and methods developed in dialogue systems and text generation. Woolf & McDonald (1984) model discourse strategies used by human tutors. They are computationally represented as augmented transition networks (ATNs). KAFITS (Murray & Woolf, 1992) uses a 4-layered representation of instructional primitives: goals, topics, presentations and responses and a template network representing a "dummy" teaching strategy. It can be seen as a general delivery plan which is instantiated by the teacher. Mohan (1991) has implemented Shuell's (1992) cognitive theory of instruction (learning functions) in IDP, an instructional delivery planner. GTE (Van Marcke, 1992) represents teaching expertise with generic *tasks*, *methods* (for carrying out these tasks) and *objects* (units of learning materials employed by the method). COCA (Major & Reichelt, 1992) uses teacher-defined rules for selection of: next concept to teach (content planning), nature of next interaction (teach, test, summarise), content of next interaction and the response to the student. McArthur et al. (1988) have focused on a mechanism for generating a task which encompasses the chosen set of skills that are the current focus of instruction — a step of delivery planning. Winkels & Breuker (1993) describe a method for generating instructional tasks from domain knowledge representation. For them a curriculum is a network of didactic goals. The curriculum is generated off-line from the domain knowledge representation and can be later used by an intelligent tutoring or help system.

2.2 Content Planning Approaches

The evolution of content planning can be traced from the days when systems were built on top of a rigid course graph. In WUSOR-III domain knowledge is represented in terms of a *genetic graph* (Goldstein, 1982), which keeps track of predictable stages in the evolution of student knowledge. The various links connect related concepts via possible cognitive operations through which one concept might be learned from another. The tutoring (coaching) strategy in WUSOR-III is to identify fringe nodes known to the student, and then use genetic relationships from these fringe nodes to suggest where coaching might help. The student's way through the genetic graph is dynamically generated based on the student model (overlay). In this way for the first time the content of what is going to be taught is decided flexibly adapting to the progress of the student.

McCalla, Peachey & Ward (1982) proposed the use of an AND/OR graph representation of domain concept and *classical planning* to structure a large CAI course. Recognising the granularity of knowledge, any node in the AND/OR graph can be broken down into a sub-graph of finer grained concepts which together constitute the superconcept. The student model in this nested AND/OR graph is a standard overlay, indicating which concepts are known or unknown to the student. Peachey & McCalla (1986) develop further the idea of applying classical planning techniques to content planning, by applying a *STRIPS-like representation* of the concept structure where the AND/OR graph is encoded in operators. The conditions and effects of the operators are logical propositions. The student model is also specified as logical propositions indicating concepts the student knows or misconceptions.

Macmillan and Sleeman (1986) pioneered the use of a *blackboard architecture* in an instructional system. Murray's system (1989) builds on this work. It uses a particular blackboard architecture called BB1 where the instructional plan is kept. The tutoring strategy essentially takes part in three phases: plan lesson objectives, select a tutorial strategy, and execute the instructional plan. During lesson planning, topics are sequenced and a current topic is selected. The lesson plan is created from stored skeletal plans and is not individualised to the student. The second phase, selecting a tutorial strategy, involves finding an appropriate presentation and assessment strategy for the current topic. The combination of the lesson plan and the presentation plan constitute the system's full instructional plan, which is stored on the control blackboard. During the third phase, plan execution, any number of knowledge sources may think they are relevant (i.e. they may "want to fire"). The central role that Murray sees for the instructional planner is as an arbiter among competing knowledge sources. In this way extremely dynamic teaching strategies can be implemented.

A comprehensive survey of how instructional planning has evolved from rigid course graphs to flexible, dynamic content planning systems can be found in McCalla (1992).

3 Instructional Planning for Various Instructional Strategies

3.1 PEPE: Content Planning and Cognitive Apprenticeship

PEPE (Wasson, 1990; 1992) is a competence-based computational framework for a content planner that views pedagogical decision making as a planning problem. A prototype based on the PEPE framework has been implemented and tested within the domain of recursive LISP programming. 51 planning rules encode a top-down prerequisite-first tutoring strategy. Content planning, as defined in PEPE, centres around content goals which are expressed in terms of the concepts the student must learn, and the various cognitive abilities a student could have in using these concepts. These can be learning outcomes as defined by, e.g., Bloom (1956), Gagné (1985), or Romiszowski (1984). For example a content goal might be "have the student learn to analyse the relationship between two concepts". The specification of a particular cognitive ability that it is expected that the student perform, constrains the type of learning situation in which the system engages the student. For example, if the goal is to have the student learn a fact, the system might direct, tell or show, whereas if the goal is to have the student analyse a concept, the system might probe, guide, observe or provide required resources (e.g. provide an appropriate cognitive tool).

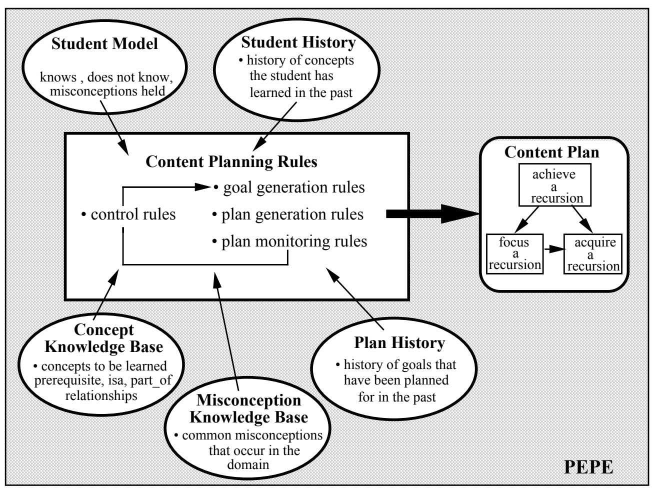

Figure 1: PEPE's conceptual architecture

Figure 1 presents the conceptual architecture of PEPE and the additional knowledge bases that provide the required knowledge. A concept knowledge base (CKB) identifies the concepts to be learned and the relationships (e.g. prerequisite, subconcept, specialization, etc...) between them. It constitutes the tutoring system's conceptual understanding of a domain. That is, the CKB abstracts, into concepts, the domain knowledge the student is to learn (i.e. the CKB does not constitute the domain knowledge, but rather it is the goal structure for learning it). The misconception knowledge base (MKB) enumerates common misconceptions that occur in the domain and relates them to concepts in the CKB. The student model (SM) keeps track of the system's beliefs about the student's understanding (and misunderstanding) of the domain. The SM used by PEPE is an overlay model where each entry corresponds to a concept in the CKB and an ability to use that concept. The student history (SH) maintains the history of concepts the system believes the student has learned, and the plan history (PH) records the goals and plans that have been generated by PEPE to date, and whether or not they were successful.

Content planning in PEPE consists of three phases: goal generation, plan generation and plan monitoring. Content planning rules (CPR) encode these three phases. First, a content goal relevant to the student's current knowledge state, the goal structure of the domain to be learned, and the pedagogical principles for manoeuvring through the goal structure, is generated. This is a system goal for generating a content plan and not a goal for the student. Once the current goal has been set, a content plan is generated based on the history of the student's learning, the current knowledge state of the student, and the pedagogical principles which express how to engage the student in achieving the content goal. Finally, the execution of the content plan (by the delivery planner) is monitored in order to determine when the current content plan has been carried out successfully, needs to be interrupted, or has failed and replanning must take place.

The CPR consist of two types of rules: pedagogical planning rules which encode pedagogical principles (PPR), and control rules (CR) which manage the interactions of the pedagogical planning rules. The pedagogical philosophy encoded in the PPR is captured in goal generation rules, plan generation rules and plan monitoring rules which correspond to the three phases of content planning. An example of the PPR can be found in Table 1. The rules shown here encode a tutor-style instructional strategy where the system controls the decision of the next concept on which to focus. The CR manage the planning cycle and thus determine when each of the phases is carried out. They consist of meta-rules that describe the interactions of the rules within each phase of the planning cycle as well as between the phases.

Table 1 Pedagogical planning rules for a content planner

1. Goal Generation
- goal prerequiste rules (e.g. achieve facts before analysis)
- subgoal generation rules (e.g. when achieving analysis of a concept achieve analysis of one of its specializations)
- subgoal selection rules
  - meta selection rules (e.g. use concept prerequisite rules before pedagogical preference rules)
  - concept prerequisite rules (e.g. do iteration before recursion)
  - pedagogical preference rules (e.g. prefer cdr recursion before car recursion - can be domain dependent preferences of an individual teacher or domain expert)

2. Plan Generation
- operator assignment rules (e.g. if the student knows concept x then review it)
- modifier assignment rules (e.g. when to plan to achieve fact, analysis or synthesis of a concept)
- plan subgoal generation rules (e.g. if planning to achieve analysis of x then focus on analysis of x and then have the student acquire x)

3. Plan Monitoring
- blocking rules (e.g. if the executor has tried twice in a row to satisfy a goal without success, assign the goal block-1 status)
- plan adaptation rules (e.g. if the currently planned for goal is not achieved and the current plan has been carried out, then the plan has failed)

In PEPE, a content plan is derived to achieve a content goal. Each node in the content plan consists of an operator, a modifier and a content. The operator encodes the notion of planning "learning events" (or "learning functions" Shuell (1992)) by indicating to the delivery planner the type of cognitive processing in which to engage the student. In figure 1, a plan for the content goal "have the student learn to analyse recursion" has been generated using three operators: achieve, focus and acquire. Achieve means have the student learn, focus refers to setting up an expectation for learning, and acquire means set up a learning situation where the student can acquire the new information. Note that it does not say anything about how to help the student acquire it, that is a delivery issue. The content of a plan node is the concept from the current content goal (recursion in figure 1). The plan node's modifier indicates which ability (e.g. f-fact; a-analysis; s-synthesis adapted from Bloom (1956)) is appropriate, and it too is taken from the current content goal. The nodes of the content plan are linked by either substep links that indicate a hierarchical organization, or by sequence links indicating an execution order. If a plan node has subnodes then execution of the node is carried out by executing the subnodes.

Wasson (1990) and McCalla & Wasson (1992) present a version of PEPE's PPR extended to allow for system-student negotiation of content goals. Furthermore, Wasson (1990) gives examples of how the instructional strategies Socratic Tutoring and Cognitive Apprenticeship have been encoded by PEPE's PPR and how rules for determining how to select an appropriate instructional strategy have been given. For example:

If the domain to be learned involves problem-solving
Then use a cognitive apprenticeship strategy.

The apprenticeship model provides encouragement and instruction when needed and lets the "master" (teacher) fade back when the student has reached an acceptable level of performance. To encode a simplified cognitive apprenticeship strategy in the PPR, the operators *(re)summarize*, *demonstrate*, and *perform* were added to enable an interleaving of system summaries (to give an overview of part of the domain to be learned) and demonstrations (which could interactively involve the student, but the system is mainly in control) with student problem-solving (which could be interrupted with local plans for giving remediation consisting of various levels of hints). The choice of "demonstrate x" or "perform x" (or a sequence of both) was encoded into the planning rules. Two examples are:

If the student model (SM) indicates the student knows concept x

Then have the student solve a problem using concept x (i.e. operator = perform).
If the SM does not contain any information as to whether concept x is known, and concept x has no specialization, Then have the system demonstrate solving a problem that requires knowledge of concept x (i.e. operator = demonstrate).

3.2 TOBIE: Reactive Planning for Coaching and Tutoring

In a real teaching process the goals are far more complex than the content goals in Peachey & McCalla (1986), Murray (1989) etc. and usually contain several dimensions. In didactic literature a distinction is made among three types: cognitive, affective and psychomotoric, and four levels of instructional goals: analysis / synthesis, knowledge, application, transfer. In addition, one can distinguish between cognitive and meta-cognitive goals. The fact-analysis-synthesis dimension has been considered by PEPE. The other types and levels of goals have not been accounted for in instructional planning so far. In practice these goals are often pursued simultaneously by teachers. For example, teachers interrupt working on a specific cognitive goal to pursue an affective goal (to increase the student's motivation) or to stress a meta-cognitive issue or which has been in the background and comes to play only when a good opportunity arises. Sometimes they even change the current content, for example, to show how the same or an similar method is applied in a different domain or problem. However, if during planning we consider all possible levels and types of simultaneous teaching goals, as well as the potential opportunities for fulfilling them, planning of the teaching session becomes too complex to be practically possible. The only solution is to plan in a limited description of the content-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.

A technical approach for this solution provides *reactive planning*. 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) which at the planning stage have not been foreseen for different reasons (e.g. because they were not known or because it would have been too expensive to consider them). The important feature of this definition of a reactive planner is that via continuous sensing it lets the dynamic characteristics of the environment define the points when a change in the plan execution may happen. Unlike in other planners, these points are not defined by the planner in advance (at planning stage). 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. We proposed a framework for reactive instructional planning called TOBIE (Vassileva, 1990), (Vassileva et al., 1991). It provides a uniform and modular language for domain knowledge representation and thus allows different levels of knowledge organisation to be represented at the same time in the system (see Figure 3).

TOBIE is an ITS-shell architecture based on content planning. It centres around a domain knowledge representation consisting of a multi-layered directed AND/OR graph. AND/OR graphs provide a mighty representation language in which curricular (concept) structures, goal (task) decompositions and problem-solving spaces can be expressed. These graphs are represented by sets of production rules, encoded in TOBIE by means of Teaching Operators (TOs), similarly to (Peachey & McCalla, 1986). The model of the student's domain knowledge is an overlay of the AND/OR graphs. A model of the student's personal characteristics represents psychological and motivational parameters, such as field-dependence, concentration, motivation, confidence, and persistence. The Pedagogical Component contains a Planner, which dynamically generates content plans in the knowledge structure to achieve certain content goals, and an Executor, which carries out the plan, re-invokes the Planner or reacts locally to arising opportunities.

During the execution of the plan a situation can arise from a combination of external (environmental) factors which can not be predicted at planning stage. For example, a violation of time-restriction, evidence that the student is no longer concentrated or motivated, 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. TOBIE provides four principle types of reactions: ignoring the situation, an opportunistic reaction without changing the plan; local plan repair and global re-planning. A set of pedagogical rules (PR) is responsible to select a reaction to situations that occur during plan-execution. A tool for creating PR and diagnostic operators is provided that allows the definition of new possible situations as combinations of factors.

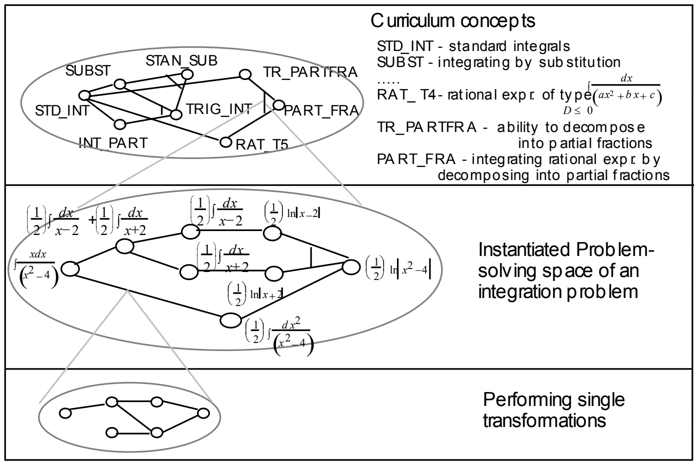

Figure 3. Representing different levels of knowledge organization

It is extremely difficult to mix interactively different teaching strategies, for example to teach new material and to give the student the freedom to explore and try solving the problems alone. In TOBIE *tutoring and coaching*, two radically opposite (with respect to the LOC) teaching styles can interact. Currently we are investigating the didactic issues involved in taking pedagogical decisions to support such a mixed initiative environment. Both strategies can be modelled as a planning process. The only difference is who takes the LOC. In the first case, it is the system who creates a plan for teaching a given concept (or solving a given problem) and leads the student to achieving the goal. In this way, depending on the level of the domain knowledge representation, the system can either present a given content (i.e. teach certain concepts) or demonstrate a problem solution step by step.

A typical use of coaching style is to support the student in practising problem solving. By using its planning mechanism in the problem-solving level of domain knowledge representation, the system can generate a solution: a problem-solving space (AND/OR graph) and a subgraph (solution plan) which leads from the initial state (problem condition) to the goal state (problem solution). If several plans are possible, it selects an optimal one according to specific domain-dependent or pedagogical criteria. Instead of executing the plan, the system only observes the student's actions and tries to match them with its own plan. In the case that the student goes along the system's plan, no reaction from the system is needed. In case of a difference, however, a situation occurs which may need a reaction from the system. The possible reactions are: ignoring the difference (keeping silent); local reaction aimed at bringing the student back on the system's plan; local plan-repair (trying to find a way to accommodate the student's solution within the system's plan); global replanning with the same goal (trying to find another plan for solving the problem that fits with the student's solution), or global replanning with a different goal (giving the student another problem, or switching from coaching to tutoring mode to demonstrate the solution of the problem step by step).

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, interfere to motivate the student or give help*. The pedagogical rules for selecting a concrete 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, choose a re-planning reaction to accommodate his way of solving (i.e. keep silent) instead of bringing him back with a remedial operator to the corresponding state in system's plan* (Cronbach, 1967). A combination of the two strategies is shown with an example from the domain of symbolic integration in (Vassileva, 1995).

Here we must stress once again that there is no difference in the planning and executing mechanisms used to implement both teaching styles. Planning is only a technique, which in the case of TOBIE is not only "content" planning (Wasson, 1990), but in some sense, "delivery" planning. It also shows that instructional planning does not presume that LOC is entirely in the System; just the other way around; it can be used to implement a "LOC in the Student" and a broad spectrum of mixed initiative behaviors.

4 Summary and Future

The notion of instructional planning was born from the desire for more flexible control in CAL systems (McCalla, 1992). Ideally, systems to support human learning should play a role in helping learners to choose their next learning activity. In traditional CAL systems, the system controlled the interaction with the learner, choosing the instructional activity and judging when, or even if, the learner was allowed to move onto new activities. With the advent of ITS, however, it became possible to provide more learner control in making instructional choices. Techniques were found that allowed the learner more freedom and gave the system the ability to act as a coach and advisor, rather than an all-knowing controller of the learner's destiny.

Instructional planning is the process of mapping out a global sequence of instructional goals and actions that provides consistency, coherence and continuity in the instructional process. It can be applied at two levels:

- **planning of content** - the process of selecting the content for an instructional goal that places the student on an appropriate learning path.
- **planning of delivery** - the process of optimal selecting and sequencing of the tutorial interactions focused on a given content.

It has been shown that instructional planning is applicable to more than just strict tutoring style ITSs. Planning approaches have been used to implement various instructional and teaching strategies such as tutoring, coaching, cognitive apprenticeship, Socratic dialogue, and combinations of these. PEPE, which focused only on content planning, demonstrated its usefulness for encoding strict tutoring -> negotiated tutoring -> cognitive apprenticeship. TOBIE, a complete ITS-shell architecture modelled strict tutoring, coaching and opportunistic pedagogical decision making. We re-emphasise that planning does not imply "LOC in SYS", but can be used to cover the whole spectrum between "LOC in SYS" and "LOC in L".

In fact, any environment to support learning can be enhanced through the use of planning techniques since they can be used as a basis for implementing certain teaching strategies. For example, a kind of planning is performed in COSOFT Hoppe (1995), Ikeda, Hoppe & Mizoguchi (1995), which is a system supporting students' collaborative problem solving. Based on modeling the individual students and comparing their progress in solving a problem, it is able to suggest partners for a team or knowledgeable students who can help with a specific problem. One can imagine such "invisible" planning taking place in a free learning environment, for example, based on simulation, by changing its parameters, difficulty etc. in order to support the acquisition of particular knowledge or training a specific skill. Planning is only a technique; it is neutral to the particular educational philosophy underlying the system. However, it can be used to implement a given educational philosophy. That is why we don't think that the age of planning is gone — *it has arrived!*

References

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

Bloom, B.S. (1956). Taxonomy of Educational Objectives, Handbook I: Cognitive Domain. New York: David McKay.

Bruner, J. (1966). Toward a Theory of Instruction. Combridge, Mass: Harvard University Press.

Dijkstra, S., Krammer, H.P.M. & van Merrienboer, J.J.G. (Eds.) (1992). Instructional Models in Computer-based learning environments. NATO ASI Series F, Vol 104. New York: Springer-Verlag.

Gagné, R.M. (1985). The Conditions of Learning and Theory of Instruction. New York: Holt, Rinehart & Winston.

Goldstein, I.P. (1982) The genetic graph: a representation for the evolution of procedural knowledge. In: Intelligent tutoring systems (D.H. Sleeman and J.S. Brown, eds.), pp. 51-77,. London: Academic Press.

Hoppe, H.-U. (1995) The Use of Multiple Student Modeling to Parametrize Group Learning, in Proceedings of the 7-th World Conference on AI and Education, Washington, August 16-19, AACE, pp. 234-241.

Ikeda, M., Hoppe U., Mizoguchi, R. (1995) Ontological Issues of CDCL Systems, the 7-th World Conference on AI and Education, Washington, August 16-19, AACE, pp. 242-249.

Macmillan, S.A. and Sleeman, D.H. (1987) An architecture for a self-improving instructional planner for intelligent tutoring systems. Computational Intelligence, 3, 1, 17-27.

Major, N. & Reichelt, H. (1992) COCA: a shell for intelligent tutoring systems, Proceedings ITS'92, 66-73.

McAlleese (1990) Locus of Control: does hypertext make adaptive / intelligent systems obsolete? Panel statement, in A. Rizk, N. Streitz & J. Andre (eds.) *HYPERTEXT: Concepts, Systems, Applications*, Cambridge University Press.

McArthur, D., Stasz, C., Hotta, J., Peter, O. & Burdorf, C. (1988). Skill-oriented sequencing in an intelligent tutor for basic algebra. Instructional Science, 17, 281-307.

McCalla, G.I., Peachey, D.R. & Ward, B. (1982). An architecture for the design of large scale intelligent teaching systems. Proceedings of the Fourth National Conference of CSCSI. Saskatoon, Saskatchewan.

McCalla, G. (1992) The search for adaptability, flexibility, and individualization: approaches to curriculum in intelligent tutoring systems. In: Adaptive Learning Environments (M. Jones and P. Winne, eds.), Nato ASI Series F, Vol. 85, Springer-Verlag, Berlin, pp. 91-121.

McCalla, G.I. and Wasson, B.J. (1992). Negotiated learning needs student modelling and instructional planning. In R. Moyse & M. Elsom-Cook (Eds.), Knowledge Negotiation. London: Chapman.

Merrill, M.D., Li, Z. & Jones, M.K. (1992). An introduction to instructional transaction theory. In S.A. Dijkstra, H.P.M Krammer & J.J.G. van Merrienboer (Eds.) Instructional Models in Computer-based learning environments. NATO ASI Series F, Vol 104, 15-41. New York: Springer-Verlag.

Mohan, P. (1991). Adapting instruction to individual learners: delivery planning for intelligent tutoring systems. MSc Thesis, Department of Computational Science, University of Saskatchewan, Saskatoon, Canada.

Mohan, P., Greer, J. & Jones, M. (1992). Enhancing the instructional capability of intelligent tutoring systems. In C. Frasson, G. Gauthier & G.I. McCalla (Eds.) Proceedings of the Second International Conference on Intelligent Tutoring Systems (ITS'92), 532-538, Montreal, Canada.

Murray, W (1989) Control for intelligent tutoring systems: a blackboard based dynamic instructional planner. Proceedings of the 4th International Conference on AI and Education, Amsterdam, 150-168.

Murray, T & Woolf B. (1992) Tools for Teachers Participation in ITS Design., in Frasson, C., Gauthier, G., McCalla, G. (eds.) Proceedings of the Second International Conference on Intelligent Tutoring Systems (ITS'92), Montreal, Canada, Berlin: Springer Verlag.

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

Pontecorvo, C. (1993). Developing literacy skills through cooperative computer use: Issues for learning and instruction. In T.M. Duffy, J. Lowyck, D.H. Jonassen (Eds.) Designing Environments for Constructive Learning, 139-160, NATO ASI Series F. Heidelberg: Springer-Verlag.

Reiber, L.P. (1994). Computers, Graphics, & Learning. Duvuque, IA: Wm. C. Brown Communications, Inc., Brown & Benchmark.

Romiszowski, A.J. (1984). Producing instructional systems: Lesson planning for individualized and group learning design. London: Kogan Page.

Shuell, T.J. (1992). Designing Instructional Computing Systems for Meaningful Learning. In M. Jones & P. Winne (Eds.) Adaptive Learning Environments. NATO ASI Series F, Vol 85, 19-54. New York: Springer-Verlag.

Simons, P.R.J. (1992) Constructive learning: The role of the learner. In T.M. Duffy, J. Lowyck, D.H. Jonassen (Eds.) Designing Environments for Constructive Learning, 291-313, NATO ASI Series F. Heidelberg: Springer-Verlag.

Van Marcke K. (1992) Instructional Expertise, in Frasson, C., Gauthier, G., McCalla, G. (eds.) Proceedings of ITS'92, Montreal, Canada, Berlin: Springer Verlag.

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

Vassileva J. Radev, R. Dimchev B., Madjarova J. (1991) TOBIE: An Experimental ICAI-Software in Mathematics. Proceedings CALISCE'91, Lausanne, 145-150.

Vassileva J. (1995) Reactive Instructional Planning to Support Interacting Teaching Strategies, Proceedings of the 7-th World Conference on AI and Education, Washington, August 16-19, AACE, pp. 334-342.

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

Wasson, B. (1992) PEPE: A computational framework for a content planner. In S.A. Dijkstra, H.P.M Krammer & J.J.G. van Merrienboer (Eds.) Instructional Models in Computer-based learning environments. NATO ASI Series F, Vol 104, 153-170. New York: Springer-Verlag.

Wilkins D. (1988) Practical Planning: Extending the Classical AI Planning Paradigm, Morgan Kaufmann Publ. Inc.: San Mateo, CA.

Winkels R. & Breuker J. (1993) Automatic Generation of Optimal Learning Routes, in Brna P., Ohlsson S. & Pain H. (eds.) Proceedings AI-ED'93, Edinburgh, AACE.

Woolf B., McDonald (1984) Building a computer tutor: Design Issues. IEEE Computer, 17(9), 61-73.

Acknowledgements

Financial support for Barbara Wasson was provided by NorFA (Nordic Research Development Academy) grant number 95.30.183-O. Julita Vassileva has been supported under contract I-406 with the Bulgarian Ministry of Science and Higher Education