Mapping Models of Cognitive Development to Design Principles of Learning Environments

to appear in: De Corte, Mandl, Glaser & Vosniadu, International Perspectives on the Psychological and Educational Foundations of Technology-Based Learning Environment. (ASI series).

Patrick Mendelsohn

Université de Genève

Abstract

In this paper, I will consider a series of questions concerning the domain of knowledge acquisition processes in human by comparing two approaches. On one hand the domain explored by designers of learning environments in their efforts to promote learning processes and knowledge acquisition and, on the other hand the field of developmental cognitive psychology. Questions raised by the various interactions between these two fields can be summarized in two sets of problems:

1. Is it possible to map the basic concepts of developmental psychology like the notions of stages, substages, transition mechanisms, etc. into concrete elements of learning environment components. What can we learn about this maping in order to design more efficient systems ?

2. What are the links between research methods and paradigms in the field of developmental psychology and in the domain of Technology Based Learning ?

These questions will be addressed in reference to the implementation and experimentation of an Intelligent Learning Environment so-called MEMOLAB. A computer based learning environment can be described as “intelligent” if its architecture, interface and components are designed according to the principles of some theory of human learning. To reach this objective, what we should adopt is the interdisciplinary approach advocated by “cognitive scientists” and build intermediary models that act as a bridge between the learning theories and the existing implementation languages. This is the approach that we have been taking with the MEMOLAB project. Its goal is to build an Intelligent Learning Environment for students studying experimental psychology. The system provides the tools for designing, simulating, performing and analyzing experiments on human memory.

Our approach allows two complementary research strategies. On the one hand, we hope to contribute to educational computing by a new design methodology and theoretical framework. On the other hand, we seek to offer developmental cognitive psychologists a test-bed for their models and theories. A particular virtue of MEMOLAB, we believe, is its complexity; a complexity that corresponds to the real-world complexity of cognitive learning: exploration and discovery, practice, analogical reasoning, tutoring and many other strategies will be allowed and supported ... just as they are in any sensible classroom or laboratory.

Postal address:
Faculté de Psychologie et des Sciences de l’Education
9, route de Drize
CH - 1227 Carouge

Introduction

The design of learning environments as a research methodology for learning from instruction

One way to contribute to progress in the understanding of processes and outcomes of learning from instruction is to create environments that elicit and produce them (Glaser, 1992). Theoretically, Intelligent Tutoring System design could represent a solution to this approach since tutoring systems can serve simultanously as instructional environments and as testbeds for theoretical principles. De Corte (1993) stated, even more recently, that the question raised by the existing systems is to decide whether they correspond to actual learning processes as these are now well established by researchers. I would like also to add the question of their compatibility with the outcomes and findings established by research in the field of cognitive development.

Some computer based instructional designers suggest building systems devoted to the promotion of metacognitive skills (like LOGO based microworlds) while others propose promoting conventional cognitive skills and domain dependant abilities (like traditional CAI and some ITS). The opposition between these two views in the design of learning environments has kept these two communities in mutual isolation for a prolonged period. This opposition mirrors the split in the field of cognitive psychology between research in learning processes and research in cognitive development. Broadly speaking, developmentalists have paid more attention to domain independent acquisition processes and learning scientists paid more attention to expertise acquisition through domain dependent mechanisms. Research on Learning Environments seems to bring these two communities closer together. On the one hand, researchers in the learning from instruction community agree now on the fact that efficient learning can no longer be described as a passive recording of knowledge: it is also constructive and self-regulated. On the other hand, cognitive developmentalists are now convinced that development is more domain dependant, goal-oriented, and cumulative than they thought in the past decades.

The emergence of the situated cognition paradigm

Operational definitions given to some of the common basic concepts previously quoted may vary from one researcher to another. Nevertheless it is now admitted, for example, that traditional ITS have little to do with a constructivist approach to learning and that LOGO programming is a useless expertise for real world problems. ITS designers neglected the fact (first established by cognitive developmentalists) that efficient learning is promoted by self-regulated processes and that it supposes an active participation of the learner in the design process of the problem-solving strategies themselves. On the other side,some microworld designers have ignored the fact that acquiring an expertise in some basic constructive set of schemes doesn’t guarantee transfer and use of these schemes to real world situations.

The emergence of the situated cognition paradigm is an opportunity for a new step towards collaboration between the two disciplines. The situated cognition view of learning reminds us that instructional strategies must respect some of the natural processes of knowledge acquisition in order to lead to effective and efficient learning. Core learning (as demonstrated by classical pre-test and post-test experimental designs) guarantees neither transfer nor the use of acquired knowledge in problem solving situations. In other words, we could say that the situated cognition paradigm proposes that instruction should be designed in accordance with the natural characteristics of knowledge acquisition attested by developmental research. The only difference being that schematas acquired by instruction occur in artificial worlds, while schematas studied by developmentalists occur in natural settings.

Cognitive development concepts to the rescue of situated cognition problems

Although the concept of stage has been much criticized in the past, it remains a useful tool to tackle some of the difficult problems raised by the theory of expertise. Providing learning environments with knowledge of relevant developmental stages could help solving two major problems met by instructional designers. First, one of the chief consequences of the “situated cognition” view was that it reminded cognitive scientists that knowledge is anchored in so-called “authentic contexts” by means of sensori-motor actions performed on physical and social reality. This is exactly what developmentalists have tried to explain for years by focusing research on transition mechanisms between sensorimotor and representational schemes. Some of the propositions they have made could be an interesting source of inspiration for instructional designers. Second, the concept of "well-organized" domain-specific knowledge bases, as stated by learning scientists, has something to do with the notion of stage used by developmentalists. In this particular domain, designers could re-discover some interesting views on the flexible application and organizational features of scheme functionning (if you are interested by such findings you can refer to previous work I have done on graphic programming activities in children). In addition to these two problems, we should not neglect the fruitful approach elicited by researchers in the field of socio-cognition (conflict resolution, ...) and their links with collaborative learning processes advocated by the situated cognition community.

Developmental concepts revisited as design principles

Knowledge representation in learning environments

Almost all cognitive developmentalists have formalized acquisition processes in a constructivist manner. This characteristic is linked to their interest in isolating the "elementary building blocks of knowledge" from which more complex behaviors can be constructed. Searching for the optimal elementary cognitive unit decomposition has been the Holy Grail of developmental cognitive psychology, just as the idea of a creating a universal programming language has inspired computer scientists.

ILE designers have also been concerned with the same problem. What should be the generic content of an elementary unit of knowledge presented in a learning environment? How can we make learners participate in the construction of expert knowledge? Should it be an analogue of the "natural" schemes as they are described by developmentalists? Many solutions have been proposed by designers, each one refering to a particular family of systems. In order to be able to represent domain-independant rules for managing pedagogical interaction, CAI designers have often used the notion of "Frame". This neutral concept allows the presentation of knowledge at the interface of a computer using their "conventional" format: definitions, facts, formulas, texts, etc. This solution has the advantage of presenting knowledge as it has to be mastered, but makes it impossible for the learner to be an active participant in the construction of that knowledge. The learner is a passive viewer of a cognitive entertainment to which they are asked to react from time to time. In a classical CAI environment, learners can only reproduce knowledge as it is represented by designers.

ITS designers have attempted to get closer to the constructivist point of view and for this purpose, they have often considered rules as basic cognitive units. This choice was mainly imposed by technological constraints and we must admit that it gave some good results in a few particular domains like the acquistion of diagnosis and troubleshooting skills. One of the main criticism that was made to this approach, advocated by cognitive scientists and some developmentalists, was that they considered knowledge as pure mental representations encapsulated and stored in a way that allowed them to be arbitrarily chosen and activated in appropriate new situations. A totally different approach was developped by microworld designers. Inspired by developmental concepts like schemes and microviews, they tried to simulate the process of scheme integration and differentiation by giving the learner some primitives analogous to those of human beingss. LOGO is the most popular of these attempts and we have already stated the limits of this approach.

With our system Memolab, we have taken a different approach. We consider that the learner's challenge is to master a knowledge base given as a network of various components including powerful transformation procedures, definitions, schemata and formulas generated by the different devices of the system. This learner's appropriation process is progressively sollicited through problem solving situations that involve manipulating these conceptual tools in the same way they would do it in real world. The constructivist approach is reflected in the way learners are not given already solved solutions to their problems so that they participate actively to the design processes of expert knowledge representation.

Stages, substages and knowledge bases

The concept of stage is tightly linked to the constructivist view on development. It supposes that cognitive abilities are self-organized in a cumulative and hierarchical way by various constraints (cognitive load, chunking mechanisms, semantic nets, etc.). These constraints determine the development of specific knowledge bases attached to different domains and lead to well-organized and flexible structures that have some common properties (automatization of recurent skills, elaborative encoding, schema construction....). In a way, we could consider that stages emerge from cognitive functioning as properties of effective learning. This position states that stages are not generative structures as it was proposed by classic constructive theories but rather descriptive structures of knowledge when it is efficiently organized for a real use.

The classical concept of "stage" centers on small subsets of knowledge. Each small piece of information is represented in a specific description language making use of a relevant grammar. This set of rules and objects can be viewed as a qualitative model. Learning environment designers, for example White & Frederiksen (1985), have used this concept of "qualitative model" in the architecture of their system as precursors of more quantitative and formal models. We all agree that expert models of a domain are often too complex to be mastered directly by a novice. They also probably wouldn't be effective for the expert if not anchored in deep sensori-motor structures.

The notion of substage could be considered as a tentative solution to the problem of continuity versus discontinuity in developmental processes (the cumulative component of learning). If development is viewed as a transition between large qualitative stages ruled by specific constraints, a complete description of acquisition processes necessarily needs intermediate transition stages to link local learning processes to broader changes in knowledge acquisition. This local description can be another qualitative model as in Piaget's theory or a quantitative description as in Case's model. We have chosen to implement this last characteristic in Memolab by proposing the metaphor of the Language shift described in a previous paper.

Some concrete implications of a constructivist approach for the design principles of learning environments

This discussion of cognitive development concepts leads to the improvement of several learning environment design principles. I would like to mention here three of them that will be illustrated with some of MEMOLAB's componants in the last section.

Learning environments should be designed as hierarchical knowledge base generators

In the field of instructional design, one major difficulty is to represent interactively the complexity of real world problems. To get round this difficulty, one temptation is to oversimplify the knowledge representation given to the student and to carefully design the solution paths that are supposed to be learned. In simplifying, designers also reduce dramatically the students's actions. I would like to propose that a powerful learning environment must allow the learner to participate in an active and collaborative way in the construction of the system's knowledge base. Effective learning is an individual and personnal implication in the process of schematization and abstraction of the complexity of reality. If this abstraction process is imposed just as a result by the instructional environment (even if it is so for technical reasons), learning will be less efficient and poorly related to meaningfull contexts in which it is supposed to be used. To reach this goal, we have implemented a hierarchy of command languages that progressively generates the complexity of the expert knowledge by a mechanism we have called "language shift".

Learning environment should be designed as powerful dedicated working environments.

In our group, we think that learning environments should be designed as real professionnal working environments favoring all sorts of mental activities such as exploration, investigation and problem solving activities. The richness and complexity of what has to be learned requires well designed knowledge bases, workbenches, integrated hypertexts, on line help. Such an environment should reflect the structural and functional properties of the real "target" working settings.

Learning environment should present knowledge as a communication system

What makes the difference between an “ordinary” professionnal working environment and one that supports efficient learning are devices that allow imitation and mutual regulation processes. These processes are supported by several agents (tutors and coach). Knowledge is communicated and negotiated in what we could call a community of practice. When a learner is stuck in his exploration, when he seems not being able to find the right information, he needs support as in real life. Kintsch (1991) proposed the idea of “unintelligent tutoring systems” just pushing the student’s performance beyond their current level of activity. Indeed, Kintsch’s proposition is very close to Vygotsky’s notion of “zone of proximal development” (Vygotskky, 1978) and this notion is largely supported by the concept of stages (and sub-stages) as stated by cognitive developmentalists.

In Memolab, domain independant knowledge about tutoring doesn’t refer to characteristics of abstract units of knowledge but are concerned with the mode of interaction between the learner and the tutor.

Memolab

Overview

MEMOLAB is a learning environment dedicated to the domain of teaching the methodology of experimental psychology. Basically, the learner’s activities form a cycle: design an experiment, run it and analyze the output data. Here is an example of an experiment that the learner can design:

Present a list of 10 words to two samples of 30 subjects. For the experimental group, the words are selected within the same semantic field. For the control groups, words have no close semantic relation. Let the subjects study the list for 5 minutes and will have to identify the first list words in another list of thirty words. Performance will be measured as the number of words correctly identified.

A challenge of our research was to articulate a computational architecture with psychological theories of learning. Our cognitive architecture divides the learning cycle into stages. In previous papers (Dillenbourg and Mendelsohn, 1992), we used the metaphor of a pyramid with several floors. Floors are discriminated by the operators provided to learners for solving the problem. In MEMOLAB, we paid attention to the relationship between successive goals and grounded our design choices in reference to Cases’s hypothesis on stage transition in children development (Case, 1985).

Actually, in MEMOLAB, each goal does not have a completely different application frame (interface). For each goal, the problem solving situation is the LAB, i.e. a workbench where learners can assemble experiments and run them. However, the set of commands available is not the same in each goal. The LAB is the central place of interaction. It is connected to subordinate frames, such as the data-tools (where learners analyse the data collected through the experiment) and the simulation. The simulation is a set of procedures that approximate the results of the learner’s experiment by comparing this experiment with the most similar experiment found in a database in which we stored experiments described in scientific literature. The simulation method is inspired by case-based reasoning techniques.

Finally, MEMOLAB includes two hypertextes. The `handbook of methodology’ includes several chapters, connected to the various goals of the system. It includes a theoretical introduction to the design of well-formed experiments. The `encyclopaedia of memory’ is another hypertext, independent from any goal, i.e. that learners can access at any time, and which provides them with the knowledge about human memory they could need, for instance, for determining the relevant variables.

Figure 1 illustrates how the learner concretely design an experiment in the first lab (we will see later on that MEMOLAB includes several labs). An experiment is constructed by positioning a set of events on a workbench. The vertical axis represents the time dimension. The activities of a same group a subjects are vertically aligned: first they encode some material,

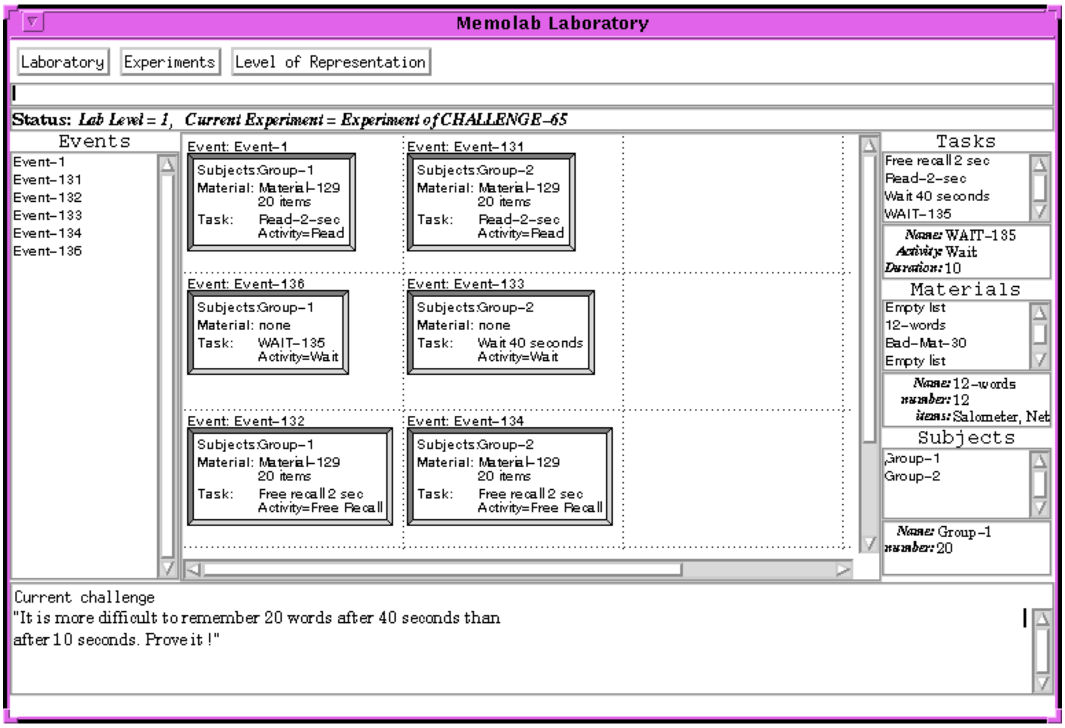

Figure 1: The learner build an experiment (at level 1) by assembling ‘events’ on the workbench. An event defines the task (read, listen, recall, wait,...) of a group of subjects with some material (e.g. a list of 12 words). Panes on both sides of the workbench provide libraries where the learner builds and select the components that he assembles on the workbench.

then there is some delay and then some attempt to recall what they studied before. When an experiment has been designed, the learner may ask to simulate the results.

MEMOLAB as a structured learning environment

If the richness of a learning environment is a quality, its complexity deteriorate learning. The designer must structure his learning environment in such a way that the learner may benefit from this richness. The idea of ‘structuring the environment’ illustrates a fundamental difference of perspective between learning environments and traditional courseware: ***the meaning of ‘teaching’ is moved from asking questions and providing feedback (although this is not excluded) to structuring the environment in way that would provide optimal learning condition***s.

The structure we have adopted is a ***sequence of increasingly complex microworlds*** (or problem solving situations, or laboratories for Memolab). The criterion for defining this sequence is that the last world should be as similar as possible to the real context where the learner will have to apply the acquired skills. For instance, the last lab in Memolab uses the languages that is commonly used by the community of experimenters in their exchanges (papers, conferences,.). However if this microworld is as complex as the real world, it will probably not be efficient for learning. It should be preceded by simpler microworlds. In the first microworld, the learner will have to solve simple problems, considering only some parameters. In the next one, he will receive more powerful operators and use the resources hence available to cope with more parameters and so forth.

The architecture of MEMOLAB includes three microworlds, referred hereafter as ‘levels’. Each level concerns a different set of experiments and uses a partially different interface:

- At level 1, an experiment compares n groups of subjects along one dimension. The interface emphasizes the concrete and chronological aspects of an experiment.

- At level 2, the interface the learner describes an experiment as a set of sequences. A sequence is a generic treatment that can be replicated several times with equivalent materials. The learners design experiments that test the interaction between two parameters.
- At level 3, the interface reifies the logical structure of the experiment as a plan.

Now, as soon as we have a sequence of microworlds, we face the issue of the transition between microworlds. To smooth this transition, we designed the interface in such a way that these microworlds partially overlap. Each microworld uses a different set of commands to create and modify objects the components of an experiment (events, sequences or plans). When an experiment has been completed at level N, the system translates this experiment into the representation that it would have at level N+1. The future representation scheme is hence systematically associated with the scheme mastered by the learner. After some time, the learner is invited to express himself directly with the new language (N+1). This principle has been called the '***language shift***' (Dillenbourg, 1992).

From Campbell and Bickhard's (1986) viewpoint, the language shift mechanism can be viewed as a process of inducing interaction patterns. An elementary interaction associates some sequence of user's actions and the computer's description of this sequence. Inferring the meaning of the description language can indeed be described as the result of inducing the relationship between the actions performed and their representation. This corresponds to a view of knowledge as a mental construction that stands in the interaction between the subject and her environment. It creates a bridge between our model and current research on situated learning (Brown,1990),an important issue in AI and Education.

Our intermediate framework also introduces the designer to the theories of Vygostsky. The apprenticeship idea is reified in the pyramid model by sharing control between the coach and the learner: when the learner is able to perform at some level L, the tutor must guide her activities at level L+1. This level L+1 corresponds to the concept of zone of proximal development (Vygotstky, 1978). At each language shift, the learner will assume a more important control of his solution process and the coach's guidance will be reduced. Moreover, Wertsch (1985) proposed a linguistic analysis of the internalization process that relates it to the language shift. He observed (in mother-child interactions) that the move from the inter-individual to the intra-individual plane was preceded by a language shift inside the inter-individual level: mothers replace a descriptive language by a strategy-oriented language (i.e. a language that refers to objects according to their role in the problem solving strategy).

We used the image of the ***pyramid*** to in order to emphasize the hierarchical relationship between operators that are used in successive microworlds. This notion of hierarchical integration is based on the neo-piagetian theories of R. Case (1985). In short, this theory states that the quantitative improvement of children schemata is limited. To by-pass this limitation, children restructure their knowledge into higher order schemata, that recombine patterns of previously existing sub-schemata.

The pyramid is a visual metaphor for the core structure of a learning environment. The pyramid represents the concepts and skills to be acquired by the learner, ranked bottom-up according to their level of "hierarchical integration". Learning consists in moving up in the pyramid. Each level of the pyramid is defined by two languages: the command language and the description language. The command language vocabulary is the set of elementary actions that the learner is allowed to do at some stage of interaction. The command language syntax defines how the learner composes sequences of elementary actions. The description language is the set of symbols (strings, graphics,...) used by the computer to show the learner some description of her behaviour. This description reifies some abstract features of the learner's behaviour in order to make them explicitly available for metacognitive activities (Collins and Brown, 1988).

The command and description languages are different at each level of the pyramid. The hierarchical nature of a pyramid implies that each level integrates its lower neighbour. This integration is encompassed in the relationship between the languages used at successive levels: if a description language at level L is used as a new command language at level L+1, then the learner is compelled to use explicitly the concepts that have been reified at level L.This is what we called the language shift mechanism (Dillenbourg, 1992): when she receives a new command language, the learner must explicitly use the concepts that were implicit in her behaviour. The meaning of the new commands has been induced at the previous level by associating the learner's behaviour with some representation. This representation is now the new command. The ILE structure can then be described as a sequence of [action language, representation language] pairs, a sequence in which the relationship between two successive pairs is described by the language shift mechanism.

The process by which properties that are implicit at some level of knowledge can be abstracted and explicitly reached at the higher level has been studied under the label of reflected abstraction (Piaget, 1971). The language shift mechanism has two uses. Firstly, it translates this psychological concept in a terminology more relevant for ILE designers. Secondly, it describes a pedagogical strategy (mainly inductive) to trigger reflected abstraction.

There is an obvious mapping between the structure defined by Case and our intermediate framework.The control structures at each level of the pyramid integrate the control structures located at the lower level. The sequence of microworlds within the pyramid is structured as Case's view of development: quantitative variations define the improvement possible within some level (or microworld or stage) while the qualitative variations define the transition between two levels.The concept of stage transition is translated into the language shift mechanism. This transition is necessary when the learner try to solve problems that have too high memory load constraints. After the language shift, the learner has at her disposal new control structures that enable her to solve the problems with a reduced cognitive load. This ILE-oriented re-expression of Case's allow us to ground the design of MEMOLAB in this theory.

MEMOLAB as a rich and complex learning environment

Any psychological or pedagogical theory regards learning from a specific viewpoint. For instance, some of them pay attention to the effect of practice, but neglect the importance of understanding. Some concentrate of discovery, and forget about imitation. However, when one observes some actual learning process, it is clear that learning does not result from a single activity but from the integration of multiple activities, based on multiple sources of information. A learning environment must takes this complexity into account by offering a wide range of learning resources that will accommodate the variety of individual learning styles and learning needs. Therefore MEMOLAB includes several tools:

- The make-material tool
  A difficult aspect in designing an experiment is to select carefully the items that the subjects will have to remember. A list of words has several features that the novice experimenter is not aware of, but affect memory performance: the length of words (number of syllabs), the frequency of words in our everyday language, the phonological or semantic homogeneity of a list of words.
- The hypertext 'Methodology'
  The very idea of a learning environment is that the acquisition of procedural knowledge is requires active problem solving (by contrast to questions-based courseware that focuses on declarative knowledge). MEMOLAB corrects this bias by giving access to declarative knowledge (theories, laws, facts,...). The hypertext structure enables the learner to find information at the level of granularity he is interested by. This hypertext includes the facilities that are now common to most similar applications (trace of read nodes, webview, book view, footnotes,...).
- The hypertext 'Encyclopaedia of Memory'
  Designing an experiment on human memory requires both methodological skills and some knowledge specific to the domain of memory. For instance, learner must know what is an independent variable but they must also know that 'words frequency' is a interesting independent variable. MEMOLAB includes an hypertext that provides theoretical and experimental data on human memory.
- The simulation.
  Research on the use of simulation in educational research has shown that the quality of this simulation is important. The role of a simulation is not simply to produce the realistic results. The simulation process must be carried out in such a way that the learner or some computational agents may understand the results. The simulation built in Memolab produces a commented trace of the simulation process that enable the learner to understand how these results have been computed.

The richness of a learning environment is not only function of the number of tools but of the extent to which those tools are integrated. This integration is possible because of the homogeneity of our representation scheme, the object-oriented approach. Let us illustrate this with the simulation. The results of an experiment are computed by analogy with those of similar experiments. When MEMOLAB explains how the results have been computed,

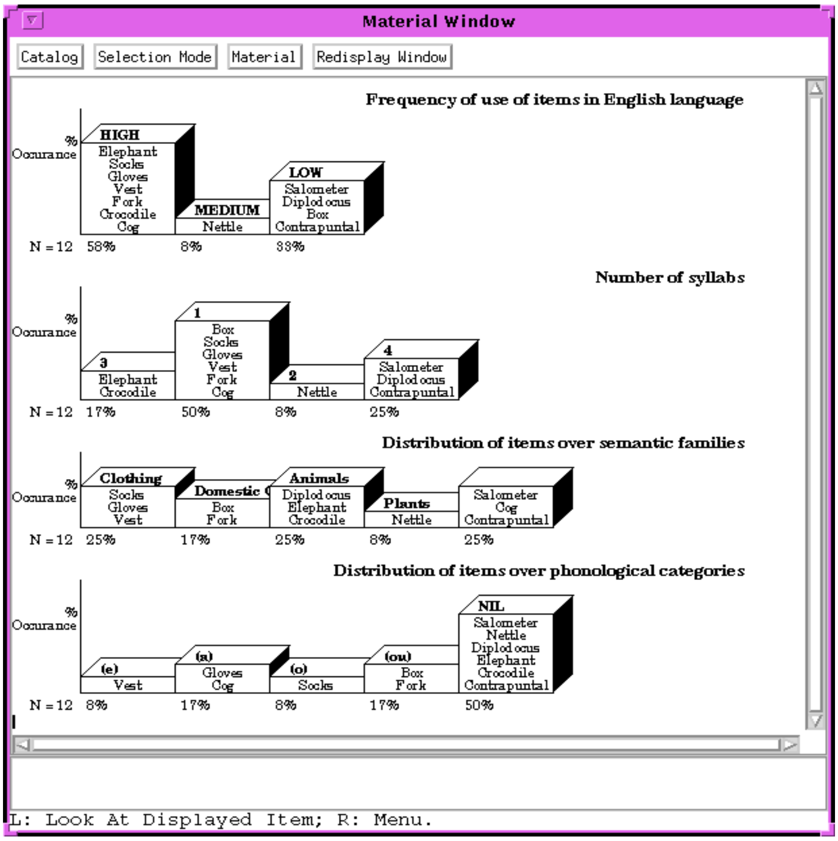

Figure 2: The make-Material tools window.

he refers to these similar experiments ('Simulation Trace'). As illustrated by figure 3, the learner may double-click on one of these references and jump directly in to the hypertext node where the referred experiment is described. The hypertext is also connected with the rule-based agents. Most current research on hypertext system has so far been concerned with the features of the hypertext itself. The originality of our approach is to have considered the place of the hypertext within a complex system, to situate the information search within a context that makes this information useful.

MEMOLAB as a community of agents

Learning does not only result from solving problems or using tools, but also from interacting with agents about these on-going activities. Even if the environment is rich and structured, some learners may need help while they solve the problem, some learners need to be pushed forward otherwise they repeatedly use the same inefficient operators. An agent can of course be human, e.g. a tutor who monitors the learners work or some peer sharing the desk. Regarding computational agents, we discriminate two types agents:

- ***Agents that conduct domain specific interactions*** (i.e. helping to build an experiment). These agents can be a expert of the domain or a computerized collaborative learner (co-learner). One can also envisage to have several experts that can solve the same problem with different viewpoints (Moyse & Elsom-Cook, 1992). The interaction with the human learner is about the domain, it is centered on how to built the solution of a problem.
- ***Agents that select and monitor learning activities*** or help the learner to manage his own learning process. In MEMOLAB, we discriminated two kinds of pedagogical agents, the tutors and the coach who is some kind of chief-tutor. The various tutors have the same function but fulfill this function differently according to their style.

This discrimination between two classes of agents has been essential to separate pedagogical knowledge from domain-specific knowledge and hence to extract from MEMOLAB the

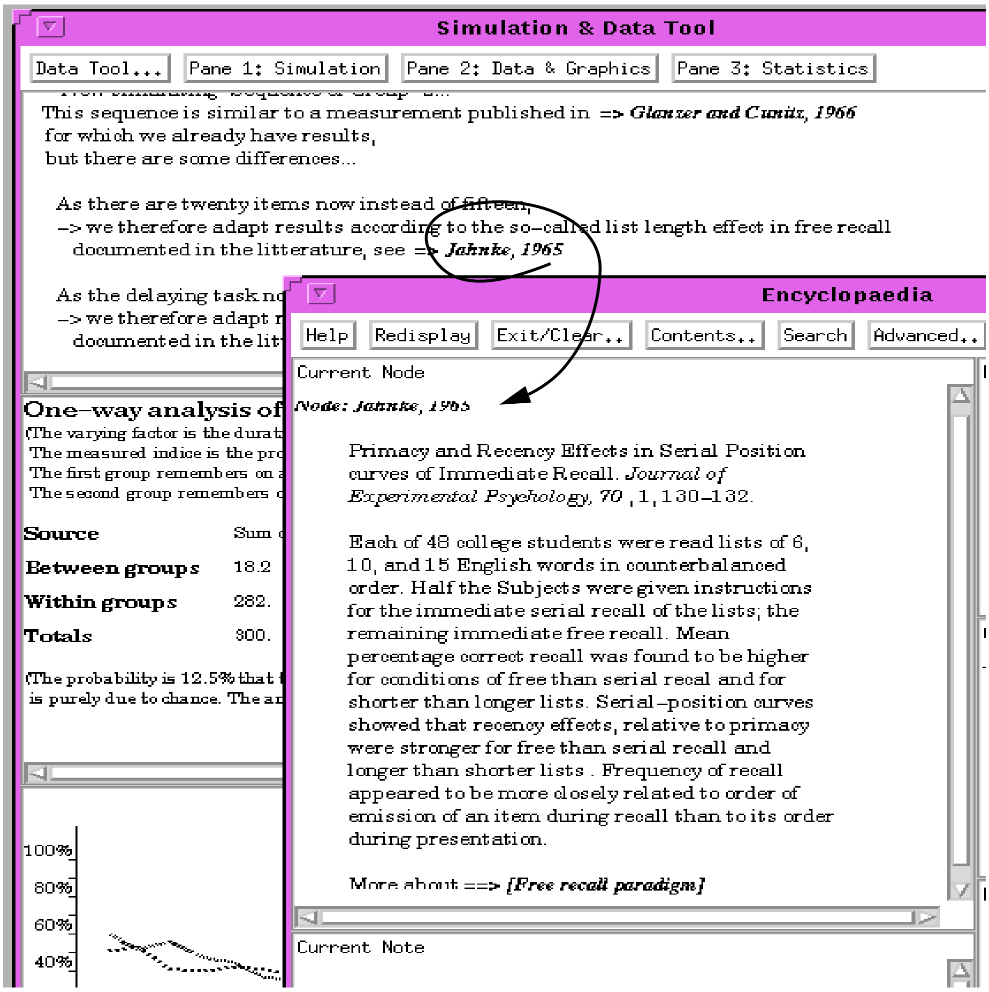

Figure 3: Integration of the hypertext and the simulation: the learner click on an experiment.

components that constitute the toolbox. This architecture will be described in the next section.

Expert-learner interaction

The interaction is tuned by the two slots labelled `local-interaction-mode' and `global-interaction-mode'.

The global-interaction-mode is an integer between 0 and 5 representing the learner's participation in the problem solving process:

- if the global-mode is 0, the learner does nothing, the expert makes indeed a demonstration;
- if the gobal mode is 5, the learner does everything, the expert simply watches him; this corresponds to a simple exercice;
- from 1 to 5, the learner will have an increasing participation in the solution.

The local-interaction-mode describes each single step within the problem solving process. Let us remember that the interaction between the learner and the expert is performed step-by-step.

- An expert's step is defined by a sequence of rules firing until one rule actions includes an interface command.
- A learner's step is defined by a sequence of commands in the problem frame, which ends when the learner uses a command that corresponds to an expert commands.

At each step, the expert may either `propose' an action or `observe' the learner's action:

- if the global mode is 0, the local mode is always `propose';
- if the global mode is 5, the local mode is always `observe';
- when the golbal mode goes from 1 to 5, the number of `propose' steps decreases and the number of `observe' steps increases.

It is the tutor who decides whether the expert's next step will be a `propose' or an `observe'. This decision is the mechanism by which the tutor controls the expert. When the expert receives the control in the mode propose, it executes a certain number of steps without returning the control to the tutor, when the local mode is `observe', the expert compares the command performed by the learner with the commands it would have performed itself

Coaching and tutoring

MEMOLAB aims to combine the flexibility of learning environments with the efficiency expected from advanced courseware. It achieves this goal by two complementary ways, some indirect help inscribed in the system architecture (as we have already seen) and some direct interventions by coaching and tutoring components

This is the traditional way to design educational support: feedback, hint, explanations, problem selection, goal management,... Our challenge was to make this help domain-independent and to combine it with the learner freedom to explore the environment. The answer to the issue of domain-dependency has been tackled creating the class of agents called `experts' and described in the previous section. The other skills have been distributed between six agents, five tutors and one coach. Those agents are described in the following.

The teaching styles

The notion of teaching style refers to the consistency of sets of pedagogical decisions. A teacher or a courseware must take many decisions. We first attempted to determine these decisions by reasoning with a set of independent parameters. We progressively understood these parameters were not independent: there are clusters of values which are more consistent. They determine what we call a teaching style. For instance, if a teacher likes to do inductive learning, he is probably also in favour of discovery learning rather than expository teaching.

We arbitrarily defined five teaching styles. We refer to them by the name of a major contributor to our discipline: Piaget, Papert, Skinner, Bloom and Vygotsky. This reference means that the style has been inspired by those famous people. It would be an exaggeration to claim that each teaching style is an implementation of the theory defended by those people.

The five styles can roughly be classified with respect to they degree of guidance, from Skinner who inspired the CAI courseware tradition, till Papert who launched the microworld wave: Skinner / Bloom / Vygotsky / Piaget / Papert. This classification will be used by the coach (among other criteria) to select a tutor.

In the specific framework of MEMOLAB, the definition of a teaching style escapes from the philosophical discussion to determine the interaction between the learner and the expert and a few other things listed below.

Each tutor:

- decides the initial value and the evolution of the global interaction mode;
- decides the value of the local interaction mode for the next step, given the global interaction mode and the expert's message (which informs the tutor whether the learner did something considered as correct or not);
- determines the learner activity which can be doing an exercise, viewing an example (mode demo) or reading the theory included in the hypertext;
- chooses the problem that the learner and the expert will try to solve jointly;
- processes the learner requests regarding the local interaction mode, the global interaction mode, the problem selection and the level of difficulty;
- determines whether the learner has the right to refuse the expert's suggestion to undo the things he just did;

- determines whether the learner has mastered the goal, and if this is the case, returns the control to the coach;
- determines whether the learner has failed a goal, and if it is the case, returns the control to the coach;
- determines if the actual interaction of the learner still fits with its personal teaching style, and if this is not the case (what we call a `pedagogical drift'), returns the control to the coach.

All these decisions result from a reasoning process which cannot be described by a few parameters but can be modelled by rulebases. These rulebases are partially overlapping. We briefly describe the style of each tutor.

SKINNER or behaviourism

Skinner is an american psychologist well-known for his development of the behaviourist theory. This theory avoids the analysis of mental states and bases its analysis with the association of environmental stimuli with the subject's responses. Some of these associations are biological, other ones can be created through a process called `conditioning'. Learning is the creation of new associations.

Programmed learning, the ancestor of CAI, is based on these principles. The main idea is to decompose complex skills into a sequence of elementary skills to be taught successively. Each behaviour that comes closer this skill must be immediately reinforced by a positive feed-back. Inversely, a negative feed-back must be presented after each incorrect behaviour.

Within the framework of MEMOLAB, Skinner manipulates mainly the difficulty parameter. He does never choose a interaction mode superior to 3. He is also deductive, i.e. when he teaches a new goal, he starts by presenting the theory, before to show any example or exercise. He does not accept any initiative from the learner (he disactivates some menus).

BLOOM or mastery learning

Mastery learning reuses the idea of programmed learning without being so extreme. It keeps the idea that skills should be decomposed into a network of sub-skills. The links in this network describe pre-requisite relations: a sub-skill must be mastered before to try to teach another one. Mastery learning also shares with behaviourism the role given to feedback. The main difference is in the granularity of learning and in the variety of approaches. The granularity of learning refers to the size of learning steps. The best size of a sub-skill is not necessarily the smallest one. Some sub-skills may be taught inductively, some by analogical methods and other ones deductively.

Within the framework of MEMOLAB, Bloom covers the range of interaction modes but prefers to increase the level of difficulty before to change the mode. For each goal, he starts by a rather free exploration, then presents some theory and finally proposes exercises until the learner masters the goal.

VYGOTSKY or apprenticeship

Vygotsky was a russian psychologist who lived at the beginning of the century and who established the basis of the so-called `socio-cultural approach'. His main hypothesis is that we learn by internalising the skills that we first acquired with more able people. The main slogan of today's defenders of this approach is that `participation changes understanding'. This means that when we collaborate with an expert in some domain, we observe how the expert appropriates our own actions into his conceptual framework. We perceive how he uses our contribution within his problem solving strategy and thereby get familiar with this strategy. This strategy is not expressed in an abstract way but receives its meaning from its deep connection to the context. Language is postulated to be the main vector of this internalisation process. The concept of apprenticeship refers to this social process by which a novice carpenter progressively acquires the skills of his master rather than learning theory at school.

Within the framework of MEMOLAB, Vygotsky will select rather difficult problems. At the outset, the expert solves the problem alone (GCM = 1) and then Vygotsky progressively increases the global interaction mode, i.e. the expert will fade out and the learner will pay a more important role in the solution process, until the learner solves the problem alone (GCM = 5).

At the difference with other tutors, the global interaction mode is the favourite parameter of Vygotsky. He will for instance increase the interaction mode rather than increase the level of difficulty of problems. With respect to Papert or Piaget, Vygotsky is more interventionist: he generally allows the expert to interrupt the learner since this intervention belongs to the natural apprenticeship process.

PIAGET or constructivism

Piaget has studied the development of intelligence in children. He claims that children pass through a rather stable sequence of stages. These stages are defined by particular structures through which the learner acts on or reasons about his environment. He noticed that these structures may change when the learner perceives a conflict between his own structure and the environment responses.

This notion of conflict has been reused in the pedagogical field as a tool to help the learner to change his structures. The main postulate is that the teacher won't change the learner's structure directly, the teacher may place the learner in a situation where he will perceive a conflict between what he knows and what he perceives and thereby modify what he knows.

The notion of conflict has been extended by Doise and Mugny to the notion of socio-cognitive conflict, in which one learner perceives a conflict between his vision of the world and his partner's vision. This social form of conflict is present in MEMOLAB. Within the framework of MEMOLAB, Piaget works only with high interaction modes (4 or 5). He leaves the learner explore his own solution and intervenes only when he notices a conflict with the expert.

PAPERT or free discovery

Seymour Papert is well-known as the author of the micro-world LOGO. The key notion is the learner involvement in a micro-world in which he may define his own project. In MEMOLAB, the tutor Papert does not take any decision: the learners select the problem he wants, the expert is maintained asleep and does not provide any feed-back. In theory, this teaching style is only selected with successful learners (but a less successful learner may ask to shift to Papert).

The Coach

The responsibilities of the coach are:
- to select a goal;
- to select a tutor;
- to select an expert.

The goal selection process is fairly trivial:
- first, the coach selects any goal which has no prerequisite (in the curriculum description provided by the author)
- then he selects any goal whose all prerequisite are mastered.

The selection of a tutor takes several factors into account:
- the tutor level of guidance (from 1 for Papert to 5 for Skinner) :if the learner mastered the previous goal with tutor-X, the expert prefers the tutors with a level of guidance inferior to the level of tutor-X. If the learner fails to the goal, a tutor with a higher level of guidance will be selected.
- the tutor goal history: the coach will never select for a goal-X a tutor who has already failed to teach goal-X;
- the tutor history of successful and failed problems: the number of times a tutor conducted a successful activity is stored in a slot, idem for the failures. Between two tutors, the coach will prefer the one who succeeded more often or failed less often.
- if the learner has `drifted' (see explanation below) away from tutor-X, the coach will select a tutor whose level of guidance is inferior or superior to the level of tutor-X, depending on the direction of the drift.

Pedagogical drift

The notion of pedagogical drift is related to the notion of teaching style. A teaching style defines a zone of the space of pedagogical decision space within which the various decisions have a high level of consistency. This implies that, between two styles there is some discontinuity. By definition, the space between two styles suffers from being unclearly defined, inconsistent.

Given the freedom offered to the learner, it may occur that the actual interaction between the learner and the tutor does not look any more as it should look inside the particular teaching style. For instance, if Bloom is active and if the learner set several times the global interaction mode to 5, the actual interaction won't look any more as what Bloom would do. This learner may have left the zone of efficiency of Bloom.

Each tutor has a set of rules to determine if such a drift occurs. All the utilities offered to the learner (selected a problem, change the difficulty level, etc...) create an instance of the class `initiative'. This class possesses two subclasses, `want-more-initiative' and `want-less-initiative' (where we mean `want-more-guidance' and `want-less-guidance'). For instance, if the learner asks to increase the interaction mode or the level of difficulty, or if he asks the expert to continue the problem by himself, we create instances of `want-less-initiative'. If the learner asks for help or reads the theory, we create instance of `want-more-initiative'. These instances are stored in the slot learner-initiatives-history and parsed by the rules which detect the pedagogical drift. If the proportion of `want-more' is twice superior to the proportion of `want-less' (after some initialisation period), the tutor interrupts himself and sends to the coach the message `drifted up' which means `the learner would rather be in a style where he has more freedom'.

Conclusion

This research has produced different types of outcomes namely at the design level and at the implementation level. We list below both the outcomes of this research project and some avenues for continuing research in this domain.

We developed an original solution for the issue of ***separating domain-independent from domain-specific knowledge***. Instead of searching for a domain-independent language that would describe any domain, ***we describe pedagogical knowledge in terms of interactions among agents***. A learning environment is a society of agents for which we specified (1) the role distribution over agents (coach, tutor, experts) and (2) protocols of communication among agents (global and local interaction mode, repair, conflicts,...). Some of the choices made in MEMOLAB are still arbitrary. For instance, the only domain-specific agents we implemented are 'experts' while we could integrate 'peer learners' or semi-experts. Another limitations is that we use the term 'agent' to refers to a simple data-structure, including namely a rulebase that drives his behavior. Current agents do not have the autonomy and goal-orientedness that would characterize a proper multi-agent system (Soham, 1993). We are on the way to redesign MEMOLAB within a pure agent-oriented approach. The specification of well-defined protocols for communication among agents would enable to integrate heterogeneous agents, namely agents which are not based on rulebases or agents that are not written in the same programming language.

By separating pedagogical expertise from domain knowledge, we are not only able to permute the domain knowledge associated to some teaching strategy, but conversely, we are also able to apply various teaching styles to a particular domain expertise. Instead of arguing on which teaching style is the best, we prefer to cover a ***large range of teaching styles***. Of course, this raises the issue of selecting the teaching styles. However, if this issue is non-trivial at the theoretical viewpoint, we can apply at a pragmatic level a simple principle: if the learner fails, change the teaching style! We would like to develop future research on this learner's perception of teaching styles.

A third outcome of our work lies in the design metaphors that have been used for developing MEMOLAB. The pyramid metaphor is in itself very simple, but it enabled us to ***structure our sequence of microworlds on the basis of some theory of cognitive development***. This metaphor is closely connected to the ***language shift principle*** which helps developers to smooth the transition between two microworlds which use different interaction languages.We would like in the near future to collect more empirical data to check the consistency between the theory and its application to MEMOLAB, and to observe the learner adaptation mechanisms when a language shift occurs.

Another achievement concerns the integration of multiple tools within a learning environment, especially the ***integration of an hypertext with rulebase agents***. Nowadays, hypertext systems is a very popular research topic. We believe that the pedagogical power of an hypertext cannot be understood without considering the context that justifies some information search in the hypertext.

A fifth lesson from this work concerns the interaction between the leaner and a computerized agent. The principle we applied is very simple: ***human-computer interaction should concern what is ON the screen***. This design of agents, especially the expert, has been determined by this principle: the rule conditions 'read' the problem state on the screen, the rule conclusions change the problem state display. This principle produces a nice opportunistic expert-learner collaboration and makes this interaction inspectable and tunable by the tutor. The experiments helped us to find to problems related to this principles, namely the difficulty of interacting about goals, but they also gave us some insight for implementing mechanisms of ***social grounding*** between a human user and a machine. Social grounding is a mechanism of joint and reciprocal modelling between two partners, based on the negotiation of shared meanings. A limitation in our current architecture is that the rule variables are instantiated by screen objects without any ambiguity. A new research avenue would be to test interactions where the computer concepts would not be directly connected to the displayed objects (the learner concepts are not either) and that this relation would be negotiated with the learner, through various pointing mechanisms.

References

Bento, C. et Costa, E. (1988). Automatic Cognitive Modelling in Hierarchical Domains. Proceedings of ECAI'88.

Bloch, G. and Farell, R. (1988). Promoting Creativity through Argumentation. Proceedings of ITS-88, Montreal, pp. 243-249.

Bradzil, P. B. (1992). Integration of knowledge in multi-agent environments. In E. Costa (Ed.). *New Directions for Intelligent Tutoring Systems*. Berlin: Springer-Verlag.

Brown, J.S. (1990). Toward a new epistemology for learning. In C. Frasson & G. Gauthier (Eds), *Intelligent tutoring systems at the crossroad of AI & Education.* Norwood, NJ: Ablex.

Case, R. (1985). *Intellectual Development form Birth to Adulthood.* New York: Academic Press.

Clancey, W. (1983). The advantages of abstract control knowledge in Expert system design. Report N° STAN-CS-83-995, Stanford.

Collins, A and Brown, J. S. (1988). The Computer as a Tool for Learning through Reflection. In H. Mandl and A. Lesgold (Eds), *Learning Issues for Intelligent Tutoring Systems* (pp. 1-18). New York: Springer Verlag.

Conklin, J. (1987). Hypertext: An Introduction and Survey. *Computer, 20*, (9), 17-41.

Costa, E. and Urbano, P (1992). Machine learning, explanation-based learning and intelligent tutoring systems. In E. Costa (Ed)*, New Directions for Intelligent Tutoring Systems.* Berlin: Springer-Verlag.

De Corte, E. (1993). Learning Theory and Instructional Science. Unpublished invited paper presented at the Final Planning Workshop of the ESF-Programme, St Gallen, Switzerland.

Dillenbourg, P. (1989). Designing a self-improving tutor: PROTO-TEG. *Instructional Science*, 18, 193-216.

Dillenbourg, P. (1992). The Language Shift: a mechanism for triggering Metacognitive Activities. In P. WINNE and M. JONES (Eds), *Adaptative Learning Environment*. Berlin: Springer-Verlag.

Dillenbourg, P. and Mendelsohn, P. (1992). The genetic Structure of the Interaction Space. In E. Costa (Ed), *New Directions for Intelligent Tutoring Systems.* Berlin: Springer-Verlag.

Dillenbourg, P. and Self, J. A. (in press) Designing human-computer collaborative learning. In C. E. O'Malley (Ed), *Computer-Supported Collaborative Learning.* New York: Wiley & Sons.

Hron, A. (1990) A diagnostic system for qualitative knowledge. Paper presented at the third ITS Seminar in Arhus, Denmark, 5 - 9 December.

Mendelsohn,,P. and Dillenbourg,,P. (1993). Le développement de l'enseignement intelligemment assité par ordinateur. In J.F. Le Ny (Ed), *Intelligence Naturelle et Intelligence Artificielle*. Paris: PUF.

Mengel, S. and Lively, W. (1991). On the Use of Neural Networks in Intelligent Tutoring Systems. *Journal of Artificial Intelligence in Education. 2.* , (2) , 43 - 56.

Moyse, R. and Elsom-Cook, M. (1992). *Knowledge Negociation*. London: Academic Press.

Piaget, J. (1971). *Biology and Knowledge.* Chicago: The University of Chicago Press.

Riesbeck, C. K. and Schank, R. C. (1989). *Inside Case-Based Reasoning.* Hillsdale, New Jersey: Lawrence Erlbaum Publishers.

Self, J. A. (1991). Formal approaches to learner modelling. Technical Report AI-59, Computing Department, University of Lancaster (UK).

Steels, L. (1990). Components of Expertise. *AI Magazine,* Vol. 11, n 2, 28 -49.

Vygotsky, L.S. (1978). *The Development of Higher Psychological Processes* (edited by M.Cole, V. John-Steiner, S.Cribner and E. Souberman). Cambridge, Massachussetts: Harvard University Press.

Wertsch,, J. (1985). Adult-Child Interaction as a Source of Self-Regulation in Children. In R. Yussen (Ed). *The growth of Reflection in Children* (pp. 69-97*).* London: Academic Press.

White, B.Y. and Frederiksen, J.R. (1985). QUEST: Qualitative Understanding of Electrical System Troublshooting, *ACM Sigart Newsletter*, 93, 34-37.