Implementing a Model of Cognitive Development in an Intelligent Learning Environment

Patrick Mendelsohn and Pierre Dillenbourg

Faculté de Psychologie et des Sciences de l'Education, Université de Genève, 9 route de Drize, 1227 Carouge, Switzerland.

Abstract. This paper addresses a series of questions concerning the domain of knowledge acquisition processes in human by comparing the approach explored by designers of learning environments with the one of developmental cognitive psychology. It starts with a brief description of the content and objectives of the project MEMOLAB, followed by an account of the learning theory we chose to implement We then discuss the problem of mapping the psychological concepts with the system's architecture.

Keywords. Learning environments, cognitive development.

1 Introduction

This paper addresses a series of questions concerning the domain of knowledge acquisition processes in human by comparing two approaches. On the one hand the approach explored by designers of learning environments in their attempts to promote learning processes and, on the other hand the field of developmental cognitive psychology. The questions raised by the multiple interactions between these two fields can be summarized in two sets of problems:

- Is it possible to map the basic concepts of developmental psychology like the notions of stages, substages, transition mechanisms, etc, with concrete elements of learning environment components ? What can we learn about this maping in order to design more efficient systems ?
- What are the links between research methods and paradigms in the field of developmental psychology and in the domain of Technology Based Learning ?

The paper starts with a brief description of the content and objectives of the project MEMOLAB, followed by an account of the learning theory we chose to implement (Case 1985). We then discuss the problem of putting the two together via an intermediary model the role of which is to map the psychological concepts with the system's architecture. The intermediate model to bridge this gap is formulated using the visual metaphor of a pyramid and the computational device of "language shift".

Our approach allows two complementary research strategies. On the one hand, we hope to contribute to educational computing an interesting 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.

A computer based learning environment may reasonably be described as "intelligent" if its architecture and interface are designed according to the principles of some theory of human learning. Under this definition, a programmed learning machine is "intelligent" with respect to behaviorist learning theory, LOGO is "intelligent" with respect to Piaget's operational theory and the same applies to Anderson's LISP tutor with respect to his own ACT theory (Anderson xxxx). The definition given thus entails a view of the engineering of "intelligent learning environments" (ILE's) as the implementation in software of a learning theory. Two major sources of difficulty are immediately evident to the designer commited to this approach. Firstly, learning theories or models are rarely developed to a point where they can be more or less directly operationalized. Secondly, there is no natural correspondence between the software engineering tools available to us and the language of psychology in which we formulate our theories or models. As researchers, what should we do? Wait for the psychologists to develop their theories to the point where they are implementable using existing tools? Or wait for the programming language designers to develop some new formalism that can capture the informal, verbal richness of psychological theory? Obviously not! What we should do is adopt 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 over the last years with the MEMOLAB project at TECFA: in the rest of this paper we shall show how our practical work on the system design reflects this methodological commitment.

In the space available, we shall not attempt a detailed exposition of the architecture and functionality of MEMOLAB (see Dillenbourg and al., 1992). Instead, we begin with a brief description of the content and objectives of the project, followed by an account of the learning theory we chose to implement (Case 1985). Having established the system requirements and chosen a learning theory, it is then possible to discuss the problem of putting the two together via an intermediate model. The intermediate model we propose to use to bridge the gap is formulated using the visual metaphor of a pyramid and the computational device of a "language shift".

2 MEMOLAB: Objectives and System Requirements

The goal of MEMOLAB is to provide an Intelligent Learning Environment for students studying experimental psychology. The system provides the tools for designing, simulating, performing and analysing experiments on human memory. A user of the system is expected to work through the following cycle of stages for each study project:

(a) Research the literature (in an online hypertext "Encyclopedia of Memory") and formulate an hypothesis to test.

(b) Design an experiment to test the hypothesis. To do this they must select one or more experimental groups; define the experimental treatment procedure and target material ; establish a performance measure and select a statistical model.

> For example: Two groups of 20 subjects each. Presentation of a list of 10 words for five minutes with subsequent recall. Group 1 is given a list of 10 semantically related words. Group 2 is given a list of 10 words that are not semantically related. The performance measure is the number of words correctly recalled. The statistical model is random assigment to groups, assumed normal distribution of scores within groups and the t-Test is used to establish a significant difference between the group mean scores.

(c) Run the experiment. This can be done by using a simulation package included in MEMOLAB.

(d) Run the statistical analysis on the data collected and draw the appropriate conclusion.

Bulding an experiment in MEMOLAB thus involves assembling on the "Workbench" a temporal sequence of discrete events. Each event has three components: a group of subjects, the target material and the task procedure. To construct an experimental design is formally equivalent to abstract an "empty sequence" from the "Workbench" and to give it content by generating an *n*-dimensional factor table, where *n* is the number of independent factors manipulated in the experiment. It is clear that the complexity of skills and knowledge implied in such a task cannot be acquired at all in a single session (see also Beauvois et al. 1990). The integration of progressively more complex facts and procedures must consequently be organized into a logical and temporal hierarchy, so that the learner can master sub-components shielded from the full complexity of the task. This pedagogic decomposition may be perfomed on an ad hoc analysis (perhaps exploiting some structure implicit in the specific domain), or it may be in accordance with a general cognitive model of skill integration. We have chosen the latter approach, with Case's neo-piagetian theory serving as the model.

3 Case's Theory of Hierarchical Integration

The key idea in Case's theory of intellectual activity and development is what he calls the "executive control structure". He believes that there is a valid general analysis of problem solving across domains viewed as the execution of a "mental plan". This plan is defined as a sequence, or program of schema. There are two types of schema: "figurative schema" represent states and "operative schema" represent transformations. The mental plan is divided into three main sub-components.

- A representation of the "problem situation": this is the set of conditions relevant to the plan. The complexity of the representation will depend directly on the complexity of the problem.
- The goals to be achieved defined as a set of new states, or "solution situation".
- The "problem solving process" to be used, stated as a set of operations that transform the problem situation into the solution situation.

These components are further analysed. Elements of the problem situation are mapped to elements in the solution situation, and both are mapped to transformations in the strategy set. The result is a well-defined formal structure that allows Case to associate specific task with problem solving processes in a rigorous way .

Case formulates his general theory with reference to developmental stages as they are identified in specific domains. One of the characteristics of his theory is that it relates quantitative changes within a stage to qualitative changes between stages: for example, an increase in the active unit capacity of working memory occurs within a stage, but helps to explain the transition to the next stage. Case distinguishes activity within a stage (i.e. a "sub-stage") by first defining what he calls "basic units of thought".

How do we explain the formation of new units and the transition between stages? According to Case, each new sub-stage within a stage is characterised by the subordination of a new basic unit to the executive control structure: the first sub-stage has two basic units, the second has three and the third has four. The complexity of subordination reached at the final sub-stage (stage n) is such that it corresponds to a basic unit at the next stage (stage n+1). When the executive control structure of stage n+1 subordinates two of these basic units passed up from below, it will enter its own first sub-stage and so on. The last sub-stage of stage n can thus be considered as sub-stage zero of stage n+1. In other words, the four-unit control structure of stage n can be translated into a one-unit control structure at stage n+1. It is this formal process which Case calls "hierarchical integration".

It is an increase in "Short Term Storage Space" (STSS) that permits the transition from one sub-stage to the next. This increase is achieved within the "Total Processing Space" (TPS) which also contains the "Operating Space" (OS) utilized to control the active schema. STSS increases with age during development as a result of the maturation of the nervous system. It also increases during the learning of schemata as the result of an increase in the efficiency of the control structure: as the learner masters a task they free up short term memory to hold new objectives. The hierarchical integration of two control structures is thus dependent on the operational efficiency of their component schemata. This efficiency depends directly on the learner's rate of activity. This activity has to pass through several stages during learning:

- Activation: an existing set of control structures must be activated within a new action sequence.
- Evaluation: the consequences and/or functional utility of the sequence must be evaluated.
- Retagging: the pertinent and salient points in the action sequence must be re-encoded so that they may be intentionally activated in the future.
- Consolidation: the re-encoded element must be practiced and consolidated as a new processing unit.

4 The Pyramid Metaphor and the Language Shift Mechanism

In Case's theory, the learner is primarily a "problem solver" (in contrast to the "tool maker" in the theories of Bruner and Vygotsky, or the "scientist" implicit in Piaget) and the didactic approach of an intelligent learning environment based on his theory will require at least the following:

- The explicit specification of the goals of the instructional unit.
- A fine-grained analysis of the control structures achieved by experts in the domain.
- An epistemology for these structures.
- Learning tasks that imply the target structures and instructional goals.
- Integration of the tasks into a wider learning environment.

These requirements together constitute a kind of methodology for the design of an intelligent learning environment: so in basing MEMOLAB on Case's theory we will be testing not only a model of cognitive skill acquisition but also a set of design principles for educational software.

The pyramid is a metaphor that we use for the architecture of MEMOLAB (see Figure 1): the concepts and know-how that the learner must master are ordered vertically according to their level of abstraction and the type of object that the learner controls. Learning within this method involves moving up the pyramid through the various levels. The role of the tutoring algorithm or program is to facilitate the learner's progress up the pyramid by taking charge (temporarily) of the higher level objectives. Each level in the pyramid is defined by a command language, a description language and a class of problems. The vocabulary of the command language represents the set of elementary actions that the learner can control at a given level. The syntax of the command language represents the ability of the learner to construct well-formed sequences of actions (in Case's model these sequences correspond to relations between schemata with a control structure). The description language comprises the set of symbols and representations used by the tutoring component to "reflect" or "mirror" the significant actions of the learner. The tutoring component can use this description to provide the learner with a continuously updated model or picture of their progress through the task, and the model may be used by the learner to assimilate control structures at a higher level than would otherwise be possible.

The command language and the description language are different at each level of the pyramid, since each level is constructed by combining elements from its predecessors. In particular, the description language at level n may become the command language at level n + 1: so the learner has to explicitly manipulate the concepts that were only available passively at the lower level. We call this mechanism the "Language Shift" (see Dillenbourg, 1992): it may be considered as an analogy of Case's process of "hierarchical integration". Unlike Case's theory, however, our model can be translated pretty directly into software structures. The command language corresponds to "compiled schemata" in MEMOLAB, while the description language refers to the effect of these schemata on the control structures.

The MEMOLAB pyramid has four levels reflecting the increasing complexity of the experimental designs required of the learner. Other necessary knowledge, such as statistics or the content of the experiments, has a special place in our model. Its acquisition is subordinated to the construction of plans in the same way that Case assumes any synthesis of two control structures is subordinated to one of them. Each level of the pyramid is consequenty defined by four components:

- A class of problems whose complexity is at the same level as the target class of experimental design: these we call the "challenge".
- Objects that can be manipulated and combined by the learner to construct the solution to a particular problem: these objects and their rules of combination form the command language.

- A description language, used by the tutoring program to "reflect" the learner's behaviour, principally in terms of the structure of the experimental design.
- An evolving body of concepts and know-how required by the learner as they progress from simple to complex designs.

Following this model it is now possible to describe the four levels of the MEMOLAB pyramid.

LEVEL 0. Pre-requisite: using the interface.

The objective here is to learn how to construct an event and activate it by positioning it on the workbench. Each event is composed of four elements (the subjects, the target material, the task and the record associated with the task). The construction of this object/event is assumed to be mastered before the beginning of the main instructional cycle at Level 1. The learner has to understand that interesting events are those which have a strong internal coherence between their components: for example, the data recorded must be relevant to the task, the target material and the type of subjects chosen. At Level 0, the learner should construct categories of event corresponding to key stages in the experimental procedure: encoding, maintenance, recall etc. The secondary control structure subordinated to this skill concerns the parameters associated with the various components of the event. The learner must acquire a clear understanding of the relevant values that may be assigned to these experimental variables: number of subjects, length of the target lists, attributes of the list items etc.

LEVEL 1. From Events to Sequences: the identification of sources of variation.

The goal for the learner at this level is to acquire knowledge and understanding of the experimental paradigms used in the study of human memory. The building blocks under the control of the learner are the events developed at Level 0, and the task is to assemble them into sequences that specify complete experimental paradigms. Sets of sequences that belong to the same basic type of experiment constitute a paradigm in MEMOLAB: for example, the "free-recall" paradigm involves four events; encoding, maintenance and recall without ordering. The tutoring component in the system will provide the learner with a representation of the process by which events are combined into paradigms. The primary control structure at this level is the paradigm, while the subordinate secondary control structure is provided by the elementary tools of descriptive statistics (frequency, mean and variance).

LEVEL 2. Comparing Groups on a Single Factor

The goal for the learner at this level is to extend the use of experimental paradigms to the comparison of groups of subjects on a single factor. The building blocks under the control of the learner are the paradigms (i.e., events combined into sequences) constructed at Level 1: these may be called up and positioned on the workbench. The learner then creates an actual experiment by changing the value of a parameter in all the events that compose the sequence. The tutoring component of MEMOLAB provides a representation (i.e., the description language) that illuminates the comparative aspect of an experiment in which a single factor may take on different values. The statistical tools required at this level are those used for comparing means (e.g., Student's t-Test).

LEVEL 3. Interaction of Factors

The goal for the learner at this level is to understand how factors may interact in an experiment and how that interaction is handled in the analysis of results. The building blocks under the control of the learner are the single factor comparisons mastered at Level 2. The command language allows the combination of these single factor comparisons into a multi-factor design. The statistical tools required at this level include those used for the analysis of variance (e.g., Snedecor's F).

LEVEL 4. N-Factor Designs

Here the learner reaches the highest level of experimental complexity supported by MEMOLAB: they can create three factor designs with seven sources of variance. We do not expect to be able to simulate the full complexity of such designs, but MEMOLAB could supply the formal notation currently used in the scientific literature. We may introduce the distinction between paired and independent factors.

In the transition from one level of the pyramid to the next, the description language of the lower level becomes the command language of the higher level. For example, at Level 1 the concept of an "event sequence" is used in a graphical description language for the learner's actions; at Level 2 the same notion and its graphical description become part of the learner's command language. In this way we oblige the learner to employ the concept in planning their new experiments. A number of workers have emphasized the importance of graphical "reifications" in supporting the learner's reflection. Our goal is to go further and use the dynamics of the interaction to constrain the user to adopt the reflective reification of one level as the active problem solivng language of the next level up the pyramid.

As we have already noted, the "Language Shift Mechanism" is repeated between each level of the pyramid. Within each level, on the other hand, the creation of progressively more complex experiments leads to a severe cognitive load imposed on the learner. As Case notes, the transition to the higher level is made necessary by the limited capacity of our working memory. After the language shift, the learner has available a new set of commands that have "automated" procedures (Case's control structures) which they previously had to explicitly execute in working memory: this frees up space for work on the higher level procedures. Our aim in MEMOLAB is to provide a an environment that supports and encourages this process.

5 Conclusion

The pyramid in MEMOLAB is no more than a metaphor, but such metaphors are often powerful heuristic tools in multidisciplinary research. In this paper we have sought to show how we are using this and other design ideas to build interfaces between diverse theoretical languages. Our approach allows two complementary research strategies. On the one hand, we hope to contribute to educational computing an interesting 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 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. Research in educational computing has often isolated acquisition processes and underestimated the importance of their interaction: one of the goals of MEMOLAB is to redress the balance a little.

References

Anderson, J.R. (19xx)

Beauvois, J.L., Roulin, J.L. et Tiberghien G. (1990). La méthode expérimentale en psychologie. Paris: PUF

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

Dillenbourg P. (1991) . The language shift: A mechanism for triggering metacognitive activities. in P. Winne and M.Jones. Foundations and Frontiers of Instructional Computing Systems. Hamburg: Springer-Verlag.