

A Genetic Structure for the Interaction Space

Pierre DILLENBOURG and Patrick MENDELSON

TECFA

Faculté de Psychologie et des Sciences de l'Education
Université de Genève (Switzerland)

Abstract

An intelligent learning environment (ILE) can be viewed as a space of interactions between rational agents. The task of the ILE designer is to analyse the structure of this interaction space. In this chapter, we propose to structure it as a sequence of hierarchically integrated sub-spaces. Each interaction sub-space associates a command language with a description language according to a simple chaining principle: the description language of each sub-space corresponds to the command language of the next sub-space. This structure instantiates the neo-piagetian theory of R.Case, which takes into account both the qualitative changes that occur between two stages (or sub-spaces) with the quantitative change that happen within a stage. The quantitative changes result from the attempt to solve increasingly complex problems and lead to the saturation of the learner's working memory. It is then necessary to move to a new sub-space that provides the learner with operators for solving similar problems with a reduced cognitive load. The ability to use these new operators is based on the reflective activities triggered by the description language.

Keywords : Learning Environment, Model of Development, Reflection, Theory Implementation.

DILLENBOURG P. & MENDELSON P. (1992) The genetic structure of the interaction space. In E.Costa (Ed), *New Directions for Intelligent Tutoring Systems*. (pp. 15-27) Hamburg: Springer-Verlag.

Introduction : The Interaction Space

Designing an intelligent learning environment (ILE) consists in defining a set of interactions that may possibly happen between rational agents : the learner, the computer and any other participant. In this chapter, we call this set the *interaction space*. Considering an ILE as an interaction space fits with the current trend to pay less attention to the internal representations (the learner's and the computer's) and more attention to how agents actually interact together (see Clancey's contribution to this book).

The interaction space can be described in two ways : either as space of interactions, or as the interaction between two spaces, the representation space and the action space. We explore the later approach because it helps the designer to define the structure of the interaction space. The *representation space* is the set of displays (text and graphics) built by the computer. It is important to notice here that the term "representation" does not refer to the computer's internal representation nor to the learner's mental representation. It designates instead screen displays such as a ball hitting another ball or the trace of some problem solving path. The *action space* is the set of activities by which the learner can do something with these representations : can they only contemplate the representation or can they transform it, and, if so, in what way ?

The total interaction space is a set of [Action Space, Representation Space] pairs. Each of these pairs defines a microworld. In order to make learning efficient, the designer has to structure this set of microworlds. We chose the simplest structure : a linear sequence of microworlds. The educational value of this sequence is based on four properties :

- successive microworlds allow the learner to solve *increasingly complex problems* with a *constant cognitive load* (that only varies within a microworld);
- the sequence must reflect some *theory* of knowledge acquisition, in our case, the neo-piagetian theory of *development* developed by R. Case (1985);
- the learner's move from one microworld to another results from their *metacognitive* activities ;
- the sequence is *transparent* , moving from one microworld to the next one is not hidden in the system but clearly visible at the interface.

This sequence of microworlds defines the core structure of an ILE. To be effective, it must be integrated with a rich learning environment, but here we will only analyze the core structure.

This chapter includes five sections. The first one describes the dimensions of interaction space. The theory of Robbie Case is briefly presented in the second section. In the third section, we present a framework that allows the designer to build a sequence of microworlds consistent with Case's theory. This framework includes two main concepts : the *pyramid metaphor* describes the *structure* of the microworld sequence and the *language shift mechanism* describes the *dynamics* of the sequence. In the fourth section, we outline how this sequence has been applied to the design of an ILE for acquiring basic skills in experimental psychology (MEMOLAB). The last section relates our intermediate framework to other theoretical positions.

1. The Interaction Space

Our study of the interaction space focuses on the dimensions of the representation space.

1.1. The Representation Space

The use of multiple representations in ILEs has been advocated by several designers. It is frequently seen as a solution to solve the incompatibility between the need for compiled and for articulated domain knowledge (Wenger, 1987; Roschelle, 1988). A psychological justification has also been advanced to support the use of multiple representations in student modelling (Self, 1988). However, these statements concern internal representations (both the computer's and learner's), while this chapter is more concerned with external representations, i.e. the objects displayed on the screen. Our concern follows the work done by researchers such as Leblanc (1988), McArthur et al (1988), Frederiksen and White (1988) to provide multiple ways to represent a problem on the interface.

We view a representation space as the result of multiplying the designer's original representation by some **generic factors**. These factors define significant differences between representations. They determine the dimensions of the representation space.

The first generic factor was isolated by Roschelle (1990) in his work on the Envisioning Machine (EM). Roschelle successively designed several representations for the same set of physical phenomena (particle movements). The first EM design focussed on the *epistemic fidelity* of the representation, i.e. the degree of consistency between the physical representation of some phenomena and the expert's mental representation of this phenomena (Wenger, 1987). However, mapping physical and mental representations is an inherently ambiguous interpretation process. Roschelle reports that the users of the early EM did not read representations as experts did. As Suchman (1987) pointed out, attempts to build representations that avoid this inevitable ambiguity are vain. Therefore, representations must not be thought of as tools for eliminating uncertainty but as tools for managing uncertainty through communication. Later designs of EM focused on the ability of representations to support activities and dialogues that will lead to some shared understanding of the representation. Roschelle called this property *symbolic mediation*. However, epistemic fidelity remains an important issue : "*it is hard to imagine that the EM would be a better learning environment if its original design had not been based on high fidelity concerns*" (Roschelle, 1990,p29). The epistemic fidelity of some display and its ability to support clarifying dialogue are not exclusive properties. The goal is indeed to have both of them. Their partial incompatibility makes difficult to imagine a single representation that have both properties. But it can be by-passed if we think in terms of multiple representations, i.e. in terms of a representation space.

The second generic factor considers *what* is represented on the screen : *the objects to be learned or the learning subject herself*. In many microworlds, the screen displays some object that the user manipulates : a bowl hitting another bowl (e.g. DIBI, Stumpf et al., 1988), a light connected to an electrical circuit (e.g. Palies, 1988), a line perpendicular to another line (e.g. in Cabri-Géomètre, Balacheff,1989), ... However, there is a growing interest in the possibility of representing the subject's solution process (e.g. how they solve an equation) in order to trigger reflective activities (Brown,1985), as illustrated by the Geometry Tutor (Anderson et al., 1985) or Algebraland (Collins and Brown, 1988). Is the designer compelled to choose between representing the object or the subject ? Is she compelled to choose between the benefits of direct manipulation or the importance of metacognition. In some cases, a single representation can served both purposes, for instances, in TAPS II (Derry, 1990), where the structure of the representation of the given word problem determines indeed the solution process itself. An alternative consists in allocating the object and the subject to different aera of the representation space.

As third generic factor, we consider the role played by representations in the process of knowledge *transfer*, a process that has not receive in ITS research the attention it

deserves. The ability to transfer knowledge from a source-problem to a target-problem implies two types of knowledge. One needs some compiled knowledge that has proved to be efficient with the source problem. One also needs some more abstracted knowledge necessary to identify the target-problem as belonging to the same class as the source-problem and to adapt the source-solution to the target-problem. Research on second generation expert systems has emphasized the complementarity of these two types of knowledge (Steels, 1990). Our point is that representations have to served two functions : to gather the context information necessary to solve a particular problem (off-loading the subject's working memory), and to detach the problem from the context-dependent features that make abstraction difficult. Once again, these conflicting requirements justify the necessity of designing a set of representations.

We cannot exhaustively enumerate the factors that generate the full representation space. Moreover, these factors are not independent. For instance, Ann Brown (1987) reported experiments that establish a relationship between reflection (second factor) and transfer (third factor). Similarly, supporting reflection can be viewed as supporting the learner's dialogue (first factor) with themselves. In front of this complexity, the designer's challenge is to structure the representation space. The purpose of this chapter is precisely to propose a structure and a mechanism that allow the learner to move within the representation space.

1.2. Interactions between the Representation and the Action Spaces

The action space is the set of actions that the user is allowed to perform. The action space is partitioned into sub-spaces. An *action sub-space* is the set of actions that the learner is allowed to do at a particular stage of the learning process. For instance, the beginner may be allowed to enter their data as in a spreadsheet, while more advanced user may be allowed to specify quantities by moving the column tops in a histogram. All the actions necessary for entering data into a spreadsheet form an action sub-space, while the mouse actions on the histogram form another sub-space. Each sub-space is defined -as it is often done in research on human-computer interaction - by a *command language*. The vocabulary of the command language represents the set of elementary actions that the learner can control. Syntactical rules determine how elementary actions can be regrouped into sequences and form complex behaviours.

Similarly, the representation space is partitioned into *representation sub-spaces*. Each representation sub-space is defined by some *description language*. For instance, you can represent a data set by a graph or an histogram. The rules for drawing a histogram constitute a description language. In the histogram sub-space, you will have many representations, one for each data set, each of them using the same description language. These analogous definitions of the action space and the representation space will allow us to specify the interaction between the two spaces in order to structure the interaction space.

We have so far presented the interaction space as a unstructured set of action-representation pairs. In order to order this set, we adapted the structure elaborated by R. Case as a description of cognitive development from birth to adulthood. We will describe Case's theory and then explain how it fits with the design of an ILE.

2. Robbie Case's theory of cognitive development

The key idea in Case's theory of intellectual activity and development (Case, 1985) is what he calls the *executive control structure*. Problem solving across domains is 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 schemata* represent states and *operative schemata* represent transformations. The mental plan is divided into three main sub-components :

- (a) 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 number of features a subject has to take simultaneously into account to solve the problem.
- (b) The goals to be achieved defined as a set of new states, or "solution situation".
 - (c) 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 tasks 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 activities within a stage, i.e. *sub-stages*, by defining basic units of thought. 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 (in stage n) leads to the construction of a new basic unit at the next stage (stage $n+1$). The four-unit control structure of stage n is translated into a one-unit control structure at stage $n+1$. The last sub-stage of stage n can thus be considered as sub-stage zero of stage $n+1$. It is this formal process which Case calls "hierarchical integration".

An increase in "Short Term Storage Space" (STSS) permits the transition from one sub-stage to the next. 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 structures: 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.

3. An Intermediate Framework

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, traditional CAI is "intelligent" with respect to behaviorist learning theory, LOGO is "intelligent" with respect to Piaget's constructivist theory ... and the same applies to Anderson's LISP tutor with respect to his own ACT* theory. Engineering an ILE means implementing a learning theory. However, learning theories are rarely sufficiently operational. Moreover, there is no correspondence between the software engineering tools available to us and the language in which psychologists 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 extraordinary new formalism that can capture the informal, verbal richness of psychological theory? Obviously not! Instead, we propose to build intermediary models that act as a bridge between the learning theories and the existing implementation languages. We will now present the intermediary framework we have built to translate Case's theory into system specifications.

We use the *pyramid metaphor* (Dillenbourg, to appear) to describe the core structure of the interaction space. The pyramid represents the concepts and procedures to be acquired by the learner, ranked bottom-up according to their level of abstraction and the type of object that the learner controls. Learning consists in moving up in the pyramid. Each level of the pyramid can be mapped to theoretical and engineering aspects :

- Each level corresponds to Case's concept of stage, i.e. the learner uses qualitatively different control structures.
- Each level corresponds to a microworld, i.e. it associates an action sub-space and a representation sub-space. This association is relevant if the description language (representation space) reifies some abstract features of the user's behaviour (action space), for instance by drawing a graphic picture of their problem solving strategy (Brown, 1985).

The pyramid metaphor includes an idea of hierarchy, i.e. that, in some interaction space, each level integrates its subordinated neighbour. At each level of the pyramid, the action sub-space uses a different command language. Similarly, each description sub-space uses a description language specific to its level. The integration between levels is encompassed in the relationship between the language used at successive levels : if a description language at some level L is used as a new command language at level L+1, then the concepts reified at level L must be explicitly used by the learner at level L+1. This is what we called the *language shift* mechanism (Dillenbourg, in press) : when one offers as a new command language the previous description language, one compels the users to use in their solution process the concepts introduced in the description language. The structure of interaction space can then be described as a sequence of pairs [action sub-space, representation sub-space], a sequence in which the relationship between two successive pairs is described by the language shift mechanism.

Let's consider a simple example from algebra. At some level of the pyramid, one can show the learner with graphics that a good heuristic is to collect the X's on the same side of the equation. At the higher level, we can offer a new "regroup X" button in order to compel them to explicitly use this heuristic in their solution. Some more complete example will be presented later.

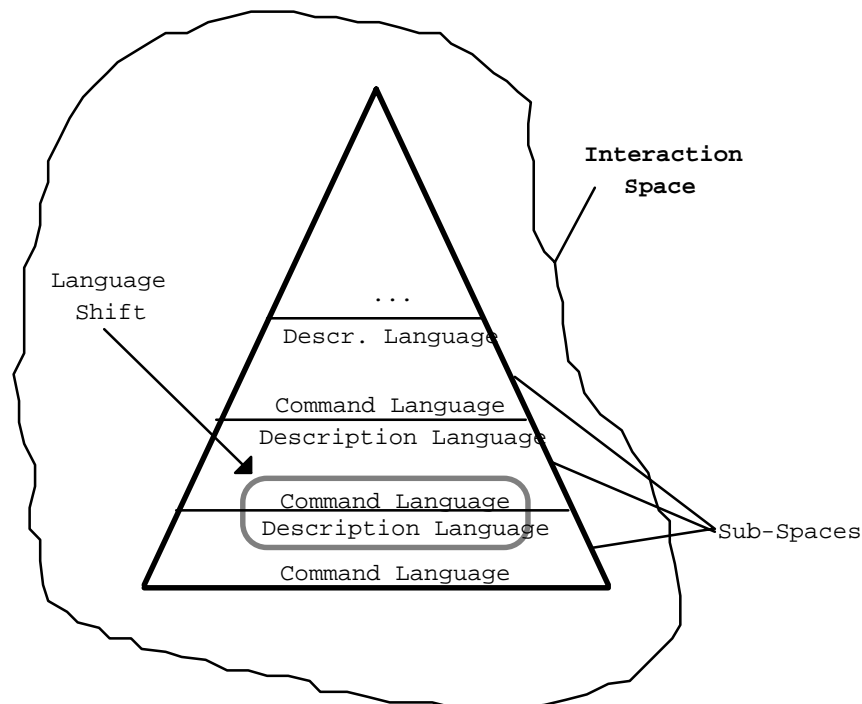


Figure 1 : The Structure of the Interaction Space

4. The Design of MEMOLAB

The goal of MEMOLAB is to provide an Intelligent Learning Environment for acquiring basic skills in 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:

- (a) Research the literature (in an on-line hypertext "Encyclopedia of Memory") and formulate an hypothesis to test, or select a challenge proposed by the system.
- (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 assignment to groups, assumed normal distribution of scores within groups and the t-Test to establish a significant difference between the group mean scores.

- (c) Conduct the experiment by using a simulation package included in MEMOLAB.
- (d) Run the statistical analysis on the data collected and draw the appropriate conclusion.

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 (human memory theory), 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 has its own command language, which means that the learner receives new tools for building experiments.

- | | |
|---------|--|
| Level 1 | Building an experiment involves assembling into a "workbench" a temporal sequence of discrete events. Each event has four components: a group of subjects, the target material, the task procedure and a performance measure. |
| Level 2 | Building an experiment means making a first sequence of events. Then, for each experimental group, this sequence is duplicated and edited. The changes between sequences correspond to the concept of independent variable. |
| Level 3 | An experiment is built in two steps. First, one creates a sequence with variable slots (e.g. group-X, word-list-Y). Then, one defines a table (a plan) that instantiate the sequence (e.g. that specifies the word-list-Y for each group-X). |
| Level 4 | As in level 3, experiments are described by a sequence and a plan, but this plan is two-dimensional and therefore allows for the treatment of complex interaction of effects. |

As presented in the intermediate framework, the concepts used in the next command language are first introduced as description language. Let's look for instance at the transition between the first and second levels. At level one, the learner puts on the workbench a set of independent events. The relationship between these events is implicit, it has to be reified in the description language to be used explicitly at the command language of second level. Therefore, we redisplay events as components of a sequence. The slots that have the same value in the events sequence (for instance those that concern the same group or the same list of words) are removed from the event frame and attached to the sequence frame. Then, after the language shift, the

learners receive the new building block called "sequence". They can build a sequence, duplicate this sequence and obtain a new one by simply editing the sequence-specific slots.

The language shift reflects the qualitative difference that exists between Case's stages (i.e. our levels). The nice point in Case's theory is that it articulates qualitative and quantitative changes. Within each level of MEMOLAB, we define different levels of problems that match Case's concept of sub-stage. These problems appear in MEMOLAB as challenges proposed to the learner. At the outset of the level one, the system proposes challenges that can easily be met with the learner's available operators. For instance, a challenge is "*Prove it : People cannot remember 200 hundreds words after 2 minutes of reading*". Progressively, more complex challenges will be proposed. These increase the cognitive load of the experiment design. In the last sub-stage of level one, learners will have to solve challenges such as "*Prove it : Long words are easier to remember than short words*". This multiplies the number of events and relations between events that the learner must consider simultaneously. This complexity justifies the introduction of the level-2 command language. The new available operators (sequences) will allow the learner to solve similar problems with a reduced cognitive load and hence to tackle later more complex challenges (until the next language shift is necessary, and so forth).

5. The Complexity of learning.

Our framework can be linked to different theoretical perspectives. Gathering the support of multiple theoretical frameworks is very important because *an ILE must address learning in its full complexity*. For instance, an ILE must account for the importance of discovery, for the role of practice and for the effect of coaching, because all of them occur at some stage of any learning in the real world. We have concentrated so far on the relationship between our framework and Case's theory because it has been our main source of inspiration. However, we now briefly describe how the same framework can be used as an "object to think about" other approaches, the apprenticeship stream and the situated knowledge trend.

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, the tutor must take decisions that the learner is not able to take, i.e. it must assume higher floors of the pyramid. At each language shift, the learner will assume more control of his solution process and the coach's guidance will be reduced. The idea of "next level up" corresponds to the concept of *zone of proximal development* (Vigostky, 1978). 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 language referring to objects according to their role in the problem solving strategy.

From Campbell's 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 actions and the computer's description of this sequence. We showed that the meaning of the description language can result from inducing some relationship between the actions performed and their representation (Dillenbourg, in press). The interest of this approach is that it create a bridge between our model and current research on situated learning, where interaction patterns play a central role in learning (Brown, 1990).

Finally, the idea of automating complex control structure into new atomic operators seems very close to Anderson's (1983) idea of knowledge compilation. We must point out that the philosophy behind the language is quite different. Of course, some automatization of subskills is required in MEMOLAB to free the cognitive resources necessary to the language shift. However, the function of the description language is make explicit concepts that were implicitly used before. For instance, the creation of a difference between groups is reified at level one in order to be used explicitly for

designing level 2 experiments. This focus on awareness is rather opposite to the idea of compilation and much closer to the Piagetian concept of "*reflected abstraction*" (Piaget, 1971).

6. Conclusions

The behaviourist approach provides the designer with a methodology for content analysis, namely decomposition into very small pieces of knowledge. The constructivist approach has suggested a radically different approach : the learner is expected to explore complex problems. Unfortunately, unstructured learning environments often fail to cause learning. This chapter proposes an intermediate grain size for analysing the content to be taught. A learning environment could be viewed as a sequence of hierarchically integrated microworlds. This sequence of microworlds partially covers the representation space. The first microworld allows a novice to solve very simple problems. In successive microworlds, the learner is pushed to tackle more and more complex tasks, with more and more powerful operators. The use of these operators implies the internalization of the major concepts that constitute the expertise in the domain.

Each microworld is defined by the association of a description language and a command language. This association also specifies the relation between successive levels: the old description language becomes the new command language. This structure also encompasses the mechanism by which learners are expected to move through the sequence. Moving becomes necessary because of the functional constraints of the problems to be solved (working memory load). It becomes possible because of the learner's reflective activities (triggered by the description language).

This empty structure of microworlds translates (a part of) Case's theory into a terminology that is relevant for the ILE designer. We therefore refer to this structure as an intermediary framework. This framework seems to be also useful for theoretical approaches. However, we did not investigate this issue deeply enough to make strong claim about the generalizability of our framework. Similarly, we have not assessed yet the generalizability of this framework to other learning environments.

References

- ANDERSON J.R. (1983) *The Architecture of Cognition*. Harvard University Press. Cambridge.
- ANDERSON J.R., BOYLE C.F. and YOST G. (1985) The Geometry Tutor. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*. Los Angeles. Vol.1
- BROWN A. (1987) Metacognition, Executive Control, Self-Regulation and Other More Mysterious Mechanisms. in F.E. Weinert and R.H. Kluwe (Eds) *Metacognition, Motivation and Understanding*. Lawrence Erlbaum. Hillsdale, NJ, pp. 65 - 115.
- BROWN J.S. (1985) Process versus product : a perspective on tools for communal and informal electronic learning. *Journal of Educational Computing Research*, vol.1 (2)
- BROWN J.S. (1990) Toward a new epistemology for learning. in C. Frasson and G. Gauthier (Eds). *Intelligent tutoring systems at the Crossroad of AI and Education*. Norwood, NJ; Ablex.
- CAMPBELL R.L. and BICKHARD M.H. (1986) *Knowing Levels and Developmental Stages*. Karger. Basel.
- CASE R. (1985) *Intellectual Development: from Birth to Adulthood*. New York: Academic Press.

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*. Springer Verlag. New York, pp. 1-18.

DERRY S.J. (1990) Flexible Cognitive Tools for Problem Solving Instruction. Paper presented at the *AERA symposium, Computers as Cognitive Tools*, Boston, MA. April.

DILLENBOURG P.(in press) The Language Shift : a mechanism for triggering metacognitive activities. To appear in P. WINNE and M. JONES (Eds). *Foundations and frontiers in Educational Computing Systems*. Springer-Verlag. Berlin

FREDERICKSEN J.R. and WHITE B.Y. (1988) Intelligent Learning Environments for Science Education, *Proceedings of ITS-88*. Montreal, June 1-3, pp. 250-257

LEBLANC M.D. (1988) Instructional Tools for Algebra Word Problems. *Proceedings of ITS-88*. Montreal, June 1-3, pp 238.-242

McARTHUR D., BURDOF C., ORMSETH T. ROBYN A., STASZ C. (1988) Multiple Representations of Mathematical Reasoning. *Proceedings of ITS-88*. Montreal, June 1-3, pp. 485-490

PALIES O. (1988) Building a Student Model without a Bug Library. *Proceedings of the Summer University on Intelligent Tutoring Systems*, Le Mans, 26 oct - 3 nov. 1988.

PIAGET J. (1971) *Biology and Knowledge*. The University of Chicago Press. Chicago.

ROSCELLE (1988) Sufficiency and Utility of Physics Problem Solving Processes. *Proceedings of ITS 88*. Montreal. June, 1-3, pp. 132-139.

ROSCELLE J. (1990) Designing for Conversations. *Paper presented at the AAAI Symposium on Knowledge-Based Environments for Learning and Teaching*, Stanford, CA, March 90.

SELF J.A. (1988) The Use of Beliefs Systems for Student Modelling. *Proceedings of the European Congress on Artificial Intelligence and Training*. Lille. Oct,88

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

STUMPF. M, OPWIS K. and SPADA H. (1988) Knowledge acquisition in a microworld for elastic impact : the DiBi System. *Proceedings of the Summer University on Intelligent Tutoring Systems*, Le Mans, 26 oct - 3 nov. 1988.

SUCHMAN L.A. (1987) *Plans and Situated Actions. The problem of human-machine communication*. Cambridge University Press. Cambridge.

VYGOTSKY L.S. (1978), *Mind in Society. The Development of Higher Psychological Processes*. Edited by M. Cole, V. John-Steiner, S. Scribner and E. Souberman. Harvard University Press. Cambridge, Massachusetts.

WENGER E. (1987) *Artificial Intelligence and Tutoring Systems. Computational and Cognitive Approaches to the Communication of Knowledge*. Loas Altos, Ca. Morgan Kaufmann

WERTSCH J.V. (1985) Adult-Child Interaction as a Source of Self-Regulation in Children. *The growth of reflection in Children*. Academic Press. Madison, Wisconsin, pp. 69-97.