Proceedings of the NACCQ 2000
Wellington NZ www.naccq.ac.nz

Concepts and Concept Mapping: Representing states and progression in a novice programmers understanding and abilities.

Chris Burrell

The Waikato Polytechnic
Hamilton
itcjb@twp.ac.nz

ABSTRACT

The physical representation of knowledge and the inter-relationship of the parts of that knowledge is fraught with the difficulties of representing the abstract in a physical way. Knowledge representation methods are investigated. The principles of concept mapping are explored together with an implementation of a student modeling and updating mechanism that is built into the Karel ++ learning environment for novice programmers.

Keywords

Computer, programming, student, modeling, microworld, knowledge.

1. INTRODUCTION

The karel ++ microworld.

The Karel ++ environment is a microworld specifically designed (Bergin, 1997)) to teach programming in a controlled environment. The purpose of a microworld is to deliberately simplify some areas that are of little or no immediate interest so that concepts and skills of more importance can be concentrated on. The Karel ++ language is similar to C++ and Java but is limited in that there is no ability to use variables and data types. The language has two classes of Robots, which can be inherited from and ability to write methods to extend the classes. The concept of "Objects", and problem solving with objects can be explored.

The robot world appears at first sight to be fairly limiting. The implementation I am using has a set of 20 Streets and Avenues at right angles to each other along which Robots can traverse under program control. The world is bounded by walls which a robot can detect but if it tries to move into a space occupied by a wall it is automatically disabled. Walls can also be placed anywhere along or across the streets and avenues. Robots can detect, place and retrieve beepers but if they try to pick up too may or place one that they do not have they are automatically disabled. Beepers can only be placed and picked up from street corners. This deceptively simple environment can lead to some complex scenarios requiring the development of problem solving skills and programming expertise.

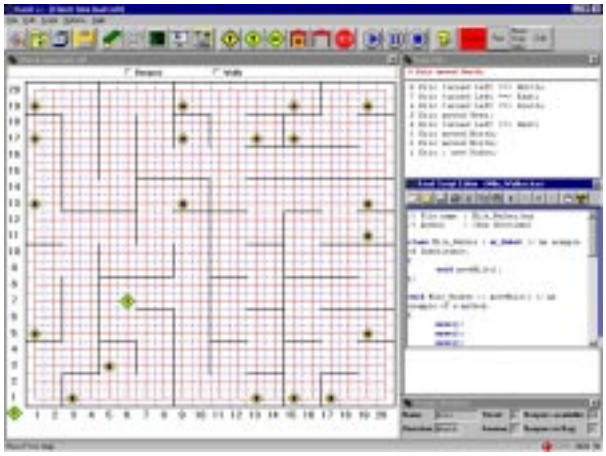

Figure 1. the Karel ++ environment.

2. THE REPRESENTATION OF KNOWLEDGE

What is knowledge, would we know it if we saw it? The Encarta World English Dictionary (1999)define Knowledge as "INFORMATION IN MIND general awareness or possession of information, facts, ideas, truths or principles. SPECIFIC INFORMATION clear awareness of explicit information, e.g. of a situation or fact". The Shorter Oxford Dictionary(1993) p 1503 definition is "The fact of knowing a thing, state, person etc; acquaintance; familiarity gained by experience. .... Intellectual perception of fact or truth, clear and certain understanding or awareness, especially as opposed to opinion."

2.1 Methods of Representing Knowledge

In representing knowledge, it is necessary to know and understand what it is about knowledge that we want to represent. The potential uses of that represented knowledge, has a significant effect on the representations we choose to use. The representations of knowledge used for teaching can be quite different to the representations used to denote understanding, and different again from representations used to elicit that understanding. The Encyclopedia of Artificial Intelligence (1992) p719 identifies a series of 5 categories, from a low to a high level, for representation in knowledge based systems.

Higher level representations may be transformed to a lower level representation (by a computer) for example a decision tree may be converted to a set of rules. Some forms of knowledge may be more easily represented in one form than another where a table may show pair, or entity and attribute relationships a semantic network can represent the relationship between objects.

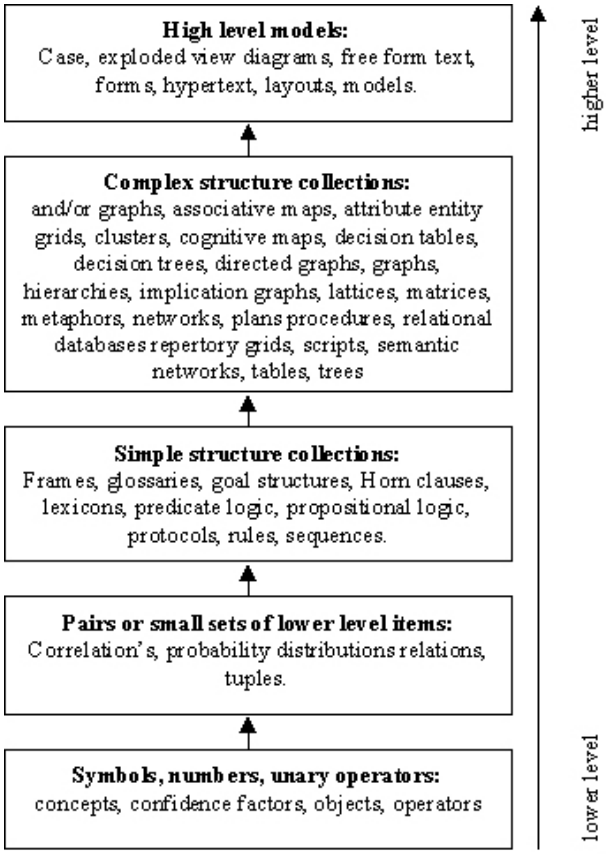

Figure 2. Representing knowledge.

2.2 Eliciting Knowledge

Manual methods, of eliciting knowledge, can range from brainstorming through unstructured to structured interviews, and Neurolinguistic programming where physical cues of eye movement and body language are used to enhance communication with an expert. Knowledge organisation techniques, such as, card sorting may be used to elicit knowledge where objects represented by cards are used to help identify a structure to that knowledge (Burton *et al.* 1987). Mediating representations (Johnson, 1989), protocol analysis, and user interface techniques, such as the wizard of Oz, are other methods that are used.

Machine learning systems derive knowledge from stored data. There are a multitude of methods that have been developed, they range from decision trees (Quinlan,

1993) where data is examined and tree structures developed, to discovery learning and explanation based learning. It is possible to generate both absolute and probability based rules from stored data. Systems have been developed to do this with names such as Induce (Michalski, 1983) PRISM (Cendrowska, 1987) and VERSION SPACES (Mitchell, 1982) have been produced.

Decision trees and rule sets can be denoted graphically, a simple example is shown below.

Fig 3 Decision Tree for the game of golf (Quinlan, 1993)

Intelligent tutoring systems have always been concerned with knowledge representation, both in terms of what the student knows and what is to be taught to the student. It is possible to use the same knowledge representation scheme for both. This paper is concerned with storing a representation of a students knowledge, what they have learned, so the storage of, and access to instructional knowledge is not considered in the detail it normally deserves.

2.3 Knowledge Representations

Minsky(Minsky, 1975) introduced the concept of "frames" for representing commonsense knowledge about ordinary things. A frame consists of a complex data structure with information about the components of the concept and links to other concepts. There may also be information about how a frame can change over time. An example of a simple frame definition is shown in Fig 4.

The structure representing a collection of facts about some domain is known as a knowledge base. An "Object" in the knowledge base denotes entities in the domain. Objects can represent abstract concepts and sets of entities. In the literature the terms "concept", and "frame" are often used to represent the same thing.

Frame	FAMILY
Is a	SOCIAL STRUCTURE
Slots	(MOTHER (a PERSON))
	(FATHER (a PERSON))
	(CHILD (a PERSON))

Figure 4. An Example of a frame definition.

```
father(terach,abraham).          male(terach).
father(terach,nachor).           male(abraham).
father(terach,haran).            male(nachor).
father(abraham,isaac).           male(haran).
father(haran,lot).               male(isaac).
father(haran,milcah).            male(lot).
father(haran,yiscah).

                                 female(sarah).
mother(sarah,isaac).             female(milcah).
                                 female(yiscah).
```

Fig 5: A biblical family database (Sterling, 1994)

Using logic to represent actions requires a multitude of explicit facts to be encoded, languages such as LISP or Prolog are used for this.

Production systems or rule based systems (Newell, 1995) represent knowledge procedurally as a series of "IF THEN Rules" which specify a condition and action, or premise and conclusion.

Concept mapping, representing knowledge in graphs is a technique developed at Cornell University (Novak, 1977.). (Novak, 1984) suggest that the initial ideas are articulated and classified into hierarchical concepts and then drawn by hand. The work is based upon theories of Ausubel ((Ausubel, 1968)).

Concept mapping has many uses in many disciplines, essentially it is a tool for thinking and learning that can be used for analysis and design. There are computer based tools that can assist with the production of concept maps however they essentially start with a word or phrase or some opposing ideas written in the middle of a blank piece of paper with related ideas on branches that radiate from the centre. The map generator

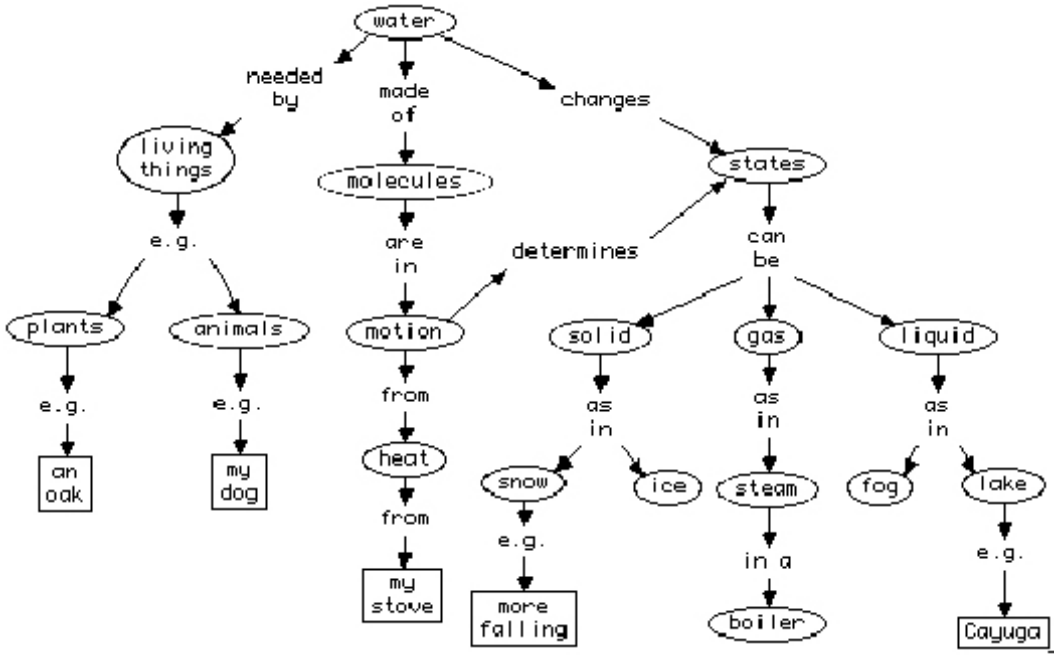

Fig 6 Concept Map representing a students knowledge (Novak, 1984)

can freely use branches from the idea to one or many others, arrows are used to join ideas from different branches. Related ideas can be grouped, notes and lists are used where and when needed. The resulting graphs of knowledge are networks of concepts, consisting of nodes (points/vertices) and links (arcs/edges). See Fig 6. Nodes represent concepts and links represent the relations between concepts. Concepts and sometimes links are labeled. Links may be non-directional, un-directional or bi-directional.

Concepts and links may be merely associative or tightly specified. They may be divided into categories meaningful to the context such as causal or temporal relations.

In 1968 Quillian (1968)developed the form of concept map that came to be known as semantic networks and which are now widely used for formal knowledge representation. Semantic networks (Sowa 1991), were motivated by cognitive models of human memory, they are a compromise between the logic "declarative" and the "procedural" forms of knowledge representation. Semantic networks represent information as a series of nodes that are connected to each other. They have been proposed as the foundation for knowledge representation in humans (Anderson, 1980; Lindsay, 1973; Brachman, 1979). They include the inheritance from nodes.

The relations (arcs) between the entities (nodes) is shown by the words in small letters. The arrows point

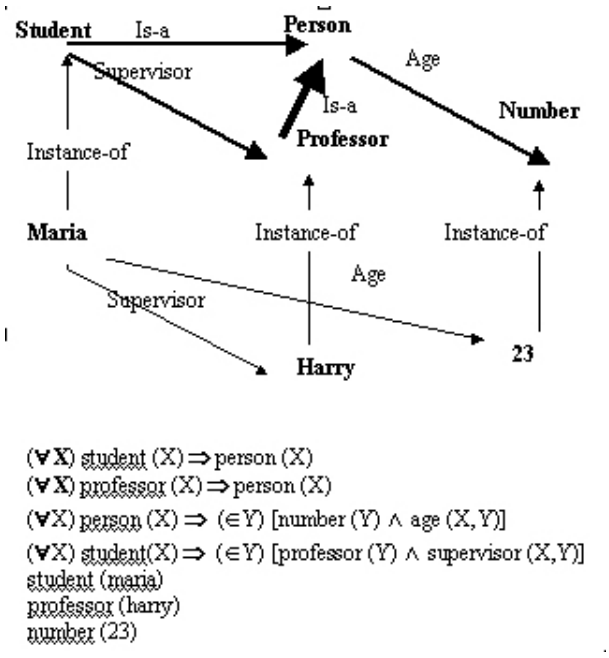

Fig 7 A Semantic network (from Encyclopedia of AI)

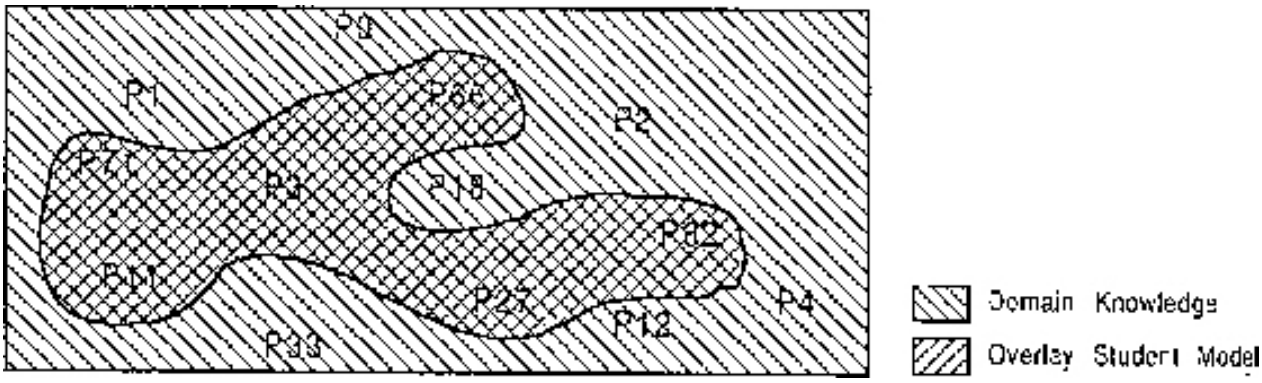

Fig 8 A graphical representation of the overlay model (Auberger, 1998)

to the entity or node that is inherited from. Prolog is a useful language with which to build semantic nets.

2.4 Processing in Semantic Nets

To be useful as a modeling tool, semantic nets must enable processing of the information they contain. Information is processed by a method known as arc traversal. This allows the identification of complex relationships between nodes. There are methods available that enable complex queries to be made about relations in semantic networks.

3. MODELING STUDENTS AND STUDENT PROGRESS

Like the construction of a concept map the construction of a student model will depend on its purpose. The depth and complexity of the map or model is reflected in the use to which it can be, or is intended to be, put.

In intelligent tutoring systems the student model should describe the knowledge and beliefs of the student (Tasso, 1992). It can also be used to derive or describe feed back specific to the student being modeled. Auberger (1998) quotes O'Shea (1979) page 35) who gives a definition of the student model:

"A student model is that component of a teaching program which is used to predict the current state of knowledge of the student. It usually is a simple algorithm operating on a database that includes a record of the student's past responses. For a teaching program with an expository teaching style a typical prediction might be that a student will always correctly answer some particular question. This prediction would be based on the student's past responses to some subset of questions previously asked by the program."

In this paper we are considering building a model only on the students responses (actions).

3.1 The Overlay Method of Student Modeling

There are three variations of student modeling that can be described in terms of overlays. The simplest is shown in the diagram Fig 8. The others can be represented by similar diagrams and are described by the terms differential modeling where only the difference in knowledge between the domain knowledge and the learners acquired knowledge is stored. The other method is known as pertubation modeling, where the learners knowledge is considered as a subset of the domain knowledge but acknowledges that it will also to include some misconceptions or faulty knowledge. In this case the student model could be envisaged as expanding beyond the domain knowledge in some areas. Fig 8 is reproduced from Auberger 1989 who credits KASS89. It represents all domain knowledge as the outer area and the current student model as an overlay. Semantic networks may be used as the basis for overlay models.

3.2 Student Modeling and the karel ++ System

The student modeling system in the Karel ++ learning environment has the primary task of capturing the students actions and recording them in a data base, a secondary action is to update the concept mapping part of the database with values derived from the actions.

The object-oriented approach to the design of the system allows events triggered by students' actions to

	Declarative Knowledge	Procedural Knowledge
Syntactic knowledge	**Knowledge of Syntactic facts related to a particular language, such as:** • Knowing that a semicolon is needed to end each statement in Pascal • The ability to explain the syntactic differences between a procedure and a function in Basic	**Ability to apply rules of syntax when programming, such as:** • Ability to write a syntactically correct REPEAT Statement in Pascal. • The ability to open a text file and read from it using BASIC
Conceptual Knowledge	**Understanding of and an ability to explain the semantics of the action that take place as a program executes, such as:** • The ability to explain what a fragment of pseudo code does. • Knowing the way in which the result of a function activation is returned in a particular language.	**Ability to design solutions to programming problems such as:** • The ability to design a procedure to compute the mean of some data. • The ability to modify a program that prints a 1-D array to print a 2-D array.
Strategic/Conditional Knowledge The ability to design, code and test a program to solve a novel problem.		

Table 1 Types of knowledge (McGill, 1997)

send messages to the object responsible for maintaining the student model. Another part of the system is responsible for interpreting the database and presenting the responses in a graphical form to the user.

The student modeling system in the Karel ++ environment has the following components.

An event trigger (database) log, and a concept map (data structure) database. The database event log is a simple record of events and parameters measured. The concept map database consists of sets of values that are updated through a set of rules derived from the context of the triggered events.

Students actions trigger events, the notification of which are sent to a mapping object. The mapping object has two tasks

- To record the event and data relevant to it on the event database.
- To process the event and update relevant concepts in the concept database.

Data	Major Concepts (to be broken into sub concepts)
Compile time errors.	Syntax knowledge. Use of the compiler as an error finding machine. Different types of error contribute to different concept deficiencies.
Run time errors. Program finishes prematurely but does not solve the problem. Program finishes but does not solve the problem. Program cannot finish (continuous loop).	Deficiency in understanding the microworld. Deficiency in understanding the language as a tool.
Correct compilation.	Valid program structure.
Correct result from running the program. Pre and post condition states met.	Valid problem solution.
Derived Program complexity metrics for a given problem.	Problem solving ability measure. Measure of the coding ability.
Complexity of the problem being tackled.	Advancement in the language and problem solving .abilities
Number of real time program steps to complete the "world" assignment	Neatness of solution appreciated.
Number of attempts and depth of solution when compiling and running programs.	Problem solving / programming skills development.

Table 2 Displaying the data and concepts

This study is interested in the changes and development of these "concept" values in time so each update includes a timestamp.

4. KNOWLEDGE REQUIRED BY PROGRAMMERS

From the education literature the following types of knowledge have been identified

- **Declarative Knowledge:** I know of something
- **Procedural Knowledge:** I know how to do something.

Programmers need to have effective command of three separate but interrelated types of programming knowledge. (Bayman and Meyer 1988 from A conceptual Framework for analysing Students' knowledge of Programming McGill and Volet 1997).

These are

- **Syntactic:** Knowledge of specific facts about a programming language and rules for its use
- **Conceptual:** Understanding of computer programming constructs and principles.
- **Strategic:** Programming specific versions of general problem solving skills.

McGill and Volet (McGill, 1997) extended this and produced and a framework which used these as a base and interprets them in terms of Syntactic Knowledge and Conceptual knowledge. The table (Table 1)also identifies Strategic or Conditional knowledge, which is the ability to design, code and test a program to solve a novel problem.

A conceptual Framework of the Various Components of Programming Knowledge

Student acquisition of these types of knowledge is usually through a series of guided computer laboratory exercises. Often these exercises involve many small programming tasks, and some larger works, through which the novice is expected to assimilate the knowledge. The abilities of students in their programming classes, are often monitored at a fairly coarse level of granularity. This may be an instructor or lab assistant. Generally, no record is kept on the detailed progress of a student. Usually only the ability to produce "correct" results and structure in the final presented product is noted.

Conventional (non OO) computer programs have a definite start, a main body and an end. Object Oriented languages tend to have smaller amounts of code in a larger number of "objects" to achieve the same end.

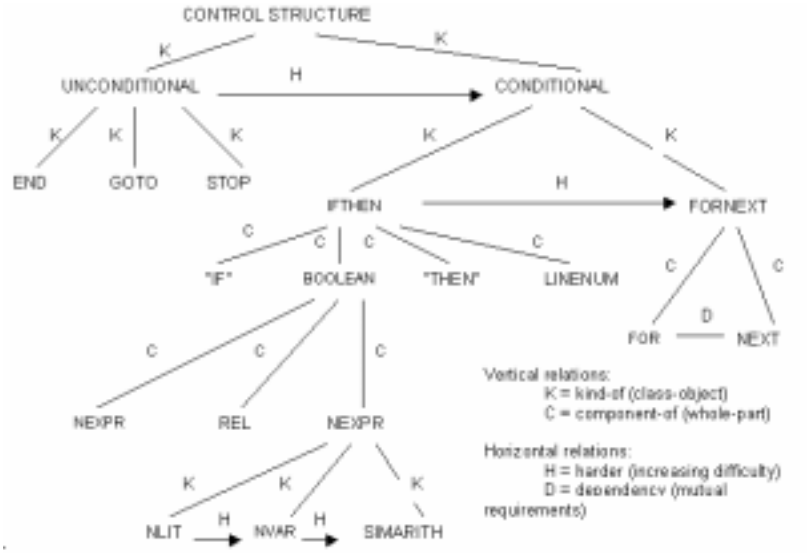

Fig 9 Representing the BASIC programming language

Immaterial of the generation of language, computer languages have three different types of construct (Loop, Choice, Sequence) that a user has to master in order to solve any but the simplest of problems.

Combinations of these constructs can be used to solve any problem that is computable. This together with the ability to divide the problem into independent modules that can be coded and that pass information or messages between them is at the core of successful programming.

Wescourt *et al*. (1997) have examined the programming language BASIC and produced a diagram representing increasing complexity. Fig 9, this representation can be extended for object oriented languages.

With object oriented techniques the level of programming has moved away from the direct control of the machine to a higher level of abstraction, the structure of the computer language has become more aligned to the problems which need to be solved.

An indication of some of the types of data captured in this environment and the concepts, derived from the data, is given in table 2.

The display of the data and the derived concepts is the subject of further study. The major problem is in representing events and the changes in concepts in time, and making the representation easily understandable and of use to both the student programmer and the lecturer.

5. SUMMARY

This paper has explored the work in progress in the development and implementation of the Karel ++ microworld. This microworld enables skills to be developed in a controlled environment and the recording of data for interpretation of learning. Methods of representing knowledge were explored and the implementation of a two level database has been outlined. The database can be used both to show progress and allow further re-examination and interpretation of the data.

6. REFERENCES

Anderson, J., R., and Bower, G. H. (1980). "Human associative memory : a brief edition". Hillsdale, N.J.: L. Erlbaum Associates.

Auberger, M. (1998). "Student modeling in educational multimedia titles using agents", [Web document]. Available: http://aif.wu-wien.ac.at/usr/geyers/archive/auberger/proposal/node57.html [2000, 11-05-2000].

Ausubel, D. P. (1968). "Educational Psychology: A Cognitive View". New York: Holt, Rinehart and Winston.

Bergin, J., and Stehlik, M., Roberts, J., Pattis, R. (1997). "Karel ++ A gentle Introduction to the Art of Object-Oriented Programming": John Wiley and Sons.

Brown, L., (Ed). (1993). "The New Shorter Oxford English Dictionary". (Vol. 1). Oxford: Clarendon Press.

Cendrowska, J. (1987). An algorithm for inducing modular rules. "International Journal of Man-Machine Studies", 27(4), 349-370.

Johnson, N. E. (1989). Mediating Representations in Knowledge Elicitation. In D. Diaper (Ed.), "Knowledge Elicitation: Principles, Techniques and Applications ": John Wiley & Sons.

McGill, T. J., and Volet, S. E. (1997). A conceptual framework for analyzing student's knowledge of programming. "Journal of Research on Computing in Education", 29(3), 276-297.

Michalski, R. S. (1983). A theory and methodology of inductive learning. In C. a. M. Michalski (Ed.), "Machine learning: An artificial intelligence approach". San Mateo; CA: Morgan Kaufmann.

Minsky, M. (1975). A Framework for Representing Knowledge. In P. H. Winston (Ed.), "The Psychology of Computer Vision". New-York: McGraw-Hill.

Mitchell, T. M. (1982). Generalization as search. "Artificial Intelligence", 18, 203—226.

Newell, A., and Simon, H. A. (1995). Human Problem Solving, Englewood Cliffs, New Jersey: Prentice Hall. 1972. In E. A. F. J. Feldman (Ed.), "Computers and thought"(pp. 279-293.). Cambridge, MA: MIT Press.

Novak, J. D. (1977). "A theory of education". Ithaca,N.Y.: Cornell University Press,.

Novak, J. D., and Gowin, D. B. (1984). "Learning How to Learn". New York: Cambridge University Press.

O'Shea, T. (1979). "Self improving teaching systems: An application of artificial intelligence to computer assisted instruction". Basel: BirkHauser Verlag.

Quillian, M. R. (1968). Semantic memory. In M. Minsky (Ed.), "Semantic Information Processing". (pp. 216-270.). Cambridge, Massachusetts,: MIT Press.

Quinlan, J. R. (1993). "C4.5: Programs for machine learning". London, England: Morgan Kaufman.

Rooney, K., (Ed). (1999). "Encarta World English Dictionary". Sydney: Pan Macmillan.

Shapiro, S. C., (Ed). (1992). "Encyclopedia of Artificial Intelligence". (2 ed.). New York: Wiley.

Sterling, L., Shapiro, E. (1994). "The Art of Prolog: Advanced Programming Techniques". (2nd ed.): MIT Press.

Tasso, C., and Fum, D., Giangrandi, P. (1992). "The use of explanation based learning for modeling student behaviour in foreign language tutoring". Berlin: Springer Verlag.

unknown. (1996). "Learning Skills Program: Concept Mapping". Available: http://www.coun.uvic.ca/learn/program/hndouts/map_ho.html [2000, 11/05/2000].

Westcourt, K., and Beard, M., Gould, L. (1977). "Knowledge-based adaptive curriculum sequencing for CAI: Application of a network representation". Paper presented at the National ACM Conference, Seattle, Washington.