1-1-1977

An alternative curriculum for computer literacy development.

Stewart Allen Denenberg
University of Massachusetts Amherst

AN ALTERNATIVE CURRICULUM FOR COMPUTER LITERACY DEVELOPMENT

A Dissertation Presented

by

Stewart Allen Denenberg

Submitted to the Graduate School of the
University of Massachusetts in partial
fulfillment of the requirements for the degree of
DOCTOR OF EDUCATION

SEPTEMBER, 1977

. EDUCATION

This work was supported in part by a grant from the Control Data Corporation.

AN ALTERNATIVE CURRICULUM FOR COMPUTER LITERACY DEVELOPMENT

A Dissertation

by

Stewart Allen Denenberg

Approved as to style and content by:

Howard A Peelle

Howard A. Peelle, Chairman

Klaus Schultz

Klaus Schultz, Committee Member

John R. Dixon

John R. Dixon, Outside Committee Member

Conrad A Wogrin

Conrad A. Wogrin, Outside Committee Member

Mario D Fantini

Mario Fantini, Dean
School of Education

ACKNOWLEDGEMENTS

To Claire who made this all possible by leaving Illinois with me and going back to work to put me through school. To Howard Peelle who accepted me as a graduate student and who constantly and unselfishly offered his encouragement and shower of intellectual sparks and fertile insights during our many discussions. To Jack Dixon who allowed me to find financial support, offered much solid advice, and who is a joy to work with. To Klaus Schultz and Conrad Wogrin who kindly consented to be committee members and in the process became friends and collaborators. To Luke Kaven whose encyclopaedic knowledge of Plato made the implementation viable. To Bill Cole and Bob Moe of Control Data Corporation who had the faith to supply funds and much good criticism and suggestions. To Sharon Ladev who made the copies from the Plato screen. To Nancy Podolak who typed the final draft of this dissertation in an incredibly short time. To all of you---my deepest thanks.

ABSTRACT

AN ALTERNATIVE CURRICULUM FOR COMPUTER LITERACY DEVELOPMENT

FEBRUARY 1978

Stewart A. Denenberg, B.S., Union College
Ph.D., University of Massachusetts
Directed by: Professor Howard A. Peelle

The design and implementation of an alternative curriculum for computer literacy development (ACCOLADE) is described. ACCOLADE is a system comprising learners, teachers, and a computer that allows learners to educate themselves about computers using the computer and teachers.

The computer provides (1) a Map (consisting of a Yellow Pages, a semantic network, and a Keyword Index) for searching out computer literacy knowledge and for showing the structure of that knowledge; (2) a Communications File for messages between teachers and learners; and (3) a Computer Managed Instruction (CMI) System for testing, recording learner-computer transactions, diagnosing, prescribing, and accrediting the knowledge acquired by the learners. The teachers facilitate the skills and behaviors of logical thinking, problem solving, learning, confidence and autonomy through the medium of computer programming.

ACCOLADE was designed with the following goals in mind:

- To demonstrate an alternative, effective teaching-learning system.
- To allow the learner to take responsibility for effecting his own education.
- To encourage a teacher-learner partnership and be responsive to their needs and requirements.
- To use computers and people as components so that ACCOLADE encourages

the human-computer partnership.

- To identify and provide resources for the acquisition of computer literacy knowledge in the areas of applications, history, social issues, hardware and software; to illuminate the structure of this knowledge space and, by doing so, shape an informed attitude about what computers can and cannot do.
- To teach the skills and behaviors of logical thinking, problem solving, learning, confidence, and autonomy through the medium of computer programming.
- To be effective; i.e., relevant, easy-to-use, reasonably inexpensive in cost and time and, above all, enjoyable for both learners and teachers.
- To be an adaptive system; i.e., a system that can modify its performance through interaction with its environment.

Computer literacy, the teacher-learner relationship, and human-computer relationships are discussed in terms of the philosophical underpinnings, background, and related research necessary for understanding the design and implementation of ACCOLADE. The general design is specified, a specific implementation using the Control Data Corportation Plato System is described, and a modest pilot study was conducted with a small group of learners and teachers.

Observations and results of the implementation and pilot study are interpreted and several inferences regarding the success of the implementation in terms of the design goals are offered. The dissertation concludes with some specific suggestions for proposed future development of ACCOLADE as well as a discussion of some educational issues and implications raised by ACCOLADE that may warrant future research.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation

My main motivation was to build something useful and enjoyable. Since my interests and competencies lie in education and in computer science, it felt quite natural and satisfying to design and build a system whereby people could use computers to educate themselves about computers. This system is called ACCOLADE--An Alternative Curriculum for Computer Literacy Development, and its philosophy, design and implementation are the subjects of this dissertation.

The philosophical foundations of ACCOLADE are eclectic in nature, being based on both eastern and western philosophies and psychologies. I have studied and practiced Buddhist philosophy and techniques over the past ten years, being heavily influenced by the works of Alan Watts, D.T. Suzuki, Christmas Humphries, Phillip Kapleau, J. Krishnamurti, and Chögyam Trungpa. Western philosophers and psychologists who have affected my growth and development include Fritz Perls, Sheldon Kopp, and Alfred North Whitehead.

These eastern and western writers seem to agree on one major point of philosophy: the individual must assume responsibility for his own actions, for his own life, for his own education. The Gautama Buddha has instructed us to work out our own salvation with diligence. If part of one's salvation is his education, then he must be willing to begin with himself, wherever he is, and accept the responsibility for educating himself. Only by accepting that responsbility can come freedom and understanding.

Fortunately ACCOLADE can draw upon much research that has been done to facilitate the process of educating oneself. Dwyer, Papert and Peelle have proposed modes of self education in the area of computer literacy that are not only pleasurable but rewarding. Dixon and Weed have suggested a separation of content and process in the engineering and medical curricula that can be easily extended to computer literacy. Bruner and Whitehead have supplied much of the pragmatic philosophy dealing with the importance of structure in knowledge. I have attempted to integrate the work of all of these men in the design and implementation of ACCOLADE.

The common conviction running through all these men's work also runs through ACCOLADE: in the final analysis it is the learner who must educate himself; if the learner is not willing to take responsibility for his own education, no educational process will work, no matter how powerful its pedagogy and curriculum. It is my hope that the learner will realize this early in his education and not have to nod agreement with that anonymous wall poster: "I lived over half my life before I realized it was a do-it-yourself job."

1.2 Goals

ACCOLADE was designed and implemented with the following goals in mind:

- To demonstrate an alternative, effective teaching-learning system.
- To allow the Learner to take responsibility for effecting his own education.

- To encourage a Teacher-Learner partnership and be responsive to their needs and requirements.
- To use computers and people as components so that ACCOLADE encourages the human-computer partnership.
- To identify and provide resources for the acquisition of computer literacy knowledge in the areas of applications, history, social issues, hardware and software; to illuminate the structure of this knowledge space and, by doing so, shape an informed attitude about what computers can and cannot do.
- To teach the skills and behaviors of logical thinking, problem solving, learning, confidence and autonomy through the medium of computer programming.
- To be an adaptive system; i.e., a system that can modify its performance through interaction with its environment.
- To be effective, i.e., relevant, easy-to-use, reasonably inexpensive in cost and time and, above all, enjoyable for both Learners and Teachers.

1.3 A Brief Description of ACCOLADE

The ACCOLADE system consists of (1) Learners, (2) Teachers and (3) a Computer.

The Learner's goal is, presumably, to acquire some level of computer literacy, and he may do this by appropriate interactions with the two other parts of the ACCOLADE system. To acquire computer literacy knowledge, the Learner uses the Computer as a tool to search through the knowledge space of computer literacy topics and as a responsive display device which exhibits the structure of the topics.

To gain the skills and behaviors (logical thinking, problem solving, learning, confidence, and autonomy) the Learner works with other Learners, a Computer, and a Teacher using a programming language to solve case-study problems associated with the Learner's discipline and interests. If the Learner wishes to accredit his experience, he may self-administer a range of criterion-referenced tests via a Computer Managed Instruction System.

The Computer Managed Instruction (CMI) System handles testing, recording, and diagnosing and prescribing for the Learner. The CMI System monitors Learner test progress and Learner-Computer transactions, issuing management reports which can be used by the Teacher and Learner, and continually performs an item analysis on each question in the test data base so that weak or ambiguous questions can be removed or improved.

A Computer Map allows the Learner to take responsibility for acquiring computer literacy knowledge. The Map shows the relations between the computer literacy topics using a linked semantic information network, Keyword Index, and Yellow Pages. The Yellow Pages also indicate the resources available for acquiring knowlede about computer literacy topics. The resources include:

1) Printed Material (books, magazines, journals, etc.)
2) University and College courses
3) Computer Assisted Instruction Lessons
4) People
5) Movies
6) Videotapes
7) Audiotapes

The Teacher is responsible for maintaining the resources in the Yellow Pages, creating criterion-referenced tests in the CMI System, and instilling in the Learner certain skills and behaviors associated with computer programming. To teach these skills and behaviors, the Teacher coordinates peer-group learning sessions and assigns practical programming problems from within the Learner's chosen discipline. The Teacher may choose more traditional methods of teaching (such as formal lectures) and testing (norm-referenced tests) but ACCOLADE affords other options. For instance, the testing can be handled by the CMI System and the formality of lecturing can be replaced by informal advising, allowing the Teacher to become more of a mentor and less of a judge to his students.

In many educational environments, Learners are viewed as raw input data to be processed by the system. ACCOLADE is structured so that the Learner is one of three equal partners within the system. An educational system should help the Learner to educate himself--its prime goal should be to liberate the Learner, not to make him dependent. It should help the Learner become a freer, more humane, more autonomous person capable of effecting his own education and taking a responsible and responsive role within society. The ACCOLADE System attempts to foster these goals.

1.4 Dissertation Overview

The first three chapters (II, III and IV) contain the background material and related research necessary for understanding the design and implementation (chapters V and VI) of ACCOLADE. The last Chapter (VII) contains suggestions for future research and development of

ACCOLADE.

More specifically, Chapter II examines the Teacher-Learner relationship in terms of the authorities and responsibilities that each may assume as part of the educational process. Associated issues, such as authoritarian vs self-directed learning, the need for structure, process vs content, and the value of heuristics are discussed; and a partnership between Teacher and Learner is proposed as the basis for the educational experience.

Chapter III deals with the human-computer relationship and discusses artificial intelligence and intellectual augmentation as conceptual tools that each can offer the other, thus providing the basis for another partnership.

Chapter IV discusses the subject matter of ACCOLADE--computer literacy--how it can be defined and justified as a worthwhile topic, as well as some of the options available for the design of the curriculum, the pedagogy, and the administrative delivery system.

Chapter V specifies the design for the ACCOLADE System, drawing upon the philosophies and conceptual tools developed and described in the preceding chapters. The goals of the system are defined and the components of the system are described in terms of their functions and the relations between them.

Chapter VI describes an implementation of ACCOLADE using the Plato System and the results of study. The methodology is accounted as the components of the actual system are described. A sample Learner interaction with the Computer part of the system and the types of reports available to the Teacher are illustrated. The observations and results concerning the Learners, the Computer, and the Plato System are interpreted and several conclusions regarding the success of the implementation in terms of the design goals are offered.

Chapter VII offers specific suggestions and recommendations for the future research and development of ACCOLADE. They include short and long range suggestions and a discussion of some educational issues and implications raised by ACCOLADE that may warrant additional research.

CHAPTER II

THE TEACHER-LEARNER RELATIONSHIP

2.1 Chapter Overview

This chapter will point out a persistent problem within the teacher-learner relationship that occurs when either the teacher or learner relinquish authority to the other in an attempt to escape responsibility for the educational process. Authority and responsibility is discussed next in terms of the issue of authoritarian vs self-directed education and some of the choices within the educational experience. Finally a solution to the problem is proposed--the forming of a teacher-learner partnership.

The issue of authoritarian vs self-directed education is framed within the context of the question, 'who knows best'?, and by the ingredients of 'discipline', 'the innate goodness or evil of man', 'trust' and 'process vs content'. This is followed by a discussion of these issues in terms of the research and thinking of several selected educational philosophies.

The educational experience is defined as choices made in the three areas of: (1) educational objectives, (2) teaching-learning experiences, and (3) evaluation. Each of these areas is explored in terms of the responsibilities and authorities inherent in the teacher-learner relationship. Educational objectives are similar to management goals or objectives, but in the educational context they constitute a statement of what the learner or teacher is to be like after an educational experience. Once the objectives are chosen, specific teaching-learning experiences are selected to actualize those objectives. Finally evaluative methods are applied to the teaching-learning experiences and to the objectives themselves to assess their validity and effectiveness.

The proposed partnership suggests that the teacher and learner view authoritarian and self-directed modes of education not as opposing philosophies but as opposite poles of a scale of options which can be used as a basis for negotiation of a particular partnership. The partnership also allows the learner and teacher to negotiate authorities and responsibilities associated with the specific choices for the educational objectives, the teaching-learning-experience and the evaluative process. Once the teacher and learner have formed a partnership, the specific educational experience can begin.

2.2 Introduction - The Problem

> "Gaudeamus Igitur department: Safely back home after a three-month lecture tour of U.S. campuses, Jorge Luis Borges, 76, Argentina's nearly blind poet-essayist, announced flunking grades for the 'extraordinarily ignorant' Yankee students. Said he: 'They read only what they must to pass, or what the professors choose. Otherwise they are totally dedicated to television to baseball and to football.'"[1]

Perhaps one of the most important difficulties present in the teacher-learner relationship is that, in many instances, the teacher and the learner perceive themselves as separate, unrelated, independent and closed systems. When the teacher and the learner act as if they

[1] Time Magazine, June 21, 1976, p. 47.

have no mutual goals then the situation depicted in the above quotation can easily arise.

In a traditional educational setting such as an undergraduate program in a large university there are, in many cases, few mutual goals between learner and teacher. The situation is, instead, that the learner (and the state) pays the teacher to educate the learner. Since the teacher works for the learner, from this perspective it appears that the learner has abdicated his responsibility to the educational relationship; it is as if the learner is saying to the teacher, "I pay you to tell me what to do--and since I pay you, 'what to do' is _your_ problem, not mine..."

This seems a common situation, not limited to the educational setting. Indeed, a great many citizens choose to be told what to do in their dealings with society. Rather than offer to work together to handle societal problems, instead they pay a civil servant to do it, relieving them of their responsibility to the problem. Unfortunately if we choose to relinquish our authority to a civil servant (so that we have no responsibilities in that area) we run the risk of the servant becoming master.

The problem, in an educational environment, then can be discussed in terms of the authorities and responsibilities of the teacher and learner.

2.3 Authority and Responsibility

Let us begin by realizing that there are more than just the teacher and learner involved in the relationship. The teacher also interacts with other teachers and to administration which provides an environment in which to teach and places certain demands back on the teacher. The learner is heavily influenced by his peers, his parents and many out-of-school factors. So, although the teacher and learner are not actually a closed system, for the purpose of delimiting the discussion we shall view the teacher-learner relationship as if it were a closed system.

Authority, in the context of this paper, will be defined as the power to control[2] the educational experience--the totality of the relationships between the learner and the teacher. The educational experience centers on three main areas: (1) educational objectives, (2) teaching-learning experiences, and (3) evaluation.

An educational objective is, expanding Mager's[3] definition, a statement of what a person (teacher or learner) is to be like when he has successfully completed an educational experience. These objectives are usually stated in terms of some criterion of performance, e.g. 'The learner will be able to write a twelve line or less APL program to solve cubic equations by the end of the semester.'

Once the educational objectives are clearly defined, a set of teaching-learning experiences can be constructed which attempts to actualize the objectives. The teaching and learning experiences consist of specific choices for curriculum (the context or topics to be investigated) and pedagogy (the mode of topic presentation).

[2] Control is a word that is often used in the educational context to denote authority and will be used interchangeably with authority.

[3] Mager, R.F., Preparing Instructional Objectives, (Palo Alto, Calif.: Fearon, 1962), p. 3.

The evaluation process comprises three parts:

1) the evaluation of the teaching, e.g. by learner ratings of the teacher.

2) the evaluation of the learning, e.g. by the teacher testing the learner.

3) the evaluation of the objectives, i.e. in terms of validity and effectiveness.[4]

The first two parts evaluate the teaching-learning experience, while the third part evaluates the educational objectives. Authority then consists of control, by either the teacher or learner over the educational experience which consists of specific choices for the educational objectives, the teaching-learning experience and the evaluation of the objectives and the teaching-learning experiences.

Authority gives rise to responsibility. Within an educational experience, the teacher and learner have certain responsibilities to one another. They have certain social responsibilities whereby they limit their behaviors in respect to the rights of the other. Each also has a responsibility to achieve the educational objectives. This includes the responsibilities inherent in the choice of the teaching-learning experiences and their evaluation; these will be discussed in more detail in section 2.4.

The authorities and responsibilities inherent in the teacher-learner relationship center on the choice of educational objectives and

[4] The evaluation of the evaluation could also be considered here but will not be--one must be careful that it not become an infinitely recursive process.

touches directly the issue of authoritarian versus self-directed education.

2.3.1 Authoritarian versus Self-Directed Education

The issue of authoritarian vs self-directed education is framed within the context of the question, 'who knows best'?, and by the ingredients of 'discipline', 'the innate goodness or evil of man', 'trust' and 'process vs content'. This is followed by a discussion of these issues in terms of the research and thinking of several selected educational philosophies.

This issue can be framed by using extreme cases. Authoritarian education, in the extreme, is a situation where the teacher has complete authority over the choice of the educational objectives within the educational experience with no responsibility to the learner--he has complete control over the learner and no responsibility to the learner. In extreme self-directed education the opposite is true: the learner is in complete control of the educational objectives within the educational experience and has no responsibility to the teacher. Of course, there exists the full range of options between these poles.

2.3.1.1 Who Knows Best?

Another way to confront this issue is by posing the question: 'who knows best -- the teacher or the learner?' Authoritarian educational philosophy would, in the extreme, take the view that 'teacher knows best', while self-directed philosophy, at the other extreme would claim that the student does or 'learner knows best'.

2.3.1.2 Discipline

Discipline is another ingredient in the issue:

> "Discipline is in fact the key to everything... And here there are two schools of thought: according to one, discipline is necessary and only thru discipline can one learn to find the right way: according to the other school of thought, things should be allowed to develop in their own way and if there is less discipline, if things are left to the individual's choice or interest, then he will develop a personal interest in the subject and there will be no need to impose anything on him. Both are extreme views".[5]

The extreme views are those of authoritarian (impose discipline upon the learner) and self-directed (let the discipline grow from within the learner) philosophies.

2.3.1.3 Attitude Towards Man

Also inherent in the two philosophies is an attitude towards man himself. Authoritarianism would stress the failability of man and the need to impose discipline and structure on the learner or he would certainly not undertake the often arduous task of learning. The self-directed philosophy might argree with man's failability but tend to be more forgiving, possibly countering with the argument that if the learner chooses not to learn, then that's fine; when he decides to become motivated, he'll learn and learn more deeply and quickly than if forced to do so. The self-directed philosophy has the implicit faith that learning, when it is self-motivated, is a joyful experience -- it

[5] Trungpa, C., Meditation in Action, (Berkeley, Calif,: Shambala, 1969), p. 46.

is its own reward.

> "A common misconception holds that intelligence by itself lacks motive force which therefore must be supplied by outside regards such as parental approval or some practical goal. While not discounting outside motivational influence, Piaget insists on the intrinsic biological need for acting and knowing."[6]

2.3.1.4 Trust

Surrounding the issue of the innate 'goodness' or 'evil' of man is the question of trust. If man in innately evil he certainly cannot be trusted to do the 'right' thing of his own volition, and therefore a heavy dose of discipline is necessary. If man is to be considered as basically good, then trust vanishes as an issue; it is assumed that the 'discipline' will grow from within, guiding the learner to do whatever is right for him. This, of course, also applies to the teacher.

2.3.1.5 Process vs. Content

Process versus content is also a key ingredient to the issue. Adherents of the accumulation of content within the educational experience usually side with the conservative authoritarian view: conserve what is good and useful in life by solidifying it into a static form so that it can be made to last and be re-used. If it can be preserved, it can be passed on to future learners. Viewed through eastern philosophy, this attitude is basically a manifestation of the ego trying to separate itself from the rest of 'What Is' by trying to control 'What Is'. It is an attempt to gather security from the risky,

[6] Furth, H.G., Piaget and Knowledge, (Englewood Cliffs, N.J.: Prentice-Hall, 1969), p. 95.

dangerous, chaotic life which constantly and turbulently crashes upon us. It tries to maximize pleasure and minimize pain by building walls (content) against the flow (process) of life. But, according to eastern philosophy, this results not only in stagnation but also in eternal frustration since this view holds that one is not separate from life; therefore it is impossible to build a wall strong enough to keep out the ever changing manifestations of life. This frustration leads one to try to build ever stronger walls resulting in man being trapped on a 'wheel' of self-generating ego.

Proponents of process would hold more with eastern concepts. Life is constantly changing and all specific forms are impermanent so that any attempt to freeze or ossify any aspects of life into a specific state of content merely kills that life. Thus, they would say that the best strategy for dealing with life in general and education in particular is to empty oneself of content and conditioned learning to leave space for life to flow in.

2.3.1.6 Related Educational Research

These related topics of 'who knows best?','discipline', the 'innate goodness of man' and the implications of 'trust', and 'process versus content' will next be briefly discussed in terms of the research and thinking of several educational philosophies.

Weed, in designing his medical education curriculum at the University of Vermont has come down resolutely on the side of self-directed education and specifically addresses the question of trust in the learner by the teacher:

> "Instead of being taught that he [the learner] can discover by himself, he is taught that he cannot get started without prerequisite courses; instead of being encouraged to ask questions based on his own observations, he is given answers to questions he never asked; instead of developing confidence in his ability to learn, he is taught that he cannot trust himself to learn without prior instruction in data and theories. Yet nothing could be more contradictory to the goals of education or more antithetical to the method of science, and nothing could leave him less well-prepared for the future or less confident in his ability to learn, than to realize that those who teach him cannot trust him to begin with his own observations."[7]

Dwyer has also expounded the self-directed viewpoint in the context of using technology to enhance education. He contends it is the learner's history of experiences or 'cultural background' that primarily determines what the learner will get out of an educational experience.[8] Dwyer goes on to say that knowledge is not transmitted (as a radio transmits signals) from the teacher to the learner -- it is always created by the learner himself. "The revoluntary goal that follows from this stance is to design a school where the students assume from the beginning that their task is to invent all knowledge"[9] -- a decidely eastern viewpoint! The authoritarian principle of outside imposition of discipline on the learner is perceived by Dwyer as a non-humanistic and futile approach to education; he speaks of "the futility of imposing subject content on

[7] Weed, L.L., PROMIS Lab, U. of Vermont, Paper given at the Macy Foundation in Aspen, Colorado, "A New Curriculum," Sept. 8-11, 1975, Appendix,.p. 3.

[8] Dwyer, T., "The Community of Learning Model for using Computers in Education," Proceedings of the ACM National Conference, Houston, 1976.

[9] Ibid, p. 5.

the student who does not perceive its acquisition as important."[10]

However, Dwyer is also aware of the merit of the authoritarian view:

> "An authoritarian view of education is not an evil one. It is, in fact, valid and essential, since it holds that each civilization presents its young with a large history of accomplishment, that there is always a significant heritage worth examining, that there is invaluable information about theories that did and did not work, and that these can (in fact must) be passed on for the profit of succeeding generations.
>
> The difficulty with this view lies not in stressing the value of accumulated know-how. The problem is with the tendency of human teachers (and even more so, authors of CAI) to imbed the information they transmit within their own personal and unalterable interpretations (models) of how to use this information. Piaget (1970) tells us that, indeed, the lesson is always bound to conform to the tendencies of the teacher, since that is by far the easiest solution."[11]

Einstein, would have agreed with Dwyer. He appears to have been forced to swallow a heavy dose of 'teacher knows best' and to have had great difficulty learning the 'significant heritage' due perhaps to the tendency described by Dwyer above:

> "One had to cram all this stuff into one's mind, whether one liked it or not. This coercion had such a deterring effect that, after I had passed the final examination, I found the consideration of any scientific problems distasteful to me for an entire year...It is in fact nothing short of a miracle that the modern methods of instruction have not yet entirely strangled the holy

[10] Dwyer, T., "Some Principles for the Human Use of Computers in Education," International Journal of Man-Machine Studies, Vol. 3, 1971, p. 219.

[11] Dwyer, T., "Heuristic Strategies for Using Computers to Enrich Education," Soloworks Paper, University of Pittsburgh, pp. 9-10.

> curiosity of inquiry; for this delicate little plant, aside from stimulation, stands mainly in need of freedom; without this it goes to wreck and ruin without fail...It is a very grave mistake to think that the enjoyment of seeing and searching can be promoted by means of coercion and a sense of duty. To the contrary, I believe that it should be possible to rob even a healthy beast of prey of it voraciousness, if it were possible, with the aid of a whip, to force the beast to devour continuously, even when not hungry--especially if the food, handed out under such coercion, were to be selected accordingly."[12]

In fairness, however it must be pointed out that there may be a large class of learners that when given the choice between authoritarian or self-directed learning will willingly decide that 'teacher knows best'. For example, the author's daughter, a junior high school student, chooses authoritarian classes over self-directed 'independent study' classes in every discipline. It is difficult to say whether this attitude has been conditioned by the present educational system or is inherent in her genetic/psychological makeup.

One of the most vociferous proponents of the self-directed philosophy with a blazing faith in process over content is John Holt. Consider the following review of Holt's position:

> "Holt's educational goal is to turn all learners into 'doers', that is, into self-directed, purposeful people who do meaningful work and lead meaningful lives. Doers, he says, should themselves decide what they will say, hear, read, write, think or dream about. He believes that the best way to accomplish this is to end compulsory education and inform people that they 'cannot have human liberty, the sense of all persons' uniqueness, dignity, and the worth on which it

[12] Einstein, A., "Examining in Harvard College," cited by Paul Goodman, Compulsory Mis-education and the Community of Scholars, (New York: Vintage Books, 1966), p. 6.

> must rest, if you give some people the right to tell other people what they must learn or know, or the right to say officially and 'objectively' that some people are more able and worthy than others."[13]

It is clear, even from this small excerpt, that Holt does not believe that 'teacher knows best', that teacher imposed discipline is wrong and arrogant, and that man in innately good and hence should be trusted.

However, the reviewer of this book, balances Holt's strong statement with one that sides more with the authoritarian view: "Few poeple who are shaping themselves outside the schools are becoming do-ers. Many of them are finding TV, drugs, and the Reverend Sun Moon."[14] Obviously the reviewer is not convinced about the innate goodness of man when the best a self-directed student can do is direct himself to escapist activities such as TV, drugs, and the Reverend Sun Moon. This may imply an authoritarian value judgement on the part of the reviewer.

Papert has long been a proponent of the self-directed philosophy. Based on this philosophy, Papert has designed and implemented computer technologies which foster the self-directed goal, particularly within the mathematics discipline. He has developed a programming language, LOGO, which can be used by elementary school children to move a robot (called a turtle) about while drawing entertaining and educational pictures. Papert is attempting to allow the children to engage in the

[13] Schwartz, S., Book Review of: Instead of Education: Ways to Help People do Things Better, by John Holt, E.P. Dutton's Co.; in Psychology Today, May, 1976, p. 101.

[14] Ibid, p. 101.

same thought processes that mathematicians do when occupied in creative problem solving.[15]

Kline is also a proponent of the self-directed mode in mathematical education, proposing that the inductive method rather than the deductive one be taught so that students can begin to feel and understand what a mathematician does.[16] In the sense, Kline is also proposing process over content in mathematical education.

The authoriatarian viewpoint seems to be almost embedded into the Puritan ethic of western culture which implies that man is basically evil and weak and only thru pain and struggle can he ever hope to become good. However, the humanistic side of the authoritarian viewpoint has been expounded by Sir Geoffrey Vickers.[17] In addressing the question, 'What is culture?', Vickers considers the purpose of education. He compares education with agriculture whose aim is to cultivate and to preserve. In agreement with Dwyer, he feels there is certainly not only a 'significant heritage worth examining' but that heritage must be preserved to be passed on to future generations so that the culture itself can be preserved. No less important is the cultivation of minds so that innovation becomes possible.

Piaget's work can be interpreted as applying to either side in this controversy.[18] One could argue that the learner must have reached the

[15] Papert, S., "Teaching Children to be mathematicians vs Teaching about Mathematics," MIT Artificial Intelligence Memo 249, 1971.

[16] Kline, M., Why Johnny Can't Add: The Failure of the New Math, (New York: St. Martin's Press, 1973).

[17] Vickers, G., at the Division for Study and Research in Education, MIT, Spring, 1976 Colloquium.

[18] Richmond, P.G., An Introduction to Piaget, (New York: Basic Books, 1971).

appropriate stage of development (perhaps the formal stage) before he is able to be a self-directed learner and that before he reaches that stage then authoritarian teaching is absolutely necessary. This is, in fact, how most parents teach their children. When the child is young, the mode of teaching is strictly authoritarian (a parent does not allow his toddler to 'discover' how to cross a busy street) and as the child grows in experience more and more opportunities are afforded to self-directed education so that ultimately the child can leave the parent as a completely self-directed and autonomous person. In this case the issue is not so much authoritarian versus self-directed philosophy so much as when it is approriate to apply either mode of instruction. Perhaps this simple parent-child interaction can be used as a basis to resolve the apparent conflict between authoritarian and self-directed education. Instead of asking the question: 'Who knows best -- the learner or the teacher?', perhaps, instead, we can ask: 'How can we best work together, learning from and teaching each other?' Before this theme is explored and expanded on the basis of a proposed partnership, let us first conclude the discussion of authority and responsibility by examining in more detail, alternatives available within the educational experience.

2.3.2 The Educational Experience

As defined in Section 2.3, an educational experience is one which comprises the three areas of educational objectives, teaching-learning experiences, and evaluation. Let us explore these areas in terms of the authoritaries and responsibilities outlined in Section 2.3.

2.3.2.1 Educational Objectives

We have, in Section 2.3 defined an educational objective as a statement of what a teacher or learner will be like when he has successfully completed an educational experience. The three parts of the educational experience: (1) choosing objectives, (2) choosing appropriate learning-teaching experiences to implement these objectives and finally (3) evaluating the experience can be viewed as management techniques useful for governing any well-defined set of tasks. That is, first we decide what it is we want to do, then we decide how to do it, then we choose an appropriate evaluative method that will tell us (a) how we are doing and (b) when we are done. Therefore, it can be argued that it is absolutely necessary to have a set of goals -- some sort of plan -- at the outset or be prepared for surprising and perhaps unpleasant results.

"I cannot emphasize too strongly the point that an instructor will function in a fog of his own making until he knows just what he wants his students to be able to do at the end of the instruction."[19]

Thus an educational objective is simply a goal that is stated clearly enough so that it can be evaluated at some later time; usually this means that the objective contains criteria that can be easily measured. For example the fuzzy objective, "to be able to appreciate music" contains less measurable criteria than "to be able to identify the composers of the 12 following pieces within 10 minutes..." Thus good objectives are usually written in terms of performance (what the

[19] Mager, p. 3.

individual will be able to do), conditions and restraints under which the individual must demonstrate competence and finally, good objectives contain within their statement the criteria by which they will be evaluated.[20] It is also possible that the clearer the goals are stated, the less valuable they may be. That is, if the objective or plan is too rigidly defined, no room is left for creativity; no opportunity is afforded to pursue interesting and possibly valuable educational side roads. Therefore it must be constantly borne in mind that objectives are similar to other plans and should be subject to change as circumstances change. We shall assume here that it is an effective strategy to have a plan or an educational objective and that one is always prepared to modify it; however one must <u>have</u> a plan before he can change it.

Furthermore, educational objectives here are not as narrowly defined as they are in Mager's book. Not only is it possible to have the learner define the objectives, the teacher can set some objectives or goals for himself. We see that instead of only one possibility there can be four alternatives for choosing the educational objectives within the teacher-learner-relationship:

1) The teacher chooses the objectives for the learner

2) The teacher chooses the objectives for the teacher

3) The learner chooses the objectives for the learner

4) The learner chooses the objectives for the teacher

Although alternative 1 is the only one usually considered in traditional educational experiences, certainly alternatives 2 and 3 are reasonable

[20] Ibid., p. 52.

and worth considering. Although alternative 4 is logically possible it would most likely never come to pass, since, at present, the teacher holds so much of the power within the teacher-learner relationship.

Let us now examine the other two components of the educational experience.

2.3.2.2 Learning and Teaching Experiences

As mentioned in Section 2.3 the learning and teaching experiences consist of making choices for the curriculum (which topics are to be taught and learned) and pedagogy (how the topics will be taught and learned). First we shall examine what other researchers have to say on this subject, then we shall explore the topics of structure, process and content, and heuristics in terms of the authorities and responsibilities associated with the learning and teaching experiences.

Dwyer recommends a high degree of control (authority) by the learner in choosing the learning experience. Dwyer encourages the 'solo' mode "based on a belief in the value of learner control of certain key aspects of his education."[21]

"The word 'solo' describes a pedagogy based on the intensity of involvement and accomplishment that occurs when something is a personal quest: the learning of a student pilot on a solo-flight; the learning of a language student in a foreign country; the learning of a blind person who must internalize the world in a completely unique way."[22]

[21] Dwyer, "Some Principles for the Human Use of Computers in Education," p. 221.

[22] Dwyer, T., "The Fundamental Problem of a Computer-Enhanced Education and Some Ideas about a Solution," SIGCUE Bulletin, Vol. 10, No. 3, July, 1976, p. 16.

With regard to the responsibility of the teacher in the teaching-learning process, Dwyer has this to say:

"Just as it is hard for a flight instructor to resist taking over the controls from a student pilot, it is difficult for a teacher to guide learning in ways different from his own."[23]

Bruner says essentially the same thing:

"...how one manages to instruct without making the learner dependent...is a very complicated question that does not yield easy answers."[24]

The question becomes: should all good teachers experience this conflict which centers on responsibility? It has been the author's experience that the only way to handle this problem is to be constantly aware that it is a problem and to be ready when the student asks for help. If the teacher is too eager to help he may destroy the learning process, and in many cases, the learner only listens with complete attention when he voluntarily comes to the teacher for guidance.

Other areas of responsibility and authority for the teacher and learner include structure, process and content, and heuristics.

2.3.2.2.1 Structure

Just as it is an important responsibility of the teacher to instruct without making the learner dependent, it is also very important for the teacher to provide the learner with insight into the structure of the topics that are to be learned. In simple terms, the teacher must be able to show the student the 'big picture' i.e. the relations

[23] Ibid., p. 18

[24] Bruner, Jerome Seymour, The Relevance of Education, (New York: Norton, 1971), p. 122.

between the topics must be made clear. As Whitehead put it: "The problem of education is to make the pupil see the wood [the forest] by means of the trees."[25] But the teacher is responsible for also providing a local view. Therefore, the teacher, since he usually has the authority to explicate the structure of an area, must be able to make the learner see the forest and the trees.

Given this authority, the teacher has the responsibility to be careful not to impose structure on a set of topics when he can let the learner derive the structure by himself. As Dwyer said:

> "The development of curricula that involve computing systems has to proceed on a basis that is open to new insights. It is important, in particular, to be wary of the "logical" sequence of fixing objectives first, and then developing the curriculum to match. The present vision of changes that can take place in learning when educational technology is properly mastered is too dim to make more than initial estimates of what our goals can or should be. We must, of course, make such estimates, but we must view them as subject to considerable refinement."[26]

Dwyer implies that we should emulate the naturalist who attempts to derive the inherent structure rather than the compulsive scientist who attempts to impose a predetermined model on a particular situation.

Weed also indicates that the responsibility for seeing structure lies primarily with the learner.

[25] Whitehead, A.N., The Aims of Education and Other Essays, (New York, MacMillan, 1967), p. 10.

[26] Dwyer, "Some Principles for the Human Use of Computers in Education," p. 224.

> "The student should see the whole of medicine before he begins to examine any of its parts. As the student learns the details of various parts of medicine, he must always make an attempt to see the relation of this part to the whole."[27]

Bruner is another scholar who has long been concerned with the importance of structure in the learning-teacher relationship:

> "Students, perforce, have a limited exposure to the materials they are to learn. How can this exposure be made to count in their thinking for the rest of their lives? The dominant view among men who have been engaged in preparing and teaching new curricula is tnat the answer to this question lies in giving students an understanding of the fundamental structure of whatever subjects we wish to teach."[28]

He goes on to say: "If earlier learning is to render later learning easier, it must do so by providing a general picture in terms of which the relations between things encountered earlier and later are made as clear as possible."[29] In stressing the importance of structure Bruner gives us a pragmatic definition: "Grasping the structure of a subject is understanding it in a way that permits other things to be related to it meaningfully. To learn structure, in short, is to learn how things are related."[30] And in regard to its importance in the transfer of learning, Bruner says,

[27] Weed, p. 5.

[28] Bruner, Jerome Seymour, The Process of Education, (Cambridge, Mass: Harvard Univ. Press, 1963), p. 11.

[29] Ibid., p. 12.

[30] Ibid., p. 7.

"The teaching and learning of structure, rather than simply the mastery of facts and techniques is at the center of the classic problem of transfer."[31] It is also clear that Bruner feels that the responsibility for showing structure lies with the teacher: "The task of the curriculum maker and of teachers is to give to the student a grasp of this underlying structure..."[32] He makes the above statement in the interest of trying to reduce clutter and to systematize a body of knowledge.

Bruner goes on to relate the importance of structure to the effectiveness of the educational process. "Good teaching that emphasizes the structure of a subject is probably even more valuable for the less able student than for the gifted one, for it is the former rather than the latter who is most easily thrown off the track by poor teaching[33] and, finally he comments on the value of structure in facilitating intuitive thinking on the part of the learner: "...we may ask whether, in teaching, emphasis upon the structure or connectedness of knowledge increases facility in intuitive thinking." Bruner raises this question and then goes on to say that he believes that the understanding of the basic structure of a discipline does lead to increased facility in intuitive thinking and that one of the reasons structure is emphasized in mathematics and physics is so that students will be able to eventually attack problems intuitively thus increasing their effectivness.[34]

[31] Ibid., p. 12.
[32] Bruner, The Relevance of Education, p. 123.
[33] Bruner, The Process of Education, p. 9.
[34] Bruner, The Process of Education, p. 62.

Closing the discussion, we quote Piaget on the relation between structure and intelligence:

> "Intelligence is identical with that type of behavior that consists in the organizing and constructing of rules, patterns and principles. If the functioning of intelligence shows structures, intelligence must also be said to be structured... Sensory stimulation as such is not knowledge and does not lead to knowledge unless there is a structured scheme prepared to assimilate it and accommodate to it."[35]

2.3.2.2.2 Process vs Content

Process vs content has been briefly discussed as an aspect of authoritarian vs self-directed education. It will be discussed again here in terms of the authorities and responsibilities implicit in the teaching-learning experience.

In traditional educational environs, the teacher is given the authority to choose process and/or content as pedagogical modes of educating his students. Since the teacher has all the authority he incurs all the responsibilities leaving the student more or less a spectator in the educational arena. This usually results in the almost overwhelming stressing of content over process by most all teachers because content is easier to teach (it is how most teachers themselves were taught) and is easier to test than process. On the other hand, content knowledge is valuable and in many disciplines absolutely necessary. However, for a teacher to stress process over content or content over process simply as a convenience can be a disservice to the student. Content and process should be used as educational tools where appropriate and this

[35] Furth, Piaget and Knowledge, pp. 179-180.

depends on the context and on the needs of the student. The discussion of this issue is continued by examining the positions of several educational philosophers.

Bruner admonishes the teacher to accept the responsibility that is inherent in his authority by sharing the process of education with the learner:

> "the acquisition of knowledge---is an active process... The individual is best viewed neither as a passive recipient of information nor as a bundle of stimulus-response connections. Rather he should be regarded as an active participant in the knowledge getting process... It is not surprising then that one important feature of Bruner's approach to education is to encourage the learner to participate actively in the process of learning."[36]

In mathematics education, Kline[37] also stresses process over content by advocating the inductive over the deductive approach to mathematics. He indicates that the intuitive inductive approach may not result in perfectly acceptable mathematical proofs but it is how real mathematicians actually do mathematics saving deductive reasoning to formally prove or disprove inductive intuitive premonitions. He feels that deductive logic is close to a content-oriented approach by forcing the learner to memorize axioms that usually appear quite arbitrary and goes so far as to quote the French mathematician Henri Lesbesque: "Logic makes us reject certain arguments but it cannot make us believe any argument." As previously mentioned, Papert is also a proponent of trying

[36] Bruner, Jerome, Seymore, Beyond the Information Given: Studies in the Psychology of Knowing, edited by J.M. Anglin, (New York: Norton, 1973), p. 397.

[37] Kline, Why Johnny Can't Add.

to teach children the inductive processes of mathematics by allowing students to actively engage in the kinds of activities performed by actual mathematicians.[38]

Dwyer has already been quoted as appreciating "the futility of imposing subject content on the student who does not perceive its acquisition as important."[39] He is of course, addressing two issues at once here: the issue of who shall control (authority) and the process vs content issue. It is clear that he favors process over content stressing the process of the learner controlling the computer (via programming).

Piaget has attempted to legitimize the idea that thinking is doing -- that mental processes are actually actions.[40] He has affirmed that knowing must involve action, that one can say little about an object without acting upon it.

Generally, most eastern writers stress that stuffing one's mind with knowledge is not only useless but that it can be harmful to one's spiritual health. "... in a real school, the student must not only be taught various subjects but also helped to be aware of the process of his own thinking."[41] "But you see our education does not teach you <u>how</u> to think [process]; it tells you what to think [content]...it is the function of right education to teach you to think for yourself..."[42] Content accumulation encourages conditioned responses to life situations rather than allowing one to think and discover for oneself--instead we are taught

[38] Papert, "Teaching Children to be Mathematicians..."

[39] Dwyer, "Some Principles for the Human Use of Computers in Education," p. 219.

[40] Richmond, <u>An Introduction to Piaget</u>.

[41] Krishnamurti, J., <u>Think on These Things,</u> (New York: Perennial Library, 1970), p. 83.

[42] Ibid., p. 81.

to rely on what others have said. Not a very effective way to create autonomous beings. (Perhaps a more appropriate place for content-addressable memory is inside computers.)

However process is not an answer unto itself. It must be relevant. "The pupils have got to be made to feel that they are studying something, and are not merely executing intellectual minuets."[43] A process must be in harmony with the educational objectives set by the teacher or learner or it is useless.

The 'discovery' method of teaching is a specific use of process in the learner-teacher relationship. Although it is generally agreed that most students enjoy this process; according to Gagne there is no actual evidence that the discovery method is any more effective than just presenting the content to the student.[44] It is clear, however, that presenting the content material to the student - in effect telling him what is true and telling him what to do - does not teach the student how to learn for himself. What it does teach him, by example, is that in order to learn, one must listen to someone in authority. A student who just absorbs content has abdicted the responsibility for his education and has lost the possibility of becoming an autonomous learner.

2.3.2.2.3 Heuristics

If the teacher feels a responsibility to help the student become an autonomous learner, then he must teach the learner how to learn which involves heuristic reasoning as a valuable component. A formal

[43] Whitehead, "The Aims of Education," p. 15.

[44] Gagne, Conditions for Learning, 2 ed., (New York: Holt Rinehart Winston, 1970).

definition of heuristic is 'serving to discover,' that is, a rule of thumb or any aid to the discovery of a solution to a problem or insight into a process. An operational definition of a heuristic would be: "Heuristic devices don't tell one what to do; they tell one how to learn what to do."[45] Heuristics are, in the language of Zen literature, fingers pointing to the moon, but not the moon itself. A useful heuristic in the game of chess might be 'play for the middle of the board.' A useful heuristic for a band of rabbits seeking a new warren is "go to the top of the hill and have a look around."[46] Heuristics are not like algorithms or formal procedures that are guaranteed to find a solution or provide insight, instead they usually take into account a few of the more important variables in a situation, apply some sort of rule to those variables, and predict a new direction to proceed in the search. In this sense, heuristics are conceptual tools that facilitate learning. A set of heuristics developed by H.A. Peelle at the University of Massachusetts is included in Appendix A.

Using heuristics to teach students how to learn also relates to the process vs content issue since heuristic strategies rely on process itself. Heuristics are techniques that let process itself be a guide to learning or to discovery. For instance, consider the process of beginning --- as in beginning to solve a problem --- which involves heuristics. The learner need not have a formal rigid plan or possible solution in mind before he begins. Surely all he need do is begin--

[45] Dwyer, "Heuristic Strategies for Using Computers to Enrich Education," p. 8.

[46] Adams, R., Watership Down, (New York: Avon, 1976).

begin somewhere. The insights gained and questions raised at any stage in the learning process may serve to guide the learner to the next stage of learning. Traveling from state to state in this manner, the learner is not only learning, he is learning how to learn. This process itself is a most useful heuristic for learning, that is to begin--to start somewhere and do something. The very action of doing starts the mind working; and possible solutions, questions, and insights may come pouring in. Action primes thinking and learning, which in turn primes action.

The decision to start and to proceed is leap of faith on the part of the learner that should be fostered by the teacher within the teacher-learner relationship. One particularly powerful method for fostering this faith is for the teacher himself to act as a model by actually demonstrating this behavior to the learner whenever possible.

Programming a computer is another method that the teacher can make available to the learner in order to nurture this attitude towards learning. This aspect will be discussed in Chapter III.

2.3.2.3 Evaluation

As mentioned in Section 2.3 the evaluation process has three parts:

1) the evaluation of the teaching
2) the evaluation of the learning
3) the evaluation of the objectives

The evaluation of the learning experience is usually performed by testing; the evaluation of the teaching experience is made by learner ratings of the teacher; the evaluation of the objectives is usually performed by the teacher.

2.3.2.3.1 Definitions

Before discussing the authorities and responsibilities inherent in the evaluation process, let us define some terms. We shall define 'evaluation' as a value judgement placed on an educational measurement. An educational measurement is usually defined as a mapping that assigns rational numbers to individuals; that mapping will be called a 'test.' If the numbers relate to what the students know then they measure the validity of the test (is the test measuring what it is supposed to be measuring?); if the numbers are consistent across different test applications, then that is a measure of the reliability of a test, (would the test-taker score the same on a given test at different times, on parallel forms, or if it were scored by different graders?). Hambleton has given a humorous example of the relationship between reliability and validity - he describes a cartoon of a small boy standing on a scale and saying "I am 30 pounds tall." The measurement of course, very reliable but invalid.[47]

Of the many types of tests extant, two will be referred to here: norm-referenced and criterion references tests.

A norm-referenced test is based on frequency distribution of test scores. Using this distribution, grades are assigned to test scores by assigning cutting points for each grade level.[48] Grading on the curve

[47] Hambleton, R., in a lecture for "Principles of Educational and Psychological Testing - ED 4530," at University of Massachusetts, Amherst, Summer, 1975.

[48] This is usually considered to be an evaluative procedure since test scores are a measurement but the conversion to grades (ABCDF) attach value judgements (in the choosing of cutting points) to those scores.

in this manner assigns grades relative to the scores of all others taking the test. It allows a ranking of test scores; and it allows one to make statements like, "Student X scored better than 80% of all those taking this test."

However, norm-referenced tests pose problems to a program of individualized instruction. It would not make sense to blindly apply norm-referenced tests which are used primarily to indicate relative differences between individuals, since one of the premises of individualized instruction is that "students differ in interests, motivation, learning rate, goals, and capacity for learning."[49] Hambleton goes on to point out that since norm-referenced test procedures are, in large part, irrelevant to the goals of individualized instruction, "An appropriate set of testing methods and decision-making procedures would facilitate the efficient movement of students through an instructional program. Norm-referenced tests are constructed specifically to faciliate making comparisons among students; hence they are not very well-suited for making most of the instructional decisions required in individualized instructional programs."[50]

Criterion-referenced tests attempt to solve this difficulty. Since they are used primarily within an environment of individualized instruction, "the pertinent question is whether or not the individual has obtained some prescribed degree of competence on an instructional performance task."[51] Criterion-referenced tests attempt to take educational

[49] Hambleton, R., "A Review of Testing and Decision-Making Procedures for Selected Individualized Instructional Programs," School of Education, University of Massachusetts, p. 1.

[50] Ibid, p. 3.

[51] Ibid., p. 4.

objectives and measure their attainment in terms of a mastery level of performance on the part of the individual. The goal is for the individual to demonstrate competency of the objectives by some criterion (usually performance regarding skills or content knowledge).

> "A very flexible definition of a criterion-referenced test has been proposed by Glasser and Nitko (1971): '... a test that is deliberately constructed so as to yield measurements that are directly interpretable in terms of specified performance standards' (p. 653)."[52]

A useful distinction between the two types of tests is that norm-referenced tests are primarily used within the context of the traditional educational experience where time is held constant (e.g., one semester) and the total amount of skills and material learned is variable across the student population--some students learning more than others within the fixed time frame: "Our system of education is based on groups of people all the same age going through classroom programs at the same rate of speed--some learning more than others," says Dr. Richard Evans."[53]

On the other hand, criterion-referenced tests reverse the priorities so that the amount of learning is made constant (mastery of the objectives) while time is allowed to become variable (an individual student can take as long as he needs to master the objectives): "Evans said if schools were reorganized "according to the way kids learn -- allowing the time for learning to vary -- we could prevent much of the failure which is evident in schools today."[54]

[52] Ibid, p. 4.

[53] Evans, R., News Item (AP), Amherst Bulletin, Wed., Dec. 1, 1976, p. 24, col. 1.

[54] Ibid.

So, while norm-referenced tests produce results that are designed primarily to assign relative rankings to individuals, criterion-referenced tests produce absolute results--either the individual has mastered the competencies measured by the test, or he has not (yet). There is of course a subjective element in creating a representative criterion-referenced test and in choosing cutting scores (below which the student is evaluated into the non-mastery category) but once their are chosen a criterion-referenced test is usually a more useful instrument, especially within an individualized program of instruction.

2.3.2.3.2 Justification

Now that certain terms have been defined let us next consider the question, "Is evaluation necessary?". In a pragmatic sense, evaluation is necessary perhaps because the educational process is embedded within a largely Puritanically-oriented society which is compelled to constantly pass judgement. "The proof of the pudding is in the eating." No one, says instead, "Here is the pudding--either it will be eaten, or not--there is no need to pass judgement on it." Unfortunately this attitude in education has been carried to the extreme where the adage is interpreted as "the proof of knowledge is in the regurgitation." So, within the current societal environment evaluation is, seen at its worst, a necessary evil. However, at its best evaluation can provide valuable feedback on the validity and effectiveness of the teaching-learning process and the educational objectives. Also evaluation usually implies testing and testing can be a powerful motivation for learning; learning can take place not only in preparation for a test but during the test itself.

Although testing is usually controlled by the teacher-authority, it is possible that control of testing can be shared by the learner. Once the choice of who controls the testing is made, then the responsibilities must be considered. Whoever designs the tests must not only be concerned with the traditional goals of reliability and validity[55] but must also be responsible to the other partner. An agreement must be reached as to what constitutes a test with the understanding that a well-designed test cannot only motivate learning but serves to classify and refine the educational objectives[56] and the processes involved in the learning-teaching experience. A flow chart illustrating this process is shown on the next page.

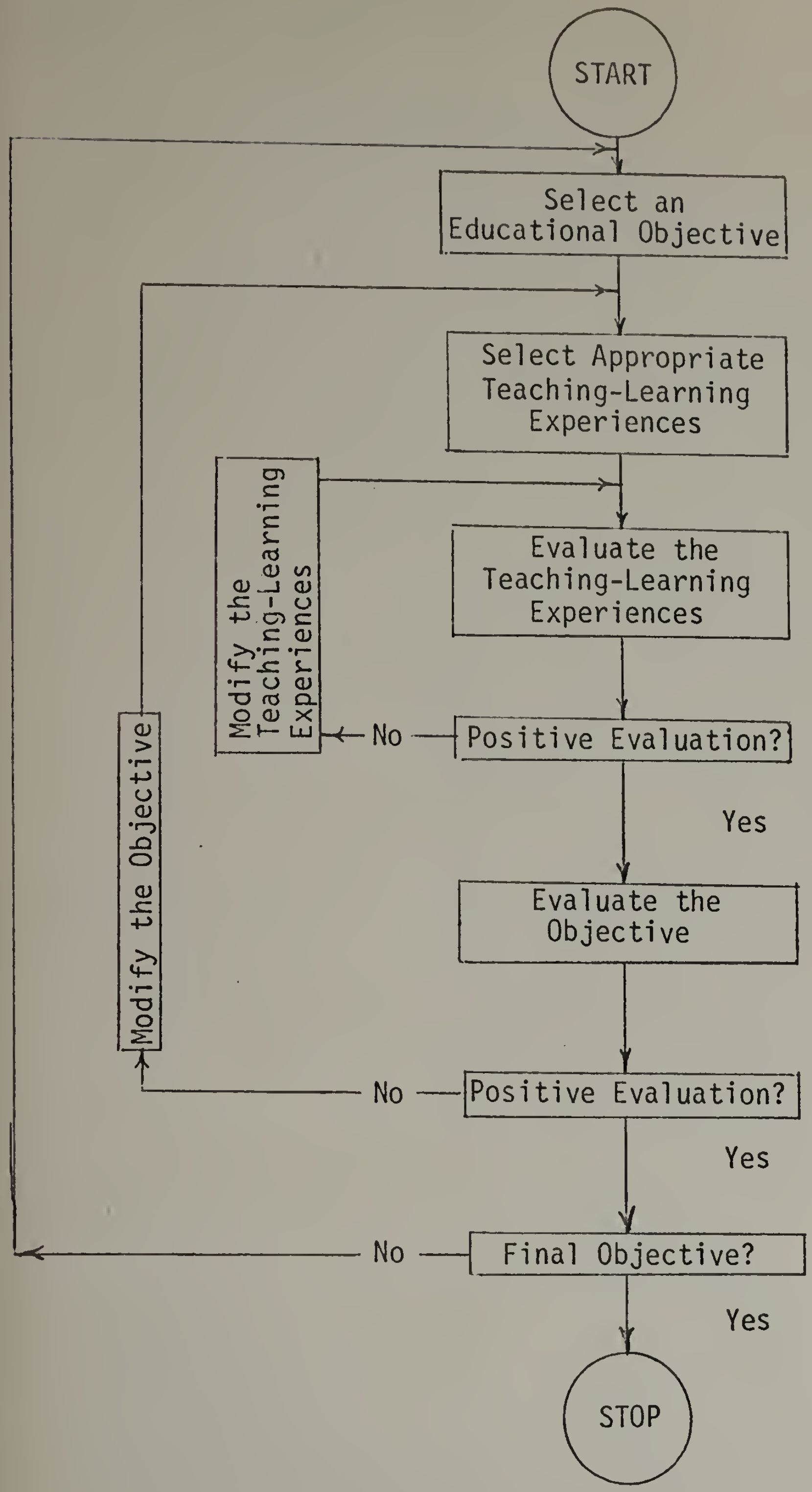
Thus we see that evaluation can be a valuable process for determining to what extent the goals or educational objectives have been reached. If they have been achieved, all is well, if not then either the teaching-learning experience designed to actualize an objective must be modified or the objective itself should be redefined. In sum, evaluation can be viewed as a useful management tool within the teacher-learner relationship and, the testing of the learner's knowledge can be thought of as an evaluation of the teaching-learning process.

2.3.2.3.3 Process vs Content

Process vs content can be considered as an ingredient of evaluation of teaching and learning. Although within most curricula in undergraduate education, process is not even considered as a possible mode

[55] Stanley, J.C. and Hopkins, K.D., Educational and Psychological Measurement and Evaluation, 5ed., (Englewood Cliffs, N.J.: Prentice Hall, 1972), pp. 101-133.

[56] Ibid, pp. 7-8.


START
Select an Educational Objective
Select Appropriate Teaching-Learning Experiences
Evaluate the Teaching-Learning Experiences
Positive Evaluation?
No
Modify the Teaching-Learning Experiences
Yes
Evaluate the Objective
Positive Evaluation?
No
Modify the Objective
Yes
Final Objective?
No
Yes
STOP

of evaluation, it certainly could be. A typical driver's test employs both content and process in its administration -- usually a multiple choice content test is followed by an actual test drive. It would certainly be unreasonable to ask beginning civil engineering students, for example, to actually construct a bridge as a process test, but that does not rule out process testing in what are usually though of as knowledge areas. Thinking and problem-solving are mental processes; however, very few teachers give their students tests designed to measure and evaluate these processes. The reason is simple--they don't know how to. Few even attempt to try since the problem is so very difficult. Bloom, however, has addressed this problem and has made some recommendations.

> "Much of the psychological research attempts to infer from the observed behavior of the individual what the mental processes must have been...
>
> Much of the work in this field is based on the use of various types of tests. In these tests the subject is presented with questions which he is to answer and problems which he is to solve. After some mental effort, the subject submits his answers or solutions for appraisal by the tester. In addition, the tester may secure observations of the behavior of the subject while taking the test. From these data the tester will attempt to draw conclusions about the mental characteristics of the subject. What is missing is information on the process [emphasis mine] by which the problems are solved. The methods of attack, the steps in the thinking process, the kinds of considerations used to make one choice rather than another, and the feelings and attitudes of the subject are neglected or given very little attention. The products of thought--the answers to the questions or the solutions to the problems--plus the observations may give the tester a fleeting glimpse into the complex processes of thought involved, but usually this is incomplete and, almost of necessity, inaccurate."[57]

[57] Bloom, B.S., and Broder, L., Problem Solving Processes of College Students, Univ. of Chicago Monograph--out of print (1950)--p. 1.

Next Bloom extends his argument from the psychologist to the educator:

> "The educator, as well as the psychologist, assumes a correspondence between mental products and processes. He usually assumes that the individual who has the largest number of correct responses on the final examination is the individual with the best or most desirable quality of mental processes. Even in classroom discussions, the teacher is generally more concerned about the accuracy of responses than about the methods by which the student arrived at his responses or solutions. In large part, this preoccupation with accuracy of responses is a consequence of the difficulties of getting information about the processes of thought, the limitations of time, the large size of classes, and the pressure for coverage of subject matter. In spite of this emphasis on the products of thought, educators usually agree that good habits (or processes) of thought are the important and significant outcomes of education. Also, they would probably agree that the particular solutions or answers given to schoolroom problems are of little consequence except insofar as they serve to indicate the quality of the student's thinking."[58]

Bloom then indicates that the educator and psychologist have three alternatives in dealing with this problem:

1) Leave things as they are where primary attention is given to the products rather than the processes of thought.

2) Design and perform new experiments which will allow relationships to be established between the processes and the products of thought so that products can continue to be used and the relationships will point to the actual processes.

[58] Ibid, p. 2.

3) Develop new techniques "which will make possible the securing of evidence on both the processes and the products of thought."[59]

Bloom holds little hope with the first two alternatives and recommends the third approach as most promising even though it is by far the most difficult and challenging. Implementing the third alternative "may also require a change from large-scale testing and mass studies to those which involve small numbers of subjects studied by rather intensive techniques."[60]

Thus it is clear that although process testing is considerably more difficult then traditional content testing, the rewards could be enormous--the teacher could tell not only if the student 'got' the correct answer and how he got it but also if the answer were incorrect, he could presumably tell why and how the learner went astray and take appropriate steps to correct the situation.

In closing this section, it is well to remember what the psychologist William James had to say on evaluation:

> "Be patient, then, and sympathetic with the type of mind that cuts a poor figure in examinations. It may, in the long examination which life sets us, come out in the end in better shape than the glib and ready reproducer, its passions being deeper, its purposes more worthy, its combining power less commonplace, and its total mental output consequently more important."

[59] Ibid, p. 4.
[60] Ibid, p. 4.

2.4 A Proposed Solution -- The Partnership

It has been previously stated in Section 2.2 that the crux of the problem in the teacher-learner relationship is that, in many cases, the teacher and the learner act as if they were unrelated, closed systems with no mutual goals. However, if an environment can be afforded which encourages the relationship to be seen as a partnership, then perhaps the teacher and learner can begin to work together. This new partnership can foster an attitude which facilitates teaching and learning and makes the relationship more enjoyable -- less that of adversaries and more of associates working together to achieve common goals.

In Section 2.3 we discussed some of the responsibilities and authorities inherent in the teacher-learner relationship in terms of Authoritarian vs Self-directed views of education and within the educational experience. Let us review these within the context of the proposed solution.

2.4.1 Authoritarian vs Self-Directed Education

The issue between authoritarian and self-directed education centers on the question: 'who knows best--teacher or learner?'. Each side of the issue can be represented by an extreme viewpoint: the authoritarian philosophy contends that the learner knows all and the teacher nothing. These are extreme views. Obviously the teacher knows some things that the learner does not yet understand and vice-versa. It is the author's contention that only by allowing an open relationship to develop -- a partnership between learner and teacher -- can effective and rewarding education take place.

2.4.2 The Educational Experience

The educational experience implies the choice of educational objectives which, once defined, lead to a specific set of teaching-learning experiences and to an evaluation of those experiences in terms of the objectives (the objectives themselves can also be evaluated as shown in the flowchart on page 41). The proposed partnership involves defining the authorities and responsibilities of the teacher and learner in terms of these three areas: educational objectives, learning-teaching experiences, and evaluation. With regard to authority (control) there exist eight alternatives where the teacher and learner completely control choices in the above mentioned three areas:

Who chooses educational objectives	Who chooses teaching-learning experiences	Who chooses the evaluation methods
1) Teacher	Teacher	Teacher
2) Teacher	Teacher	Learner
3) Teacher	Learner	Teacher
4) Teacher	Learner	Learner
5) Learner	Teacher	Teacher
6) Learner	Teacher	Learner
7) Learner	Learner	Teacher
8) Learner	Learner	Learner

For instance, the first option represents the strict authoritarian mode of education while the eighth option is freely self-directed.

There exist only eight options when the three areas are treated as Boolean variables that can only take on two values: 'complete control' or 'no control' with no possibility of a value in between. However, in a real-life situation there is a great deal of room for negotiation between the two partners to find a point somewhere between 'no control' and 'complete control.' Therefore let us extend our representation from the binary form with only eight alternatives to one that allows a continuity along the three dimensions of:

1) Educational Objectives:

1——————————————————————————1

Teacher Controls ... Learner Controls

2) Teaching-Learning Experiences

1——————————————————————————1

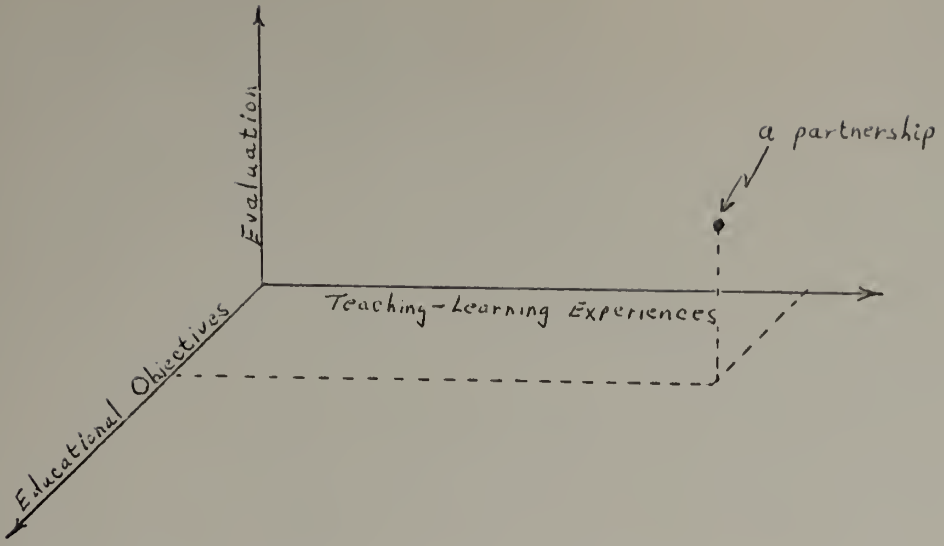
Teacher Controls ... Learner Controls

3) Evaluation

1——————————————————————————1

Teacher Controls ... Learner Controls

and allow the teacher and learner to negotiate a point on each scale that is appropriate to their particular partnership. If these scales are placed as orthogonal axes, then the partnership reached by a particular learner-teacher pair can be viewed as a point in 3-space:



The zero point of each axis represents complete teacher control of all three variables while positive extension along the axis is proportional to increased learner control. The above example partnership is one in which the learner has more control than the teacher in choosing the teaching-learning experiences but the roles are reversed for choosing the educational objectives and the evaluation. Of course, more control or authority in the choices made along a particular dimension inplies more responsibility. Whoever has the authority to define the majority of the educational objectives must be responsible for their relevance and their sharp definition so there will be no misunderstanding later. Whoever has the majority of authority in controlling the teaching-learning experiences must be responsible for their relevance to the educational objectives and for their effectiveness. Whoever is primarily responsible for the evaluation must be especially careful: any tests used must be reliable and valid, and if both partners agree, the evaluation should be a useful tool for feedback on the effectiveness of both the objectives and the teaching-learning experiences.

No attempt is made here to quanitize these scales since the particular point that a teacher-learner pair chooses is arbitrary. The important consideration is that some point be negotiated as a starting place and that each partner be willing to renegotiate that point to reflect the changing nature of the partnership with time. Once a particular option has been agreed upon by both partners and the authorities and responsibilities clearly defined, the educational experience, as a partnership, can begin.

CHAPTER III

THE HUMAN - COMPUTER RELATIONSHIP

3.1 Chapter Overview

This chapter will describe and develop certain aspects of the human-computer relationship as a context for understanding the ACCOLADE system. Just as the previous chapter proposed that a most useful teacher-learner relationship would be a partnership, this chapter proposes that a reasonable and effective human-computer relationship would also be a partnership. The patnership is described in terms of a set of conceptual tools that each partner can offer the other. When the computer supplies these tools to a human, a process called "intellectual augmentation" (IA) takes place; when the human offers these tools to a computer "artificial intelligence" (AI) can take place.

3.2 Introduction--Conceptual Tools

The previous chapter dealt with some of the problems inherent in the typical teacher-learner relationship and proposed a partnership between the teacher and learner as a possible solution to those problems. Since ACCOLADE is a computerized system for acquiring computer literacy, this chapter will discuss some of the assets and liabilities incurred when one uses a computer to assist in the teaching-learning process. There is an implicit premise to this chapter that it is reasonable and effective to actually use a computer in a curriculum that attempts to teach computer literacy. It is almost axiomatic that one can learn only so much by reading and thinking, for example, about how to ride a bicycle--at some point one must actually mount the bicycle and begin to learn to ride it. This learning-by-doing approach is an axiom embedded within the ACCOLADE system. Rather than building a case for a partnership as was done in Chapter II, we shall begin with the premise that a good, effective human-computer relationship within a curriculum for computer literacy is one of a partnership. We shall begin by defining what we mean by a partnership between man and machine followed by a more detailed discussion of the conceptual tools each partner can offer the other.

Usually the relationship between man and machine is assumed to be in the master-slave paradigm rather than a partnership, but the computer is a very special kind of machine: The computer can be understood as a 'universal machine' in the sense that it can simulate most any other machine. It can be an airplane flight simulator, a rapid transit system or a rocket ship. It can also simulate organic systems ranging from a single cell to a society of people. Not only can the computer afford insights into organic and inorganic systems, it is a media (m) in itself, it can draw, animate, print, compose and play music so that:

> "The computer, viewed as a medium itself, can be all other media if the embedding and viewing methods are sufficiently well provided. Moreover this 'new medium' is active--it can respond to queries and experiments--so that the messages may involve the learner in a two-way conversation. This property has never been available before except through the median of an individual teacher. We think the implications are vast and compelling."[1]

[1] [No Author cited], Personal Dynamic Media, Xerox Palo Alto Research Center, Learning Research Group, Palo Alto Calif., Feb., 1976, p. 4.

It is clear that the computer has much to offer -- indeed it has the potential to become a companion to man, and as a companion or partner has much to give to a man-machine relationship. But what is the coin of exchange? In the teacher-learner relationship, the coin was minted from the alloy of authority-responsibility. Although control is certainly an issue in the human-computer relationship, the basis for exchange that will be discussed here centers on the idea of "conceptual tools."

Simply and pragmatically defined, a conceptual tool is a tool that makes the user smarter. When the computer offers man conceptual tools such as computer assisted instruction (CAI), computer managed instruction (CMI), programming languages for generalized problem solving and other mechanisms for organizing and seeking out information, and this can be called intellectual augmentation (IA). Conversely, when man augments the capabilities of the computer by providing it ways to represent knowledge or techniques for problem solving he is, in effect, making the computer smarter. This process has been systematized into a branch of computer science called Artificial Intelligence (AI). Therefore we shall refer to the flow of knowledge from man to machine as AI and the reverse flow from machine to man as IA. This flow is the basis for the human-computer partnership and will now be described in terms of the conceptual tools that effect that partnership and are the basis of the ACCOLADE System. The conceptual tools of AI that will be described are heuristic search techniques and knowledge representations; the conceptual tools of IA are programming languages, Computer Assisted Instruction and Computer Managed Instruction systems.

3.3 Artificial Intelligence (AI)

As previously noted, AI is a new and growing field in computer science. It includes the study of problem solving techniques (including problem representation, state-space description, and heuristic search), representation of knowledge, pattern recognition (including scene analysis), natural language processing, learning systems, automatic theorem proving, and robotics.[2,3]

The components of AI which directly influence the design and implementation of ACCOLADE are: (1) problem solving and (2) representation of knowledge. Let us first investigate the branches of problem solving technique and theory that will be useful to ACCOLADE.

3.3.1 Problem Solving

Within the field of AI, several problem solving techniques have been developed. One of these techniques uses a specialized programming language to describe the problem and to transform goals into subgoals, thus solving problems by the problem reduction or successive simplification method.[4] Other techniques apply state space frameworks[5] to represent knowledge and solve problems. Since ACCOLADE utilizes this latter approach both in philosophy and design, state space frameworks and heuristic search strategies will be described in this section.

[2] Course outline for COINS 783 Artificial Intelligence, at University of Massachusetts, Amherst, Spring, 1976.

[3] Nilsson, N.J., Problem-Solving Methods in Artificial Intelligence, (New York: McGraw-Hill, 1971), p. 2 and p. 9.

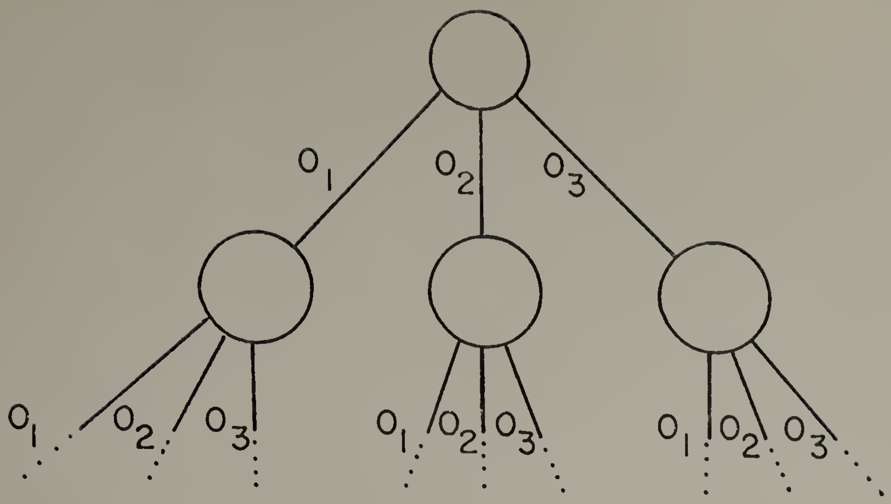
[4] Newell, A. and Simon, H., Human Problem Solving (Englewood Cliffs, N.J.: Prentice-Hall, 1972), p. 414.

[5] Nilsson, pp. 17-79.

> "A state space is all possible states which could be encountered in the search for a solution to a problem. A state is a particular situation or configuration encountered in a problem. Each state in the state space is represented as a node. (A node is a representation of a state.) Nodes are connected by operators which transform one state into another. An operator is a procedure which describes precisely how a given state is changed into a new state.
>
> The initial statement of a problem is represented by the initial node and the solution is represented by the goal node. The objective is to apply operators in the proper sequence which generates a path from the initial node to the goal node. The entire state space, then, is represented as the complete set of all nodes which can be generated using the operators given, or alternatively, by the initial state and the set of operators.
>
> The diagram of a state space looks like a tree (inverted so that the initial node is at the top and is called the root node), with branches at the nodes. A circle is commonly used to designate a node and a line connecting two nodes to designate an operator. If an operator transforms a state into a state which is already represented as a node somewhere else on the tree then that representation is a graph; in this material however, we will be concerned only with searching a special kind of graph--namely a tree."[6]

For example, the following diagram can be used to represent a state space with three operators (0_1, 0_2, and 0_3):

[6] Denenberg, S.A. and Peelle, H.A., "An Introduction to Search Procedures for Problem Solving -- Using APL Recursive Programs," University of Massachusetts, School of Education Monograph--draft, June, 1976.



Thus, if the problem is amenable to state space representation we have furnished the computer with a very powerful conceptual tool with which to solve problems. Once we have represented the problem symbolically as a tree in state space with a certain initial node and a certain set of operators, we can program the computer to search the tree for a goal node or the solution to the problem.

The art/science of searching the tree of possibilities is under constant study within the AI field of problem solving; the goal is generally to find techniques which minimize the time (computer time) and space (computer memory) consumed during the search to a goal node. There are several exhaustive search procedures which will guarantee a solution if one exists. These include breadth-first search (across the tree proceeding from the root node level to deeper and deeper levels of the tree until a solution is found) and depth-first search (down the tree, applying the same operator to generate new nodes to some depth bound at which point the search is backed up to the last node investigated and the next operator is applied ... repeating this procedure to greater and greater depth bounds until a solution is found).

However, while the breadth-first and depth-first search techniques are exhaustive (they will eventually uncover all possible nodes in the state space) they can be very costly in terms of computer time for large problems, i.e., those with many nodes in their state space representation. What is needed is some sort of strategy that will prune the tree and suggest the most promising directions for search. This strategy is called heuristic search and relies heavily on the idea of a heuristic. (A heuristic is a "rule of thumb" serving to guide the search toward solution of a problem.) However, heuristic search is more precisely defined within the field of AI. The heuristics themselves are formulated mathematically in terms of a heuristic evaluation function which can be computed for each node as it is generated. Those nodes with high heuristic values are searched before those with lower values. If the heuristic evaluation function is good, then the total number of nodes investigated and the corresponding time of investigation will be substantially reduced. In short, a heuristic search procedure attempts to include any information about the problem that would minimize the time and effort put into the search--it is itself a valuable conceptual tool.

Dreyfus[7] however has made a strong case that human beings do not in fact solve most problems by searching through a state space for a goal

[7] Dreyfus, H.L., What Computers Can't Do: A Critque of Artificial Reason, (New York: Harper Row, 1972).

node and that the heuristics people use are not the mathematical evaluation functions that direct the search in an AI computer program. While this may or may not be true, it is not the issue that will be discussed here. What is important is that now people can use this extremely powerful problem solving technique thanks to the progress of AI. Man now has at his disposal a way of representing and working through a problem that was not previously available. In this sense, an advance within the field of AI has increased the intellectual capabilities of man and, as such, can be considered to be a conceptual tool under IA. It is interesting to note that the fear that computers will become more intelligent than man is considerably lessened by this argument: once man has provided the computer with a conceptual tool to make it smarter, he has automatically added that same tool to his own problem solving repretoire--hence bootstrapping his own intellectual capabilities.

The state space search technique also helps to clarify and enrich the usefulness and power of heuristics as a problem solving technique. As we mentioned in the previous chapter, heuristics allow process itself to be the guide to solution. One of the most important things one can do to solve a problem is to begin the search. The insights gained and questions raised at any node in the state space can be heuristically evaluated so that further choices can be made. It will be seen in Chapter V that the design of ACCOLADE is in part based on the premise that it is the user (learner) himself who is in the best position to guide his search for a solution to his problem (in this case the problem is to acquire knowledge about particular computer literacy topics). The learner is aided in his search by a Computer Map that can be viewed as a conceptual tool having its roots in the problem solving technique of AI. This Map can also be understood as a particular representation of knowledge. Representation of knowledge is the next conceptual tool of AI to be discussed.

3.3.2 Representation of Knowledge

How to represent knowledge is currently one of the most challenging issues in the field of AI and is confronted by the ACCOLADE system. The basic epistemological question, "what is knowledge?" will not be addressed here since it is much too borad. However, Drucker's definition: "knowledge is the systematic organization of information and concepts"[8] will be useful as an operational definition (without operational definitions of "information" and "concepts"). Organization (or "structure") is the key word with regard to knowledge--some pattern must be discernable to earn the label of "knowledge".

Within AI, there are currently two methods for representing knowledge: the procedural representation and the declaractive representation.

3.3.2.1 Procedural Representation

The Procedural representation of knowledge has been described by Minsky,[9] Hanson and Riseman,[10] and Winograd,[11] in detail. For the purposes of this dissertation we shall view a procedural representation of knowledge generally as a set of procedures (which could be computer

[8] Drucker, P.F., The Effective Executive, (New York: Harper Row, 1967), cited by Engelbart, D., et al., in "The Augmented Knowledge Workshop," American Federation of Information Processing Societies (AFIPS), Vol. 42, June 4-8, 1973, p. 9.

[9] Minsky, M., "A Framework for Representing Knowledge," M.I.T.: Artificial Intelligence Memo No. 306, June, 1974.

[10] Hanson, A.R., and Riseman, E.M., "The Design of a Semantically Directed Vision Processor," University of Massachusetts COINS Technical Report 75c-1, Feb., 1975.

[11] Winograd, T., "Five Lectures on AI," Stanford AI Lab, AIM Memo No. 246, Sept., 1974.

programs) which by itself can search a set of data for the presence of certain patterns; if a match is found then the content and/or structure of that data set is usually modified. For example, in a large computer program that attempts to simulate "vision", certain subprograms may scan the data in a picture for the presence of a feature such as "the horizon." Once the horizon is found, the data representing the horizon might be changed to strengthen that interpretation; e.g. it might be made perfectly straight where before it was slightly jagged. In this sense, the subprogram that recognizes horizons in a procedural representation of the horizon.

3.3.2.2 Declarative Representation

The declarative representation of knowledge is closer to the conventionally accepted meaning of knowledge, and has been more commonly implemented on computers. A declarative representation can be symbolized by a statement such as "All cows have four legs" and can be stored within a computer as data rather than as a program. These data can be represented hierarchically similar to the tree structure used in state space representation or more generally in a graph so that associations between the data can be richer.

The declarative representation of knowledge, within the context of an educational system like ACCOLADE offers two possible advantages over the procedural representation:

1) the structure may be revealed more easily to the user than the procedural method which requires an understanding of algorithms and procedures in order to "see what's happening." Many people, however, can understand structured data. Brunnstein and

> Schmidt also favors this method arguing that if the learner must supply his own procedures for searching out knowledge then this would seem to be useful for forming his own internalized mental model of the knowledge and its structure.[12]

In short, the procedural representation is useful for supplying knowledge to an artifically intelligent program while the declarative representation makes knowledge more easily available to the user-learner.

2) From a system builder's viewpoint, it is usually easier to implement a data base for the declarative representation of knowledge than it is to create a network of computer programs for the procedural representation since the "intelligence" is supplied by the user rather than by the computer programs themselves.

With the above in mind, we will next describe two commonly used forms of the declarative representation of knowledge -- hierarchical and heterarchical.

3.2.2.2.1 Hierarchical Representation

A hierarchical structure for representing knowledge or information is similar to a topical filing system or a card catalog in a library. Major topics (categories or headings) are broken down into subcategories within each major category. These subcategories are then broken down again until the level of detail is fine enough to encompass all of the information in the total system. In the instance of a library, a hierarchical structure such as the Dewey Decimal System or the Library

[12] Brunnstein, K., and Schmidt, J., "Structuring and Retreiving Information in Computer-Based Learning," International Journal of Computer and Information Sciences, Vol. 2, No. 2, June, 1973.

of Congress Category Codes may be imposed so long as every resource (books, magazines, maps, records, etc.) in the library can be assigned some unique category code.

Yellow Pages are another common example of a hierarchical structure. In this case, the resources to be classified are consumer products and services. The local telephone company supplies the hierarchical structure, and if the taxonomy is a good one then any product or service in the area can be assigned a particular category heading. Also, this dissertation itself is organized hierarchically and this section (3.2.2.2.1) is an illustration of a particular labelling scheme.

There are several disadvantages to using a hierarchical structure to represent information or knowledge. Category codes, no matter how carefully chosen will most likely change, especially in such a dynamic field as computer literacy. Category codes are usually arrived at inductively: after looking at the mass of information carefully and for a long enough period of time, the categorizer arrives at a set of categories or generalized pigeon holes which will span the set of information and, if well chosen, will be compact and concise. A set of category codes that is as large as the set of information to be classified is of no value whatsoever whereas too few can be frustrating for the user. It is a usual circumstance that a given set of category codes becomes outmoded or outdated after a certain period of time. Category codes must be deleted, new ones created, some expanded, and others combined. (The author must add that this is not a serious disadvantage so much as a fact of life to be accepted when attempting to impose structure on chaos).

A second disadvantage to the hierarchical structure is possible replication of information. The same information may appear under different categories. For example, in a library card catalog, the information describing a book on Computer Assisted Instruction (CAI) may appear under both the headings "Computers" and "Education." This example illustrates only the possibility of duplication of information; in more severe cases the information may be replicated many more times-resulting in wasted space and wasted time on the part of library employees. This could become an important consideration when the time comes to computerize the information -- precious space will be wasted and this translates directly into dollars.

Perhaps the most important disadvantage to hierarchical structures for representing information is that only the hierarchical structure of the information is illuminated. There is only one relationship that is implicitly shown between a category and its subcategories in a hierarchical structure: that is, the 'general-specific' relation. For example, in the Yellow Pages, when 'George Washington Federal' appears under the category of 'Banks,' all the user can surmise is that 'Bank' is the generic term for 'George Washington Federal' and conversely a specific instance of a bank is 'George Washington Federal.' Not apparent is the connection between 'George Washington Federal' and the category 'Automobiles' if 'George Washington Federal' issues automobile loans. Certainly not shown is a list of automobile loan interest rates for each bank under the category 'Banks'--very useful information for anybody in the market for a car.

The three above-mentioned disadvantages associated with hierarchical structure information schemes are largely overcome by the use of particular heterarchical structures called semantic information networks, and these will be discussed next.

3.2.2.2.2 Heterarchical Representation

The term "heterarchical" was coined by Minsky and Papert[13] to delineate the contrast to hierarchical structures and deals more to the procedural representation of knowledge than the declarative representation. Implied is that basic building blocks of a knowledge framework, whether they be represented as data or procedures, are not necessarily arranged in some hierarchical form but, rather, the building blocks are to have equal status, (perhaps a more useful terminology for this representation might be "equ-archical").

Now if the knowledge is declaratively represented as heterarchical topics or nodes in a knowledge space, what mechanism is used to organize the topics? Some organization or structure must be present to meet our operational definition of knowledge. One way to structure the topics is to associate them by means of a set of relationships--when this is done, that structure is called a semantic information network.

Semantic Information Networks

Semantic information networks or "semantic nets" evolved as attempts to model the associative way knowledge is perhaps stored and retrieved in the human mind. This representation has been described by

[13] Minsky, M., and Papert, S., MIT: "Artifical Intelligence Report," Jan. 1, 1973.

Quillian[14] and applied by Carbonell[15] in a computer assisted instruction system called Scholar which, using a semantic net to represent knowledge, can answer as well as ask questions about a particular knowledge space -- geography in this instance. Wexler[16] and Koffman[17] have also designed and implemented similar systems based on semantic nets that are able to interact with student users.

As mentioned above, semantic nets are a declarative representation of knowledge consisting of a set of relationships between a set of topics. A convenient symbolism for a semantic net is a graph (briefly described in the state space search discussion) where the topics are nodes and the relations between the topics are arcs. A semantic net can be understood as a graph representation of topics with relationships between these topics.

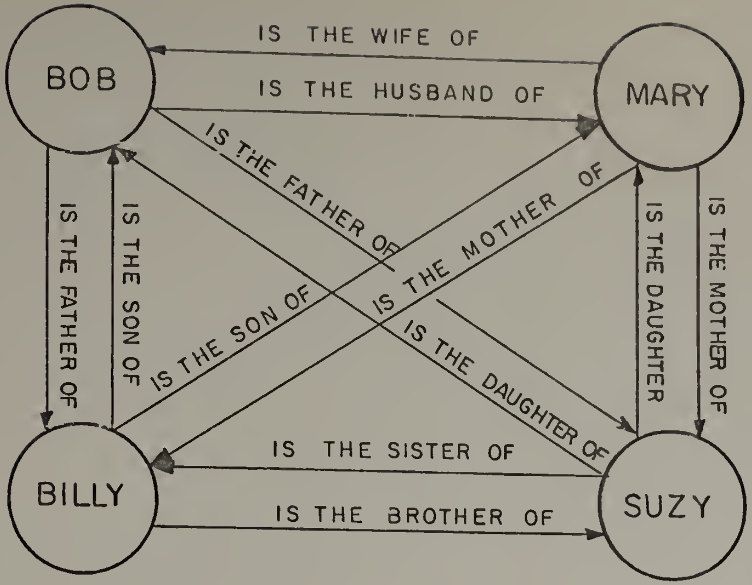
A simple illustration to clarify how a semantic net actually represents knowledge may be useful. Consider the following miniature world consisting of a family of four -- a father named Bob, a mother named Mary and two children, a boy, Billy and a girl, Suzy. A semantic net representation of this knowledge could look like this:

[14] Quillian, M.R., "Semantic Memory" in Semantic Information Processing, editor Marvin Minsky, (Cambridge, Mass: MIT Press, 1968).

[15] Carbonell, J., "AI in CAI: An AI Approach to CAI," IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, Dec., 1970, pp.190-202.

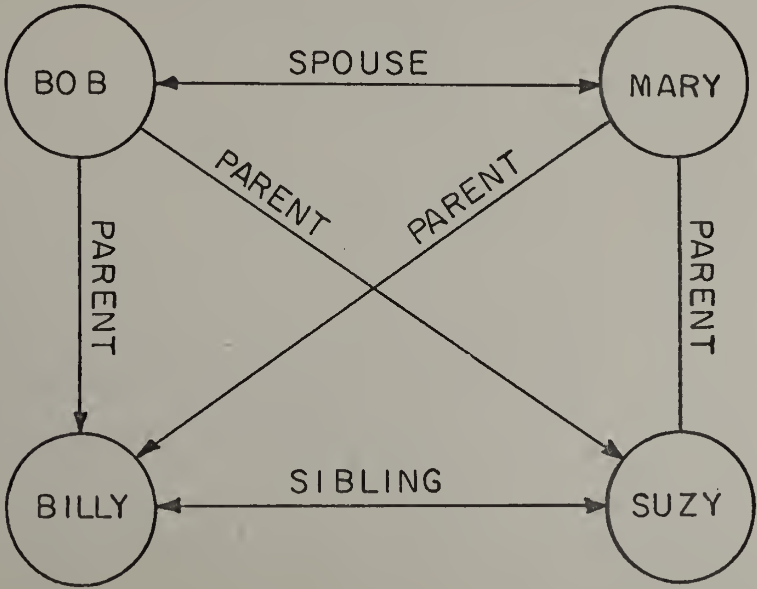
[16] Wexler, J.D., "Information Networks in Generative Computer Assisted Instruction," IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, Dec., 1970, pp. 181-189.

[17] Koffman, E.B. and Blount, S.E., "Artificial Intelligence and Automatic Programming in CAI," Third Joint International Conference on AI, Stanford, Calif., Aug., 1973, pp. 86-94.



The above semantic net has four nodes (Bob, Mary, Billy and Suzy) and eight relationships ('wife-of,' 'husband-of,' etc.) that make associatons between the nodes. Together, the nodes and the relationships form a semantic net which is a representation of knowledge about this family.

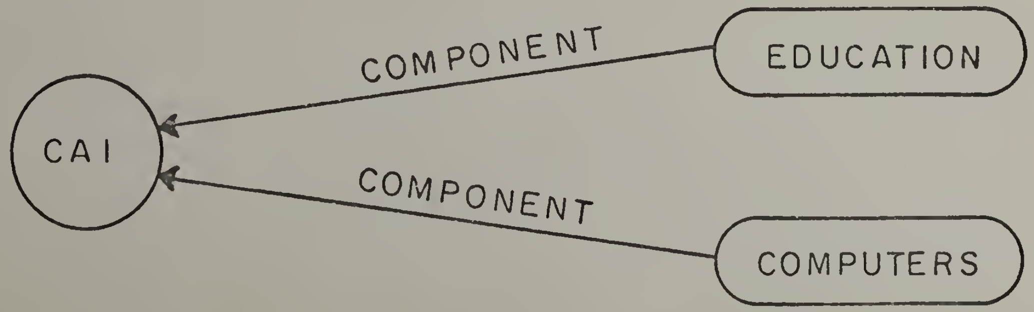
A simpler semantic net could be used to represent the same mini-world:



Here we have the same four nodes in the net but only three relations; two of the three are symmetric or reversible relations, i.e., spouse and sibling. The third is the inverse relation-pair 'parent-child'; that is, if Node A is a parent of Node B then it is implied that Node B is a child of Node A. Either version of the semantic net can be used to represent the knowledge inherent in the mini-world; one is more detailed than the other, and the choice of which to use depends on how we wish to use the representation as well as economic constraints. The important point is that there is more than one way to represent knowledge using a semantic net.

It is now possible to discuss why the semantic net representation of knowledge overcomes the second and third disadvantages of the hierarchical representation noted in section 3.2.2.2.1.

Redundancy or replication of information is not a problem in a semantic net. Information need not be duplicated at more than one node in the network -- all of the information which defines a node can reside at that node only; and the relations can be used to refer to other nodes, which also contain that information. Using the previous example of the library book on 'CAI' which appeared under both the headings 'Computer' and 'Education,' we may now use a semantic net to represent that knowledge:



Where the relationship shown by the arcs is the inverse relation-pair: 'system-component.' In this case the diagram would be interpreted as: "The topic 'CAI' is a component of the topic 'Education' and the topic 'Education' is a system which contains 'CAI' as one component." A similar relationship holds between the topics 'Computers' and 'CAI.'

The other weakness in the hierarchical structure that is overcome by the semantic net representation is that very little of the structure of the knowledge was revealed. A semantic net can show structure to a much richer and deeper level through the use of appropriately chosen relationships; the more types of relations that can be identified, the more the structure between the topics is revealed.

It is interesting to note at this point that we can think of a semantic net as a collection of hierarchical representations each with different relations that impose the hierarchy. Conversely, we may think of a hierarchical structure as a one-relation semantic net where that relation is usually 'general-specific.' A keyword index is another specific instance of a semantic net that has only the one weak symmetric relation --- 'related' where 'related' means that the two nodes are related somehow but the exact relationship is not specified. For example, when a book is reviewed it can be classified by a set of keywords and these keywords are "related" to the book itself. On the other hand, a dictionary can be viewed as a very large and powerful semantic network with many nodes (words) and relations that associate these nodes to give them their meaning.

Of course, the semantic network representation also has its disadvantages and they will be mentioned briefly now:

1) Resistance to change. This is the same problem described in Section 3.2.2.2.1 that is associated with a hierarchical structure, but here the problem is compounded; not only will the topics or nodes change as a function of time, so too will the relations. The problem of maintenance (adding and deleting information) induced by change will have a new dimension. The number of possible connections between nodes increases as the square of the number of nodes in the network. Under this constraint, the designer of a semantic net must be very careful to initially choose the topics and relationships so that they are not only valid and useful but also flexible.

2) Identifying the salient relationships. It is difficult enough to organize knowledge into a space of topics and subtopics, but it is even harder to determine the correct and important relations which associate these topics with each other. Furthermore it appears that different knowledge spaces (e.g. engineering, geography, computer literacy) require their own unique relationships. So, the problem is context-dependent.

3) Size of the data base. It is clear that for a semantic net like a dictionary to have any meaning there must be some minimum number of words to it. This same consideration is also true for any semantic net---it appears that there is a threshold in terms of number of nodes present that must be reached before meaning can be realized.

It is clear however, that a semantic net is an extremely useful and powerful way to represent knowledge; when represented as a graph, it comes very close to being a "picture" of the reality it attempts to represent. Another useful feature of the semantic net is that it can be used as a map of a knowledge space --- not only does it reveal the structure of the space via the relationships, but it can also be used as a search tool to find topics that are of interest to the user. This map aspect of a semantic net will be described in detail in Chapter V.

Finally, the use of a semantic net in itself can be an educational experience for the user. Since every topic in a semantic net is connected by certain relationships to various other topics in the net, the user can begin to see that the meaning of any topic is not absolute but relative to its relationships with other topics. As a matter of fact, the meaning of any topic can be defined as the rest of the semantic network as seen through that topic.[18] This insight is similar to the one described in Buddhist literature as the Vajracchedika Sutra[19] or "Diamond" Sutra where each manifestation of reality is represented as a many-faceted diamond reflecting all other diamond-manifestations. Not one single aspect of reality stands alone; everything is interdependent. Current environmental concerns are more and more confirming this intuition.

In sum, like state space and heuristic search techniques, semantic nets can themselves become a part of the repertoire of human conceptual tools for understanding.

[18] Quillian, "Semantic Memory."

[19] Humphreys, C., Buddhism, (Pelican Books, A228), 3rd ed. (Baltimore, Md.: Penguin, 1962), p. 58.

3.4 Intellectual Augmentation

Engelbart has defined intellectual augmentation in the following way:

> "by 'augmenting human intellect' we mean increasing the capability of a man to approach a complex situation, to gain comprehension to suit his particular needs, and to derive solutions to problems."[20]

He then goes on to suggest that one of the means for extending the intellect is through the use of artifacts such as computers and specifically 'computer-controlled information--storage, information-handling, and information-display devices."[21]

This section will deal with those aspects of the computer-human relationship that enhance and augment man's intellect and form an integral part of ACCOLADE. They are: programming, Computer Assisted Instruction (CAI), and Computer Managed Instruction (CMI).

3.4.1 Programming

Programming a computer via a programming language is the basic means by which humans and computers communicate, so that, in this sense, it is at the very heart of the human-computer relationship. Programming is usually viewed as a method of AI rather than IA; this is, it is through the use of programs supplied by the human to the computer that the computer is made smarter and consequently able to solve man's problems more effectively. In this section, however, we will show how programming can be a conceptual tool provided by the computer that augments the intellect of human user. Two aspects, problem solving and creativity will be discussed.

[20] Engelbart, D.C., "Augmenting Human Intellect," Stanford Research Institute, Summary Report AFOSR-3223, Oct. 1962, p. 1.

[21] Ibid, p. 9.

3.4.1.1 Problem Solving

Just as the act of writing down one's thoughts onto paper forces organization upon and insight from those thoughts, so also does the act of writing a program in a programming language. The act of programming (which is akin to explaining the problem to the computer and then teaching it the solution) forces one to be careful and thoughtful that one cannot help but understand the problem better in the process. Once a program is written, it models the problem to be solved; any simplifications or constraints are explicitly visible, to the programmer at least, and can act as a constant reminder as to the limited validity of any "solution." One cannot help but be aware of multiplicity of problem representations and the relative arbitrariness of choosing a particular representation. It is also extremely useful to write a program that carries out the steps for a problem solution without even running the program on the computer -- the insights gained by being able to describe the problem to a computer are worthwhile in their own right. This is a powerful intellectual augmentation technique. Not only does programming facilitate problem solving, it has allowed problems that were previously considered too immense or complex to be attacked and solved. For example, problems ranging from describing the growth of a single cell to predicting the weather on a global scale[22] can be modelled by computer programs and, by empirically checking the results, these models can be gradually improved.

There are many who feel that a person does not really understand a problem until he can teach it to (write a program for) a computer.

[22] Slotnick, D.L., "The Fastest Computer," Scientific American, Feb., 1971, p. 87.

Certainly programming is at least a valuable life-long conceptual tool for problem solving:

> "The most valuable acquisitions in a scientific or technical education are the general-purpose mental tools which remain serviceable for a lifetime. I rate natural language and mathematics as the most important of these tools, and computer science [including programming] as a third."[23]

Since the mid-sixties several major efforts have been made by educators to use computer programming as a problem solving tool. The results have been overwhelmingly positive and will be briefly reviewed next.

The Dartmouth Project, in addition to establishing time-sharing as a powerful access mode to the computer, began the process of using a programming language (BASIC) to allow students to become problem solvers across a wide range of disciplines.[24] The title of the final report for the project sums up the philosophy neatly: "the computer as pupil" -- let the student become the teacher and explain the problem to the computer-pupil through the medium of programming. Not only is the problem thereby solved but valuable problem solving techniques are also learned (teachers know that one of the best ways to learn is to teach). In the words of the final report:

[23] Forsythe, E., "What to do till the Computer Scientist Comes," American Mathematical Monthly, Vol. 75, No. 5, May, 1968, p. 456.

[24] Nevison, J.M., "The Computer as Pupil: The Dartmouth Secondary School Project," Final Report (NSF Grant GW-2246), Kiewit Computation Center, Dartmouth College, Oct., 1970.

> "The experience of the project over the last three years would suggest that perhaps a new and excellent way to teach the art of problem solving is to give students the responsibility of teaching various tasks to computers. This kind of learning is as important as any of the current improvements that rapid calculations afford traditional classes."[25]

Papert has also contributed to the trend of allowing the student to become an intellectual agent in the problem solving process. Utilizing the LOGO programming langugage, young children (ages 6-12) are given a wide variety of tasks to solve in the area of computational geometry.[26] Not only are the students learning how to use a powerful programming language to solve problems, they are able to invent and solve problems using some of the same mental processes that highly educated mathematicians also use.

Project Solo at the University of Pittsburgh is another example of utilizing the problem solving power of programming. Although students (high school) are started off in the "dual mode" where they share control with already written computer programs, the ultimate goal is to allow them to "solo" and write their own programs for problem solution.[27] A side benefit to this approach that the teachers also gain: "We have seen high school teachers make genuine scholarly discoveries at terminals."[28]

[25] Ibid, p. 21.

[26] Papert, S., "Teaching Children Thinking," MIT: AI Lab Memo No. 247, LOGO Memo 2, Oct., 1971.

[27] Dwyer, T.A., "Some Principles for the Humane use of Computers in Education," Int. J. Man-Machine Studies, Vol. 3, 1971, pp. 219-238.

[28] Ibid, p. 237.

Peelle has capitalized on the work of Papert and Dwyer and added the "glass box" as a technique for problem understanding and solution.

> "In this approach short, quickly comprehensible computer programs are given to students for their direct viewing. Each program embodies a concept, a procedure, or an inter-relationship and is written as simply and clearly as possible. Here the workings of the program are visible and, hence, become the basis for learning."[29]

Peelle has proposed a teacher-supplied "glass box" in contrast to the non-transparent "black box" program as a starting point in the problem solving process. After an initial period of 'mind-tinkering' (examining, analyzing, predicting and experimenting with a glass box program), the student is encouraged to modify, extend and creat glass boxes of his own designs.

"Debugging" is another exceptionally useful problem solving tool that stems from computer programming. Debugging is the process of identifying and correcting mistakes found in computer programs. It is the process of learning profitably from mistakes.

> "We often see bugs as rather good things because we can learn from them. Other people see everything as either 'right' or 'wrong.' For them, if it has a bug it is wrong and bad. But for us, this might make it interesting. There are many ways to react to a buggy situation. Can you recognize the bus? Is it a new one? Is it worth putting in our collection? We learn to appreciate bugs. They are telling us something."[30]

[29] Peelle, H.A., "The Computer "Glass Box": Teaching Children Concepts with A Programming Language," Educational Technology, Vol. XIV, No. 4, April, 1974.

[30] Solomon, C.J., "Leading a Child to a Computer Culture," Joint Bulletin (SIGCUE and SIGCSE) of ACM Conference on Computer Science and Education, Feb., 1976, edited by R. Coleman and P. Lorton, Jr., Anaheim, Calif., p. 80.

Indeed bugs do tell us something, we have only to learn to listen:

> "Listen and it will tell you its story; do not translate or act, but listen without interruption or interpretation right to the end of the story. Then only will there be action."[31]

Pirsig makes the same point regarding problems encountered while working on a motorcycle:

> "Just live with it for a while. Watch it the way you watch a line when fishing and before long, as sure as you live, you'll get a little nibble, a little fact asking in a timid, humble way if you're interested in it. That's the way the world keeps on happening. Be interested in it.
>
> "At first try to understand this new fact not so much in terms of your big problem as for its own sake. That problem may not be as big as you think it is. It may not be the fact you want but at least you should be very sure of that before you send the fact away. Often before you send it away you will discover it has friends who are right next to it and are watching to see what your response is. Among the friends may be the exact fact you are looking for.[32]

Of course the motorcycle can represent any problem situation in life and the skills acquired in the process of debugging programs can be transferred to problem solving in general --- a very valuable augmentation of the human intellect.

3.4.1.2 Creativity

Many definitions of intelligence include creativity as a component. It is interesting to note that creativity is also often given as part of the "reason" that computers are not really intelligent. Be that

[31] Krishnamurti, J., Commentaries on Living, (New York., Harper Brothers, 1956), p. 165.

[32] Pirsig, R.M., Zen and the Art of Motorcycle Maintenance, (New York: Bantam Books, April, 1975), pp. 305-306.

as it may, computers allow humans to program, and programming is a dynamic and potent instrument for expressing creativity.

Programming itself is a creative activity. The programmer can enter into the same experience any creative artist does while he is creating--the experience of spontaneous control of being a vessel that is filled from an outside source while knowing that the outside source is within himself--a mystical experience very hard to describe but intensely satisfying. Like the artist, the programmer may not have a complete idea at the outset what his creation will be like when it is complete. At the start he may have some amorphous ideas about the form and content of his intended creation, but once he enters the process of creating a program he is carried along, sometimes swept into the process not controlling but being controlled by the process until at its completion he surfaces, awakens, and holds his creation up to his mind's light. Watch a programmer programming and see if there is any difference between his activities and that of an artist or craftsman holding his creation in the palm of his hand.

Consider also what a programmer from the grey flannel company (IBM) has to say on the same subject:

> "There is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures."[33]

[33] Brooks, Frederick, The Mythical Man-Month, (Essays on Software Engineering) (Reading, Mass: Addison-Wesley, 1974).

There are at least two other aspects of creativity that are nourished by programming.[34]

1) Programming allows high school and college undergraduate students to perform real research in areas that previously required doctoral and even post-doctoral education and experience. For instance areas of mathematics (computational geometry) have been explored with the LOGO programming by young students and original creative work has been done.

2) Programming creates an active environment which embodies the content to be taught. Rather than passively receiving knowledge, the student is able to create inexpensive, responsive models via the medium of computer programming.

Whatever the creative advantages engendered by computer programming, it is the process of programming--analyzing, synthesizing and debugging that is so rewarding and in many cases enjoyable for the student. In programming as in life, the process can be more important than the finished product. "It is life, life that matters, life along-the continuous and everlasting process of discovering it--and not the discovery itself."[35]

3.4.2 Computer Assisted Instruction (CAI)

CAI is, as its name implies, an attempt to use the computer to assist in and enhance the instructional process; CAI does not include

[34] These two creative advantages of computer programming were mentioned at a talk by Hal Abelson of the MIT AI Lab at the University of Massachusetts on Nov. 19, 1976.

[35] Dostoyevsky, F., The Idiot, (Baltimore: Penguin Books, 1970), p. 433.

administrative or non-instructional uses of computers in schools such as budgeting and accounting, payroll, pupil census, class scheduling, student grade reports, etc.[36]

In this sense, the whole ACCOLADE system could be considered as CAI since it utilizes computers to assist in computer literacy education. In this section however we will deal with CAI as a conceptual tool to be used within ACCOLADE and avoid issues such as cost-effectiveness, history, etc.--interesting as they may be.[37]

Next we shall attempt to synthesize various proposed CAI classification schemes to show where ACCOLADE is situated within the spectrum and to indicate the various levels of intellectual augmentation associated with each classification.[38]

[36] For a full accounting of noninstructional uses of computers in education, see: Goddard, A.R., "Computer Applications for Prospective Public School Administrators," SIGCSE Bulletin, Fifth Symposium in Computer Science Education, Vol. 17, No. 1, Feb., 1975, pp. 51-55.

[37] For the reader who wishes a general survey of the field of CAI, the following references are useful:
a) Peelle, H.A., "Pygmalion's Computer," in Controversies in Education, editors Allen and Hecht, (Philadephia: Saunders, 1974).
b) Allen, J.R., "The Cybernetic Centaur: Advances in Computer Assisted Instruction," Computers and the Humanities, Vol. 7, No. 6, Sept.-Nov., 1973, pp. 373-387.
c) Atkinson and Wilson, editors, CAI: A Book of Readings, (New York: Academic Press, 1969).

[38] CAI classification were synthesized from:
a) Dwyer, T.A., "Some Principles for the Human Use of Computer in Education."
b) Peelle, H.A., Instructional Applications of Computers, (Menlo Park, Calif.: Addison-Wesley, to appear).
c) Control Data Corp., Plato Lesson, "aids."
d) Camerino, et al., Report of the Academic Computer Needs Committee, University of Massachusetts, Amherst, June 30, 1975).

Modes of CAI

1) Drill and Practice. The student is drilled on material previously learned in order to practice, improve and reinforce specific skills. This mode can include "page-turner" lessons (similar to programmed textbooks) and testing. Classic examples are foreign languages and arithmetic drills.

2) Tutorial. Here the student learns new material not previously presented to him (in contrast to Drill and Practice), usually by techniques similar to those used in programmed instruction. This mode can also include testing to determine the direction of the presentation and may use the "discovery" method as well as just presenting the information.

3) Gaming and Simulation. The student uses the computer as a model of some system or process such as a chemistry laboratory or a population growth model. By varying the input parameters to the system the student can view the related outputs and gain insight to the working of the system.

4) Utility Packages. These are "canned" programs that provide the student with problem solving tools such as statistical routines, matrix manipulation routines, and information systems for searching, storing and retrieving information.

5) Problem Solving. In this mode, the student writes his own programs to solve self- or teacher-assigned problems. This approach is, in essence, grounded in the Dartmouth "computer as pupil" philosophy.

Although no attempt is made to rate or rank these five modes of CAI, it should be clear that they have been presented in order of increasing learner control over the computer: in the "drill and practice" mode the computer almost completely controls the interaction, while the opposite is true for the "problem solving" mode.

In Chapter V we will see that in addition to using CAI lessons as learner resources, ACCOLADE can be considered within 4) above as an "information system" system mode of CAI. This method of enhancing CAI through information retrieval techniques is discussed by Huyck[39] who makes an interesting distinction between programs which computer and those which store and retrieve information:

> "A computation system has data to be found in the future embedded in logic prepared in the past; an information system has logic to be found in the future embedded in data prepared in the past."[40]

Huyck goes on to argue the future promise of the human-computer relationship.

> "CAI technology holds out the possibility of building a hybrid system using the computational power of machines and the metaphysical power of man."[41]

Perhaps the most anbitious vision of CAI as a conceptual tool for intellectual augmentation is captured in the phrase, "AI in CAI" where AI is "artificial intelligence" and CAI is "Computer Assisted Instruction." This phrase appeared in Carbonell's paper[42] and was discussed in section 3.2.2.2.2 under semantic information networks. It is also discussed

[39] Huyck, P.A., "CAI Techniques for Information Retrieval," Datamation, Feb., 1973, pp. 91-92.

[40] Ibid, p. 91.

[41] Ibid, p. 92.

[42] Carbonell, J., "AI in CAI: An AI Approach to CAI."

speculatively by Peelle and Riseman[43] who envision four phases of student-CAI interaction:

1) "A benevolent mentor" where either the computer or student can ask and answer questions about a particular knowledge space. As a mixed initiative dialogue CAI system, the computer has strategies for asking and responding to questions based on the history of previous interactions with the student.

2) A set "cognizant tools" which amplify problem solving powers and make the student-user aware of the limitations of these tools. The goal is to help the student teach himself to use certain computer supplied cognizant tools such as statistical packages, pattern matchers and logical deductive and inductive functions.

3) "A problem solving partner" where both the computer and the student formulate and critique their problem solving strategies.

4) A "computer-as-learner" in which the student teaches the computer how to solve problems expressed in a formal "meta-language." (This differs from the Dartmouth "computer as pupil" approach in which students solve problems in a programming language).

These four phases ("faces") are certainly ambitious and have yet to be entirely realized. However, progress in AI is beginning to actualize parts of each as the progress in intellectual augmentation proceeds.

[43] Peelle, H.A., and Riseman, E.M., "Four Faces of HAL: A Framework for using AI Techniques in CAI," IEEE Transactions on Systems, Man and Cybernetics, May, 1975.

3.4.3 Computer Managed Instruction

Computer Managed Instruction (CMI) is an attempt to use the computer to help manage the instructional process by automating testing, recording, diagnosis and prescription. The primary goal of CMI is to allow individualized instruction for the students [44,45,46] where individualized instruction has been defined as "the assignment of appropriate learning tasks to students according to their needs."[47] To this end, a curriculum is usually defined in terms of behaviorally-stated instructional objectives[48] and related objectives are grouped into units. Associated with each objective in a unit is a set of learning resources for acquiring knowledge that will allow the user to master that objective. Finally, criterion-referenced tests are constructed which measure mastery of the objectives in each unit. In practice, the computer will administer these tests to the student, diagnose which objectives were mastered and which were not, and, for the unmastered objectives prescribe a set of learning resources. This process is shown in the following flowchart:

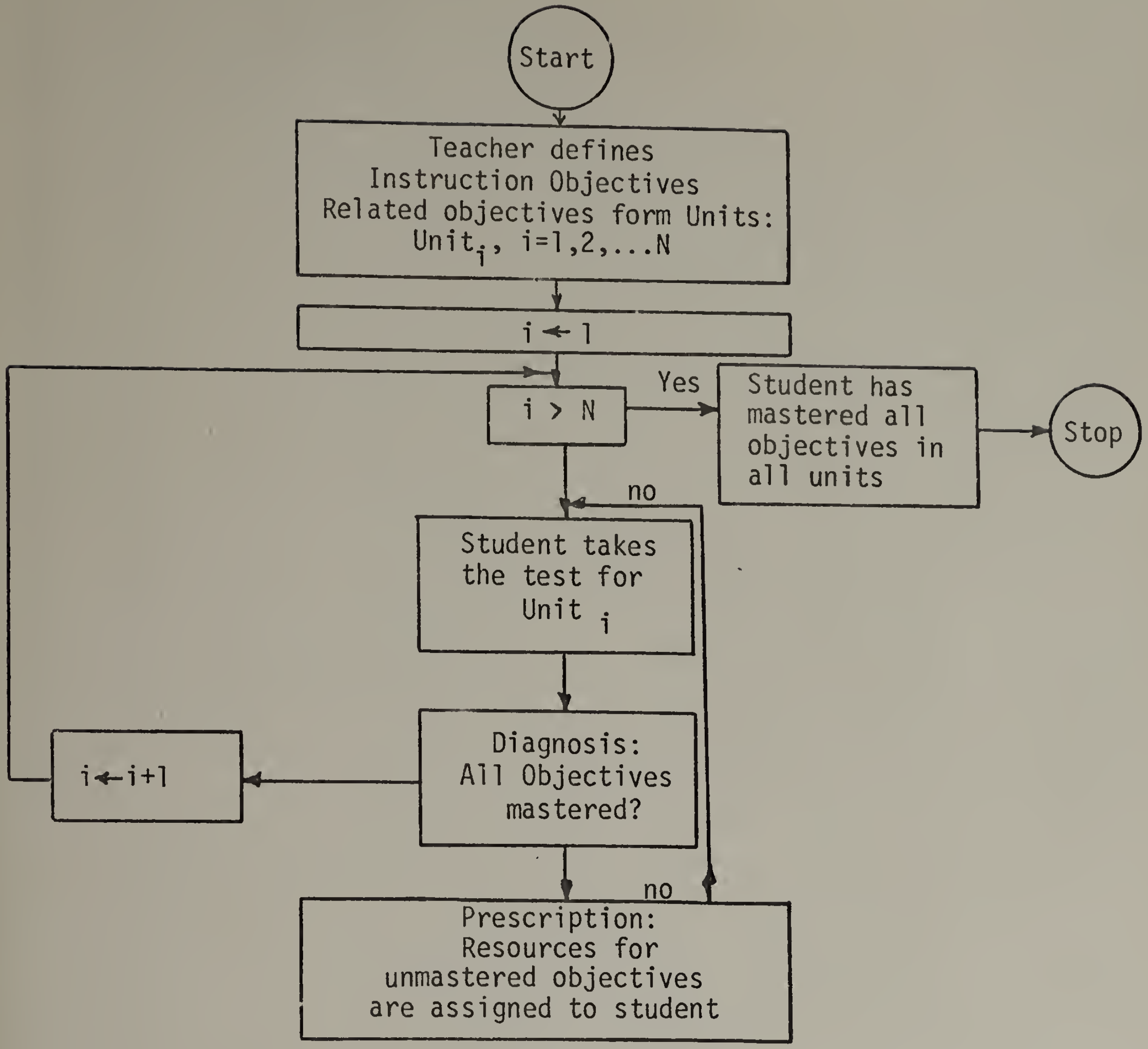
[44] Allen, M.W., et al., A Model for the Computer Management of Modular, Individual Instruction, Ohio State University, 1973.

[45] Brudner, H.J., "Computer-Managed Instruction," Science, Vol. 162, Nov. 29, 1968, pp. 970-976.

[46] Cooley, W., and Glaser, R., "An Information and Management System for Individually Prescribed Instruction," CAI: A Book of Readings, edited by Atkinson, R.C., and Wilson, H.A., (New York: Academic Press, 1969), pp. 95-117.

[47] Shanner, W.M.,: from a brochure by Westinghouse describing project PLAN.

[48] Mager, Preparing Instructional Objectives.


Start
Teacher defines Instruction Objectives Related objectives form Units: Unit$_i$, i=1,2,...N
i ← 1
i > N
Yes
Student has mastered all objectives in all units
Stop
no
Student takes the test for Unit $_i$
Diagnosis: All Objectives mastered?
i ← i+1
no
Prescription: Resources for unmastered objectives are assigned to student

In this flowchart representation, no distinction is made between pre- and post-tests as is the case in most CMI systems.[49] However, the first time a student takes a test can be viewed as the pretest and all subsequent times as post-tests. Also not shown in the flowchart are the

[49] Allen, M.W., et al., "A Model for...Individualized Instruction."

management reports that the computer generates for the teacher. Generally, these reports are of two types: (1) student progress reports which indicate the current status of each student as he proceeds through the curriculum units and (2) item analysis reports which help to measure the reliability and validity of the test questions.

The relation between CAI and CMI can now be clarified: CAI is a possible learning resource within the larger CMI system.

Next we shall list some of the advantages and disadvantages of CMI.[50]

Advantages

1) Allows the teacher more time to spend in personal interaction with the students since he can now spend less time administering and grading tests and the bookkeeping tasks to monitor student progress. Computerized testing also places most of the burden of "judger" on the "objective" computer so that the teacher can, if he wishes, become more of a friend and mentor to the student.
2) Allows self-paced or individualized instruction. Each student may "march to his own drummer" through the content of the curriculum and, when ready, can take the tests. Since the tests are criterion-referenced mastery is held constant and time is a variable (as discussed in Chapter II) so the student may, if need be, take a test more than once.
3) Accountability is facilitated. Not only are the objectives for a course explicitly stated, the tests are based on these objectives so that grading can be standardized across a group of teachers. Additionally, the exercise of objective formulation

[50] See also: Denenberg, S.A., MONITOR: A Computer Managed Instruction System, Dept. of Mech. Eng., ESIC Program, University of Massachusetts, Amherst, May, 1975, pp. 2-4.

can be a rewarding and enlightening experience.

4) Student and teacher morale can be improved since students receive their grades from the computer immediately (not the usual one week wait) and teachers need not worry about make up tests since students take the tests when ready and not at specified times.

Disadvantages

1) Allows the teacher to spend more time personally interacting with the students. This could be an anxious experience especially for authoritarian-mode teachers who may prefer to keep a "proper distance" between themselves (the source of knowledge) and the students (the receivers of knowledge). Also, since computerized testing places the burden of "judger" on the computer, the teacher is afforded an easy opportunity to shirk his responsibility for actually guiding the student's learning process.

2) Accountability is facilitated. Identifying and formulating instructional objectives can be an arduous and painful experience. An authoritarian teacher may not wish to have his methods questions--even by himself.

3) Student morale is lowered since in most CMI systems they have no part in formulating the instructional objectives or rating the suitability of the instructional resources.[51] (It simply

[51] For an example of this authoritarian type of thinking, see: Smith, J.O., and Smith, I.K., "Role of the Learning Resource Center in the Selection of Mediated Instructional Material," The Journal, Vol. 4, No. 5, May/June, 1977, pp. 21-24.

does not occur to most students to offer to assist in test formulation due to the predominate authoritarian environment).

4) Reliability of a computer system. Computers are machines and are subject to mechanical/electric malfunction and breakdown--if the computer is located remotely from the user and access is via telephone lines, the reliability problem is further compounded. Unfortunately, machines seem to breakdown more as humans come to depend on them more.[52]

In Chapter V these above considerations will be taken into account in the design of ACCOLADE.

Computer Managed Instruction systems allow intellectual augmentation indirectly for student users by affording them individualized (self-paced) instruction. If a student can learn at his own pace, the learning process should be more enjoyable and perhaps that enjoyment of learning will extend into the student's attitude toward learning in general. Another important lesson the student may gain is the concept that computers are <u>tools</u> that can be easily used to acquire information.

Certainly CMI appears to be more of a conceptual tool for teachers, and therefore a useful intellectual augmentation device. The teacher, is freed from tedious bookkeeping tasks and is able, if he chooses, to spend more time interacting with students and reflecting on his pedogogical and curriculular choices. Given this additional time, a teacher might come to realize that the structure of knowledge is perhaps arbitrary and

[52] A correlary, no doubt, of Murphy's Law.

to share this insight with his students. Students could then be encouraged to actively participate in the process of structuring course content knowledge as well as knowledge acquired outside the curriculum.

3.5 Conclusion

In conclusion, intellectual augmentation and artificial intelligence can be viewed as complementary processes, each enriching the quality of existence of man and machine. Whether or not the next stage of evolution will be a man-computer gestalt[53] it is not clear at this time. What is clear is that the human-computer relationship is dynamic and is continuing to grow, changing the definition of what man is and what a computer is.

A partnership between man and machine may be easier to facilitate than a teacher-learner partnership. Several thousand years have tended to make ossify the dynamics of the teacher-learner relationship. Computers have only been in existence for the past thirty years and the relationship has grown from master-slave to equals where each partner continues to develop conceptual tools for the other's use. From the western standpoint, perhaps the question, "What is the purpose of man?" can finally begin to be answered; from the eastern view, the purpose of man is largely irrelevant and, if it exists at all, it is to transcend the inclination to ask the question---to stop asking questions and get on with it.

[53] Sagan, C., The Dragons of Eden, (New York: Random House, 1977).

CHAPTER IV

COMPUTER LITERACY

4.1 Chapter Overview

The previous two chapters discussed the teacher-learner and human-computer relationships in order to explore the philosophical issues and to lay the groundwork for designing and building a computer literacy curriculum. This chapter will examine more precisely what is meant by the term 'computer literacy' and will discuss the educational value of computer literacy, as well as delineate some of the options for a curriculum, a pedagogy and administrative delivery systems for computer literacy.

4.2 Definition: What is Computer Literacy?

Computer literacy can have a spectrum of definitions. In a naive sense it implies mere recognition of the words "computer" and "literacy." In its most sophisticated meaning it might imply highly developed skills in the art of computer programming plus broad and deep knowledge in the areas of history, applications, social issues, hardware, and software. Of course, the word "literacy" means the ability to read and write but it has a secondary meaning of "possession of education."[1] This secondary meaning is relevant for the contest of this dissertation.

This leads us to consider the question of how much and what kind of education the student should possess in order to be considered literate

[1] Stein, J., ed., Random House Dictionary of the English Language, (New York: Random House, 1967), p. 536, col. 3.

about computers. Unfortunately, since computer literacy is a new topic having emerged only within the last five years, the "how much and what kind" question has been answered only operationally: computer literacy is what is taught in a computer literacy course.

Many of these courses are of the "computers and society" type and cover only certain aspects of the following topics: history, applications, social issues, hardware and software.[2] Other computer literacy courses include the teaching of programming skills.[3,4] The author will, in this chapter, attempt to make a case for at least affording the student the option to acquire programming skills. Damerell has operationally defined "illiteracy" as being the case when the student can speak but cannot read and write "well enough."[5] Extending this reasoning to computer literacy would mean that just being able to talk about computers (their history, applications, etc.) is not sufficient for true computer literacy--one must be able to read and write programs. In order to be able to do that, programming should be taught and learned within the confines of a computer literacy course.

With that in mind, the author chooses to begin to answer the question "how much and what kind of education should constitute computer literacy?" as follows: Computer literacy should entail an introductory knowledge of the areas of computer applications, history, social issues,

[2] Horowitz, R., et al., "Computers and Society: A Proposed Course for Computer Scientists," Communications of the ACM (CACM), Vol. 15, No. 4, April, 1972, pp. 257-261.

[3] Morsund, D., "What is Computer Literacy?", Creative Computing, Nov.-Dec., 1976, p. 54-55.

[4] DAT 101 at Greenfield Community College, Greenfield, Massachusetts.

[5] Damerell, R.G., "Let's Not Confuse 'Visual Literacy' with Notational Systems," Audiovisual Instruction, May, 1976, pp. 11-12.

hardware and software as well as beginner's skills in the art of computer programming.[6] Computer literacy courses which do not involve acquisition of the skills and behaviors of computer programming are, in the opinion of the author, basically just "Computer Appreciation courses--worthwhile but may be less rewarding and provide less fertile ground than a course which includes programming. In support of this view let us consider an analogy to the field of music.

A "Music Appreciation" course might include a study of the history of music, how it influences and is influenced by the culture, as well as a study of the various tupes of musical composition. The educational outcome of such a Music Appreciation course is usually intended to be a heightened awareness and sensitivity to music which presumably results in a richer and more enjoyable life. However, a course intended to impart "musical literacy" must add another dimension. The student must also actually enter into the process of creating music. He must be able not just to listen knowledgeable, he must be able to play some musical instrument or perhaps even compose a beginner's level of competence. Only then can a student be considered "musically literate." Of course this literacy should further expand the student's appreciation of music; the history, social implications, hardware and software can take on new, richer shades of meaning when enhanced by the experience of actually playing music on a musical instrument. Viewed in this way, computer

[6] Donald Knuth contrasts the terms 'science' and 'art' as they pertain to computer programming in: Knuth, D.E., "Computer Programming as an Art,: CACM, Vol. 17, No. 12, Dec., 1974, pp. 667-673.

literacy subsumes and enhances a computer appreciation on the part of the student.

Arthur Leuhrmann has written a clever parable comparing the invention of reading and writing with the invention of computing. Although he is essentially attacking the position of conventional Computer Assisted Instruction (drill and practice and tutorial) by asking the question, "Should the computer teach the student or vice-versa?", his final argument can be easily applied to a computer literacy course:

> "Computing constitutes a new and fundamental intellectual resource. To use that resource as a mere delivery system for instruction, but not to give a student instruction in how he might use the resource himself, has been the chief failure of the CAI effort. What a loss of opportunity if the skill of computing were to be harnessed for the purpose of turning out masses of students who were unable to use computers."[7]

Leuhrmann suggests that merely using computers <u>on</u> students is wasteful of the full potential of these machines. Similarly, merely telling students about some of the history, applications, social issues, hardware, and software of computers may build an appreciation but, lost is the opportunity to actively engage computers through the medium of programming. Is a person who has been told about the history of reading and writing, shown some books with words in them, and told about some of the useful applications of reading and writing - is this person to be considered literate? Appreciative perhaps, but most likely not literate.

[7] Leuhrmann, A., "Should the Computer Teach the Student or Vice-Versa?", <u>Proceedings of the SJCC</u>, 1972, pp. 407-410.

Therefore, besides educating the student in the areas of computer knowledge, it is the author's contention that a computer literacy curriculum should include the option for the student to master some of the skills of programming. Additional justifications for including programming in the computer literacy course will be offered in the next two sections.

4.3 Justification: Is Computer Literacy Worthwhile?

The question, "Is computer literacy necessary, or worthwhile?" will be discussed from two viewpoints: from that of the faculty and that of the student. The question will be answered in the positive--if computer literacy includes programming as part of its definition.

Data has been collected at the University of Massachusetts which measure faculty attitude toward the necessity of computer literacy for themselves. Specifically, the following question appeared as part of a questionnaire sent to all faculty of the University of Massachusetts at the three campuses in Amherst, Boston and Worcester:

> There is a viewpoint that in today's world, an educated person needs some knowledge about computers. What do you believe is a minimum degree of "computer literacy" for you?
>
> (please check one in each column)

	Now	5 years from now
a) no knowledge of computers needed	6	2
b) some general appreciation of computers	33	16
c) some general experience with computer p rogramming	30	28

	Now	5 years from now
d) fluency in at least one computer programming language	24	29
e) fluency in several computer programming languages	5	18

The numeric entries indicate the percentage of respondents that checked each category. (98% checked 'now' and 93% checked '5 years from now.') The median response of the degree of computer literacy required today fell somewhere between category 'b' and 'c'; the median computer literacy to be expected five years from now fell about midway between categories 'c' and 'd.' This, coupled with the fact that this question drew the largest response from the faculty, indicates that the area of computer literacy is of great interest to the University of Massachusetts faculty. Further, the report containing the questionnaire made the recommendation: "Computer literacy should be promoted."[8] In the same report, a group of University of Massachusetts students were asked the same question and allowed only choices a), c), and e). Over half (54.9%) chose category c) indicating that the students felt programming was an important aspect of computer literacy.[9]

A more nationally representative group of faculty made the following recommendation as early as 1967: "We recommend an expanded faculty

[8] Camerino, et al., "Report of the Academic Computer Needs Committee," University of Massachusetts, June 30, 1974, p. 2.

[9] Ibid., p. 58.

training program to provide adequate faculty competence in the use of computing..."[10] The same report recommended that all students at college level be required to have basic understanding of digital computation.

The National Advisory Committee on Mathematical Education has also recommended that computer literacy be promoted at an early stage of education:

> "We recommend the preparation of a junior high school course in 'computer literacy' designed to provide students with enough information about the nature of a computer so that they can understand the roles which computers play in our society."[11]

It is reasonably clear that most faculty feel that computer literacy is worthwhile not only for themselves but for their students also. But computer literacy cannot be considered worthwhile only because most of the faculty "feel" that it is. There are additional reasons:

- As computer costs decrease, personal computers may become commonplace in homes, schools, and businesses.
- Computer programming is becoming a prerequisite for many jobs that a few years ago required no computing experience at all.
- Computer programming is fast becoming a survival skill in many university courses.
- An informed attitude concerning the capabilities and non-capabilities

[10] Report of the President's Science Advisory Committee, Supt. of Documents, U.S. Govt. Printing Office, Feb., 1967.

[11] Recommendations of the Conference Board of Mathematical Sciences Regarding Computers in High School Education, (National Advisory Committee on Mathematical Education), 1972.

14. of computers is daily becoming more important in order to function in modern American society.

Theodore H. Nelson has stated this latter concern in very strong language:

> EVERYONE SHOULD UNDERSTAND COMPUTERS
> Computers are simply a necessary and enjoyable part of life, like food and books. Computers are not everything, they are just an aspect of everything, and not to know this is computer illiteracy, a silly and dangerous ignorance.[12]

Another important justification for teaching computer literacy centers on the mythology that has grown along with computer technology.

> "The public image is a mixture of fact and fiction: computers which make out payroll checks merge with the 'giant brains' and robots of fantasy. For above all, the computer in the popular mind has become the symbol of the ever-increasing world mechanization that is impinging on human life."[13]

"Much of the mythology deals with the threat they pose to the individual's freedom: People at present generally believe that computer systems are used on them rather than for them."[14] Since many people may have minimal computer literacy, they perceive only the threatening effects computers can have on their lives and are often not able to see the causes. How many times have we read in newspapers, heard on the radio or seen on television instances where the computer was blamed for some administrative 'foul-up?' If people were literate enough to realize that in many instances it was an administrator (a person) who instructed

[12] Nelson, T.H., Computer Lib, (Chicago: Hugo's Book Service, 1974), p. 2., col. 1.

[13] Horowitz, R., et al., "Computers and Society; a Proposed Course for Computer Scientists," p. 259.

[14] Colstad, K., and Lipkin, E., "Community Memory," Joint Bulletin (SIGCUE and SIGCSE) of ACM Conference on Computer Science and Education, Anaheim, Calif., Feb. 1976, edited by R. Coleman and R. Lorton, Jr., p. 199.

a programmer (another person) to program the computer to behave as it did, then perhaps people could see that the problem is not so much with the "*!?*! computer" as it is with people. The computer, shrouded in myth, is a perfect scapegoat--it has no need to defind itself (yet).

The threat to the individual's privacy is another common fear that is often exploited in newspaper articles:

> "COMPUTER HEADACHES -
> ...In blunt terms, the congressmen claim that the federal computers have violated citizens' privacy and wasted the taxpayers money. At one point, the bureaucracy schemed "to link up all the federal government's computers." This would have created a national data bank, which could have extinguished individual privacy. It also, incidentally, violated congressional restrictions."[15]

Certainly, the computer threat to privacy is a very real one, but statements like "extinguish individual privacy" serve only to polarize opinion and increase fear. Perhaps what is needed is less colorful reporting and more emphasis on education.

Another important misconception surrounding computers is that, somehow, in order to understand them, one must have a "mathematical mind."

> "The layman regards the computer as somehow 'being mathematics.' This erroneous concept has an unfortunate consequence in that those administrators, faculty, parents, and students who feel they have no aptitude for mathematics, and hence avoid it. shy away from the computer in the same way. That misconception is probably the single largest factor inhibiting the infusion, and diffusion, of informational technology (of which the computer is only a part) throughout our society. Yet that misconception is reinforced again and again in part because computer programming courses are usually offered by teachers of mathematics."[16]

[15] Jack Anderson, Item in Daily Hampshire Gazette, Northampton, Mass., July 23, 1977.

[16] Lykos, Peter G., "The Computer Illiteracy Problem; A Partial Solution," The American Mathematical Monthly, Vol. 81, No. 4, April, 1974, p. 398.

It is the author's experience from teaching people about computers over the last dozen years that while a "mathematical mind" is an asset in learning about computers and programming (it is a valuable extensible conceptual tool) it is by no means a necessary condition for the acquisition of computer literacy.

A balanced course in computer literacy can show the computer as the conceptual tool it is, a partner in the problem solving process, and a potential companion for man. Using Illich's terminology,[17] a computer can be used "convivially" as well as manipulatively. If the learner is allowed to control the computer by programming it in a computer literacy course, much of the perceived threat of computers can be mitigated.

This leads us to what the author feels is the most important justification for computer literacy: especially if a computer literacy curriculum includes programming it can begin to teach the skills and behaviors of logical thinking, problem solving, learning, confidence and autonomy. For the purposes of this dissertation, skills and behaviors are defined as follows: all skills are expressed behaviorally (they have associated with them an observable set of human responses or behaviors), but not all behaviors are skills (not everyone is skilled in all behaviors). In this sense, logical thinking, problem solving and learning are skills, while confidence and autonomy are behaviors. A course in computer literacy has the potential to teach a person to think for himself, solve problems in a creative way, and develop heuristics for learning so that confidence and eventually autonomy can be

[17] Illich, I., Tools for Conviviality (New York: Harper Row, 1974).

realized--if the curriculum allows and encourages the learner to actually program the computer.

Computer programming may help the student develop more logical and organized thought processes because the final program presented to the computer for execution must itself represent a logical unambiguous process. Minsky has said that "Eventually programming itself will become more important even than mathematics in early education"[18] for this reason.

Programming also fosters intuitive as well as logical thinking since the act of programming is essentially creative. And the creative act, a combination of logical and intuitive thought processes, can foster self-confidence in the creator. Intéréstingly, Bruner has identified the inverse cause and effect relationship---that self confidence can cause effective intuitive thinking:

> "Yet it seems likely that effective intuitive thinking is fostered by the development of self-confidence and courage in the student. A person who thinks intuitively may often achieve correct solutions, but he may also be proved wrong when he checks or when others check him. Such thinking therefore, requires a willingness to make honest mistakes in an effort to solve problems. One who is insecure, who lacks confidence in himself, may be unwilling to run such risks."[19]

It may be that intuitive thinking and self-confidence are reversible cause-effect processes.

Programming also lends itself to sharpening a student's problem

[18] Minsky, M., "Form and Content in Computer Science," 1970, ACM Touring Lecture.

[19] Bruner, J.S., The Process of Education (Cambridge, Mass.: Harvard Univ. Press, 1963), p. 66.

solving skills. Arthur Whimbey has characterized good problem solvers in the following four ways:

1) They are confident. They are not afraid to try an innovative method on the problem. They are not afraid to fail or look "dumb."

2) They perceive the struggle to solve a problem as pleasureful rather than painful.

3) They are sticklers for accuracy. Not only do they go over the given data more than once, they constantly check and recheck each step they have made toward problem solution.

4) They are careful. If the problem is a word problem they read it over more than once carefully. Steps taken towards problem solution are made with care and without attempt to force the solution into some preceived mold.[20]

Good programmers, in many cases, exhibit the same characteristics. They are confident (due perhaps to a history of successful experience in writing programs to solve problems) and take pride in especially innovative programming techniques and methods. They also enjoy the creative struggle to write a program to solve a problem--they seem to be absorbed in the process with the same intensity as a serious game-player. Programmers must also be extremely careful and sticklers for accuracy because this demand is placed upon them by the computer: the computer programs which compile, assemble and interpret the programmer's programs

[20] Whimbey, A., at his talk: "Teaching Problem Solving" at the University of Massachusetts., Amherst, Nov. 19, 1976.

demand complete syntactic accuracy.

It appears that programming and problem-solving skills feed back and nourish each other in the same way that intuitive thinking and confidence are the cause and affect of the other.

Programming can also foster the growth of learning-heuristics on the part of the programmer. One of the most valuable areas where these heuristics can be learned is in debugging which can be defined (as in section 3.4.1.1) as the process of learning from mistakes. For example, as Papert[21] has pointed out, debugging a program can lead to the realization that "mistakes" made during the trial steps to the solution of a problem are not "bad," they are just bugs to be uncovered. After some period of time, one realizes that mistakes or bugs are just wrong paths to the solution and in many cases, are necessary and useful as guides to the correct solution. An "unproductive" path is not necessarily a waste of time--usually new insights into the problem come as one proceeds or backs out of that wrong path. The student can learn that uncovering bugs is a natural part of the process of problem solving and that accepting this fact is an important ingredient to problem solving.

Another valuable learning skill that may be gained with programming experience is that it is not too terribly important where or how to start, but just to start somewhere and that the debugging process itself will begin to suggest useful strategies that lead to problem solution. To paraphrase Papert: We learn by doing--doing and thinking about what

[21] Papert, S., "Teaching Children Thinking," MIT AI Lab Memo No. 249, LOGO Memo 2, Oct., 1971.

we do and what we have done. Programming can facilitate this type of learning.

Writing programs to solve problems can also build confidence in the student in ways other than those previously mentioned. The student learns to consider alternative paths to solution, to select among them and inevitably to take responsibility for the consequences of his actions (the computer is sublimely unforgiving). If an appropriate set of programming problems can be offered by the computer literacy teacher, student confidence can be gradually increased.

Programming can also increase self-confidence when it reinforces the student's perception of himself as an "intellectual agent."[22] In this situation, the student, not the computer, is in control---effecting his own problem solving and building self-confidence. It is the author's experience that skill in programming is an effective leveller of age barriers between people. Young programmers have equal status to older programmers within the context of computers and the communications between them reflects this. The young programmers are treated with the same respect as any programmer of equal knowledge and skill, and this translates into an enormous boost of self-confidence on the part of the younger programmer. This self confidence, however it is engendered, can help the student become an autonomous person and competent member of society.

In conclusion, it is the author's contention that the skills of logical thinking, problem solving and learning and the behaviors of

[22] Ibid.

confidence and autonomy are encouraged and fostered by computer programming. A computer literacy course requiring computer programming can be a joy, stimulating creativity and intellectual challenge.[23] Not only can the course be enjoyable but the student is afforded the experience of learning about learning--learning to debug and solve problems which may lead to confidence and eventual autonomy.

Whitehead has said that "education is for life" and as such, it should be useful, "But if education is not useful, what is it?".[24] Writing a computer program actualizes this philosophy and when computer literacy includes an ability to program then it is good education and most certainly worthwhile.

4.4 Curriculum and Pedagogy

Given that computer literacy is worthwhile, we next consider the content to be taught and the methods for teaching.

4.4.1 Curricular Topics

Many computer literacy textbooks[25,26,27] have organized computer literacy knowledge into five general topics:

1) History of Computers

2) Applications of Computers

[23] Knuth, D.E., "Computer Programming as an Art." Knuth talks about the ability of a program to create joy in its creator, the intellectual and emotional satisfactions of computer programming as well as the discipline gained from working within tough constraints.

[24] Whitehead, A.N., The Aims of Education, p. 3.

[25] Rothman, S., and Mosman, C., Computers and Society, (Chicago: SRA 1972).

[26] Kochenburger, R. and Turcio, C., Computers in Modern Society, (Santa Barbara, Calif: Hamilton, 1974).

[27] Sanders, D., Computers in Society, (New York: McGraw-Hill, 1973).

3) Social Issues surrounding and caused by Computers

4) Hardware

5) Software (includes programming knowledge)

There are, of course, alternate ways to organize a computer literacy curriculum where specific topics are stressed. For example, some computer literacy courses stress social issues and applications[28,29] while others concentrate on programming skills.[30] An exhaustive curriculum for educating prospective computer scientists has been proposed[31] and could be used as the basis or a computer literacy curriculum. Indeed the ACM category codes[32] (the classification system in computer science) could be used as basis for selecting the topics in a computer literacy curriculum.

Which of these schemes is best? The fact is no one knows in an absolute sense the best structure or organization for the computer literacy knowledge space. What is "best" is more likely relative to the individual student's unique needs and the condition in which he arrives to be education. If we educators are to be completely honest with ourselves, we have no absolute knowledge concerning the correct way to

[28] Molnar, A.R., "Design of a Course in Social Applications of Computers," AEDS MONITOR; Feb., 1975, pp. 12-13.

[29] Geisler, R.G., "Teaching Computer Science," Computers and People, Vol. 23, No. 7, July, 1974.

[30] For Example COINS 122 (Intro. to Problem-Solving using the Computer) at the University of Massachusetts, Amherst.

[31] "Recommendations for Academic Programming in Computer Science." (Curriculum '68) CACM, Vol. 11, March, 1968, pp. 151-197.

[32] "Categories of the Computer Sciences," CACM, Vol. 15, No. 2, Feb., 1972, p. 70.

teach about anything. All we have to guide us are sets of premises which feel natural, premises based on our experiences and judgement which hopefully will lead the student to the acquisition of certain knowledge and skills. But as Weed[33] points out, the premises are completely arbitrary and time dependent and we should not fool ourselves or our students as to their immutability. If both the teacher and the student realize that the organization of the knowledge space is arbitrary, discussions about that space can be externalized and learning can begin to flow freely with room to expand.

In the author's view, a reasonable initial breakdown for computer literacy knowledge includes the five areas: history, applications, social issues, hardware and software (which includes programming knowledge, but not the skills and behaviors which will be discussed in section 4.4.2). We still hold with an update of Cicero's view that to be educated one must have knowledge not only in the areas of jurisprudence, literature, rhetoric, and history but in the skill of oratory itself.[34] Jurisprudence would correspond to practical application of the subject of computer literacy, history remains history, the literature and rhetoric can be interpreted as software and hardware, and finally oratory, the integration of philosophy and eloquence is, of course, the art of programming.

In addition to the five areas mentioned above, it might be useful

[33] Weed, L.L., "A New Curriculum."

[34] Cicero, M.T., De Oratore, translated by E.W. Sutton, (Cambridge, Mass.: Harvard University Press, 1959).

to add interdisciplinary topics:

1) The Future
2) Games
3) Education
4) Government
5) Artificial Intelligence

Education and Government and Artificial Intelligence could be considered as Applications, but they are of such magnitude and general interest that they, like Games and the Future, cut across and draw upon the original five topics so that they seem to deserve unique classification.

When these classifications are added, students desiring to learn specifically about, say, the role of computers in education would not be forced to search for education subtopics in the five main areas. Instead they could go directly to the topic of 'Education.' Similar considerations hold for the other four suggested interdisciplinary topics.

4.4.1.1 Alternatives

We have already suggested the arbitrary structure of computer literacy knowledge. This assumption, coupled to the hypothesis that all students are unique in their educational skills and needs, requires that alternative forms of computer literacy curricula be considered.

Many computer literacy courses offer few alternatives and are no more than FORTRAN language programming courses:

> "The most frequent exposure to computing in higher education is an introductory course in computing offered as a service course by the computer science or engineering department. This course is usually a course in FORTRAN programming."[35]

There is a need for these types of computer literacy courses where programming is stressed over all other possible topics. However, if this is the only educational food on the menu, many students have only two choices: starvation or malnutrition. What of the student who 'wants to learn about computers' but does not necessarily wish to learn the FORTRAN language?

Other options can be offered, and it appears that the market for alternative forms of computer literacy courses exists--at least at large universities. For example, at the University of Massachusetts, over 10% of the student population[36] takes some sort of an introductory course in computer literacy. At the University of Massachusetts, this 10% translates to 2000 students and if only 5% of these desire an alternative to a FORTRAN or other programming language courses then this means that there are over 100 students who want and need some alternative form of computer literacy education.

What then, are some of the alternatives available? One set of alternative offerings can come from specific academic disciplines and the

[35] Molnar, A.R., "The use of Computers in Education," THE Journal, Vol. 3, No. 2, Feb., 1976, p. 21.

[36] In 1977 over 2000 of 20,000 undergraduate students took introductory computer courses from within the computer science, engineering or business schools at the University of Massachusetts, Amherst. Data gathered from secretaries at each department.

corresponding departments. For example, the mathematics department can offer mathematically-oriented computer literacy courses for their students, and similarly for statistics, physics, business and engineering. Although the main thrust of these discipline-oriented computer literacy courses has been in the above areas, it is also possible that other academic disciplines such as the humanities could make available their own specialized curricula.

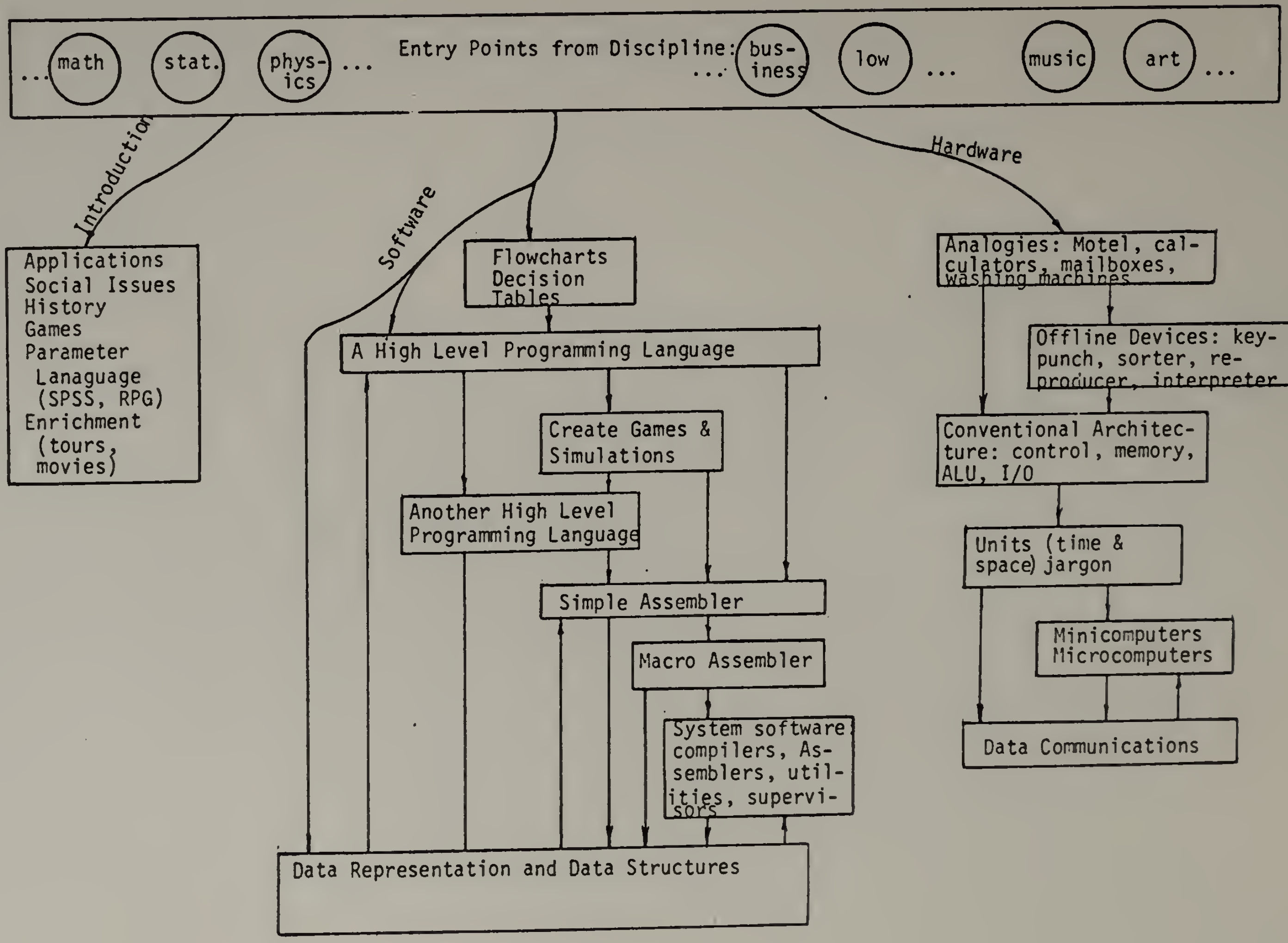
Another set of alternatives is possible when the curriculum stresses either authoritarian or self-directed education. In an authoritarian computer literacy course where the teacher is in control of the educational experience we would probably find the traditional lecture/lock-step instruction/norm-referenced test cycle with the topics chosen by the teacher. In a self-directed situation, the student would control the choices as to which computer literacy topics he would study, learn through individualized instruction and perhaps be evaluated by criterion-referenced tests. If the instruction were individualized, then the student could enter the knowledge space at any point and explore topics in that space at his own pace. We shall see in the next chapter that ACCOLADE is designed for this latter type of student.

An option between self-directed and authoritarian education also exists. The teacher can design modules which instruct the student about various computer literacy topics and then impose a sequence (where certain topics are a prerequisite to others) that should be followed by the student. Then the teacher can allow the students to traverse this network at their own pace. So, although the knowledge space and the paths through it have been somewhat constrained by the teacher, the student can still acquire knowledge and skills at his own pace and choose from several suggested learning paths. A possible network of this type is shown on the next page. The network coarsely represents a set and sequence of possible topics within a long-range computer literacy curriculum showing alternative learning paths among which students may choose. Each topic can, of course, be further expanded to form a subnetwork. The entry points at the top of the network are to be chosen by the student as a function of his discipline or interests. There are three main paths: Introductory (which includes history, social issues, and applications), Software, and Hardware. Any or all of the three paths can be traversed simultaneously and in any sequence subject to the directions imposed by the arrows.

In closing the discussion on curricular considerations, we feel it important to reiterate that no matter which educational alternative is used to teach computer literacy, the overriding concern should be on literacy itself. That is, the important issue is not so much what the topics are, in what sequence they are covered, or if they are taught under an authoritarian or self-directed philosophy, but rather that the student should gain a more informed set of conceptions about computers in general.

> "...it's not my job as a teacher-educator to offer the straight technical skills that are undoubtedly part of an information science course. It is far more important to teach about what computers can and cannot do in a context where the meaning of what is to be done is constantly referred to."[37]

[37] Marsh, B., "Teaching Teachers about Computers--A Course Description," Joint Bulletin of ACM, Feb., 1976, p. 86.


Entry Points from Discipline:
... math
stat.
phys-ics ...
... business
low ...
music
art ...
Introduction
Software
Hardware
Applications
Social Issues
History
Games
Parameter Lanaguage (SPSS, RPG)
Enrichment (tours, movies)
Flowcharts Decision Tables
A High Level Programming Language
Create Games & Simulations
Another High Level Programming Language
Simple Assembler
Macro Assembler
System software compilers, Assemblers, utilities, supervisors
Data Representation and Data Structures
Analogies: Motel, calculators, mailboxes, washing machines
Offline Devices: keypunch, sorter, reproducer, interpreter
Conventional Architecture: control, memory, ALU, I/O
Units (time & space) jargon
Minicomputers Microcomputers
Data Communications

Bronowski[38] has talked about fire as a "transformer"-- a process that illumines the "hidden structure of things." Perhaps that is how a computer should be viewed in a computer literacy course--as modern fire--an information transformer than can reveal the hidden structure of things.

4.4.2 Pedagogy

Some pedagogical considerations have, of necessity been discussed in Section 4.4.1, Curricular Topics--specifically Authoritarian versus Self-Directed education and what they mean in terms of individualized instruction, and norm versus criterion-referenced testing. We shall continue by discussing a teaching strategem particularly well suited for computer literacy and an obvious technology (the computer itself) as a delivery system for computer literacy.

4.4.2.1 A Teaching Strategy

The strategy is simply stated: it consists of analyzing a curriculum and separating it into two areas--a skills and behaviors part and a knowledge part; if that separation is reasonable for the particular content area, then it is implemented in the pedagogy. The advantage of this separation is that the skills and behaviors are usually taught through the process philosophy (by doing) and the knowledge part by more traditional cognitive activities such as reading, listening, and attending lectures. A curriculum that can be divided into process and content activities not only can capitalize on the advantages of both

[38] Bronowski, J., "The Ascent of Man," PBS TV series, 1975.

philosophies but can in and of itself act as a model for the student to use in deciding whether he is basically a "process" or a "constant" type learner--or some combination of the two.

The idea of dividing a curriculum into a knowledge component and a skills/behaviors component is not a new one. Dr. Lawrence Weed has done so in the medical school at the University of Vermont; and at the University of Massachusetts, Professor John Dixon has extended Weed's work to the engineering curriculum[39,40]. The author will describe Dixon's approach to dividing the engineering curriculum into a skills/behavior part and a knowledge part and then describe how Weed's and Dixon's pedagogy can be transferred to the area of computer literacy.

Dixon's engineering curriculum is called "Professional Practice-Directed Engineering Education" or PPDEE. The philosophy of PPDEE is that, in professional education, the emphasis should not be on teaching a core of knowledge. That is knowledge of itself should not be the main concern of engineering education. Instead, PPDEE proposes a core of functional skills and professional behaviors together with a required but subordinate knowledge base. The knowledge base includes typical engineering courses such as statics, mechanics, and thermodynamics. The skills and behaviors are: reliability, thoroughness, analytical soundness, productivity and social concern. To teach these skills and behaviors,

[39] Dixon, J.R. and Nelson, C.W., "Practice-Directed Engineering Education," Engineering Education, Oct., 1973, pp. 39-42.

[40] Denenberg, S.A. and Dixon, J.R. "A Computer-Monitored Instruction System for Practice-Directed Education," Proceedings of the Fifth Annual Frontiers in Education Conference, (ASEE and IEEE), Georgia Inst. of Technology, Oct. 20-22, 1975, pp. 315-321.

the students are assigned real engineering problems (obtained from industry) and work through them with a faculty auditor who acts like a demanding coach--modelling the skills and behavior he wishes to instill the student. The student's performance on the engineering project is judged entirely in terms of the core of behaviors and skills. The student can acquire the knowledge base in any way that suits him--e.g. self study, formal classes, movies, videotapes, and CAI lessons. The knowledge base is described by the auditor to the student in terms of instructional objectives. When the student feels ready, he can take a computer-generated and computer-graded test in any area within the knowledge base portion of PPDEE. The administering, testing, grading, diagnosing, prescribing and recording of student progress through the knowledge base has been computerized into a system called MONITOR; the teaching of the behaviors and skills, however, are done with human auditors. Experience gained from the PPDEE program shows that it is a viable and useful form of alternative engineering education--the students find it personally fulfilling and the faculty find working in the program to professionally enlivening.[41]

Once the curriculum has been partitioned, it is relatively easy to incorporate peer groups of learners attempting to acquire problem solving skills into the skills/behaviors part of the curriculum. Peer group learning can be an extremely powerful pedagogical tool. It has been used effectively by Dwyer in his Solo Project[42] and by Whimbey who has

[41] Dixon, J.R., "Professional Practice-Directed Engineering Education: Progress and Observations II," Final Report to Exxon Foundation, University of Massachusetts, Dept. of Mechanical Engineering, April 2, 1975.

[42] Dwyer, T., "Some Principles of the Human Uses of Computers in Education," p. 223.

designed a course around a problem-solving workbook within a peer learning group.[43] Let us briefly discuss Whimbey's method.

Whimbey uses the following methodology: The learners form dyads and the pair shares a workbook which consists of "IQ test-type problems" in order of increasing complexity. On the first problem, one student acts as problem solver and the other as listener. The problem solver is forced to vocalize or think aloud as he solves the problem. The listener's job is to enforce that vocalization as well as to stop the problem solver when he does not follow a step in the solution; the listener is not to help or offer advice. On the next problem, the roles of listener and problem solver are reversed and this sequence continues throughout the workbook. In addition, after each problem is a detailed transcription of how an experienced problem solver (a professor or graduate student) attacked and solved the same problem. The learners are not only afforded a model of the 'experts' solution, they get to see the total problem solving process-the wrong paths and bugs and the usually very simple techniques (making sketches, using one's fingers) that even experts use to solve problems. Also, vocalizing forces the problem solver to bring his thinking to conscious awareness so that he and the listener may analyze and perhaps improve it: "You will be able to see exactly what strategies you use, and what difficulties you have in solving a problem.[44]

[43] Whimbey, A., Development of Problem Solving Skills for Vocational and Educational Achievement, Washington State Superintendent of Public Instruction, Division of Vocational-Technical and Adult Education, 1975.

[44] Whimbey, p. 36.

Can this approach be extended to computer literacy education? At first glance it might appear that the type of student entering a computer literacy course is substantially different from a doctor or engineer-to-be and that the goals for a professional program of education differ considerably from the more modest goals of a program for imparting computer literacy. However, it is clear that programming encompasses a set of skills and behaviors (logical thinking, problem solving, learning, confidence and autonomy) and that the rest of a computer literacy curriculum (history, applications, social issues, hardware and software) constitutes a knowledge base. Therefore computer literacy, like the professional areas of medical and engineering training, is also suited to the stratgem of dividing the curriculum into skills and behaviors component and a knowledge component. This will, in fact, be incorporated into the design of ACCOLADE as described in Chapter V.

4.4.2.2 A Delivery Technology

While most courses tend to use no more advanced technology than chalk/blackboard, overhead projectors, or movies, a course in computer literacy is afforded a golden opportunity to use one of the highest forms of technology--the computer itself. Three opportunities for using a somputer as part of the pedagogy of a computer literacy course will be discussed: Computer Assisted Instruction, Computer Programming and Computer Games.

4.4.2.2.1 Computer Assisted Instruction (CAI)

CAI has been discussed in detail in Section 3.4.2 The purpose of mentioning it again here is to remind the reader that it can also be a very powerful pedagogical tool in any curriculum as evidenced by the popularity of the CAI aspect of the PLATO system.[45] Although research results fail to show compelling evidence that CAI is cost-effective, studies have shown that a gain in the performance of students taught by CAI over students taught by traditional methods.[46] A gain in performance means either the students scored higher on tests of the material or they covered it faster, or both. The largest single drawback to CAI, however, seems to be its high cost. When one includes system development, training, lesson preparation, and hardware and telecommunication costs, CAI costs range from one to five dollars per student contact hour.[47] Although this cost is currently too high, it should decrease as computer costs in general continue to decrease and could be "affordable" within the next decade.

4.4.2.2.2 Programming

Programming has been discussed as a necessary part of the computer literacy curriculum and as a valuable mode of CAI. In this section, several pedagogical strategies for teaching programming will be mentioned as well as a discussion of comparing possible programming languages for computer literacy.

At the heart of a computer is the computer program--the sequence of instructions which "drives" the computer. If we use the word "program"

[45] The author has experienced instances of students breaking into locked rooms to use CAI lessons PLATO terminals--as opposted to the relatively small number of students who break into lectures.

[46] Simonsen and Renshaw, "CAI: Boon or Boondoggle?", Datamation, March, 1974, pp. 90-102.

[47] Simonsen and Renshaw, Ibid.

as a verb rather than a noun, it emphasizes that a program represents a process and connotes that programming itself is a process rather than a static body of knowledge that can be taught by traditional lecture methods. A useful technique for teaching the programming process would be the case-study approach. Carefully circumscribed problems related to the student's discipline or interest area can be assigned as case-study programming projects. In this way, the student not only learns programming, but he can also learn more about the problem itself. Since the learning is by doing, it is usually effective and enjoyable.

A useful adjunct to the case-study approach is the "program log" similar to Weed's "problem-oriented medical record" or Dixon's "engineering record." Mason[48] has used a program log in his introductory course with excellent results. In its simplest form, a program log is a chronological annotated listing of all student transactions with the computer; every time a change is made to the program (which represents the solution of the problem) the student carefully annotates the reasons for the change on the program listing itself. Thus, the program log not only affords the teacher some insight into the student's problem solving process, it can also be a useful learning tool for the student since it forces him to externalize and hopefully examine his problem solving heuristics.

One final pedagogical tactic that the author has found to be quite useful in teaching programming should be mentioned: Go from the simple

[48] Mason, T.W., DP 200 Computer Programming, Florida A & M, Business School Publication, 1975.

to the general--from the concrete to the abstract. They may not be the best method for some disciplines, but to teach the process of programming, it seems to work quite well. Perhaps adults are inclined to learn the concrete before the abstract in the same way Piaget indicates that children learn.[49] As Whitehead says,

> "There is a proverb about the difficulty of seeing the wood [the English word for forest] because of the trees... The problem of education is to make the pupil see the wood by means of the trees."[50]

Since programming is such an important component of a computer literacy development let us conclude this section by briefly examining some of the alternative programming languages that might be used. First assembler languages, then high level languages will be considered and finally some recommendations will be made as to which are most suitable for computer literacy development.

Assembler Languages

The advantages of using an assembler language such as COMPASS for CDC machines or BAL for IBM model 360 machines centers around considerations of efficiency, capability and ease-of-use. Assembler language programs usually utilize the system resources more efficiently--programs generated are smaller and take less storage space. Additionally, the assembler process is faster than the 'compile' or 'interpret' process used for a high level language. Another important advantage is the increased capabilities of an assembler language over high level language.

[49] Richmond, P.G., An Introduction to Piaget.

[50] Whitehead, A.N., The Aims of Education, p. 10.

Assembler language allows the user access to the full range of hardware capabilities of the machine: e.g. individual bits in a word or register can be manipulated. A possible definition of what a computer is, then, is simply a list and description of the assembly language repertoire. The assembler user can begin to see what a computer is as well as gain some understanding of what it does.

There are, however, several ease-of-use problems associated with assembler language which will be mentioned here briefly (since they are described next as advantages of high level languages). They include: difficulty of writing, debugging, documenting and maintaining programs, lack of compatability between machines of different manufacturers, and a more detailed knowledge of the hardware structure is required by the user--a serious disadvantage in a computer literacy course. There do exist stripped-down assemblers which are easier to learn and use than conventional assemblers[51] but these are, in most cases, so simplified that only very trivial programs can be written.

High Level Languages

High-level or process-oriented languages have been popular since FORTRAN was invented in the 1950's. Although high-level languages are a layer away from the actual capabilities of the computer, they provide more insight into what a computer can do (as opposed to what it is). The user does not have to know very much about the inner workings of the machine. One statement of a high level language usually represents

[51] For example, Knuth's MIX or Forsythe's SAMOS.

many assembler language instructions. They are generally easier to learn, use and debug than assemblers but they do not execute as quickly. They are transportable; while assemblers usually run only on the computer of one computer manufacturer, high level language can run (with minor modification) on most all computer systems. High-level languages are easier to document (the language is usually close to a flow chart in representation), and since they are easier to document they are easier to maintain. In general, a high level language gives the user more convenience at the expense of the flexibility and greater control of an assembler language.

Since high level languages are easier to learn, use, debug and are more convenient then assembler languages, it is this writer's opinion that a high level language should be the <u>first</u> or introductory language taught in a computer literacy course. Following is a list of candidates, together with brief qualitative descriptions:

1) ALGOL. A very nice, powerful language selected by the ACM as the formal method for describing computer algorithms. However, its syntax is difficult to learn (Bachus-Naur form may be easy for a computer to check if a statement is syntactically correct, but it is cumbersome for a human being who wants to learn the language). Further, an adequate working version of ALGOL is difficult to find except on Burroughs computers.

2) APL. Developed by Iverson at IBM as a unifying mathematical notation as well as "<u>A</u> <u>P</u>rogramming <u>L</u>anguage," APL is well suited as a computer literacy language. A subset of the available constructs could be taught and as the user grows in literacy,

he should be able to tackle the more sophisticated and powerful concepts. The advantages of APL include its conciseness, versatility, elegance and potential for growing with (sometimes stretching) the user. Also, the structure of the language encourages "top-down" programming---an essential ingredient of using the problem reduction method to solve problems. Disadvantages often cited are its cryptic notation, the 'backwards' evaluation of expressions, the lack of an explicit conditional jump construct, and that it is too mathematically oriented. APL is usually implemented as an interpreter so that it runs interactively and reasonably fast. Its use seems to be growing since IBM has begun to actively market it.

3) BASIC. Developed by Kemeny and Kurtz at Dartmouth College for the express purpose of affording the student with an easy-to-learn, easy-to-use interactive language. It is just that. It is implemented on many machines and exists in many mutated forms.[52] The author has used BASIC as an introductory computer literacy language and found the student response to be very good. However, since it is so very simple and easy to learn (Jr. High School students have learned it in a weekend)[53], it is not rich and lacks the power to solve a broad range of programming problems---leaving the user to learn another

[52] Lientz, B.P., "A Comparative Evaluation of Versions of BASIC," CACM, April, 1976, Vol. 19, No. 4, pp. 175-181.

[53] "The Computer as Pupil: The Dartmouth Secondary School Project Final Report (NSF Grant GW-2246)," Kiewit Computation Center, Dartmouth College, Oct., 1970.

more powerful, programming language.

4) COBOL. One of the oldest languages, COBOL was developed primarily for business data processing applications. Besides being specialized in its application, it is difficult to learn since the user must be aware of computer data representation formats and seems to run adequately only on IBM computers.

5) FORTRAN. The first-developed high level language. FORTRAN is relatively easy to learn (except for the notorious FORMAT statement) and is probably the most widely used computer language in the world. FORTRAN runs well on most all computers and has been implemented in the batch mode (as a "quick-and-dirty" compiler or a "slow-but-efficient" compiler) as well as in the time sharing mode. However, unless the goal of the computer literacy course is to provide the student with a job-oriented skill, there are several better languages available (ALGOL, APL, and BASICS thus far).

6) LISP. In a class by itself. It has been utilized as a computer literacy language,[54] and its main advantages include teaching the concepts of recursion and "top-down" programming. However, the main application of LISP has been in AI research. Since it is rather difficult to use for solving numerical problems and is somewhat difficult to read (the syntax is full of parentheses) it might best be taught as a second or follow-on computer literacy language.

[54] Friedman, D.P., The Little Lisper, (Palo Alto, Calif.: SRA, 1974).

7) LOGO. Developed by Papert at MIT and Feurzeig at BBN, LOGO is basically an offshoot of LISP that attempts to teach computational geometry and recursive techniques using either a real or a graphic turtle. LOGO might make a very nice introductory computer literacy language especially at the level for which it was designed (grades 1-6) but is so unique in structure that it may give the college student misconceptions about programming languages in general. It is not implemented widely.

8) PL/I. Developed by IBM as a combination of ALGOL, COBOL, FORTRAN, it is a very extensive and powerful language. PL/I seems to be the replacement for COBOL as the business data processing language (State Farm headquarters in Bloomington, Illinois which has three large computer systems to handle all their fire, life, and car insurance policy updates is now writing all programs in PL/I). Although a subset of PL/I could be used as an introductory computer literacy language[55], it is only as good as its components.

Rather than choose one of the above described languages as the single computer literacy programming language, the choice will be made on two levels: as the first language of a computer literacy course, and as a possible follow-on language for the student who wishes to extend and enhance his literacy. As a first language this author would choose

[55] Holt, R.C. et al., "SP/K: A System for Teaching Computer Programming," CACM, May, 1977, Vol. 20, No. 5, pp. 301-309.

APL or BASIC with a very slight edge to APL (let me put it this way: if I were stranded on a desert island with just a computer terminal I would prefer APL as my companion); in the follow-on category the author suggests a stripped-down assembler first, followed by FORTRAN and LISP in that order.

Other choices can be made depending on the characteristics of the student population. For example, primary school children would likely prefer LOGO or BASIC while for adults, APL would be a more suitable choice for a computer literacy programming language. If the breakdown is by academic discipline, then for business or data processing students the choice would be FORTRAN or COBOL, FORTRAN or APL for scientists and engineers, and APL, BASIC, or LOGO for humanities students.

4.4.2.2.3 Games

Computer games can be an entertaining and useful pedagogical tool in a computer literacy curriculum. Not only is play an enjoyable way to learn, it often provides the initial path into the more academic and practical pursuits associated with computers. This is not to imply that all computer games are frivolous. There is, for example, a two player inter-terminal game on the PLATO system called MOONWAR where each player attempts to shoot at his opponent by specifying the angle of the shot; this game teaches the concept of a directed angle quickly, easily, forcefully and enjoyably. Unfortunately there exits a barrier against games in most organizations which administrate the scarce resources of computer time and memory space.[56] However it has been the experience of

[56] Denenberg, S.A., "An Evaluation of the PLATO IV System," Dec., 1974, unpublished manuscript, available from Stewart Denenberg, Dept. of Computer Science, SUNY, Plattsburgh, N.Y.

the author that many professional programmers were first attracted to computer programming through computer games. After the student has "played" a game for some time, he may become interested in the program which runs the game. Computer games can also be justified as an entry into the topic of artificial intelligence. Many computer games[57] embody AI techniques to make the game more satisfying for the user. Indeed, games have furnished AI research with rich, complex and constrained environments well suited for experimentation.[58]

The final section of this chapter will deal with the delivery of computer literacy at an administrative level within a university such as the University of Massachusetts at Amherst.

4.5 Delivery Systems

If one accepts the worth of computer literacy and one has considered the alternative pedagogies and curricula to effect this literacy, then one must next consider the delivery system. What is the best way to "deliver" computer literacy to the students?[59] What we wish to consider here are the administrative (as opposed to pedagogical) questions dealing with computer literacy: For example, should efforts be made to persuade faculty to integrate a modicum of computer literacy into their established courses? Should specialized computer literacy courses be offered by each department? Should computer literacy be taught

[57] Burton, R.R., and Brown, J.S., "A Tutoring and Student Modelling Paradigm for Gaming Environments," Bolt, Berenak and Newman, Cambridge, Mass., 1975.

[58] Checkers, Chess, Go, and Kalah, for example.

[59] Note: the comments in this section are paraphrased from the "Report of the Academic Computer Needs Committee," but were originally written by the author.

from a general perspective through a central source?

Fortunately these three approaches to delivering computer literacy are not mutually exclusive, so there exists some mix of them that can be used to begin to address the problem. Let us consider each approach separately.

Persuading faculty to become literate themselves so they can transmit that literacy to their students is a very difficult problem because, at present, there is little incentive for the individual faculty member to do so. He already has an overload of demands on his time; why should he take more of that precious time to engage in an endeavor that offers no direct rewards (promotion, prestige, publications, power)? There is no easy answer here short of persuading the university administration to make money or release time available for the pursuit of computer literacy. However, all that does is raise the problem one level--how can the administration itself be persuaded? After all, administrators are themselves people, usually former faculty, who must allocate scarce resources (money, time, and people) to strongly competing ideas, mechanisms and philosophies. How does ten dollars for computer literacy stack up against a new book for the library? Not very well, based on a recent survey of faculty attitudes toward the funding priorities of competing university resources.[60]

[60] Camerino, et al.,"Report of the Academic Computer Needs Committee," June 30, 1975, p. 53. Actually, the central library was rated against a central computing facility which is only one measure of support for computer literacy. Out of ten proposed resources, the library ranked second, the computer ranked about fifth.

In any case, some effort could be made to encourage computer literacy among the faculty especially within the School of Education. It is not enough to educate the elementary school teachers in computer literacy; one must go one step higher and teach the people who teach the teachers--the faculty or computer literacy may never reach the students.

Next, consider the question of centralization versus decentralization of the teaching of computer literacy. That is, should separate courses be offered through each department or through a central source such as the Computer Science Department, the Computer Center, or a Computer Literacy Center? The problem of centralized versus decentralized delivery of computer literacy is similar to the problem that arises when a service course if offered by a specific discipline to students from other areas. The problem is that the faculty who teach these service courses tend to teach them at a more abstract level than the students appreciate and, as a result, these courses and the departments for whom they are offered are oftimes disliked by the students. Consequently the "served" departments find it necessary to offer their own tailored and applied courses in order to satisfy their student's unique needs. In many cases what has evolved out of this "problem" is a compromise between centralization and decentralization in which students take several service courses and a few applied courses within their own departments. In some cases these courses are taught jointly by members of each department. For instance, a technical writing course for engineers at the University of Massachusetts is taught jointly by members of the rhetoric and engineering faculties.

A reasonable middle road for delivering computer literacy is some type of mix of centralization and decentralization. A centralized service could be provided to assist departments in developing their own computer literacy programs. Additionally, this service could offer more general literacy courses that would be of interest to a large number of people. At this central source, professional, full-time people could be employed who would presumably be more qualified to produce an excellent course than would a faculty member who was teaching this course as part of an already overloaded teaching and research schedule. A centralized general course can alleviate the problem of varied student backgrounds by assigning programming projects within the student's discipline.

A final observation concerning the administration of computer literacy: there is no reason to assume that a university or college is the best environment to delivery computer literacy. Perhaps, if the curriculum is self-directed, the most appropriate place would be the public library system or, in the future, from a videodisc attached to one's home television-computer set.[61]

In conclusion, the author would like to restate his contention that, since no one knows for sure precisely what computer literacy is, let alone the correct curriculum and pedagogy it becomes extremely important to provide the student with alternative curricula by which he can acquire computer literacy himself. ACCOLADE is one such option, and its design and implementation are described in the following chapters.

[61] Bork, A., "Videodiscs - The Ultimate Computer Input Device?", Creative Computing, Mar.-Apr., 1976, pp. 44-45.

CHAPTER V

DESIGN OF THE ACCOLADE SYSTEM

5.1 Chapter Overview

This chapter deals with the proposed design of ACCOLADE in terms of the issues raised and discussed in Chapters II, III, and IV: authority and responsibility, process and content, structure, heuristics artificial intelligence and intellectual augmentation. The design is specified by identifying the major components of the system: Teacher, Learner and Computer and describing the function of each. A scenario which describes the operation of ACCOLADE from the point of view of the Teacher, the Learner, and the Computer closes the chapter.

5.2 Introduction

This chapter proposes an apotheosis for ACCOLADE--it is an ideal and is a response to the need for an alternative curriculum for computer literacy suggested in the previous chapter. The actual implementation of the proposed design which takes into account the mundame realities of time and human effort will be discussed in the next chapter.

5.2.1 Issues

Before describing the components of the ACCOLADE system let us first review the issues raised in Chapters, II, III, and IV and indicate their relationship to the design of ACCOLADE.

For the most part ACCOLADE is a self-directed educational system although certain aspects are definitely authoritarian in nature. For example, ACCOLADE allows the Learner to take responsibility for effecting his own education by encouraging the Teacher-Learner partnership and by affording the Learner individualized instruction, evaluation, computerized tools and communications media. On the other hand, the choice of the educational objectives, the teaching-learning experiences and the evaluation are largely under the control of the Teacher. However, this authority is mitigated by two factors: (1) It is the responsibility of the Teacher to provide a wide range of alternative objectives, learning experiences, and evaluative tests among which the Learner can choose freely. (2) The Learner can himself evaluate the Teacher's choices of objectives, learning experiences and evaluative tests so that the partnership can flourish (assuming the Teacher chooses to listen to this Learner feedback).

One of the most important features of the design is the separation of the curriculum into programming skills/behaviors and knowledge where the skills and behaviors fostered by programming are: logical thinking, problem solving, learning, confidence and autonomy, and the knowledge areas are history, applications, social issues, hardware and software. Thus, neither process nor content is chosen as the only path for learning; but rather, each is used in its appropriate place.

The design also assures that man in innately "good," at least good enough to be able to impose enough discipline upon himself to learn what he chooses to learn. In the ACCOLADE system the responsibility for learning rests squarely on the Learner.

With regard to artificial intelligence, ACCOLADE allows and facilitates the Learner's experience of heuristic and state space search techniques as he actively searches out computer literacy knowledge. If the Learner becomes more aware of the heuristics he uses to solve problems or to search a space then he has at his disposal a very powerful skill, namely thinking about thinking; and, hopefully this will increase his problem solving effectiveness. The techniques of state space search are tied closely to AI models for representing knowledge. The design of ACCOLADE includes a hierarchical structure (a Yellow Pages) and a heterarchical structure (a Semantic Network) to act as search tools and to illumine the underlying structure of the computer literacy knowledge space for the Learner.

Intellectual augmentation (IA) is provided to the Learner via CAI lessons, a CMI system, and, again, via programming itself. The CAI lessons are one type of resource available for acquiring computer literacy knowledge. The CMI system not only performs the functions of testing, recording, diagnosing and prescribing, it captures Learner interactions with the computer for later analysis.

Programming is offered, using the case-study approach through the auspices of a Teacher. It is very difficult to teach programming skills and behaviors exclusively by means of computer although work continues in this area.[1,2,3] However, much progress remains to be made before programming skills and behaviors can be taught by computer and in the near future the best delivery agent for those skills appears to be a competent experienced human teacher. This aspect will be discussed

[1] Danielson, R.L. and Nievergelt, J., "An Automatic Tutor for Introductory Programming Students," SIGCSE Bulletin: Fifth Symposium on Computer Science Education, pp. 47-50.

[2] Wilcox, et al., "The Design and Implementation of a Table Driven, Interactive Diagnostic Programming System," CACM, Vol. 19, No. 11, Nov.,1976, pp. 609-616.

[3] Fenichel, R., et al., "A Program to Teach Programming," CACM, Vol. 13, No. 3, March, 1970, pp. 141-146.

again in Chapter VII---"Suggestions for Future Research and Development."

Finally, the design implicitly assumes that ACCOLADE is an alternative form of computer literacy education (viz: the 'A' in ACCOLADE). It is directed primarily to undergraduate college students who want more flexibility than traditional courses offer and who are willing to supply some of the direction in their own education. There is, of course, no reason why ACCOLADE cannot be used by advanced high school students, graduate students or faculty. As a matter of fact, since ACCOLADE is amenable to delivery from either separate departments or some central source with a college or university, faculty and graduate students could use it to develop their own computer literacy and in a particularly non-threatening way--away from large classes full of undergraduates, and in a secluded room furnished only with a terminal connected to the computer system.

5.2.2 Spirit

The following two quotations speak to the essence of the spirit motivating the design of the system:

> "My purpose is to get people at least to begin to think in another direction, to look for an organization of education less wasteful of human resources and social wealth than what we have. In reconstructing the present system, the right principles seem to me to be the following: To make it easier for youngsters to gravitate to what suits them, and to provide many points of quitting and return. To cut down the loss of student hours in parroting and forgetting, and the loss of teacher hours in talking to the deaf. To engage more directly in the work of society, and to have useful

> products to show instead of stacks of examination papers. To begin to decide what should be automated and what must not be automated, and to educate for a decent society in the forseeable future."[4]

> "So do I feel kind of mystical about education, I feel that somehow the answers come to you through your own inquiry."[5]

As much as possible ACCOLADE Learners are responsible for their own inquiry, and as much as possible ACCOLADE Teachers listen to that inquiry and make it as easy as possible for the Learner to "gravitate to what suits them."

5.2.3 Goals

ACCOLADE has the following design goals:

- To demonstrate an alternative, effective teaching-learning system.
- To allow the learner to take responsibility for effecting his own education.
- To encourage a Teacher-Learner partnership and be responsive to their needs and requirements
- To use computers and people as components so that ACCOLADE encourages the human-computer partnership.
- To identify and provide resources for the acquisition of computer literacy knowledge in the areas of applications, history, social issues, hardware and software; to illuminate the structure of this knowledge space and, by doing so, shape an informed attitude

[4] Goodman, P., Compulsory Mis-education and the Community of Scholars, (New York, Vintage, 1964), p. 153.

[5] Mitchell, J.L., "The Education of Joni Mitchell," The Co-Evolution Quarterly, Summer, 1976, p. 137.

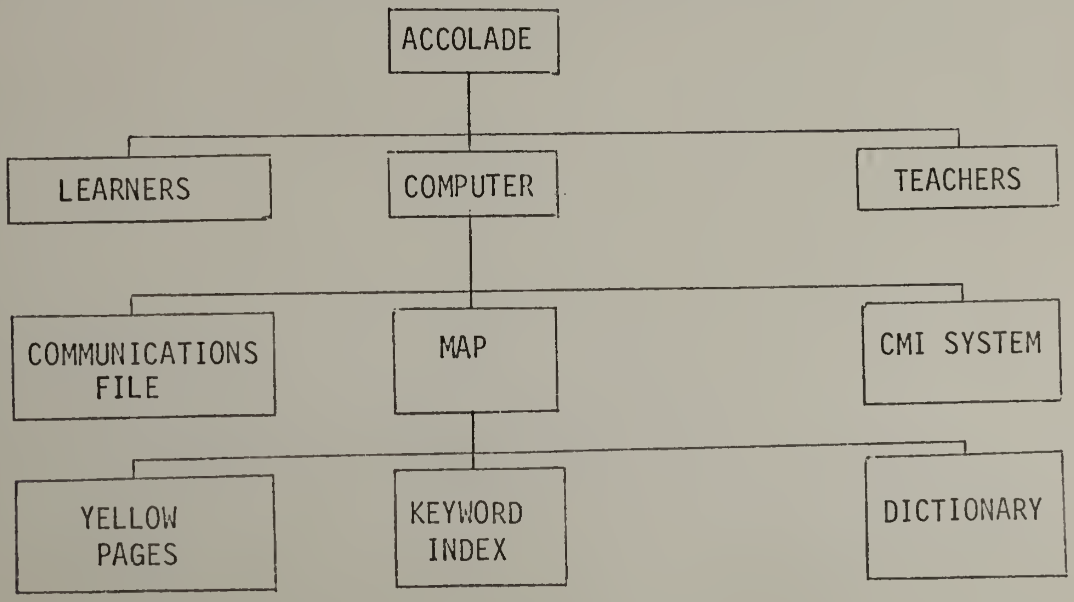
about what computers can and cannot do.

- To teach the skills and behaviors of logical thinking, problem solving, learning, confidence, and autonomy through the medium of computer programming.
- To be effective; i.e., relevant, easy-to-use, reasonably inexpensive in cost and time and, above all, enjoyable to Learners and Teachers.
- To be an adaptive system; i.e., a system that can modify its performance through interaction with its environment.

With the issues, spirit and goals of ACCOLADE in mind, let us now describe the components of the system, the function of each component, and how they interact with each other.

5.3 System Components

The three primary components of ACCOLADE are: Learners, Teachers and a Computer. The Computer is broken down into three parts: a Communications File, a Map, and a CMI System. The Map, in turn consists of three parts: a Yellow Page, a Dictionary, and a Keyword Index. The organization of the system is shown in the diagram below.



5.3.1 Learners

The Learner acquires computer literacy by appropriate interaction with the Teachers and the Computer. For instance, the Learner uses the Map part of the Computer to gain access to resources which will allow him to acquire knowledge about particular computer literacy topics. Additionally, the Map provides the Learner with a search tool for exploring and displaying the structure of the topics which comprise the knowledge space. The Communications File is a tool which allows the Learner to converse with Teachers and other Learners. If the Learner wishes to accredit his knowledge, he may self-administer criterion-referenced tests via the CMI system, and, if he scores high enough, the Learner is invited to become a resource within the ACCOLADE system.

To gain the skills and behaviors of logical thinking, problem solving, learning, confidence and autonomy, the Learner works with other Learners, the computer, and a Teacher who uses a programming language to foster these skills and behaviors. This process is facilitated through the use of Peer Learning Groups and case-study problems in the Learner's discipline and interests.

5.3.2 Teachers

There are four types of Teachers in the ACCOLADE system: Coordinator, Programming Guide, Peer Learning Group Facilitator, and Terminal Room Consultant.

1) The Coordinator is responsible for coordinating the activities of the other Teachers in the system and is the initial contact with ACCOLADE for the Learner. At their initial meeting, the Coordinator interviews the Learner and gathers Learner profile

data such as age, sex, academic major, grade point average, measures of self-confidence, interests, and past history. On the basis of the profile data and the interview, the Coordinator and Learner negotiate an initial learning plan for the Learner.

The Initial Learning Plan (which is subject to continuous negotiation) specifies the computer literacy topics to be mastered, the approximate times to be spent mastering those topics, the evaluation methods (if any) and includes recommendations for a Programming Guide, a Peer Learning Group, and the Terminal Room Consultant for the Learner. The plan also suggests an initial value for a set of depth/breadth-of-presentation parameters for the Learner. These parameters govern the number of topics and relations seen in the Map as well as the amount of help the Learner will be given when he requests assistance from the Computer. Future meetings between Learner and Coordinator and reports from the CMI System will serve to adjust the depth/breadth parameters. A sample plan is described in section 5.4.1.

The Coordinator is also responsible for maintaining the resources in the Yellow Pages, the computer literacy topics in the Map, and the test questions in the CMI system. The Coordinator used information supplied by the computer to aid in forming mental methods of the learners; this aspect is illustrated in section 5.4.

2) The Programming Guide is responsible for developing in the Learner the skills and behaviors of computer programing. He may do this through traditional methods such as formal lectures and norm-referenced tests, but he is encouraged to employ learner-

directed methods to conform with the system goal of allowing the Learner to take responsibility for his own education. Using this self-directed approach, the Guide assigns interesting, practical programming problems from within the Learner's chosen discipline and meets with the Learner on a periodic basis for intensive one-on-one training sessions. Since one of the skills the Guide wants to teach the Learner is "how to learn," the Guide does not act as a tutor; instead he critiques the Learner's work and progress in terms of the skills and behaviors and makes recommendations to help the Learner improve those skills and behaviors.

3) The Peer Learning Group Facilitator is responsible for helping the learner acquire problem solving skills. A workbook of computer programming problems is provided to the learning group. Using Whimbey's approach, (discussed in more detail in the section 4.4.2.1) the Learners form listener-speaker dyads and attempt to bring their problem-solving heuristics and strategies up to a conscious level by vocalizing and listening with care. The Facilitator constantly encourages the learners to not be afraid to make mistakes (to "look dumb") and to perserve (to "hang in there").

4) The Terminal Room Consultant is responsible for the initial Learner contact with the Computer part of ACCOLADE and is always available if Learners have questions regarding the operation of the Computer and the computer terminals. We assume here that the Learner has access to the Computer remotely via graphics terminals and that most of these terminals are centralized in one

physical location for the Learners' use. This central location can be the college or university library since many of the resources for acquiring computer literacy knowledge will also reside there.

5.3.3 The Computer

The Computer is used by the other two parts of the ACCOLADE system--the Learners and the Teachers. It is used by the Learner for two purposes: (1) to acquire the programming skills and behaviors and (2) to acquire computer literacy knowledge; to this end, each component of the Computer is self-instructing and able to engage in a helpful interactive dialogue with the Learner. It is used by the Coordinator to monitor Learner-Computer interactions, Learner test progress, to provide an item analysis of the question data base in the CMI system, and to assist the Coordinator in forming models of each Learner in the system.

The Computer part of ACCOLADE consists of three components: The Communications File, the Map, and the CMI System.

5.3.3.1 The Communications File

The Communications File allows communication via the Computer between Learners and Learners, Learners and Teachers, and Teachers and Teachers. It functions in two modes: as a Mailbox or as a Bulletin Board.

As a Mailbox it allows the Learner and Teacher to write and respond to personal messages to one another. The contents of the communication are private and only for the eyes of the sender and receiver of the messages.

As a Bulletin Board, messages and responses to the messages can be viewed by everyone. Teacher and Learner may use this mechanism to announce events of interest (such as a visiting speaker or an upcoming

Star Trek convention), or to make comments on the educational effectiveness of any of the components in the system.

5.3.3.2 The Map

The Map allows the Learner to take responsibility for acquiring computer literacy knowledge. The Computer Map is analogous to a road map: a road map not only provides the user with a search tool for getting from one place to another, it also shows the geographic structure of an area. One might not choose the shortest route from Boston to Washington, D.C. but instead choose the most scenic or the cheapest route. Thus, the computer Map has two functions:

1) To be a search tool for acquiring knowledge about particular computer literacy topics.
2) To reveal the structure of the computer literacy knowledge space by showing the relationships between the topics.

The Map consists of three parts: A Yellow Pages, A Semantic Network, and a Keyword Index.

5.3.3.2.1 The Yellow Pages

The Yellow Pages is a hierarchical, declarative representation of knowledge as described in Section 3.2.2.2.1 and is patterned after the telephone company's Yellow Pages. While the Yellow Pages of the telephone company are a hierarchical tree of consumer products and resources, the Yellow Pages of the Map constitute a tree of computer literacy topics where the terminal nodes contain a list of resources for acquiring knowledge about the topic that node represents. The resources are of seven types:

1) Printed Material (books, magazines, journals, etc.)

2) University and College Courses

3) Computer Assisted Instruction (CAI) Lessons

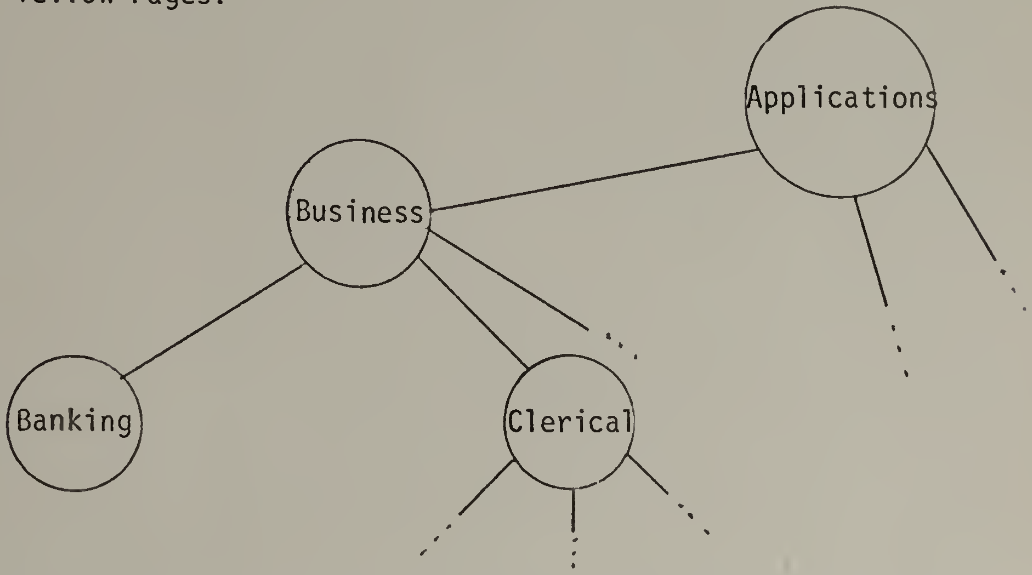
4) People

5) Movies

6) Videotapes

7) Audiotapes

Therefore the resource data base consists of a dynamic set of the above seven resource types, and each terminal node topic in the Yellow Pages contains references or pointers to the appropriate subset of the resource data base. For example, the following diagram represents a subtree of the Yellow Pages:



'Banking' is a terminal node and, as such, has associated with it a set of resources (books, courses, people, etc.) for acquiring knowledge about the topic of banking as a business application. Appendix B contains a list of suggested computer literacy topics that were synthesized from the author's experience and ACM suggested topics.[6,7] Appendix C contains a list of resources for acquiring knowledge about those topics.

In order to assist the learner in selecting appropriate resources, the Yellow Pages afford:

1) Learner effectiveness ratings (on a scale from 1 to 5) supplied by the ACCOLADE Learners. Not only can the Learner request to see the cumulative average Learner rating for any particular resource, he can himself enter a rating which will modify the average rating for that resource.
2) Teacher ratings in terms of effectiveness and degree of difficulty.
3) Expected times of completion for each resource--supplied by the Teachers and compared empirically with actual Learner experience.
4) An abstract (supplied by the Teacher) of the content of each resource. The ratings not only allow the Learner to select resources appropriate to his learning style, they provide a rationale for the Coordinator to use in maintaining the resource data base---when new resources should be added, and computer memory space is scarce, older resources with very low ratings can be displaced first.

[6] "Categories of the Computer Sciences," CACM, Vol. 15, No. 2, Feb., 1972.

[7] McCracken, D., et al., "A Problem List of Issues Concerning Computers and Public Policy," by the ACM Committee on Computers and Public Policy, CACM, Vol. 17, Sept., 1974, p. 9.

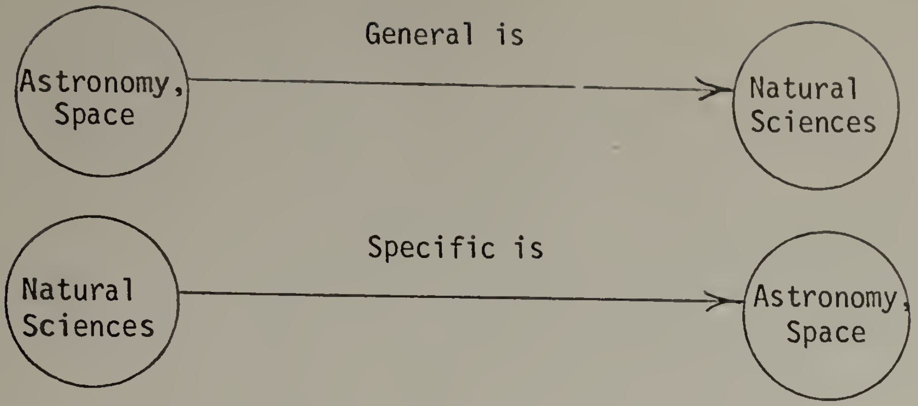
Although a Yellow Pages representation of computer literacy knowledge may be quite tractable for Learner use (most people have considerable experience using a Yellow Pages index), and is a natural way of providing access to the resources, it does not illuminate much of the structure of the knowledge space since it can only show the 'general-specific' relationship between topics. As described in Section 3.2.2.2.1, this shortcoming is overcome by the use of a semantic information network.

5.3.3.2.2 The Dictionary

The Dictionary is basically a semantic network discussed in Section 3.2.2.2.2 as a heterarchical representation of knowledge. It is called the Dictionary for two reasons: (1) "Dictionary" is a less foreboding term for the beginning Learner than "Semantic Network" and (2) in addition to being a graph representation of the computer literacy topics and the relations between them, it contains short definitions of each topic. The topics are the same ones as in the Yellow Pages and are listed in Appendix B. Since the Dictionary is a semantic network, it may contain more relations between the topics then just the 'general-specific' of the Yellow Pages. Seven relation types are used:

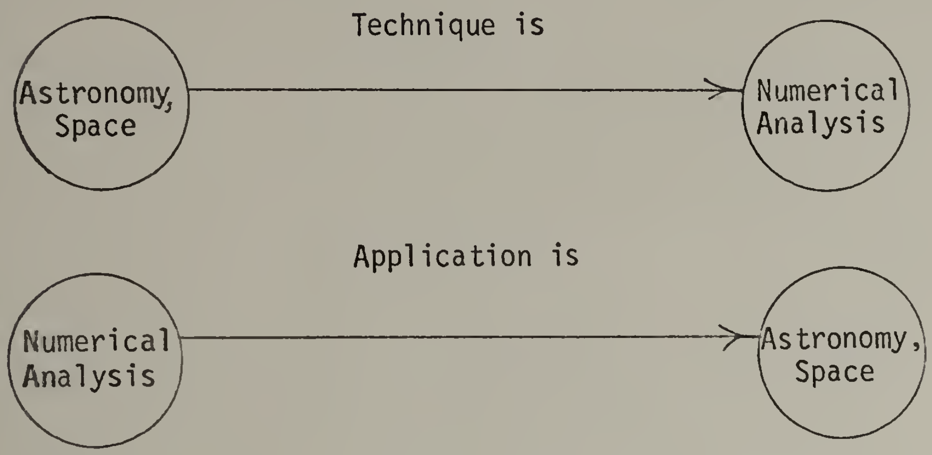
1) General-Specific is an inverse relation-pair where 'general' implies a more abstract categorization and 'specific' implies a more concrete one. If topic A is the general case for topic B, then topic B is a specific instance of topic A.

   Example:



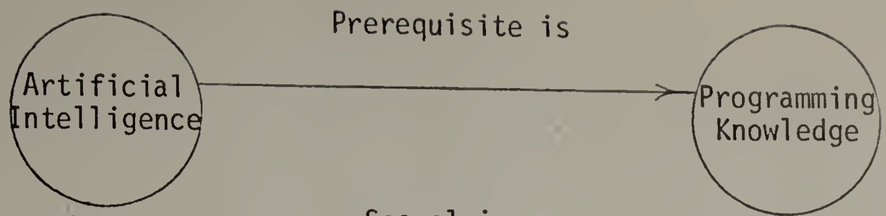
2) Technique-Application is an inverse relation-pair where 'technique' refers to a body of knowledge or set of technical skills and methods that are utilized for a particular purpose or 'application.' If topic A is a technique used in the application of topic B, then topic B is an application of topic A.

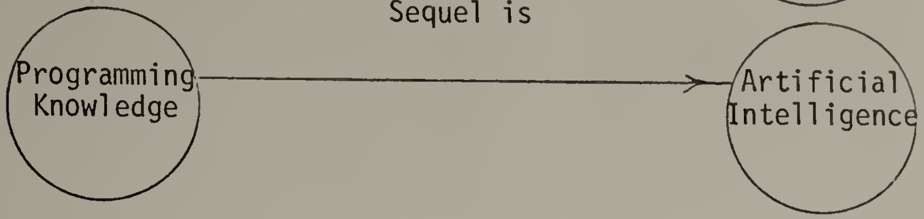
Example:



3) Prerequisite-Sequel is an inverse relation-pair where 'prerequisite' suggests that certain knowledge or skills should be mastered before the 'sequel' topic is attempted by the Learner. If topic A is a prerequisite for topic B, then topic B is a sequel to topic A.

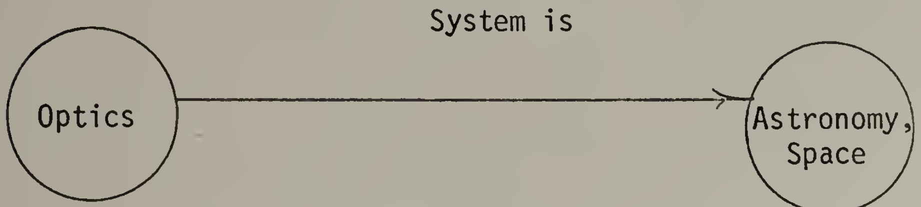
Example:

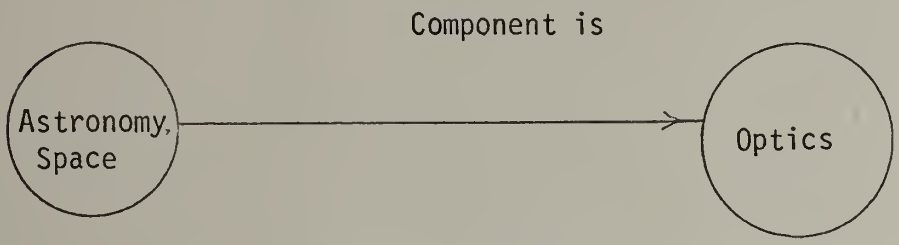




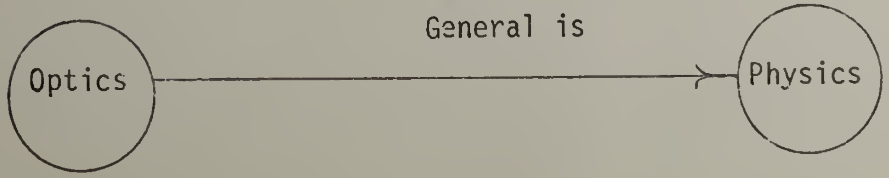
4) System-Component is an inverse relation-pair where 'system' is a set of interdependent 'components' such that each component exerts an influence on the operation or definition of every other component and the components taken as a whole comprise the system. If topic A is a system containing topic B, then topic B is a component of topic A.

Example:



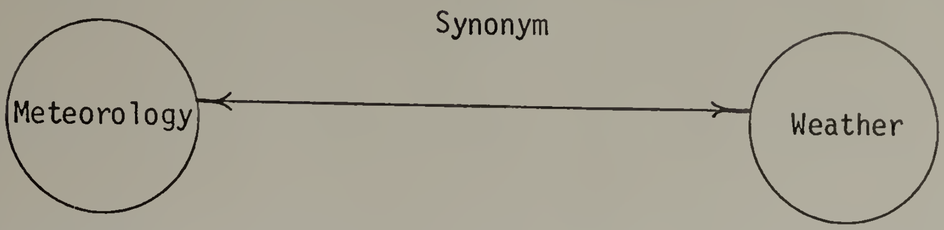


To distinguish this relation pair from 'general-specific' note that the 'general' for optics in physics:



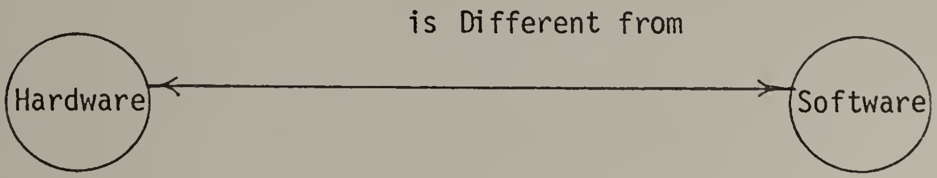
5) Synonym is a symmetric relation which implies equality or equivalence between two topics. If topic A is a synomym for topic B, then topic B is a synonym for topic A.

Example:



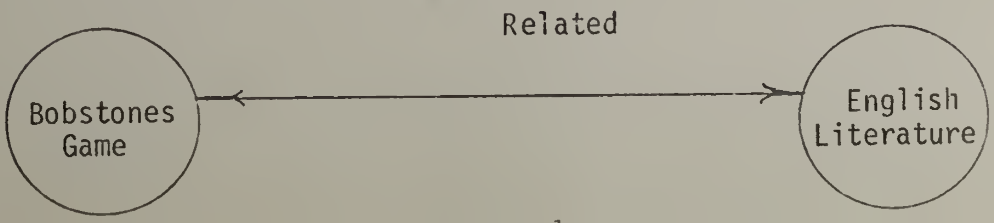
6) Different is a symmetric relation which supplies information by counterexample and implies that two topics are different enough to be considered antonyms for each other. If topic A is different from topic B, then topic B is different from topic A.

Example:

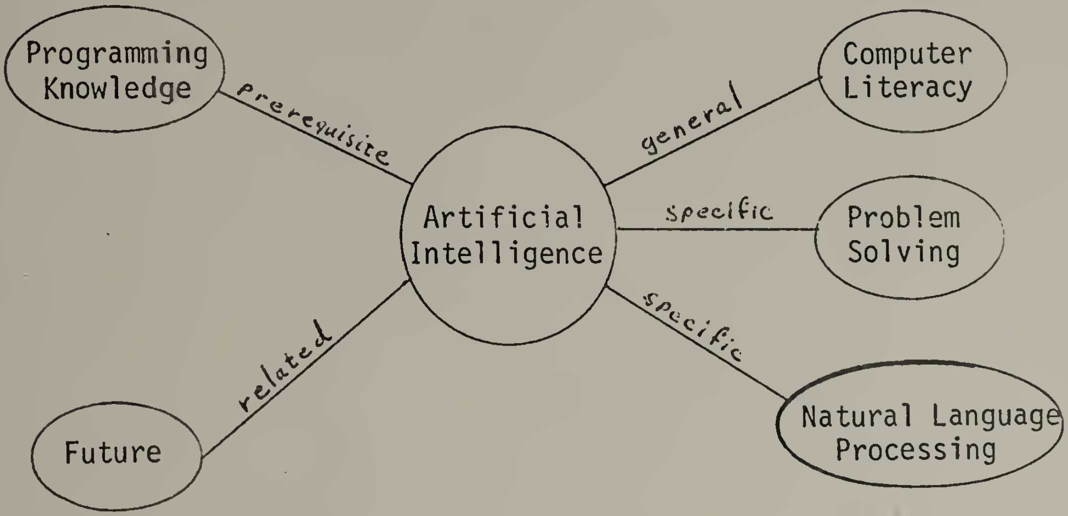


7) Related is a symmetric relation used when it is clear that there is some sort of relation between two topics but that one cannot yet specify it. At a later time, the relation may be identified and would replace the last-resort 'related' relation. If topic A is related to topic B, then topic B is related to topic A.

Example:



Since a semantic network with N nodes and M relations can have $\frac{NM}{2}(N-1)$ possible associations (which means a net with 100 nodes and 7 relations could contain as many as 34,650 possible connections), it is necessary to devise a method to show the inherent structure without overwhelming the user with too much information. Therefore, instead of displaying the whole Dictionary or even two levels of relations as Eland does in his Guide system[8] the Dictionary shows only the structure around one node or computer literacy topic at a time. For example, if the Learner wishes to see the topics one relationship away from 'artificial intelligence' in the computer literacy knowledge space, he is shown a display similar to the diagram below:



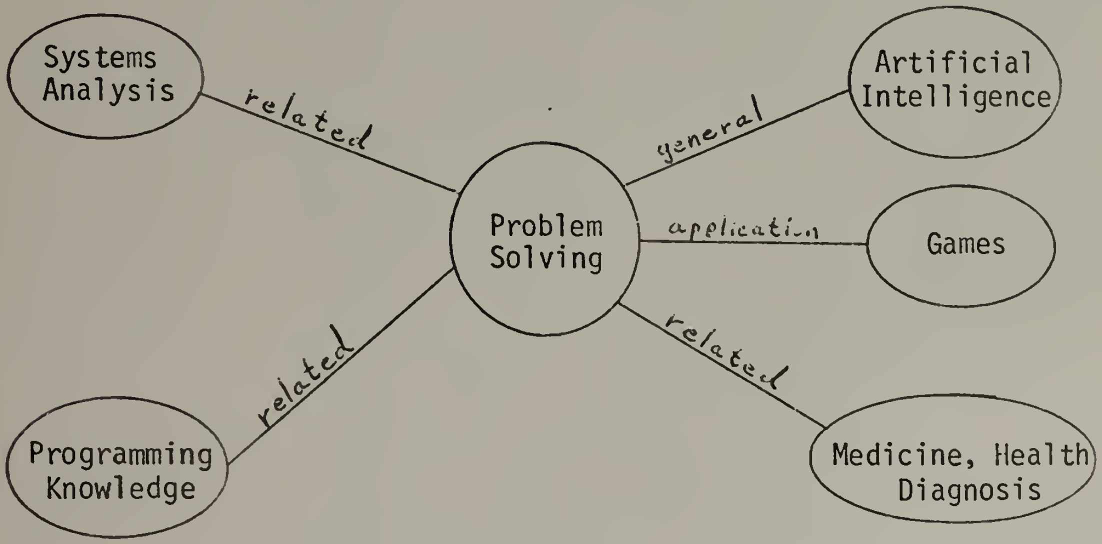
There are no arrowheads on the relation arcs between topics since the convention followed is to read from the central topic outward. For

[8] Eland, D.R., An Information and Advising System for an Automated Introductory Computer Science Course, Dept. of Computer Science Doc. #UIUCDCS-R-75-788, University of Illinois, June, 1975.

example, in the above diagram we read:

> "For the central topic of artificial intelligence:
> (1) the general topic is Computer Literacy
> (2) a specific instance is Problem Solving
> (3) another specific instance is Natural Language Processing
> (4) a related topic is the Future
> (5) a prerequisite topic is Programming Knowledge."

Next the Learner may request to see the structure around one of the noncentral topics such as 'Problem Solving' and that would place 'Problem Solving' in the center thusly:



In this manner, the Learner may see all of the topics in computer literacy knowledge space and all of the relationships between them. The viewing mechanism is similar to a spotlight--the entire Dictionary resides in the darkness inside the Computer, but the Learner may shine his spotlight on any one topic and see the topics in the immediate neighborhood (one relationship away). Topics n-relations away could be seen by increasing the size of the spotlight. The depth/breadth parameter associated with each Learner is also used to delimit the amount of information displayed. For example, the Learner may set this parameter so that he sees only one of the seven relations on any display, or he may set it to include any proper subset of the relation set. Similarly, he may reduce the topics displayed to one particular area such as Business or Engineering. In this way, the Learner can discover the structure of computer literacy knowledge without being overwhelmed by it. An illustrative use of this and other features of the Computer is shown in section 6.4.

One of the seven relationships in the dictionary is deemed to be important and useful enough to warrant special attention: the symmetric 'synonym' relation has been broken out of the Dictionary and expanded to become the Keyword Index---the third component of the Map.

5.3.3.2.3 The Keyword Index

The Keyword Index contains the names of all the topics in the Yellow Pages and the Dictionary along with synonyms for each name. For example, the topic 'meteorology' is synonomous with 'weather' and even 'whether' (a possible Learner misspelling). The reason for expanding the 'synonym' relation of the Dictionary into a separate component of the Map is to aid the Learner in his search for computer literacy knowledge. He may go directly to the Keyword Index and type in a request such as:

"Please tell me about weather applications of computers"

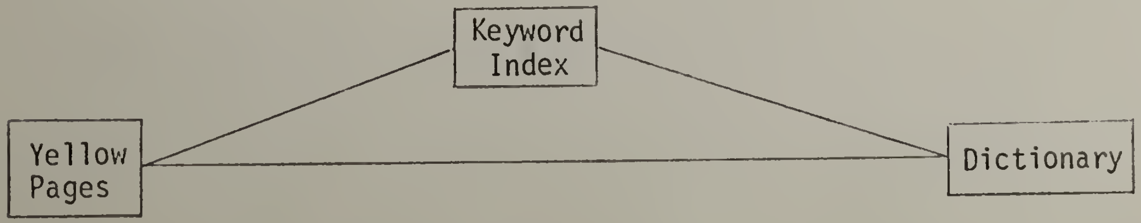
and the Keyword Index will send him directly to that topic in the Yellow Pages; if it is a terminal topic he may view the resources for acquiring knowledge about meteorology as it applies to computers.

If a keyword request is unrecognized, it is saved by the Computer. When that request appears three times or more across the total Learner population, it is brought to the attention of the Coordinator who must decide whether to incorporate it into the Keyword Index.

They Keyword Index can also save the Learner search time through the Yellow Pages tree or the Dictionary graph. Suppose for example, yesterday the Learner was viewing the topics surrounding 'meteorology' in the Dictionary. Today he wishes to get back to that display but cannot remember exactly how he got there. Rather than spending time searching, he may leave the Dictionary, enter the Keyword Index, type in the word 'meteorology' and be sent directly to the correct display in the Dictionary--where meteorology is the central node and all topics one relation away are displayed. This is possible since the Keyword Index is linked to both the Yellow Pages and the Dictionary: if a Learner requests a Keyword topic from the Yellow Pages, he is returned to that topic in the Yellow Pages; if he requests from the Dictionary, he is returned there; if he enters the Keyword Index first, the default is to the Yellow Pages.

Finally, the Yellow Pages are also linked to the Dictionary so that all three components of the Map can be entered from any of the other two. For example, the learner may be exploring the structure in the Dictionary and is "shining his spotlight" on the topic 'Banking'---with one keypress of his computer terminal he is transferred to 'Banking' in the Yellow Pages so that he may view the resources for acquiring knowledge about that topic.

Thus the Map, consisting of the Yellow Pages, the Dictionary, and the Keyword Index and connected as shown in the diagram below:



Thus, the Map fulfills its stated functions:

1) to act as a search tool for acquiring knowledge about a particular topic in the computer literacy knowledge source and

2) to reveal the structure of that knowledge space

and allows the learner to become a "navigator"[9] through the relational data base of the Map.

5.3.3.3 Computer Managed Instruction (CMI)

While many traditional CMI systems usually give a student a test to determine where to place him in the curriculum and perhaps even which types of learning resources to recommend, the ACCOLADE CMI system returns these functions back to the Learner.

The CMI system has three functions:

1) Generating and Grading tests

2) Diagnosing unmastered objectives and Prescribing remediating resources.

3) Recording and Reporting Learner-Computer Transactions, Learner Test Progress, and an Item Analysis of the test question data base.

5.3.3.3.1 Test Generation and Grading

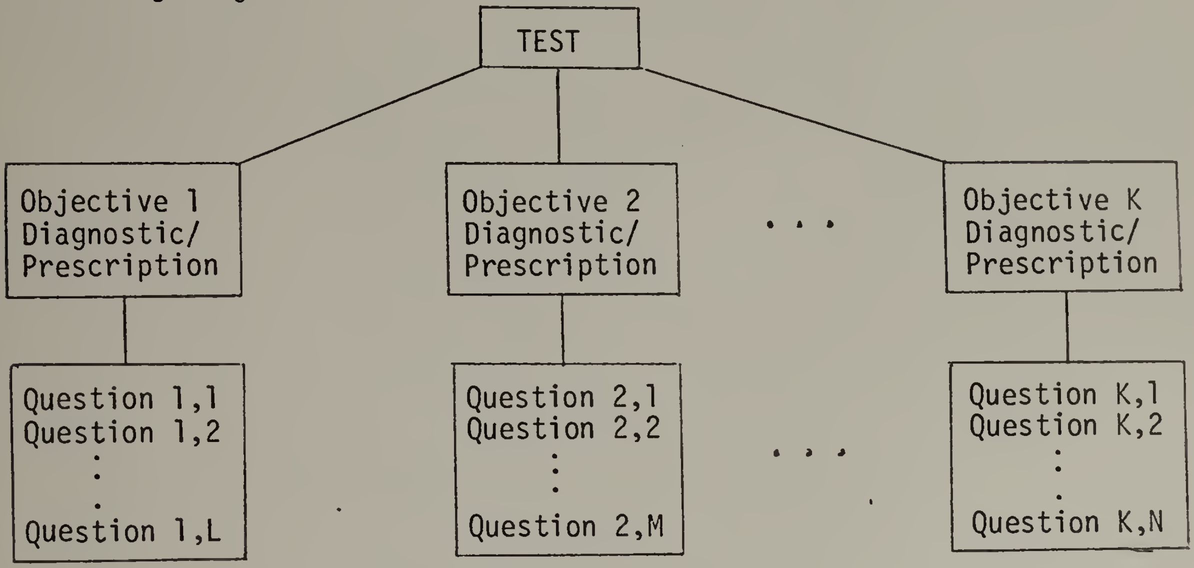
Whenever the Learner feels he has mastered the knowledge about a particular computer literacy topic, he may request a test of his knowledge. The CMI system generates a criterion-referenced test, the Learner takes the test, and the CMI system returns a grade (mastery or not)

[9] Bachman, C.W., "The Programmer as Navigator," 1973 ACM Turing Lecture, CACM, Vol. 16, No. 11, Nov., 1973, pp. 653-658.

at the Learner's request. The Learner may take a test on a topic as many times as he likes, subject to an upper bound specified by the Teacher.

5.3.3.3.1 Diagnosis and Prescription

Since the tests are criterion-referenced, it is possible to determine which objectives the Learner has not yet mastered (diagnosis) and recommend a set of resources (prescriptions) to correct the situation. The structure of a criterion-referenced test can be illustrated by the following diagram:



A test is a collection of related objectives where each objective has associated with it a diagnostic and a prescription in terms of the resources necessary to master that objective. Each objective has associated with it a pool of questions which measure mastery of the objective. The collection of all question pools across all test constitutes the question data base.

The Teacher in charge of this test specifies the passing criterion for the test (how many of the K objectives must be mastered to pass the test) and the objective mastery criterion (how many questions are to be given and, of these, how many must be answered correctly to master the objective).

The questions are drawn randomly without replacement so that a learner never receives the same test questions twice; this implies a question pool "large enough" to stand repeated test administrations for the same student.

5.3.3.3.3 Recording and Reporting

The CMI system records and gathers data on the Learners as they interact with the Computer. These data are available for Teacher (usually the Coordinator) viewing via various types of management reports.

Learner-Computer Transactions

Learner-Computer Transactions are kept at two levels: by individual Learner and cumulative summaries across the total Learner population.

By individual Learner, the following types of reports are available:

1) Computer Usage as shown in the diagram below:

Learner Name:	J.A. Doe
Last on Computer:	05/31/77
Days:	7
Hours:	4
Sessions:	16

Using the data displayed in the above report, the Teacher can see that as of the time he requested this report, Learner J.A. Doe last visited the

Computer component of ACCOLADE on 05/31/77. Doe has used the Computer on 7 different days, for a total of 4 hours, doing 16 sessions with the Computer. The Teacher can calculate that Doe, on the average, uses the Computer for 15 minutes per session (hours÷sessions), about one-half hour for each day on the computer (hours÷days) and about 2 sessions per day. The Learner's usage can give the Teacher a measure of the Learner's interest and determination in using the Computer to acquire computer literacy knowledge.

2) Total Visits and Time of Visit to the Computer components of ACCOLADE as shown below:

Learner Name: J.A. Doe

Component	Total Visits	Total Time (min)
Yellow Pages	12	80
Dictionary	6	60
Keyword Index	18	20
Communications File	25	60
Tests	2	20
Totals	63	240

Here the Teacher may infer which parts of the Computer are of most interest and usefulness to the Learner.

3) Computer Path and Visits to Specific topics in the Yellow Pages and the Dictionary.

The Computer Path is a chronological account of the Learner's interaction with the Computer. Where the previous report showed only the summary data for the Learner's path through the components of the Computer, the Computer Path shows a sequence of snapshots of the Learner's progress as he moves through the five main parts of the Computer. Additionally, specific topics visited in the Yellow Pages and Dictionary are also recorded as shown below:

Learner Name: J.A. Doe

Date	Time	Component Visited	Topic (if applicable)
5/31/77	0132	Keyword Index	Meteorology
5/31/77	0133	Yellow Pages	Meteorology
5/31/77	0134	Yellow Pages	Natural Sciences
5/31/77	0135	Yellow Pages	Mathematics
5/31/77	0137	Dictionary	Mathematics
⋮		⋮	⋮

The Teacher may also specify only selected data in the path display, e.g., just this Learner's interactions with the Dictionary, just Mathematics, or just between the dates of 5/28/77 and 5/31/77. Cumulative visits to each topic in the Dictionary and Yellow Pages may also be displayed:

Learner Name: J.A. Doe

Topic	Dictionary	Yellow Pages
Computer Literacy	12	6
Applications	4	2
Natural Sciences	3	0
.	.	.
.	.	.
.	.	.

Careful viewing and analysis of the Computer Path can help the Teacher form a better model of the Learner: which topics seem to interest him most and whether he is a depth-first or breadth-first searcher. (See also section 7.2 for a discussion of the Learner model.) If the Teacher sees similarities in the paths of two Learners, he may recommend similar types of resources for them or perhaps that the Learners meet one another. Finally, the Computer Paths across the total Learner population could be used to form composite paths based on specific profile characteristics in order to recommend paths to individual Learners when requested. For example, a Learner may ask the Teacher for a suggested path to follow for someone "like himself." If the Learner is an engineer, female, age 23, etc., etc., the CMI system can display the "average" path for the composite Learner type requested.

Across the total Learner population the same three types of reports as described above are available by selected groups of Learners and/or by averages. To illustrate, consider the Computer Usage Report by a Selected Group of Learners:

Name	Last on Computer	Days	Hours	Sessions
J.A. Doe	5/31/77	7	4	16
B.P. Han	5/18/77	4	4	7
C.A. Moe	5/20/77	3	3	6
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
Average		5	3.1	8

Learner Test Progress

Learner Test Progress Reports are also kept at two levels: by individual Learner and across all Learners in ACCOLADE. That is, the Teacher may request to see the progress of a particular learner and would see the following type of report.

1) Summary History of Learner Test Progress

Learner Name: J.A. Doe	
Test	Status
1	P
2	F,F,P
3	F,F,F
4	*
5	*

Where 'P' means Passed and 'F' means not yet passed and '*' means the Learner has not taken that test yet. The Teacher may also pinpoint a particular test and request to see more detailed information; for example, he may wish to have more information about Test 3 for J.A. Doe since Doe has not passed that test in three tries. In this instance he would see the following type of report.

2) Detailed History of Learner Test Progress

Test 3 for J.A. Doe

Date	Question No.	P/F	Time (sec)
3/3/77	14	P	15
	16	F	62
	18	F	82
	27	F	89
	29	F	93
3/4/77	15	F	10
	17	F	15
	19	F	10
	20	F	12
	21	F	13
3/8/77	13	F	10
	24	P	10
	25	F	9
	26	F	7
	27	F	7

This report allows the Teacher to see when the Learner took each of the three tests, which questions were drawn from the question pool, whether or not the Learner answered those questions correctly, and how much time the Learner spent on each question. This data could be useful in a Teacher-Learner conference.

If the Teacher is interested in comparing a particular Learner's test progress with the rest of the Learner population he may request the following type of report:

3) Total Learner Population Test Progress

Learner	Test Number 1	2	3	4	5
Doe, J.A.	P	P	F	*	*
Evans, P.B.	P	P	P	P	*
Farley, S.S.	F	*	*	*	*

where the entries represent the last grade on a particular test.

Item Analysis

The Item Analysis is available to the Teacher so that he may identify overly difficult, tricky, weak or ambiguous questions in the question data base and take steps to rectify the situation. A sample Item Analysis is shown below:

Question Number	All			Passes Only			Fails Only			Avg. Time (sec)
	R	W	No.	R	W	No.	R	W	No.	
1	100%	0%	55	100%	0%	53	100%	0%	2	5
2	50%	50%	100	100%	0%	50	0%	100%	50	27
3	50%	50%	100	0%	100%	50	100%	0%	50	136
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.

The first column contains the question number that is being analyzed. The next three columns shows how all learners who received that question fared on it--shown is the percentage who got it right, wrong and the total number of Learners. The next three columns show the same information but only for those Learners that also passed the test of which this question was a part. The next three columns show the same information again, but this time for those who did not pass the test. The last column shows the cumulative average time spent on this question.

Looking at the figures in the sample item analysis, the Teacher might conclude that:

1) Question No. 1 is too easy since everyone (55 different students) is getting it correct and the time to solve the problem is very short.

2) Question No. 2 is a highly-positive discriminating question since only the Passes are getting it right and the Learners who fail the test are also failing this question. In other words, this question is highly representative of the total test.

3) Question No. 3 is a highly-negative discriminating question since all of the Passes are getting it wrong and the Fails are getting it right. This coupled with the large value for the average time spent on the question warrants a closer investigation of this question by the Teacher.

5.4 Scenario

A scenario will perhaps clarify the relationships and illustrate some possible interactions between Learner, Teacher and Computer.

5.4.1 Learner View

Mary Mary, a sophomore at the Western New England Multiuniversity, has decided she wants to "know more about computers" but she is not sure where to start or even what it means, so she elects to enroll in ACCOLADE, having been told by a friend that it was enjoyable for her.

Mary's entry into the ACCOLADE system consists of an initial interview with one of the Teachers in the system, Judith Friendly, an ACCOLADE Coordinator. For the first fifteen minutes Friendly explains the goals of ACCOLADE to Mary and makes sure she understands that the system is primarily learner-directed. Mary indicates that she feels comfortable with the self-directed mode of learning and so Friendly invites Mary to view a fifteen minute videotape which explains ACCOLADE in greater detail and illustrates a typical Learner interaction with the system. After the videotape presentation Mary indicates interest in using ACCOLADE to pursue her computer literacy goals and so Friendly begins the in-depth portion of the interview. As Friendly gathers Mary's profile information (age, sex, etc.) she begins to form a model of her as a Learner in the ACCOLADE system.

At the close of the interview Mary and Friendly have negotiated and agreed upon an Initial Learning Program (subject to renegotiation and change) which specifies that Mary will begin her ACCOLADE experience by exploring the topics of management, marketing and systems analysis as business applications of computers. Mary's first inclination is to spend approximately six hours per week for 14 weeks acquiring knowledge in these computer literacy topics in return for three multiuniversity credits.

The Learning Plan also includes the evaluative procedure for gaining the three credits. Two of the three credits will come from successful acquisition of the programming skills and behaviors under the guidance of Sara Swift--a professor of computer science that Friendly judges to be a particularly good Teacher-Learner match-up for both Mary and Swift. The other credit will be gained when Mary has passed three criterion-referenced tests in the areas of marketing, management and systems analysis. Friendly furnishes Mary with a list of objectives for each of the topics and informs her that each test will be ten questions long and that she must pass eight of these questions to demonstrate mastery of the topic. Mary may take a test whenever she feels ready and can retake a test up to three times---after that she must meet again with Friendly to try to determine what the problem is.

The Initial Learning Plan also recommends that Mary join Peer Learning Group No. 19, which meets Tuesday evenings from 11 PM to 1 AM (since she has indicated in the interview that this is a time that suits her learning style) for the purpose of sharpening her problem solving skills. This Peer Learning Group is facilitated by Ms. Sue Smart, an expert in communication skills.

Finally, the learning plan has set the depth/breadth parameters for Mary at their lowest levels so she will receive the greatest possible amount of help from the Computer part of ACCOLADE when she requests it and so that only business topics and the relations 'general-specific,' 'technique-application,' and 'different' will be displayed by the Map.

Before leaving, Mary is reminded that the learning plan will most likely change with time since she will be changing in knowledge, interests and confidence and since the Coordinator, Ms. Friendly will be forming a more comprehensive and useful model of Mary as a Learner. In any case, Mary is also free to override the depth/breadth parameters herself at any time and explore any part of Map that interests her as she forms her own model of herself.

Friendly makes two appointments for Mary--one with Sara Swift, the Programming Guide and one with Sally Shot, the Terminal Room Consultant for ACCOLADE. Mary and Friendly sign both the Initial Learning Plan and each keep one copy. Mary leaves Friendly's office, somewhat excited to begin the next stage of her intellectual journey.

The next day finds Mary in the office of Dr.Swift. Mary is explaining that she knows nothing about programming let alone APL and that the prospect of learning a computer language is somewhat confusing and frightening. Swift nods in agreement having already glanced at Mary's profile provided to him by Friendly earlier that morning--both Swift and Mary are aware that Swift has read her profile, indeed in order for Swift to obtain it required Mary's permission. Swift reassures Mary that programming will probably not be terribly difficult and tells her of several dozen other students who came to ACCOLADE in situations similar to hers who had no problems and even enjoyed the experience. Swift recommends a case-study in the area of market forecasting: Mary will be given demographic data, including income patterns, transportation routes and real estate values and will be expected to write an APL program to predict the best placement for a store of her choice. Mary chooses a wine and cheese shop, tentatively beginning to accept the idea of writing a program to solve a problem of interest to her. Swift gives Mary the access numbers for the set of videotapes she has made to teach the rudiments of APL and sets up a meeting time with her on Mondays from 9-10 AM to discuss any programming problems she is having. Swift cautions Mary that she does not have the time to act as an APL tutor but that she will instead recommend resources and will help her to solve problems at their meeting times; additionally she reminds Mary that one of the purposes of the Peer Learning Groups is for the Learners to help each other learn to debug programs. Swift emphasizes that she is most interested in helping Mary achieve the skills and behaviors of logical thinking, problem solving, learning, confidence, and autonomy through the medium of computer programming. Mary leaves, ready to begin her encounter with ACCOLADE.

Two days later Mary meets Sally Shot at Computer Center in the Terminal Room. Shot is a teaching assistant who will give Mary her introductory demonstration of ACCOLADE. Shot shows Mary how to sign onto the computer terminal and request the ACCOLADE system; from that point on, as Shot indicates, ACCOLADE is self-explanatory--special function keys such as---HELP--are always active so that if Mary is confused she need only press HELP and enter into dialogue with ACCOLADE.

After showing Mary how to use the MAP, take tests, and use the Communications File on the terminal, Shot moves aside and lets Mary "play" with and explore the various components of ACCOLADE herself. Mary is confused at times and asks Shot for help; however, instead of tutoring her, Shot advises Mary to just press the HELP and ask ACCOLADE instead. After several times Mary begins instead to ask ACCOLADE. The answers provided are detailed but easy to read since they are suited to Mary's depth/breadth parameters. After about an hour Mary leaves the terminal room, excited about her first encounter with the Computer.

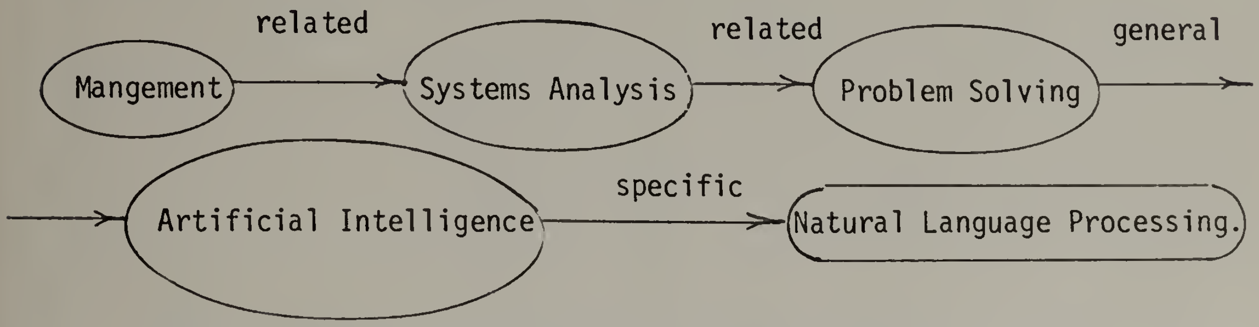
During the first week, Mary spends only 2 hours at the computer terminal because she must study for several tests in other subjects. From the second week on, Mary spends over 12 hours per week on the Computer in addition to the 2 hours in the Peer Learning Group and the hour with her Programming Guide. The Learning Plan has been modified to reflect this fact. Mary comes and goes as she pleases and she especially enjoys this aspect of ACCOLADE.

As Mary acquires knowledge about her three chosen topics, the computer invites her to rate the resources she has used---in terms of effectiveness, degree of difficulty, and ease of accessibility---so that Mary's ratings can be used by future ACCOLADE Learners.

After only six weeks Mary has completed the knowledge acquisition part of her contract having successfully passed the three tests on marketing, management and systems analysis--although she is still learning APL under the guidance of Professor Swift. She also feels quite comfortable using the computer part of ACCOLADE and as of the third week has been setting the depth/breadth parameters of the MAP by herself.

After she has completed the last test the Computer informs Mary that she has scored so high on the systems analysis test that she is eligible to become a resource for that topic if she chooses. Flattered, Marry accepts and supplies the Computer with her name, address and telephone number. The fact that Mary is now a resource for the topic 'systems analysis' is automatically posted by the Computer on the Bulletin Board for the next week. Next, Mary is requested to rate the ACCOLADE system itself so that the Teachers can receive constructive feedback from the Learners and modify the system if necessary.

After Mary has rated ACCOLADE, she is invited to use her terminal to play computer games if she so wishes. She is provided with an index to the games, some of which she can play against the Computer, some of which she can play either competitively or cooperatively with other users on the Computer system. Mary is intrigued by this prospect and in particular with a "twenty questions" type game which accepts her queries and responds in "natural language." Mary is aware that natural language processing is a research area in the field of artificial intelligence because one day while she was in the Dictionary part of ACCOLADE she travelled the relation path:



Mary is amazed to realize that this game program is behaving as if it understood her! She is now literate enough to realize that somehwere in the computer there is just a program that is processing her input and producing output--but the output appears intelligent! On her own, Mary returns to the Yellow Pages in ACCOLADE and finds resources for acquiring knowledge about natural language processing in the field of artificial intelligence.

Four years pass. As a result of her ACCOLADE experience, Mary has graduated with a bachelor's degree in Computer Science and taken several courses in Linguistics. She is now a graduate student in Computer Science Education working on her dissertation--her topic is the design and implementation of a speech understanding system which she plans to incorporate into ACCOLADE. The wheel has turned once--the computer has given Mary conceptual tools for her intellectual augmentation. Now Mary is upholding her end of the partnership by supplying the computer with artificial intelligence.

5.4.2 Teacher View

As we saw in the scenario from Mary's point of view, there are several types of Teachers in the ACCOLADE System. There is Mary's Coordinator--Judith Friendly, her Programming Guide--Sara Swift, the Terminal Room Consultant--Sally Shot, and her Peer Learning Group Facilitator--Sue Smart. Each week these four Teachers meet and spend approximately one-half hour discussing the progress of the Learners they have in common. Let us observe that part of the first meeting dealing with Mary's progress--it has been one week since Mary began ACCOLADE.

The Coordinator, Friendly, presents her report first. She passes out CMI system-generated copies of Mary's progress to the other Teachers. The Learner-Computer Transaction reports indicate that Mary has spent one day, two hours, and one session on the computer--indicating a lower than average usage. They also show that Mary is spending most of her time in the Yellow Pages and Keyword Index components of the Computer--a typical beginner's pattern since these parts are easy to use and have common analogies in the Learner's general world-experience. It is still too early to try to compare Mary's Computer Path with other Learners to see if her learning style is similar to any specific group of ACCOLADE Learners, however this analysis will be performed before the next Teacher's Meeeting. Since Mary has not yet taken any of the criterion-referenced tests at this early date, no Learner Test Progress reports are yet available.

Friendly finishes her report by noting that although Mary seemed to be alert and quick to follow, she showed a very low level of confidence at their initial interview meeting. Shot, the Terminal Room Consultant confirms this view, she indicates that although most students approach a computer rather tentatively, Mary seemed to be especially reticient. Shot goes on to note, however that Mary did seem to gain in self-confidence as the interaction proceeded and she was able to leave Mary for the last ten minutes of the initial session with ACCOLADE.

The Programming Guide, Sara Swift, disagrees somewhat in her impression of Mary. Although she appeared a bit confused at their first meeting, Swift indicates that this reaction is entirely normal for most Learners. Swift indicates that she will be seeing Mary tomorrow at their weekly meeting and will inquire about Mary's initial low computer usage. "I'll ask if she's having a specific problem, and if I can help," she says. Swift has nothing further to report regarding Mary's progress in the acquisition of the programming skills and behaviors since it is still too early to even speculate.

Sue Smart, the Peer Learning Group Facilitator, agrees with Friendly and Shot--Mary seems to be lacking confidence at this early stage in her ACCOLADE experience. Mary is experiencing difficulty in vocalizing the problems from the Programmers' Problem Solving Workbook with her partner. During the class, Smart talked with Mary and found that the problem seemed to be that Mary felt embarrassed when asked to talk in front of others. Smart reminded Mary that no one else was listening besides her problem-solving partner and that, soon it would be Mary's turn to play listener.

This advice seemed to be helpful for Mary--she began to loosen up a bit by the end of the session. Smart concludes her report with the assurance that she will observe Mary closely for the next few sessions and undertake to build Mary's self-confidence.

Based on the observations of the group, Friendly recommends that the Teachers spend a few extra minutes with Mary over the next two weeks and to report back to him if Mary's confidence does not seem to improve. Friendly also indicates that she will "keep taps on Mary's computer usage". Swift, not previously aware of Mary's low level of self-confidence, makes a note to that effect in her file for Mary and the group moves on to review the program of the next Learner on the agenda.

After the meeting is concluded, Friendly signs on to the Computer at the terminal in her office. The Coordinator has several things to do. First, she leaves Mary a note in her personal Mailbox to the effect that she is aware that Mary may be having problems and that all of the Teachers are willing to offer added assistance whenever Mary requests it.

Next, Friendly wants to enter a new resource (a recently published textbook) into the Yellow Pages but she is informed by the Computer that currently memory is very scarce and that she should consider dropping an older, low-rated, resource first; she is furnished a list of the least-used and lowest-rated resources, broken out by the same category of resource and for all resources. After reading this report, Friendly instructs the Computer to drop a movie that has received consistently low ratings from the Learners over the last year and that she herself has seen and had judged to be unorganized and overly detailed with much unexplained computer jargon. Now that there is room for the new resource, Friendly adds it, and with prompting from ACCOLADE, specifies which topics will use the resource.

Finally, Friendly requests an exception report of the item analysis on the question data base of the criterion-referenced testing system. She wishes to see all questions that are highly negatively disciminating for a Learner population greater than 100. Seven questions are produced on the hard-copy device and Friendly puts them in her briefcase for her afternoon meeting with her friend, Sylvia Fromm, an educational psychologist with a special interest in computer literacy--together they will attempt to rephrase or replace these questions.

Before she can sign off, the Computer reminds Friendly of the quarterly meeting of New England ACCOLADE Coordinators next week at Lake Placid.

5.4.3 Computer View

We shall discuss here several adaptive aspects within the design of ACCOLADE made possible by the Computer. Each component (Computer Teacher, Learner) adapts based on inputs received from the other two components.

After the initial interview with Mary, the Coordinator, Ms. Friendly, initialized her depth/breadth parameters, thus specifying the amount of information the Computer would furnish Mary. As time progressed, however, these parameters were adjusted both by Mary and Friendly as Mary learned more about ACCOLADE from the Teachers and the Computer, and Friendly learned more about Mary from Mary and the Computer.

The Computer furnishes Friendly with some of the information needed to form a Learner model of Mary in the form of the Computer Path Report.

From this report Friendly is able to discern that Mary is basically a depth-first searcher, preferring to exhaust the depth of a topic before exploring related topics. At their fourth meeting Friendly makes Mary aware of this fact and they spend over an hour discussing various searching heuristics and strategies as well as the affective personality characteristics and attitudes that seem to underly search strategies. Mary is curious to see a typical path based on her profile data and Friendly has the Computer display that path and gives Mary a copy to take away with her. Friendly indicates that this particular path has always correlated highly with good evaluation and continued interest in computer literacy.

The Computer also learns from Mary. After Mary has rated a resource, that information is incorporated into an overall cumulative average rating across the Learner population by the Computer; later Friendly will use these ratings to make maintenance decisions on the resource data base. Mary also supplies the Keyword Index section of the Computer with adaptive data: whenever a keyword is unrecognized not only is Mary informed, but that unrecognized keyword is saved by the Computer. When an unrecognized keyword appears three times or more, the computer notifies Friendly who must decide whether or not to incorporate this new keyword into the Computer. With Mary's permission, the Computer will supply Friendly with her resource ratings and her personal list of unrecognized keywords so that Friendly may continue to improve his Learner model of Mary.

The Computer keeps all data associated with Mary until she leaves the multiuniversity; at that time Friendly is notified and has the option of deleting her records or keeping them on file. In any case, the data that Mary has contributed to ACCOLADE as one of the Learner population has permanently modified the rest of the system--the Computer and the Teachers.

CHAPTER VI

AN IMPLEMENTATION OF THE ACCOLADE SYSTEM

6.1 Chapter Overview

This chapter will discuss a specific implementation of ACCOLADE as it was specified in the previous chapter and a pilot study of that implementation. The implementation is described in terms of the components of ACCOLADE: Learners, Teachers, and Computer and an illustrative session with a Learner is given to show more clearly the relationship between the components and how the system actually operates. The results of the implementation are interpreted from the viewpoints of the Learners, Teachers and the Implementor (the author).

6.2 Introduction

The term 'implementation' not only means the programming of the Computer part of ACCOLADE but carrying out a modest pilot study. The main purpose of the pilot study was to debug and improve the system and to explore its feasibility--not to measure its effectiveness. Therefore, a carefully controlled experiment testing the educational effectiveness of ACCOLADE was not performed but is suggested in the next chapter "Suggestions for Future Research and Development."

Methodology does not appear in this chapter as a separate section but instead is discussed under the appropriate ACCOLADE component. The methodology concerning the procedures used by the people in the ACCOLADE system is described in the sections on 'Learners' and 'Teachers'; the methodology used by the author in actually writing the necessary computer programs is described in the 'Computer' section.

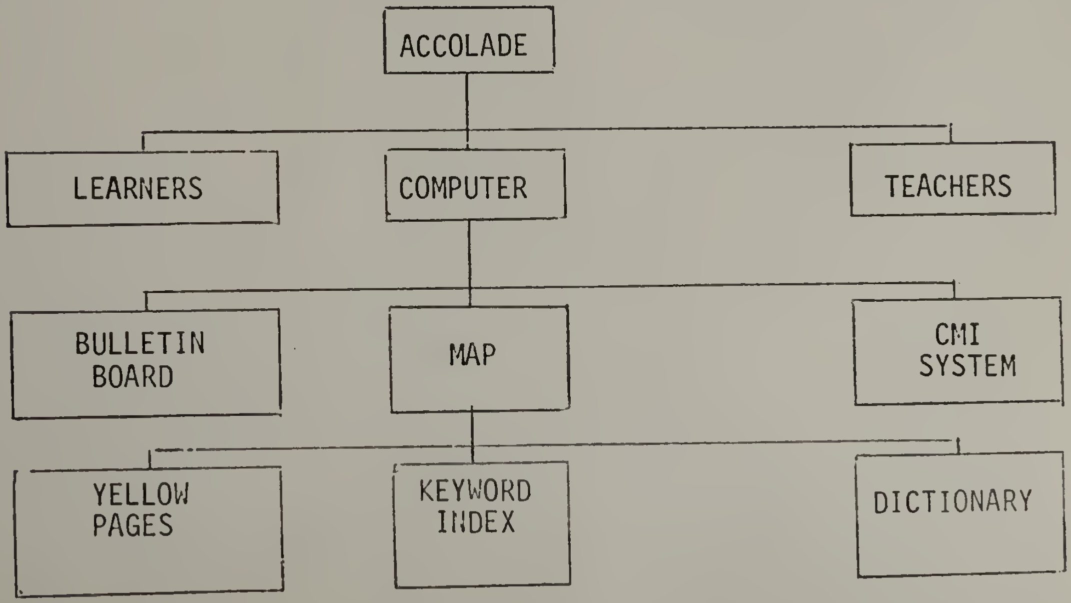
Although the design of ACCOLADE includes as a very important feature the splitting of the computer literacy curriculum into a knowledge part and a skills and behavior part, because of time constraints, only the knowledge part was implemented in detail. That is, the Computer components of ACCOLADE responsible for allowing the Learner to search out and understand computer literacy knowledge were almost completely implemented as specified in the design; however the skills and behaviors acquisition through the medium of computer programming and a Programming Guide--Teacher were not piloted-tested as effectively. It also become clear at the outset of pilot study that only one of the Learners had no previous programming experience and therefore it would be extremely tenuous to test the hypothesis that computer programming could build the skills and behaviors of logical thinking, problem solving, learning, confidence and autonomy using only one student as a population. However, two ACCOLADE Learners did choose to learn programming languages and that part of the study will be described under the 'Teachers' section of this chapter.

Finally, the topics comprising computer literacy knowledge were also prioritized in their development. Rather than attempting to provide knowledge for the full 224 topics suggested in Appendix B, the decision was made to concentrate on topics that dealt with computer applications. Since there would not be enough time to provide resources for all aspects of the topics (applications, hardware, history, software, social issues, education, future gaines, government, and artificial intelligence), the decision was made to choose the most important of these topics, namely applications and to show that in detail while only breaking down the others to a general or survey level. If only one of the main computer literacy topics had to be taught, it is the author's judgement that 'applications' would be the appropriate choice. Therefore 127 computer literacy topics are available in the MAP and these are denoted by an asterisk (*) in Appendix B.

Thus, the implementation of ACCOLADE is not the complete system described in the design, It should be considered instead a prototype, or perhaps, a molecule--the smallest part of a system that still exhibits the properties of the full system.

6.3 System Components

The ACCOLADE System comprises three components: Learners, Teachers, and a Computer. The Computer consists of three parts: A Bulletin Board, a Map, and a CMI System. The Map, in turn, consists of three parts: A Yellow Pages, a Dictionary, and a Keyword Index as shown in the diagram below:



6.3.1 Learners

Ten Learners enrolled in ACCOLADE at the University of Massachusetts, Amherst for the Spring semester, 1977; five Learners through the School of Education (EDUC 390/690) at the beginning of the semester and five from the School of Engineering (ENG 104) two-thirds through the semester having just finished a two-credit introductory FORTRAN course.

Most of the Learners elected to take only the knowledge part of ACCOLADE for one credit, however two Learners took a programming skills and behaviors module for one or two extra credits--only one of these two had no prior programming experience. The Learners who registered through the School of Education earned their credits on a pass-fail basis while those coming from Engineering were to earn grades of either A, B, or Incomplete. The evaluation was based on three criterion-referenced tests given by the CMI system and are discussed in that section. After a one-hour general introduction to the ACCOLADE system, the Learners signed up for individual one-hour introductory sessions with the Terminal Room Consultant who showed them how to sign onto the Computer and how to use the five components of the Computer: The Yellow Pages, Dictionary, Keyword Index, Bulletin Board, and CMI Testing System. Additionally, each Learner went through a 30 minute CAI lesson on the Computer which provided an introduction to the single terminal they were to use. From this point on, the Learners signed up for blocks of terminal time to pursue their acquisition of computer literacy knowledge. The terminal was available from 9 AM to 4 PM on Tuesdays, Wednesdays and Thursdays and each Learner generally reserved about two hours per week of this time. (As might be expected, many Learners waited until the end of the course to sign up for their time, so Mondays and Fridays were also opened up as the course drew to a close). The Terminal Room Consultant was present whenever a Learner was using the Computer.

The Learners could use the Computer part of ACCOLADE to acquire computer literacy knowledge or to play a wide range of computer games. After two months, the Coordinator (using statistics gathered by the CMI system) determined that fully three-quarters of the Learner Computer time was being spent playing games and at that time the game playing option was taken away. The Learners were told that when they completed the three evaluative tests, the game playing privileges would be returned. Only two of the ten Learners finished the tests in time to return to the games. Nine of the ten Learners passed the course with grades of Pass, A and B; one Learner withdrew from the course.

6.3.2 Teachers

During the pilot study, the role of the Coordinator and the Terminal Room Consultant was played by the author; the role of Programming Guide was played by Dr. Howard A. Peelle, Director, Instructional Application of Computers, School of Education, University of Massachusetts, Amherst. No Peer Learning Group Facilitator was available and the Learners did not attend any Peer Learning Group meetings.

Since both the Coordinator and Terminal Room Consultant were the author, he had ample opportunity to form models of each Learner in the pilot study. The author was present during all Learner interactions with the Computer and a detailed log book was kept to help in preparing the 'Observations, Results and Interpretations' section of this chapter. The author was present not just to observe Learner behaviors but to offer assistance when necessary. We also encouraged the Learner to think out loud while taking the criterion-referenced tests so that his thinking process could be observed and so that weak or ambigious questions could be improved.

The Programming Guide worked with two Learners. One, who had no previous programming experience elected to take a one-credit introduction to the APL programming language. Using a self-teaching workbook[1] and self-scheduled meetings between Learner and Guide, the Guide attempted to transmit the skills and behaviors of logical thinking, problem solving, learning, confidence and autonomy to the Learner. The case-study approach was used, and in this case the Learner choose to program a three-dimensional tic-tac-toe game. The criterion for earning the one credit was to tackle a problem of interest and to hand in a complete, working program. The Learner passed, earning one credit.

The second Learner had already learned BASIC and FORTRAN but wished to take a two-credit programming module through ACCOLADE so that he could learn APL. In addition to completing the introductory credit by writing a computer dating-matching program this Learner went on to master more advanced APL constructs by designing and writing a permutation algorithm. This Learner also earned his two credits.

The Coordinator and the Programming Guide did not hold weekly meetings reviewing the progress of the Learners they had in common. Instead the programming part of ACCOLADE was handled more as a separate

[1] Peelle, H.A., APL U-Programs (Poughkeepsie, N,Y.: SECOS, 1974).

module that could be elected by the Learner. At the end of the semester the Coordinator met with the Learners who had elected programming and the Programming Guide. While the Learners were split on whether programming had increased the desired skills and behaviors, the Guide felt that he had seen increased performance in all areas and especially in the behavior of autonomy.

6.3.3 The Computer

The Computer component of ACCOLADE was implemented on the Control Data Corporation (CDC) PLATO IV Computer System[2] using the TUTOR Programming Language.[3] The primary reason for the choice of this system was that it was readily available: the author, through the offices of Dr. John Dixon, procured a research assistantship funded by CDC to develop an ACCOLADE system. Other factors make the PLATO system an especially attractive medium for ACCOLADE:

1) Many CAI lessons are already developed and available as resources for a program of individualized instruction.

2) The PLATO terminal is capable of graphics, animations, and is highly interactive--it makes it easy for the user to explore a knowledge space. Special function keys, such as HELP further facilitate user convenience. Perhaps since the terminal acts so much like a TV set, Learner attention span seems to be

[2] Control Data Plato System Overview, Doc. #97406700, 1975, available from: Control Data Corp., Publications and Graphics Division, 4201 North Lexington Ave., St. Paul, Minn.

[3] Sherwood, B.A., The Tutor Language, Computer-based Education Research Laboratory, University of Illinois, Urbana, Ill., June, 1974.

enhanced, especially when compared to a conventional teletype terminal.

3) Powerful, easy-to-use, program editors are available which facilitate creation of programs (called "lessons" on the PLATO system) by the implementor (called an "author").

4) Some of the CMI recording and reporting functions described in Chapter V (section 5.3.3.3.3) already exist on PLATO and the others are relatively easy to implement since CMI was a design consideration of the TUTOR language and the PLATO system.

The primary disadvantage to PLATO is the costs--both in time and in dollars. It takes anywhere from 100-200 hours of author time to develop a one hour lesson for a Learner, and it costs over $1000 to produce and maintain that hour.[4] The terminal currently rents for approximately $800/month exclusive of communication line costs, but this is not out of line with present computer costs.[5]

Although the Computer part of ACCOLADE is logically organized as shown in the diagram in section 6.3, it is presented more simply to the Learner. After the Learner signs on to the PLATO system, he is automatically placed in a "Router" lesson where he sees displayed on his screen a menu of the following form:

[4] CDC Plato Lesson "aids," option V.

[5] If the terminal is available for 16 hours/day, this means over 400 hours/month bringing the Plato cost to approximately $2.00 per student contact hour.

1. Use the Yellow Pages
2. Use the Dictionary
3. Enter a Keyword request
4. Use the Bulletin Board
5. Take a Test
6. Leave PLATO

By typing the appropriate number, the Learner is routed to that part of ACCOLADE. When he is finished using a component, he is automatically returned to the Router to make another choice.

Therefore the Computer component implementations will be discussed in the following six sections as the Learner sees them, beginning with the router lesson itself. Before beginning these descriptions the reader should be aware of the special function keys on the PLATO keyboard that the Learner uses to move among the Computer components. Basically, there are five function Keys and a SHIFT key as shown below:

NEXT LAB SHIFT
BACK DATA
HELP

These function keys allow the Learner ten unique ways to respond and direct the search through the Computer components. For example, to move back one node in the Yellow Pages tree of computer literacy topics, the Learner just presses the BACK key; to move all the way back to the root node or front page of the Yellow Pages, he would press SHIFT-BACK (the shifted BACK key), to jump from the same topic in the Yellow Pages to the same topic in the Dictionary, he presses SHIFT-NEXT.

6.3.3.1 The Router

The Router has already been briefly described in the previous section---it is a special PLATO lesson that acts as a choice page or a central position from which the Learner conducts his exploration through the Computer parts of ACCOLADE.

Also available from the Router lesson is a "help lesson" which gives a short tutorial overview of the ACCOLADE system--to invoke this lesson, all the Learner need do is press the HELP key. This help lesson can also call a more primative help lesson--if this is the Learner's first time on PLATO, a lesson can be invoked (again by pressing the HELP key) that provides an introduction to the Plato terminal, keyboard and special function keys.

The Router also gathers data for the CMI system. Collected are the total Learner sessions (the time from a Learner sign on to a sign off the computer), cumulative visits to each computer component and cumulative times associated with those visits. Exactly what the Router page looks like to a Learner is shown in section 6.4 and the report showing total sessions, number and times of visits to specific components are shown in section 6.5.

6.3.3.2 Yellow Pages

What shall be described here is how the Plato lesson implementation of the Yellow Pages differs from the specifications given in the design section 5.3.3.2.1.

1) Of the 127 computer literacy topics identified for inclusion in the Yellow Pages, most fit under the topic of 'applications' as

explained in the introduction of this chapter. See the asterisked topics in Appendix B for a complete list.

2) The only ratings mechanism available are Learner ratings, no Teacher ratings, expected topic completion times or abstracts are furnished. Furthermore, the ratings are coarse when they could be handled in finer detail. For example, when a Learner rates a certain group of pages in a book as a resource for a particular computer literacy topic, the rating is assigned to the book as a whole rather than to just that part of the book that dealt with the topic. Although this imperfection is a suggested topic for further ACCOLADE development (in the next chapter), the coarse rating mechanism does serve the purpose of assisting Learners in selecting appropriate resources when the Learner-rater population is sufficiently large.

3) A help lesson explaining the detailed operation of the Yellow Pages is active upon pressing the HELP key.

4) The Yellow Pages gathers data for the CMI system. Collected are the Computer path of each topic visited and the time spent there for each Learner, and the cumulative number of visits to each topic across the total Learner population.

5) The DATA key links the Learner to the Keyword Index. SHIFT-NEXT jumps the Learner to the same topic in the Dictionary. BACK moves the Learner back to the previous topic viewed. SHIFT-BACK moves the Learner Back to the first page of the Yellow Pages. Continued pressing of the BACK key will eventually move the Learner back to the Router lesson. Otherwise, pressing

the letter beside a topic will move the Learner to that topic.

6.3.3.3 Dictionary

What shall be described here is how the Plato lesson implementation of the Dictionary differs from the specifications given in the design section 5.3.3.2.2

1) To match the Yellow Pages, the Dictionary also contains the 127 asterisked computer literacy topics as shown in Appendix B.

2) The display of a central topic and the topics one relationship away in an outward direction as described in section 5.3.3.2.2 is replaced by a more verbal description. The central topic along with a short definition, is displayed at the top of the page and the related topics are numbered for reference and displayed below the central topic. For example, if the Learner is 'shining his spotlight' on the topic 'problem solving' he would see a display like:

Problem Solving***

includes searching through trees and graphs like this Dictionary.

1. General is Artificial Intelligence
2. Application is Games
3. Related is Systems Analysis ***
4. Related is Programming Knowledge
5. Related is Medicine, Health-diagnosis ***

Thus, as described in the design, the Learner would read the above as:

"For the central topic of 'problem solving':

1. the general topic is Artificial Intelligence
2. an application is Games

etc
etc
etc

To place one of the related topics at the top of the page, the Learner types its number. The asterisks signify that topic is a terminal node in the Yellow Pages, and, as such, has resources associated with it.

3) A help lesson explaining the operation of the Dictionary is active upon pressing the HELP KEY.

4) The Dictionary gathers data for the CMI system. Collected are the Computer path of each topic visited and the time spent there for each Learner, and the cumulative number of visits to each topic across the total Learner population.

5) The DATA key links the Learner to the Keyword Index. Pressing the number of a topic followed by pressing NEXT places that topic at the top of the page in the Dictionary; pressing the number of a topic followed by SHIFT-NEXT jumps the Learner to that same topic in the Yellow Pages. BACK moves the Learner back to the previous topic viewed.[6] SHIFT-BACK moves the Learner back to the front page of the Dictionary--from there pressing the BACK key will return the Learner to the Router lesson.

[6] Since the Dictionary is a graph and not a tree, it was necessary to write the software to implement a stack to allow proper backing up.

6.3.3.4 Keyword Index

What shall be described here is how the Plato lesson implementation of the Keyword Index differs from the specifications in the design section 5.3.3.2.3.

1) As the Keyword Index lesson parses each word of the Learner's request, it is underlined to provide visual feedback to the Learner that the Computer is in fact doing something.
2) A help lesson explaining the detailed operation of the Keyword Index is active upon pressing the HELP key. Also included in the help sequence is an alphabetized list of valid Keywords should the Learner request to see it.
3) The Keyword Index saves unrecognized requests for later viewing by the Coordinator but does not count repeated instances of the same request--only a list of all unrecognized requests is provided.
4) After a request is entered it is sent with the NEXT key. The BACK key will return the Learner to either the Yellow Pages or the Dictionary depending where he came from. If he came from the Router lesson the default is to return to the Yellow Pages.

6.3.3.5 Bulletin Board

What shall be described here is how the PLATO lesson implementation of the Bulletin Board differs from the specifications in the design section 5.3.3.1.

1) As is clear from the name of this Computer component, only the Bulletin board part of the Communications File was

implemented. Actually part of the Mailbox function (personal messages from the Teacher to the Learner) is handled by the PLATO system; this mechanism was used to cue Learners to read important notes on the Bulletin Board. For example, a personal note was left to each Learner to be sure to read the note on the Bulletin Board which advertised a movie to be shown.

2) This facility is entirely provided by the Plato system in what is called a "group notes file." HELP is provided and Learners are afforded an easy-to-use, powerful editor with which to compose notes.

6.3.3.6 CMI System

The implementation of the CMI System consists of three parts:

1) Testing, Grading, Diagnosing and Prescribing.

2) Recording and Reporting Learner-Computer Transactions, Learner Test Progress and an Item Analysis of the test question data base.

3) Learner Feedback.

6.3.3.6.1 Testing, Grading, Diagnosing and Prescribing

Whenever a Learner feels ready, he may request to take one of the three criterion-referenced tests in the area of:

1) Natural Science Applications

2) Business Applications

3) Humanities Applications

Each test has five objectives. Mastery of each objective is measured by one multiple-choice question. To pass each test, the Learner must pass 4 of the 5 objectives. Each Learner may take each test up to three times.

The School of Education Learners in order to gain a pass grade for their one credit had to pass all three tests. The School of Engineering Learners were required to be assigned letter grades so the following scheme was followed:

To gain an 'A' grade the Learner had to pass at least two of the three tests with all five objectives answered correctly--the other test had just to be passed.

To gain a 'B' grade the Learner had to pass all three tests (same as pass for School of Education Learners).

Otherwise the grade was to be 'Incomplete.'

The Learners could review the objectives without having to take a test and were encouraged to do so. All of the objectives and the questions measuring mastery of those objectives were drawn from three of the resources--all books[7,8,9] which required approximately 150 pages of reading. These three books were available in the same room as the terminal and could be read any time the terminal room was open or could be checked out overnight or over the weekend by the Learners. All Learners were allowed to take notes with them to the tests.

Since each Learner was allowed to take a test up to three times and each test contained five questions it was necessary to create a question pool of 15 questions for each of the three tests so that a Learner would receive new questions if he had to take a test over; thus the total

[7] Rothman, S. and Mosman, C., Computers and Society (Chicago: SRA 1972).
[8] Kochenburger, R. and Turcio, C., Computers in Modern Society (Santa Barbara, Calif.: Hamilton, 1974).
[9] Sanders, D., Computers in Society (New York: McGraw Hill, 1973)

question data base comprised 45 questions.

The test administration was on the Plato terminal so that the Learner could be informed immediately whether or not he passed each objective comprising the test. If he passed the objective, he received a "Correct" response; if he did not he received an "Incorrect" diagnosis along with a prescription in terms of the three resource books (which pages to read over). If at any point in the test administration the Learner missed two objectives, the test was over, the Learner received his "grade" (how many objectives he mastered of those he was given) and a message to the effect that he would have to re-take the test. This strategy was followed to discourage Learners from taking the tests merely to learn the content of the question data base so that they could share this knowledge with each other.

When a Learner successfully mastered the required number of objectives on all three tests he received a large graphic CONGRATULATIONS message. If a Learner attempted to take a test he had passed he was locked out, receiving again a CONGRATULATIONS message.

6.3.3.6.2 Recording and Reporting

The CMI System consisting of a Plato-supplied lesson "umass" and several author-supplied lessons, records and gathers data on the Learners as they interact with the Computer. These data are available for author (the Coordinator) viewing via various types of management reports. These reports have been copied directly from the Plato terminal using a Tektronix 4632 Video Hard Copy Unit and are shown in Figures 1-11.

Learner-Computer Transactions

1) Computer Usage

The Computer usage by Learner is shown in Figure 1. The Learner's names appear on the left; the 's' means 'student' (as opposed to author) and the rest of the columns are interpreted as described in the design section 5.3.3.3.3 with the exception of the last column 'CPU' which is a measure of the user's share of the time sharing system (below 2 is considered to be adequate to insure 1000 simultaneous users of the system with less than a 2 second response time).

Figure 2 shows Computer Usage Averages for all ten students. Also shown are averages for Plato authors (programmers). Figures 1 and 2 are produced by Plato lesson "umass."

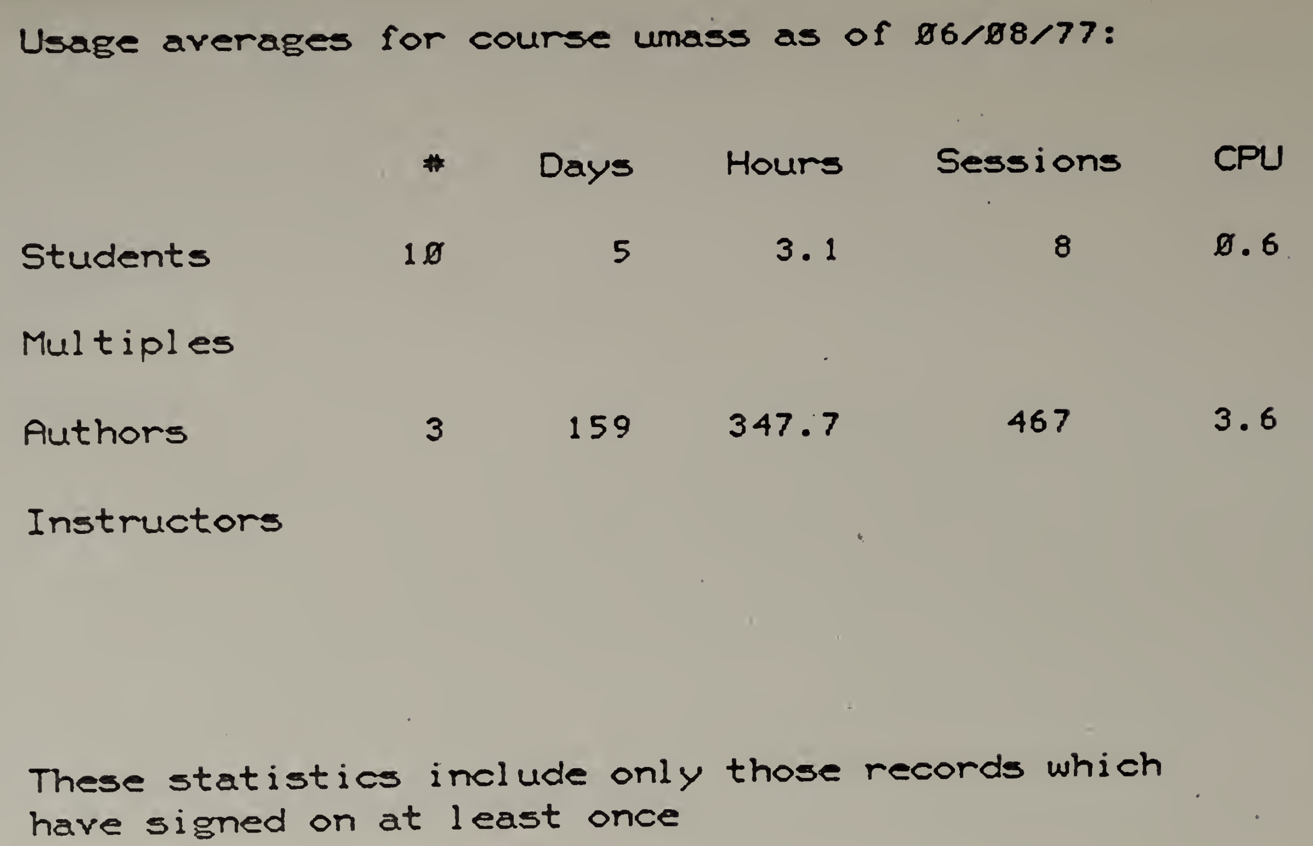
2) Visits to Computer Components

The Router lesson (named "accolade") gathers data of the type displayed in Figure 3. Every time a Learner signs onto Plato counts as a separate session. Shown are the cumulative number of visits across the total Learner population to each Computer component as well as the time spent there. When the figure was copied "net" had not yet been replaced by "dictionary" and "enotes" by "bulletin board." Although the figures were taken at the end of the course they are not to be interpreted as firm data since the statistics were initialized part way through the course when a bug was found in the program that gathers this data; also the "cmi quiz" figures which represent the three criterion-referenced tests include many visits by a 'dummy'

Record usage for course umass as of Ø6/Ø8/77:

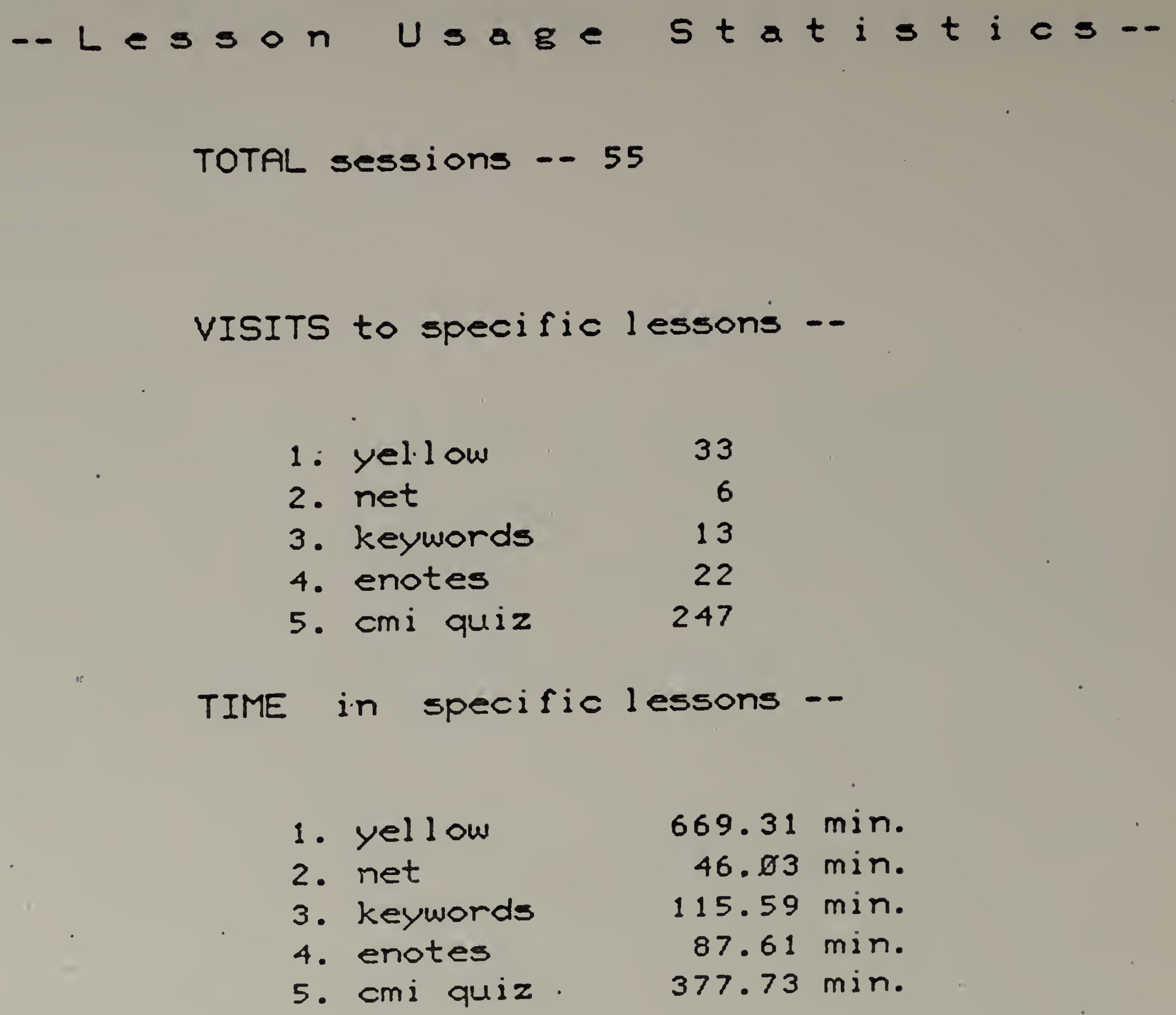
		Last On	Days	Hours	Sess.	CPU
andy	s	Ø5/31/77	4	4	7	Ø.5
bernard	s	Ø5/18/77	3	3	6	Ø.5
bob	s	Ø5/17/77	7	4	16	Ø.9
cathleen	s	Ø5/19/77	3	2	5	Ø.7
cyrus	s	Ø5/18/77	3	2	4	Ø.4
ed	s	Ø5/12/77	11	4	15	Ø.6
harvey	s	Ø4/19/77	5	2	8	Ø.5
mark	s	Ø4/14/77	5	5	6	Ø.6
mike	s	Ø6/Ø1/77	4	3	7	Ø.9
pat	s	Ø5/25/77	4	2	5	Ø.4

Figure 1. Computer Usage by Learner



```
Usage averages for course umass as of Ø6/Ø8/77:

                     #      Days     Hours      Sessions     CPU

Students            1Ø        5        3.1             8     Ø.6

Multiples

Authors              3      159      347.7           467     3.6

Instructors



These statistics include only those records which
have signed on at least once
```

Figure 2. Computer Usage Averages



```
--Lesson Usage Statistics--

     TOTAL sessions -- 55


     VISITS to specific lessons --


          1: yellow          33
          2. net              6
          3. keywords        13
          4. enotes          22
          5. cmi quiz       247

     TIME  in  specific lessons --


          1. yellow         669.31 min.
          2. net             46.Ø3 min.
          3. keywords       115.59 min.
          4. enotes          87.61 min.
          5. cmi quiz       377.73 min.
```

Figure 3. Visits to Computer Components

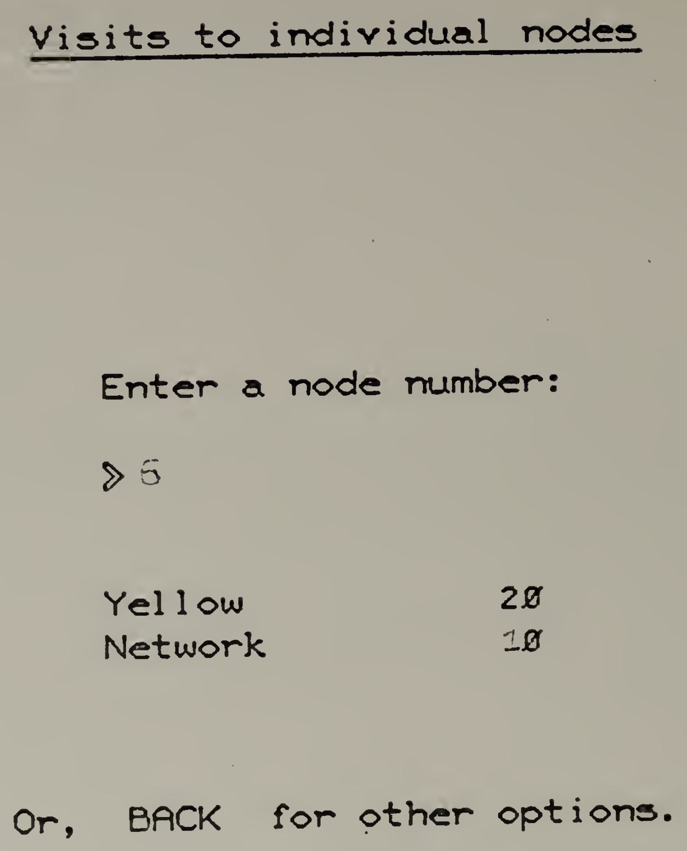
student for the purposes of debugging the CMI tests. However, if the system is ever used again, the data should be accurate.

3) Visits to Specific Topics

Each of the 127 computer literacy topics was assigned a number from 1 to 127 so that the Teacher could enter a node number and see the cumulative number of visits to this node by the total Learner population in both the Yellow Pages and the Dictionary. Figure 4 shows the visits to node 6 (Social Issues) in both the Yellow Pages and the Network (Dictionary). The data is not accurate since a bug was discovered in the data gathering program well into the pilot study. However, the bug has been fixed and if the system is used again, this data will be accurate. This report is produced by Plato lesson "correlate."

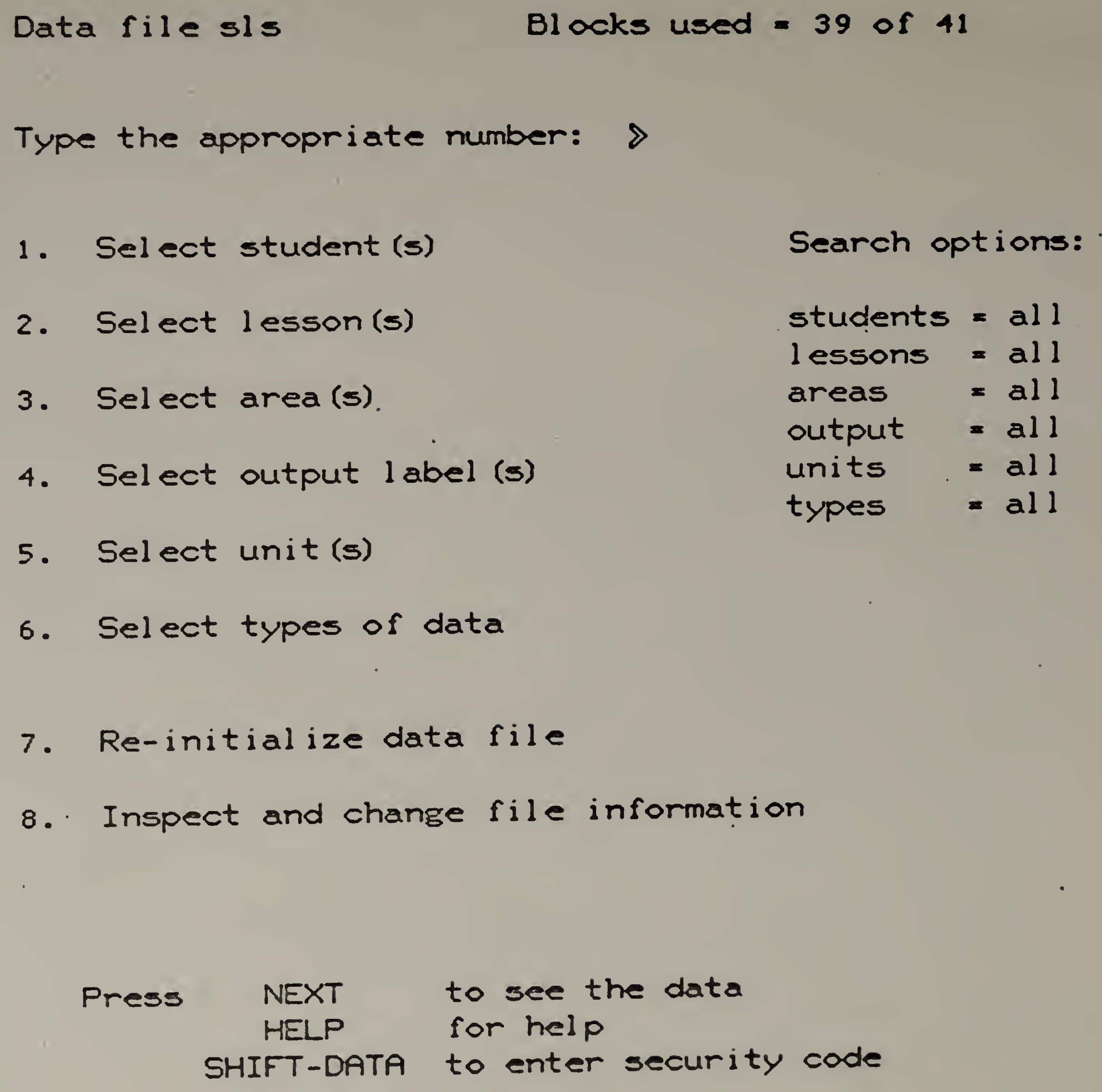
4) Computer Path

The Computer Path is a chronological account of each Learner's interactions with the topics of the Yellow Pages and the Dictionary. This data is captured in Plato datafile "sls" (for 'student learning styles') and various selection options are available as shown in Figure 5a. If no selection is made, the Teacher will see Figure 5b where an index is displayed which divides the chronological Learner paths by fixed size computer storage blocks. If the Teacher selects block a, then he will see Fig. 5c which is part of the Computer Path for Learner 'bob.' The second line shows the name of the Learner, the lesson (the Yellow Pages or the Dictionary), and the date and time the Learner signed in to the lesson. Each succeeding pair of lines shows: the time in minutes from sign in time, the Learner name,



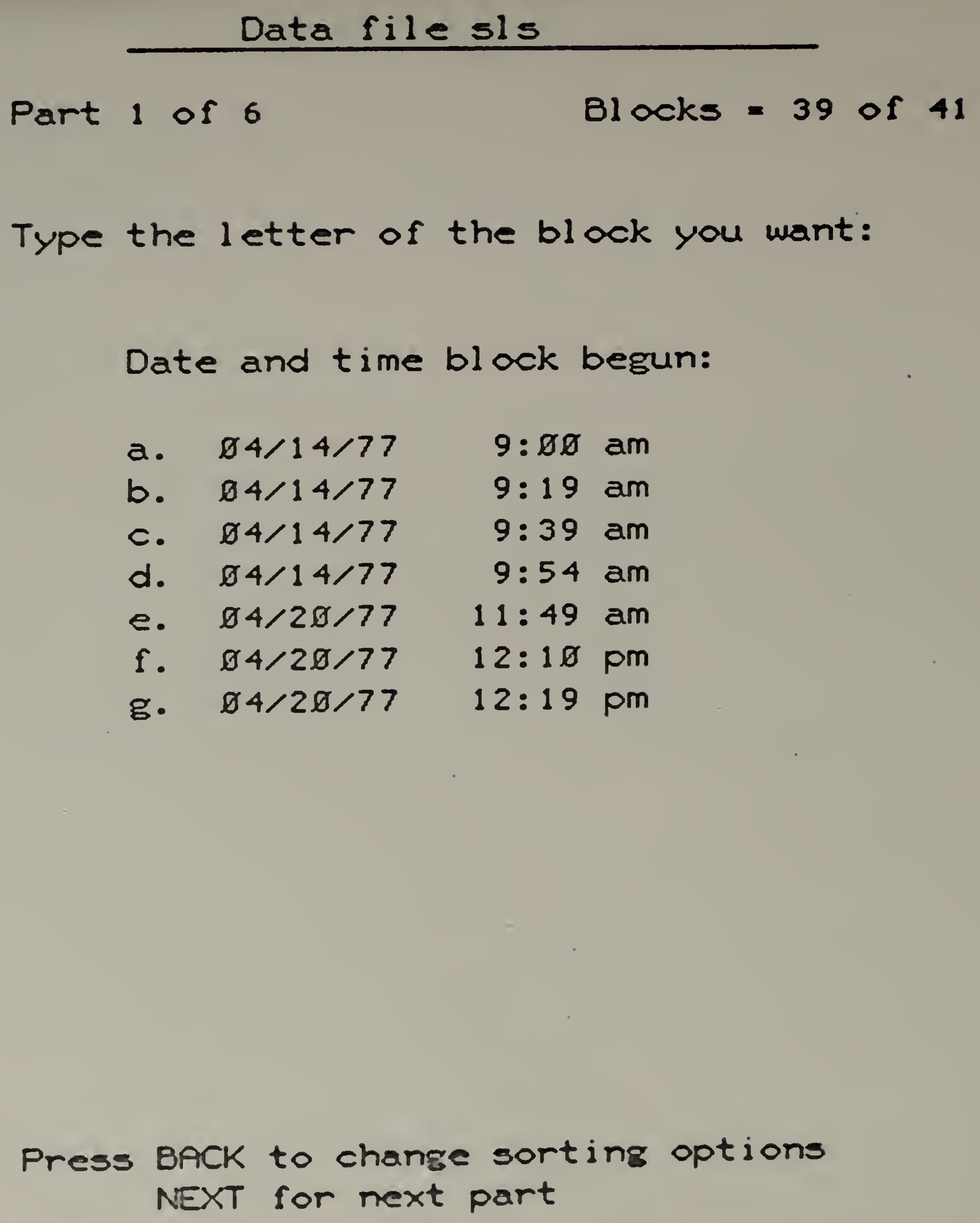
```
Visits to individual nodes


Enter a node number:

> 6

Yellow              2Ø
Network             1Ø

Or,  BACK  for other options.
```

Figure 4. Visits to Specific Topics



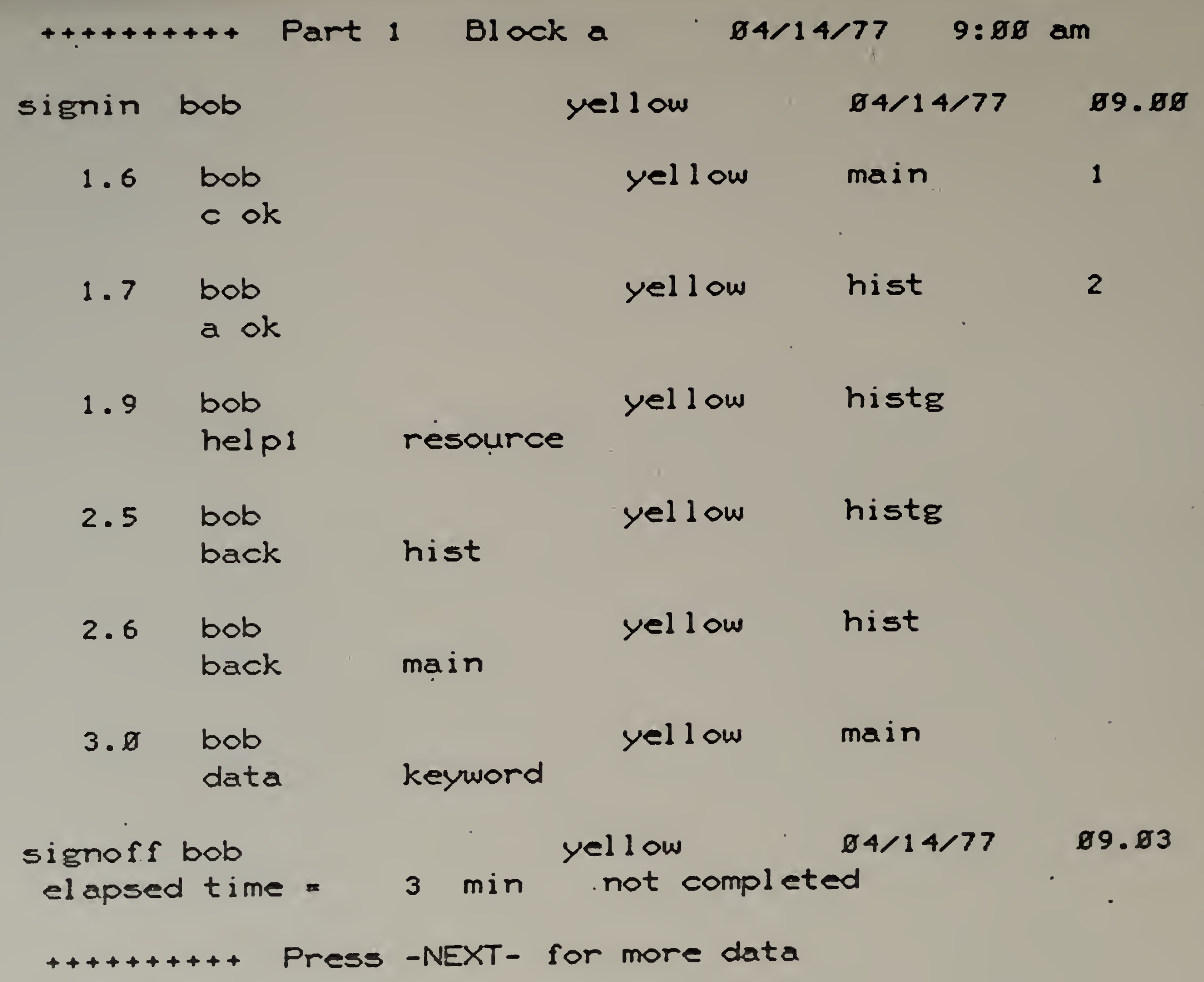
```
Data file sls                     Blocks used = 39 of 41


Type the appropriate number:   >


1.   Select student(s)                          Search options:

2.   Select lesson(s)                           students  = all
                                                lessons   = all
3.   Select area(s)                             areas     = all
                                                output    = all
4.   Select output label(s)                     units     = all
                                                types     = all
5.   Select unit(s)

6.   Select types of data


7.   Re-initialize data file

8.   Inspect and change file information



   Press     NEXT      to see the data
             HELP      for help
        SHIFT-DATA     to enter security code
```

Figure 5a. Computer Path: Selection Options



```
              Data file sls
Part 1 of 6                   Blocks = 39 of 41

Type the letter of the block you want:


     Date and time block begun:

     a.   Ø4/14/77     9:ØØ am
     b.   Ø4/14/77     9:19 am
     c.   Ø4/14/77     9:39 am
     d.   Ø4/14/77     9:54 am
     e.   Ø4/2Ø/77    11:49 am
     f.   Ø4/2Ø/77    12:1Ø pm
     g.   Ø4/2Ø/77    12:19 pm





Press BACK to change sorting options
      NEXT for next part
```

Figure 5b. Computer Path: No Selection



```
 ++++++++++  Part 1    Block a     Ø4/14/77   9:ØØ am

signin  bob                yellow        Ø4/14/77    Ø9.ØØ

   1.6   bob                  yellow     main        1
         c ok

   1.7   bob                  yellow     hist        2
         a ok

   1.9   bob                  yellow     histg
         helpl     resource

   2.5   bob                  yellow     histg
         back      hist

   2.6   bob                  yellow     hist
         back      main

   3.Ø   bob                  yellow     main
         data      keyword

signoff bob                yellow        Ø4/14/77    Ø9.Ø3
 elapsed time =    3  min  not completed

 ++++++++++  Press -NEXT- for more data
```

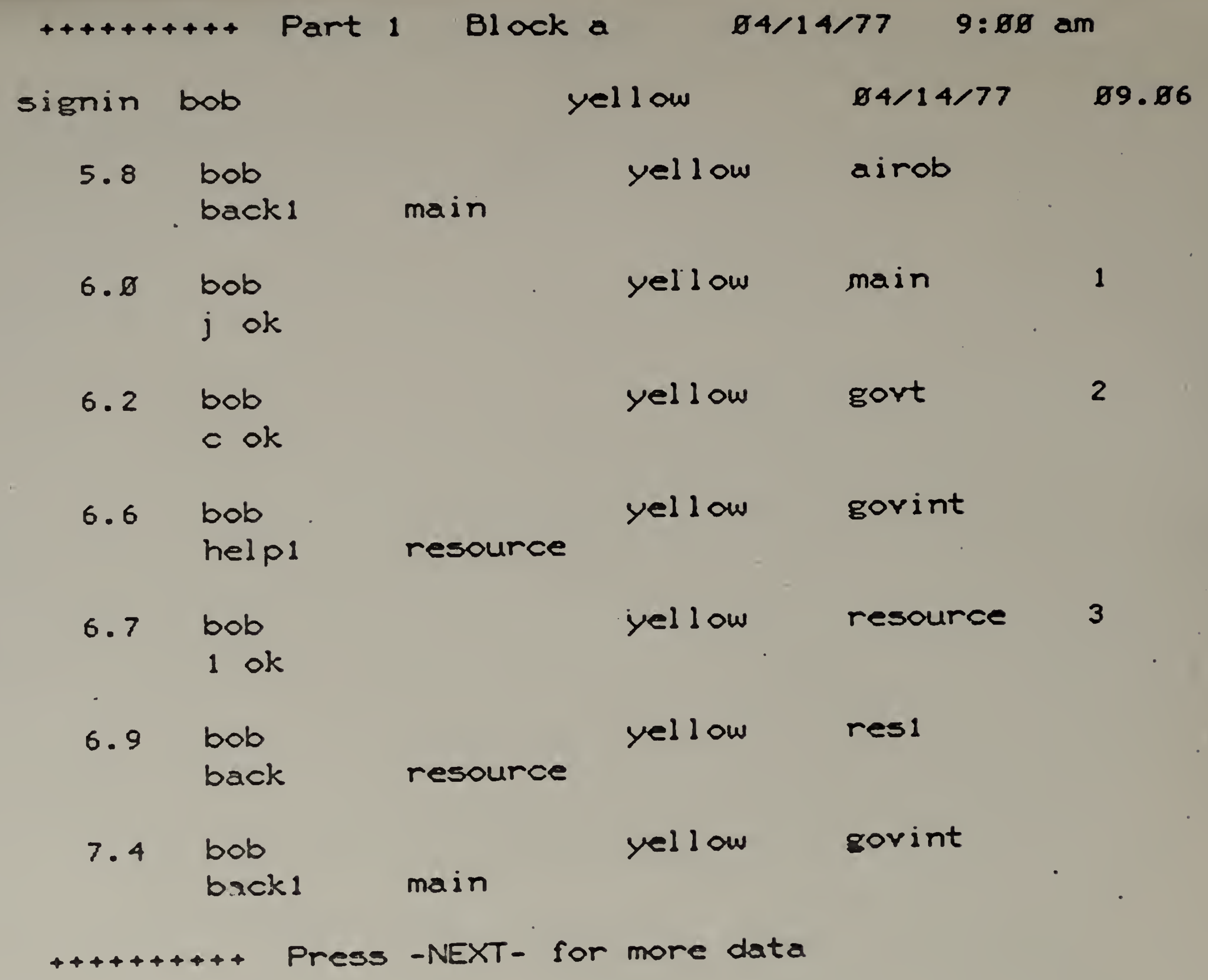
Figure 5c. Computer Path: Learner 'bob'

the lesson name, and the topic visited; the second of the pair of lines shows which key the Learner pressed followed either by a Plato judgement or the name of the next topic visited. The last lines show sign off data. Figure 5d shows a picture of the PLATO screen after the Teacher presses NEXT for Figure 5c. A detailed interpretation of this data is available by pressing the HELP key in Plato data file "sls."

Learner Test Progress

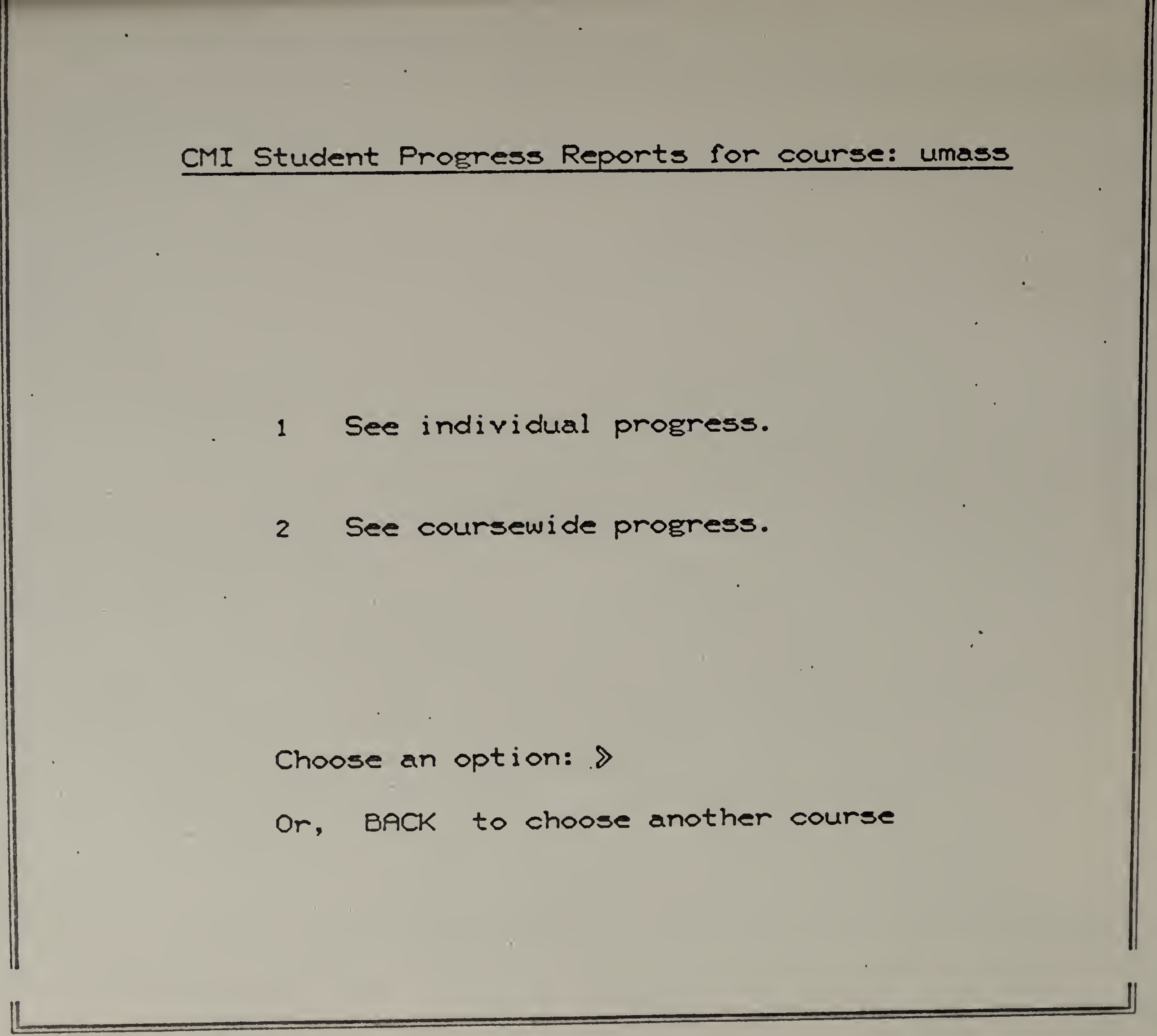
Learner Test Progress Reports are kept at two levels: by individual Learner and across all Learners. When the Teacher enters Plato lesson "anal" (short for 'analysis') he sees the choice afforded in Figure 6a. If he chooses option 1 ("See individual progress") the next display he sees is Figure 6b, and when he types in the Learner name "cyrus," he sees the Learner Progress Report for that Learner in Figure 6c. Figure 6c shows that the last time 'cyrus' took a test it was test number 3 (Humanities Applications) and that he passed with a score of 5 out of 5 in 467 seconds.

Shown next is how 'cyrus' performed on each question that was given him--the question number from the pool, his score (R=Right, W=Wrong) and his time in seconds on each question. Since 'cyrus' took test number 3 which consists of 5 objectives each measured by one question, Figure 6c indicates that 'cyrus' received question 312, 322, 332, 343, and 351 where each 3 digit question number is a code for the Test No., Objective No., and Question Pool No. Finally, Figure 6c shows the current status of 'cyrus' on all three tests: he has, at this point, passed all three tests, having had to take test 3 twice before passing it.



```
++++++++++  Part 1   Block a        Ø4/14/77    9:ØØ am

signin  bob                    yellow         Ø4/14/77    Ø9.Ø6

    5.8    bob                     yellow    airob
           back1     main

    6.Ø    bob                     yellow    main         1
           j  ok

    6.2    bob                     yellow    govt         2
           c  ok

    6.6    bob                     yellow    govint
           help1     resource

    6.7    bob                     yellow    resource     3
           1  ok

    6.9    bob                     yellow    res1
           back      resource

    7.4    bob                     yellow    govint
           back1     main

 ++++++++++  Press -NEXT- for more data
```

Figure 5d. Computer Path: Learner 'bob' continued



CMI Student Progress Reports for course: umass

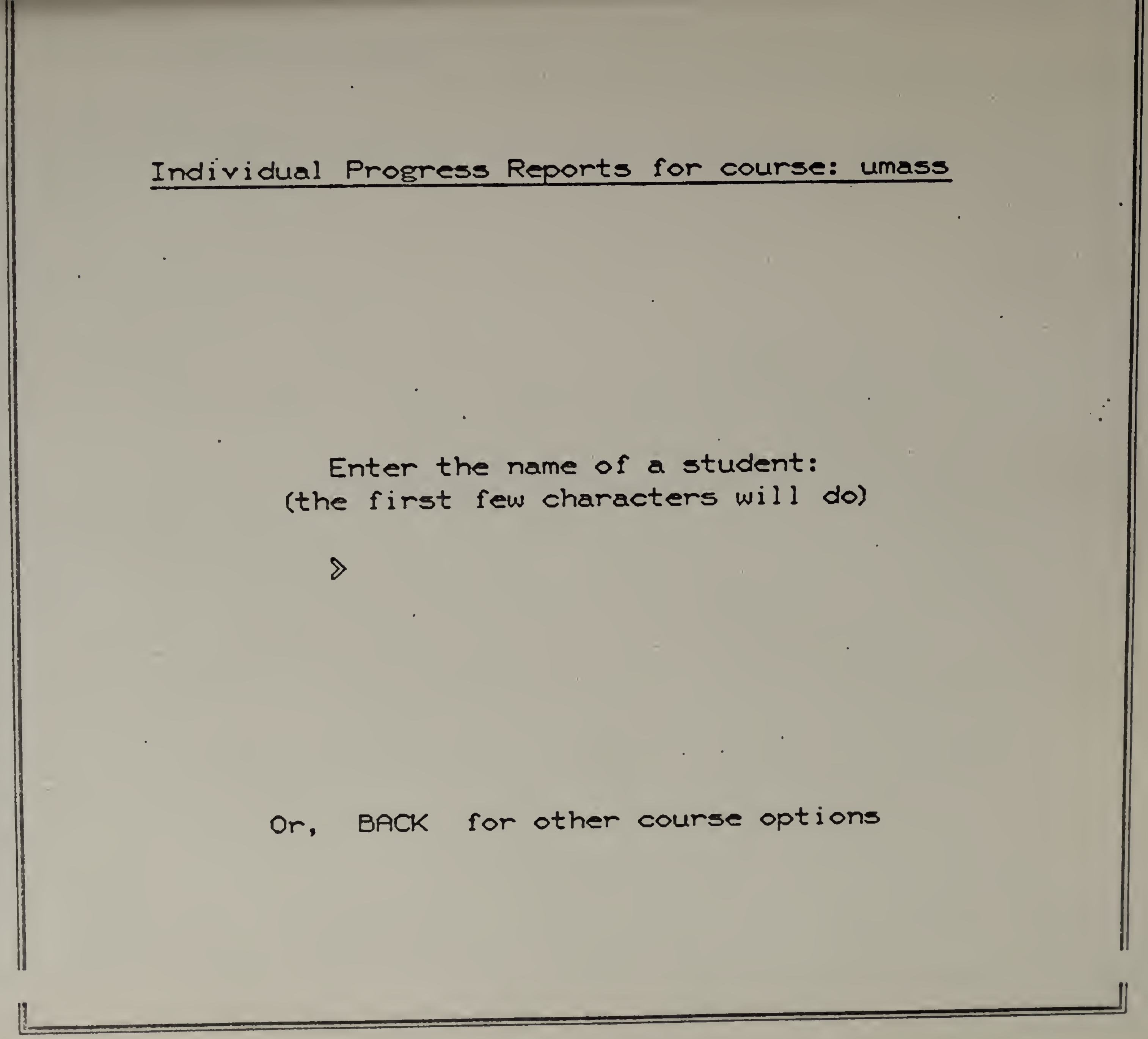
1   See individual progress.

2   See coursewide progress.

Choose an option: ≫

Or, BACK to choose another course

Figure 6a. Learner Test Progress: Choice of Individual or All.



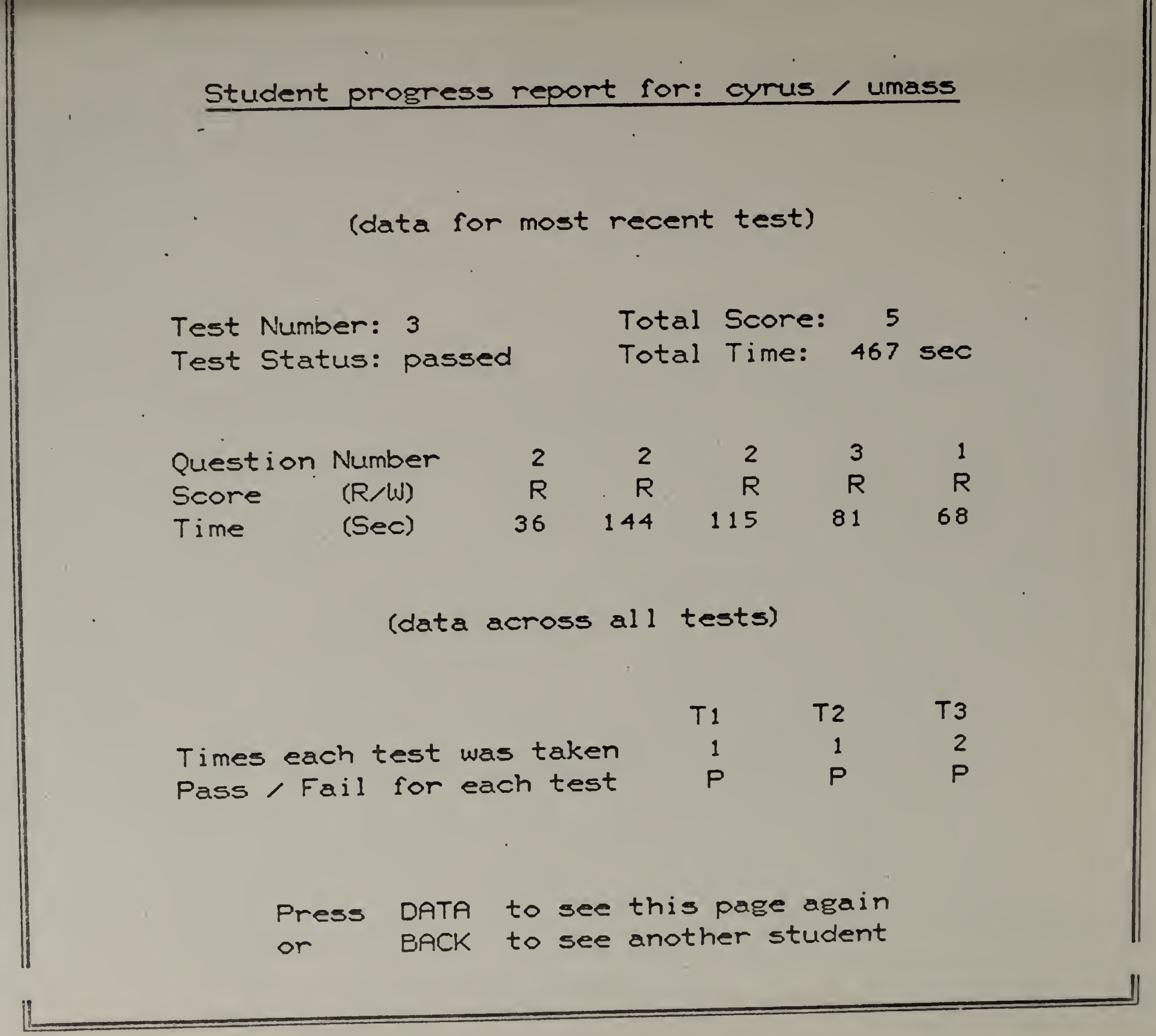
Individual Progress Reports for course: umass

Enter the name of a student:
(the first few characters will do)

>

Or, BACK for other course options

Figure 6b. Learner Test Progress: an Individual



```
      Student progress report for: cyrus / umass


              (data for most recent test)


Test Number: 3              Total Score:    5
Test Status: passed         Total Time:   467 sec


Question Number       2     2     2     3     1
Score      (R/W)      R     R     R     R     R
Time       (Sec)     36   144   115    81    68


              (data across all tests)


                                 T1     T2     T3
Times each test was taken         1      1      2
Pass / Fail for each test         P      P      P


     Press   DATA   to see this page again
     or      BACK   to see another student
```

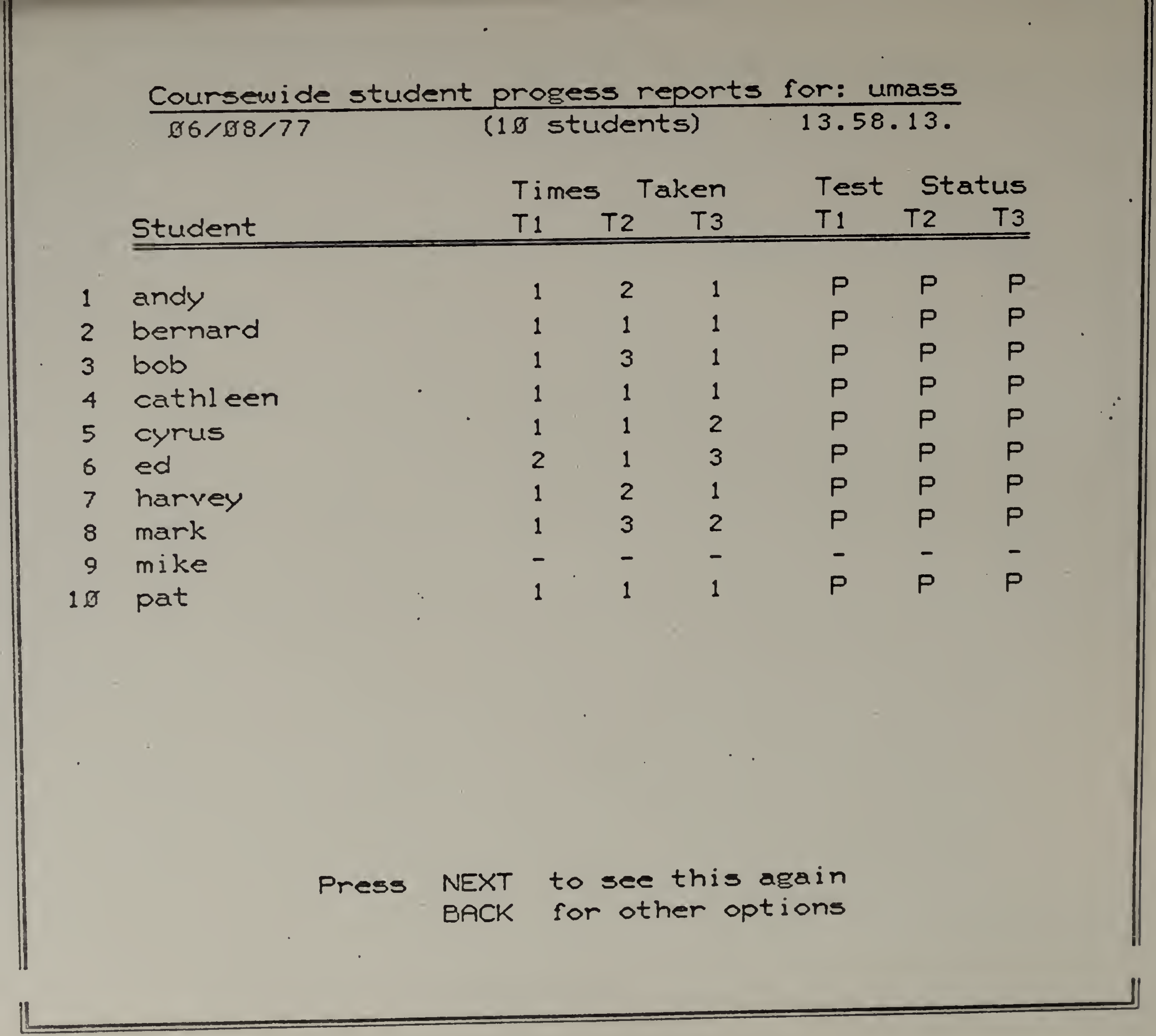
Figure 6c. Learner Test Progress: Learner 'cyrus'

If the Teacher had chosen option 2 at Figure 6a, he then would see the display of Figure 6d, where the progress of all Learners is shown simultaneously. From this report the Teacher can see how many times each test was taken by each Learner and whether or not it was passed the last time the Learner took the test. It is clear from Figure 6d that 'mike' is the Learner who withdrew and that probably test 2 (Business Applications) is more difficult than the other two. These figures were printed after the course was concluded but the reports were displayed many times during the progress of the course to monitor the Learner Test Progress.

Item Analysis

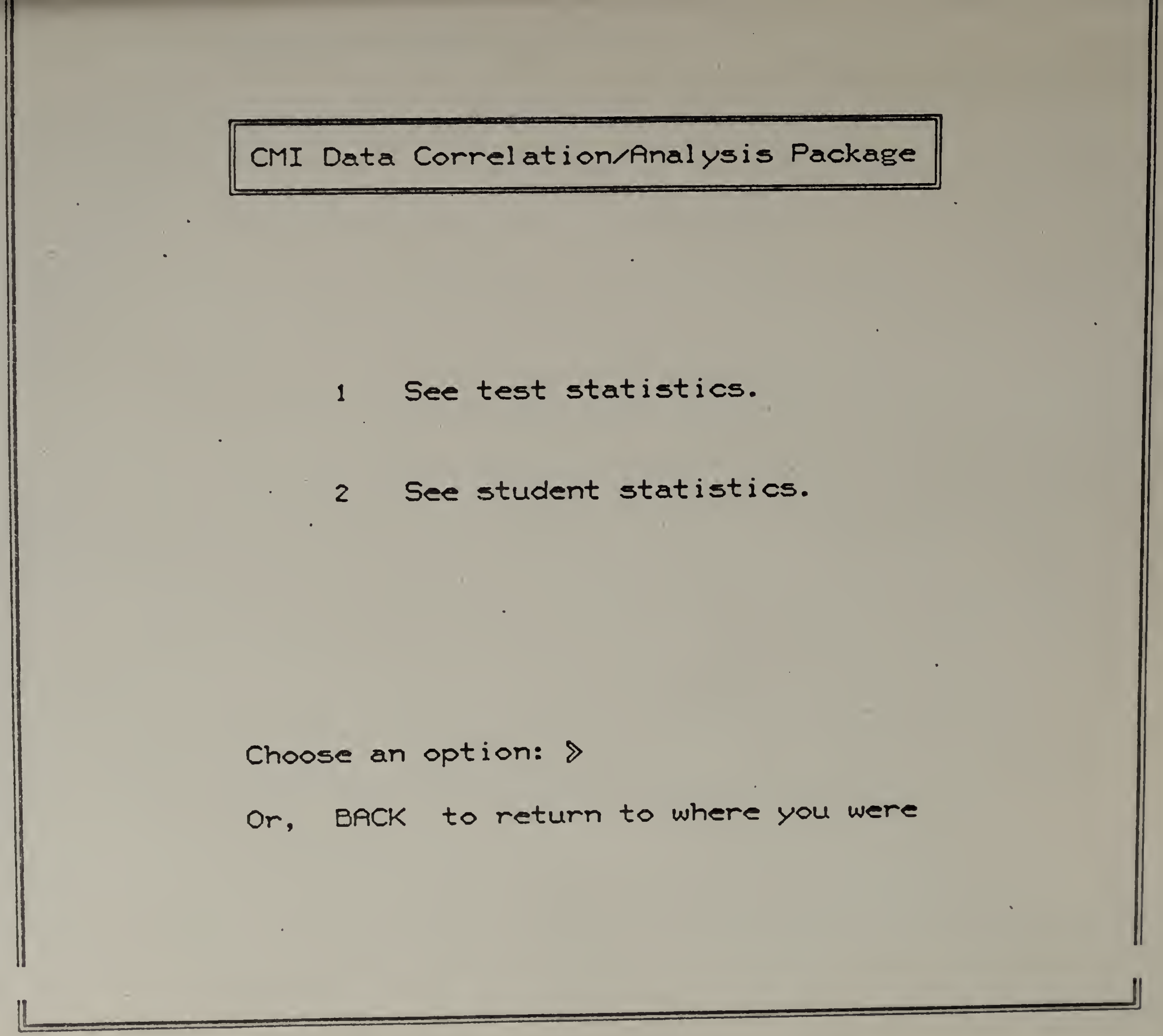
Another option available in Plato lesson "anal" is the Item Analysis as described in the design section 5.3.3.3.3. The first page the Teacher sees in lesson "anal" is the choice page shown in Figure 7. If the Teacher chooses option 2 ("See student statistics") he will see Figures 6a-6d; however when he chooses option 1 ("See test statistics") he sees an item analysis for each test as shown in Figures 8a-8c. The interpretations for each column are the same as those discussed in the item analysis description of design section 5.3.3.3.3 with the following exceptions: (1) the first two columns denote the objective number and question pool number, which with the test number, uniquely determine the question number. (2) the total number of Learners is in the 'Times Given' column, the total number of 'Only Passes' is in the '#P' column, and the total number of 'Only Fails' is in the '#F' column.

A cursory inspection of the figures seems to indicate that questions 132 and 332 may be too easy while question 212 appears to be discriminating positively and question 342 discriminating negatively. Of course



```
    Coursewide student progess reports for: umass
     Ø6/Ø8/77               (1Ø students)     13.58.13.

                            Times  Taken      Test  Status
    Student                 T1    T2    T3    T1    T2    T3

 1  andy                    1     2     1     P     P     P
 2  bernard                 1     1     1     P     P     P
 3  bob                     1     3     1     P     P     P
 4  cathleen                1     1     1     P     P     P
 5  cyrus                   1     1     2     P     P     P
 6  ed                      2     1     3     P     P     P
 7  harvey                  1     2     1     P     P     P
 8  mark                    1     3     2     P     P     P
 9  mike                    -     -     -     -     -     -
1Ø  pat                     1     1     1     P     P     P


              Press  NEXT  to see this again
                     BACK  for other options
```

Figure 6d. Learner Test Progress: All



CMI Data Correlation/Analysis Package

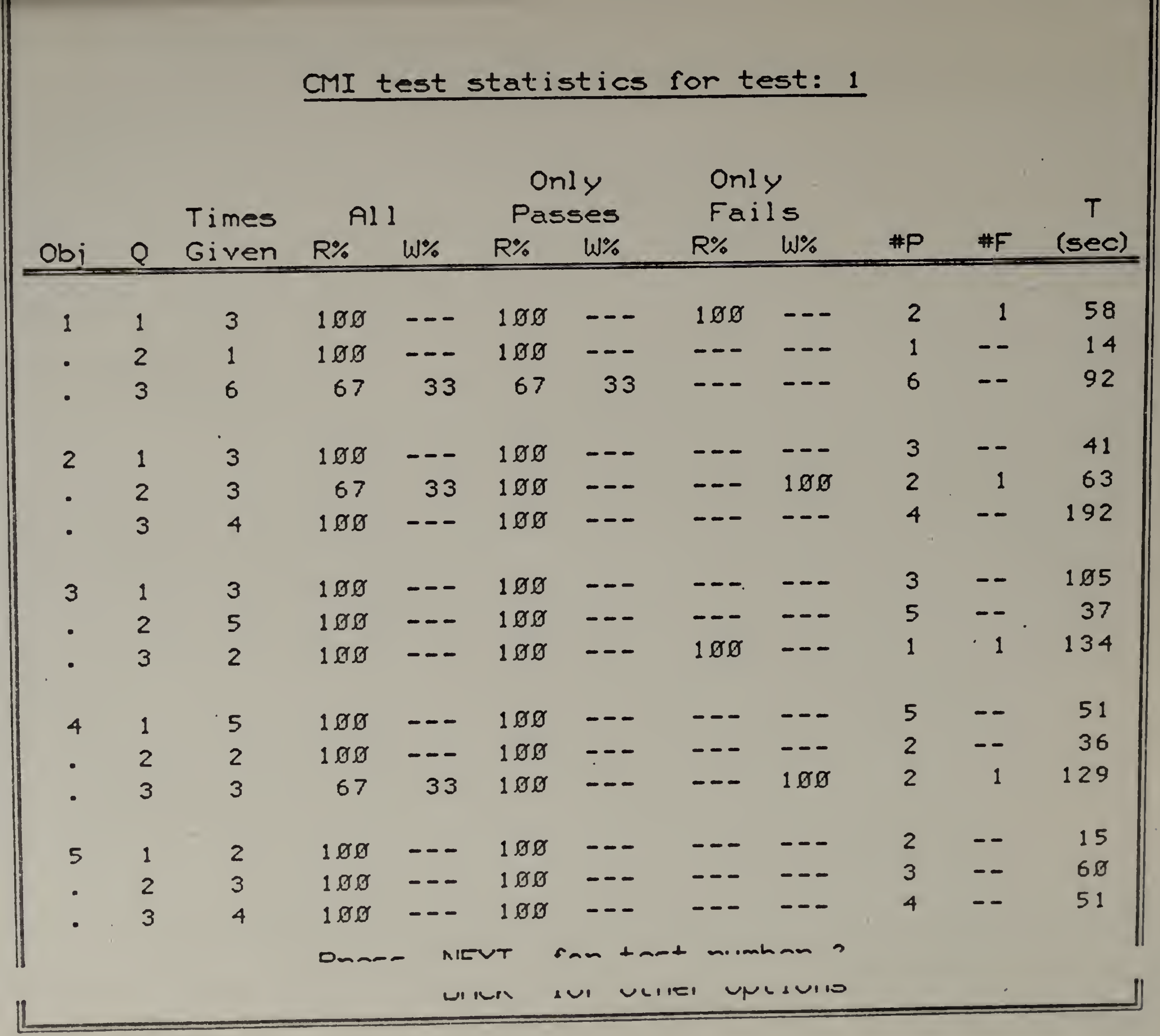
1 See test statistics.

2 See student statistics.

Choose an option: »

Or, BACK to return to where you were

Figure 7. Item Analysis - Test Progress Choice Page



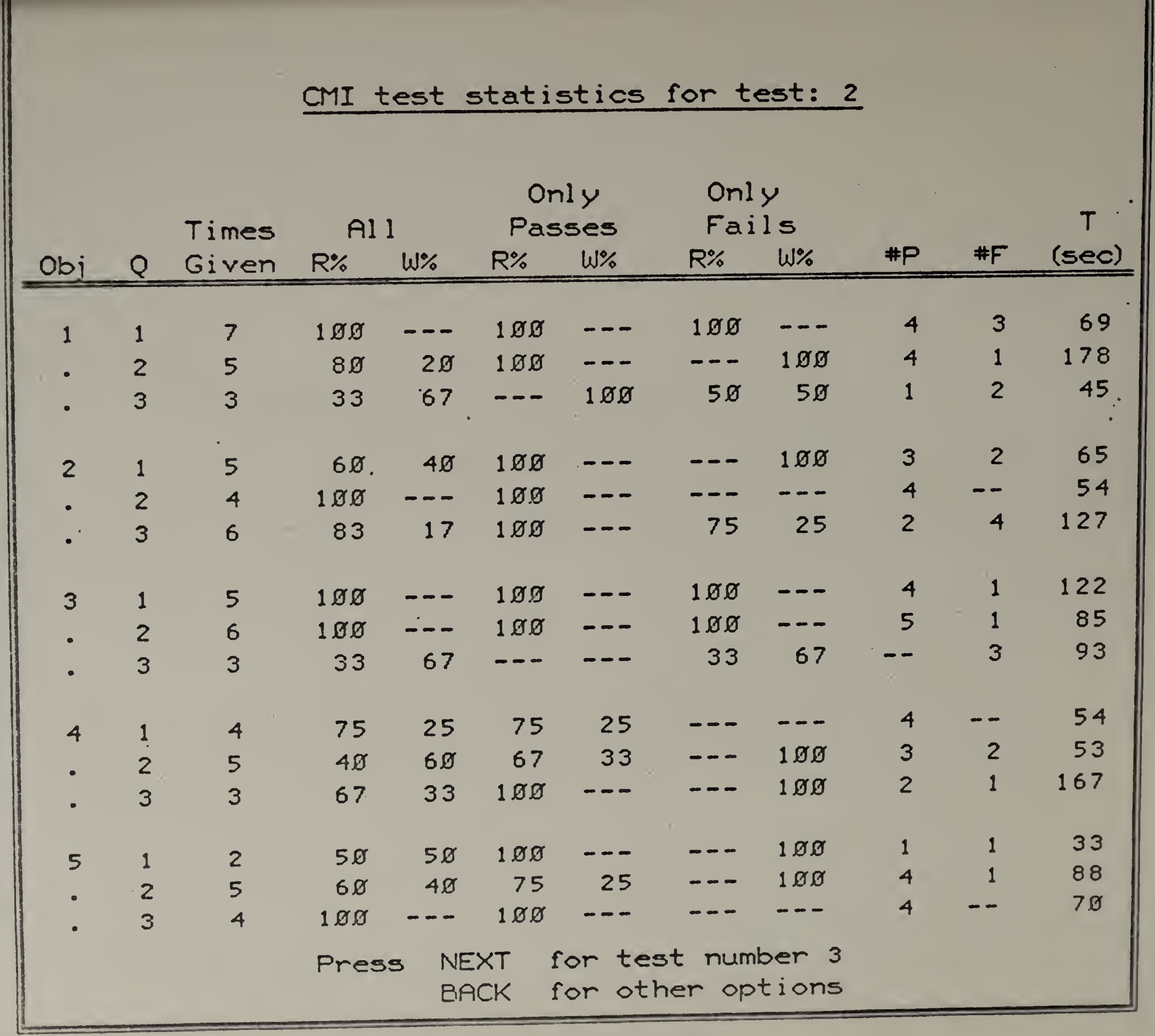
CMI test statistics for test: 1

		Times	All		Only Passes		Only Fails				T
Obj	Q	Given	R%	W%	R%	W%	R%	W%	#P	#F	(sec)
1	1	3	1ØØ	---	1ØØ	---	1ØØ	---	2	1	58
.	2	1	1ØØ	---	1ØØ	---	---	---	1	--	14
.	3	6	67	33	67	33	---	---	6	--	92
2	1	3	1ØØ	---	1ØØ	---	---	---	3	--	41
.	2	3	67	33	1ØØ	---	---	1ØØ	2	1	63
.	3	4	1ØØ	---	1ØØ	---	---	---	4	--	192
3	1	3	1ØØ	---	1ØØ	---	---	---	3	--	1Ø5
.	2	5	1ØØ	---	1ØØ	---	---	---	5	--	37
.	3	2	1ØØ	---	1ØØ	---	1ØØ	---	1	1	134
4	1	5	1ØØ	---	1ØØ	---	---	---	5	--	51
.	2	2	1ØØ	---	1ØØ	---	---	---	2	--	36
.	3	3	67	33	1ØØ	---	---	1ØØ	2	1	129
5	1	2	1ØØ	---	1ØØ	---	---	---	2	--	15
.	2	3	1ØØ	---	1ØØ	---	---	---	3	--	6Ø
.	3	4	1ØØ	---	1ØØ	---	---	---	4	--	51

Press NEXT for test number ?

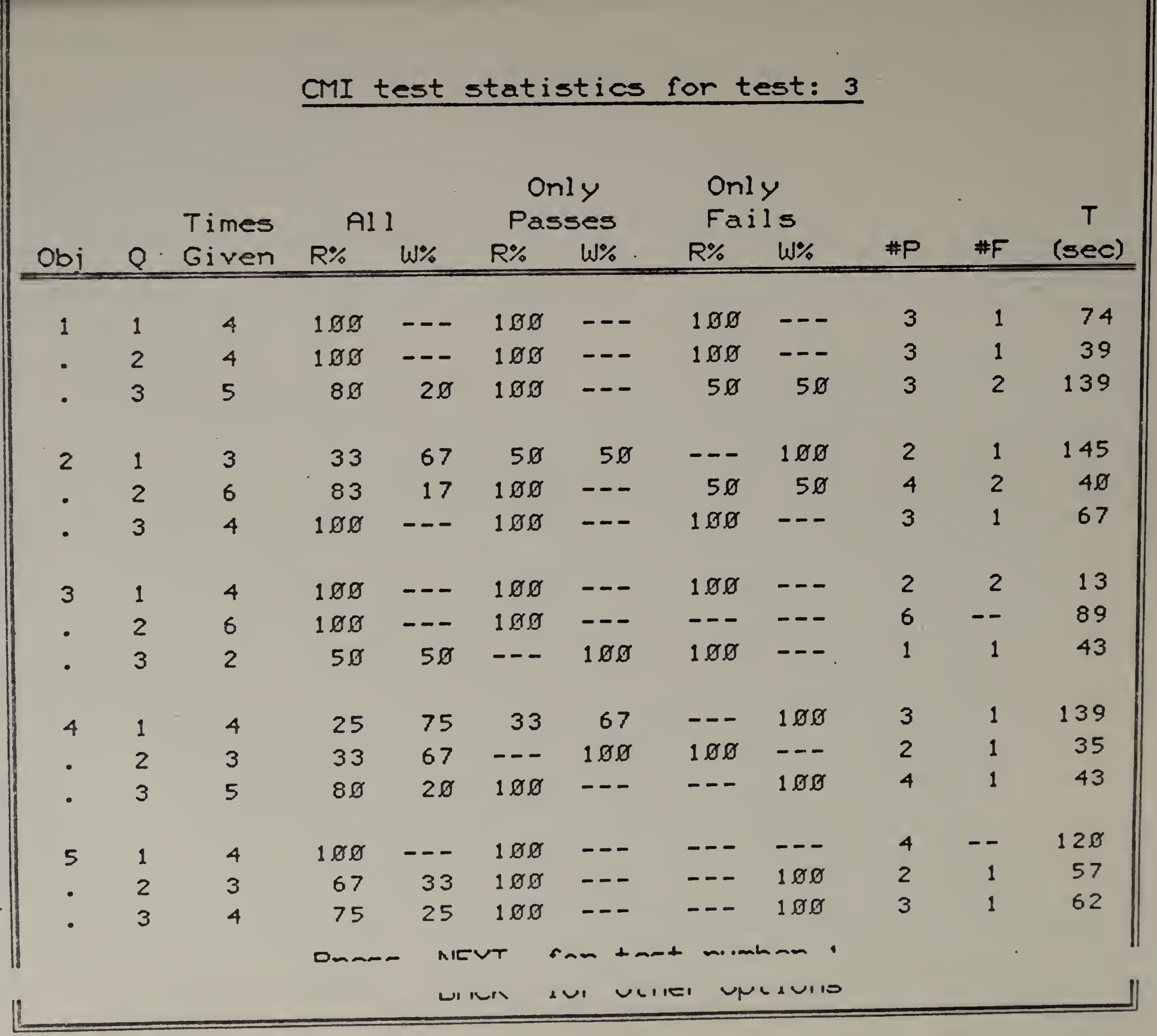
BACK for other options

Figure 8a. Item Analysis: Test 1



```
                CMI test statistics for test: 2


                                 Only        Only
            Times     All        Passes      Fails                 T
Obj    Q    Given  R%    W%    R%    W%    R%    W%    #P   #F   (sec)

 1     1      7    1ØØ   ---   1ØØ   ---   1ØØ   ---    4    3     69
 .     2      5     8Ø    2Ø   1ØØ   ---   ---   1ØØ    4    1    178
 .     3      3     33    67   ---   1ØØ    5Ø    5Ø    1    2     45

 2     1      5     6Ø    4Ø   1ØØ   ---   ---   1ØØ    3    2     65
 .     2      4    1ØØ   ---   1ØØ   ---   ---   ---    4   --     54
 .     3      6     83    17   1ØØ   ---    75    25    2    4    127

 3     1      5    1ØØ   ---   1ØØ   ---   1ØØ   ---    4    1    122
 .     2      6    1ØØ   ---   1ØØ   ---   1ØØ   ---    5    1     85
 .     3      3     33    67   ---   ---    33    67   --    3     93

 4     1      4     75    25    75    25   ---   ---    4   --     54
 .     2      5     4Ø    6Ø    67    33   ---   1ØØ    3    2     53
 .     3      3     67    33   1ØØ   ---   ---   1ØØ    2    1    167

 5     1      2     5Ø    5Ø   1ØØ   ---   ---   1ØØ    1    1     33
 .     2      5     6Ø    4Ø    75    25   ---   1ØØ    4    1     88
 .     3      4    1ØØ   ---   1ØØ   ---   ---   ---    4   --     7Ø
                   Press  NEXT  for test number 3
                          BACK  for other options
```

Figure 8b. Item Analysis: Test 2



```
              CMI test statistics for test: 3

                               Only        Only
           Times     All       Passes      Fails               T
Obj   Q    Given   R%   W%    R%   W%    R%   W%    #P   #F   (sec)

 1    1     4     1ØØ  ---   1ØØ  ---   1ØØ  ---    3    1     74
 .    2     4     1ØØ  ---   1ØØ  ---   1ØØ  ---    3    1     39
 .    3     5      8Ø   2Ø   1ØØ  ---    5Ø   5Ø    3    2    139

 2    1     3      33   67    5Ø   5Ø   ---  1ØØ    2    1    145
 .    2     6      83   17   1ØØ  ---    5Ø   5Ø    4    2     4Ø
 .    3     4     1ØØ  ---   1ØØ  ---   1ØØ  ---    3    1     67

 3    1     4     1ØØ  ---   1ØØ  ---   1ØØ  ---    2    2     13
 .    2     6     1ØØ  ---   1ØØ  ---   ---  ---    6   --     89
 .    3     2      5Ø   5Ø   ---  1ØØ   1ØØ  ---    1    1     43

 4    1     4      25   75    33   67   ---  1ØØ    3    1    139
 .    2     3      33   67   ---  1ØØ   1ØØ  ---    2    1     35
 .    3     5      8Ø   2Ø   1ØØ  ---   ---  1ØØ    4    1     43

 5    1     4     1ØØ  ---   1ØØ  ---   ---  ---    4   --    12Ø
 .    2     3      67   33   1ØØ  ---   ---  1ØØ    2    1     57
 .    3     4      75   25   1ØØ  ---   ---  1ØØ    3    1     62

           Press   NEXT   for test number !
                  BACK   for other options
```

Figure 8c. Item Analysis: Test 3

the sample sizes are much too small to allow any valid interpretations to be drawn, but if the Learner population were, say greater than 100, and the same trends were apparent, the Teacher could begin to draw these types of conclusions about his question data base.

6.3.3.6.3 Learner Feedback

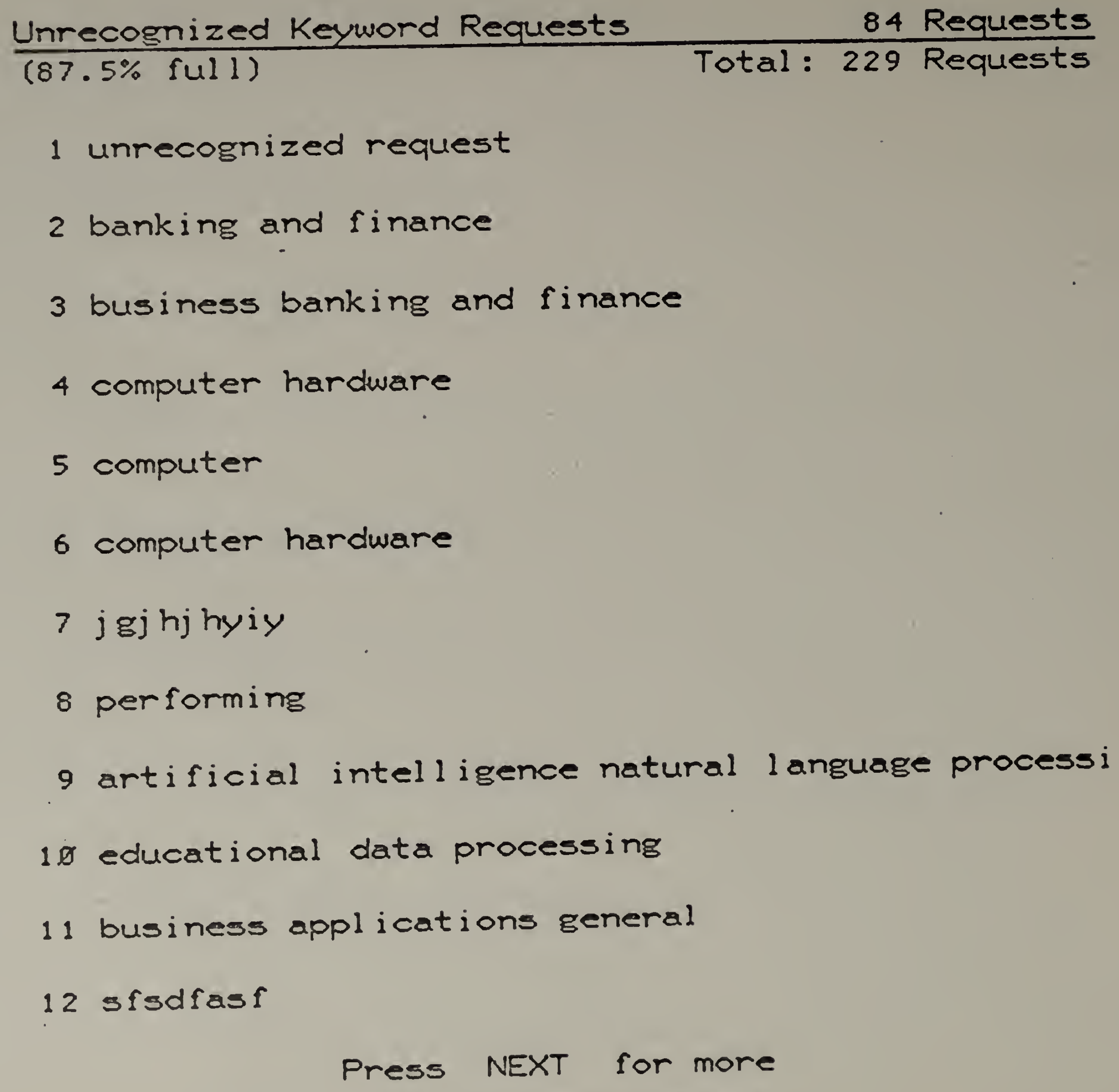
Three more types of reports are available to the Teacher which can be used to monitor Learner feedback on aspects of the operation of the Computer parts of ACCOLADE.

1) Unrecognized Keyword Requests

As mentioned in section 6.3.3.4, unrecognized Keyword requests are saved for later viewing by a Teacher (the Coordinator). Figure 9 shows one page of unrecognized keyword requests, some of which were incorporated into the list of recognizable synonyms for the Keyword Index. This report is displayed by Plato lesson "keywords."

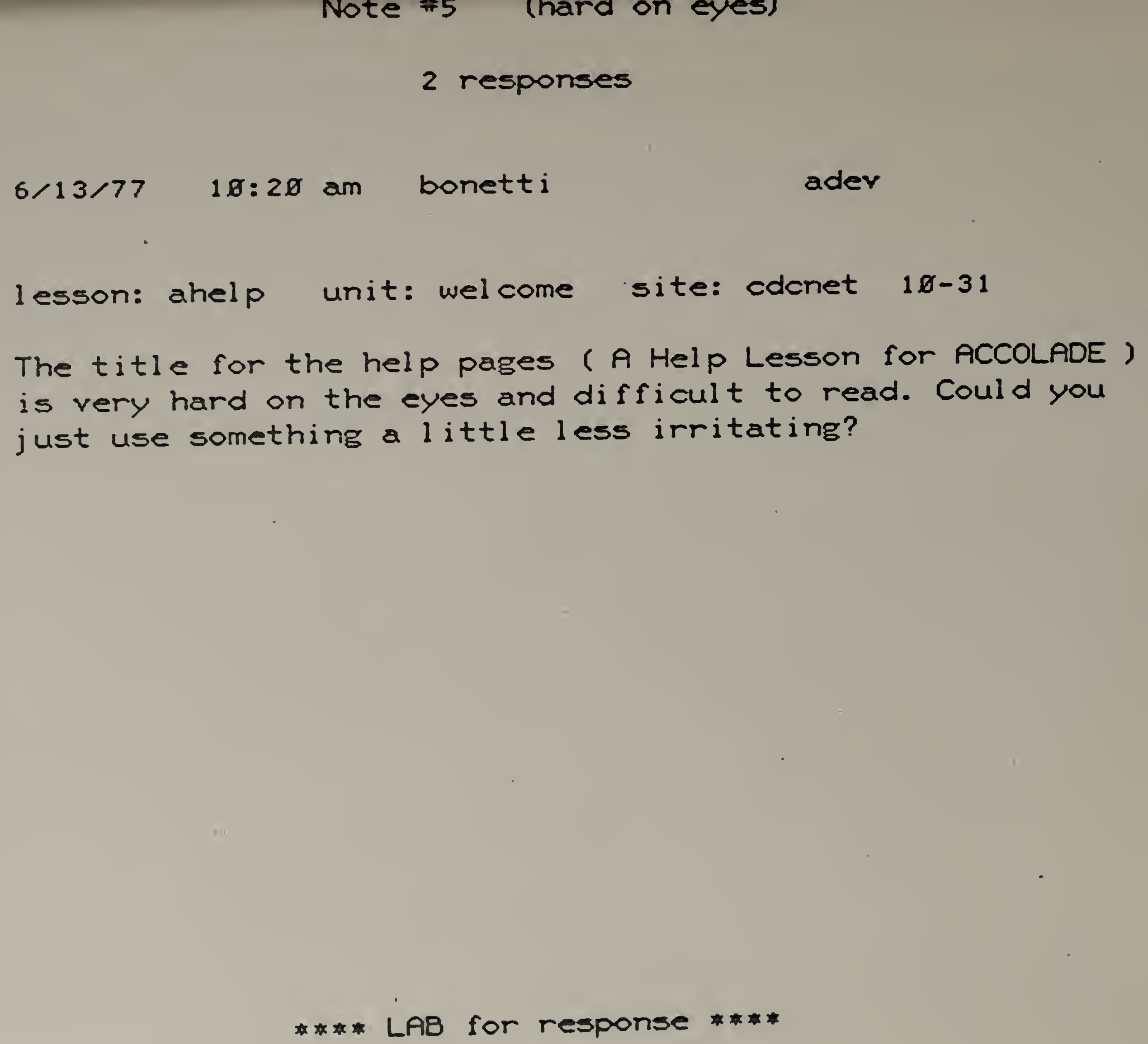
2) Learner Comments

Attached to each Computer component (Yellow Pages, Dictionary, Keyword Index, Bulletin Board, and each CMI Test) is the Plato lesson router file "unmassln." Whenever the Learner is in one of these components and wishes to leave the Teacher a comment without having to return to the Bulletin Board, he need only press the Plato TERM key (yet another Plato special function key not previously mentioned) and then type "comment(s)" and his name and which component he was writing from, along with his comments, will be saved in Plato file "umassln." Figure 10 shows just such a comment from a professional programmer on the Plato system who was acting as an ACCOLADE Learner to help the author



```
Unrecognized Keyword Requests                84 Requests
(87.5% full)                          Total: 229 Requests

 1 unrecognized request

 2 banking and finance

 3 business banking and finance

 4 computer hardware

 5 computer

 6 computer hardware

 7 jgjhjhyiy

 8 performing

 9 artificial intelligence natural language processi

1Ø educational data processing

11 business applications general

12 sfsdfasf

              Press  NEXT  for more
```

Figure 9. Unrecognized Keyword Requests



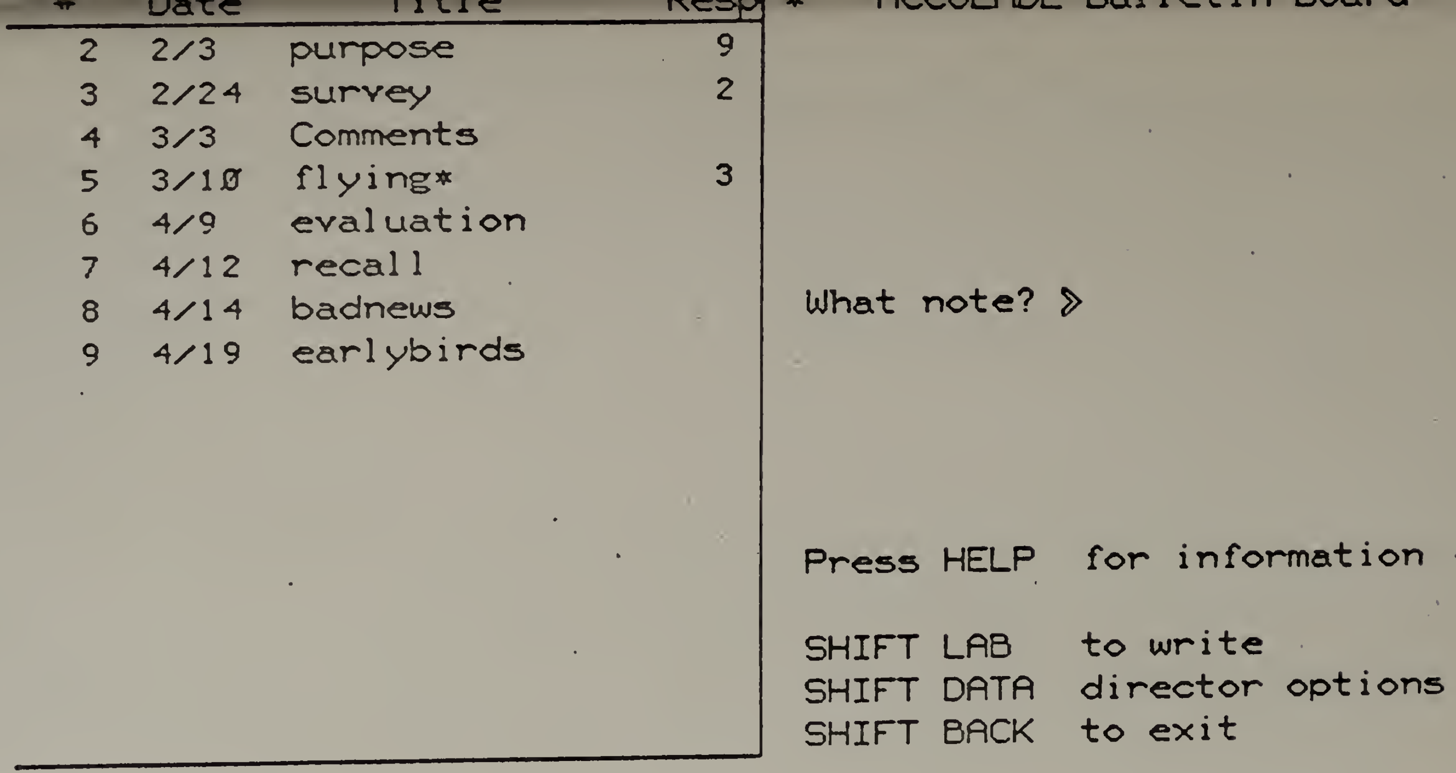
```
                 Note #5    (hard on eyes)

                      2 responses


6/13/77   1Ø:2Ø am   bonetti              adev


lesson: ahelp   unit: welcome   site: cdcnet   1Ø-31

The title for the help pages ( A Help Lesson for ACCOLADE )
is very hard on the eyes and difficult to read. Could you
just use something a little less irritating?








                **** LAB for response ****
```

Figure 10. Learner Comments

improve the help lesson for the Router. In fact, the ACCOLADE Learners did not use this feature at all during the course but several Plato authors did contribute valuable suggestions through this file.

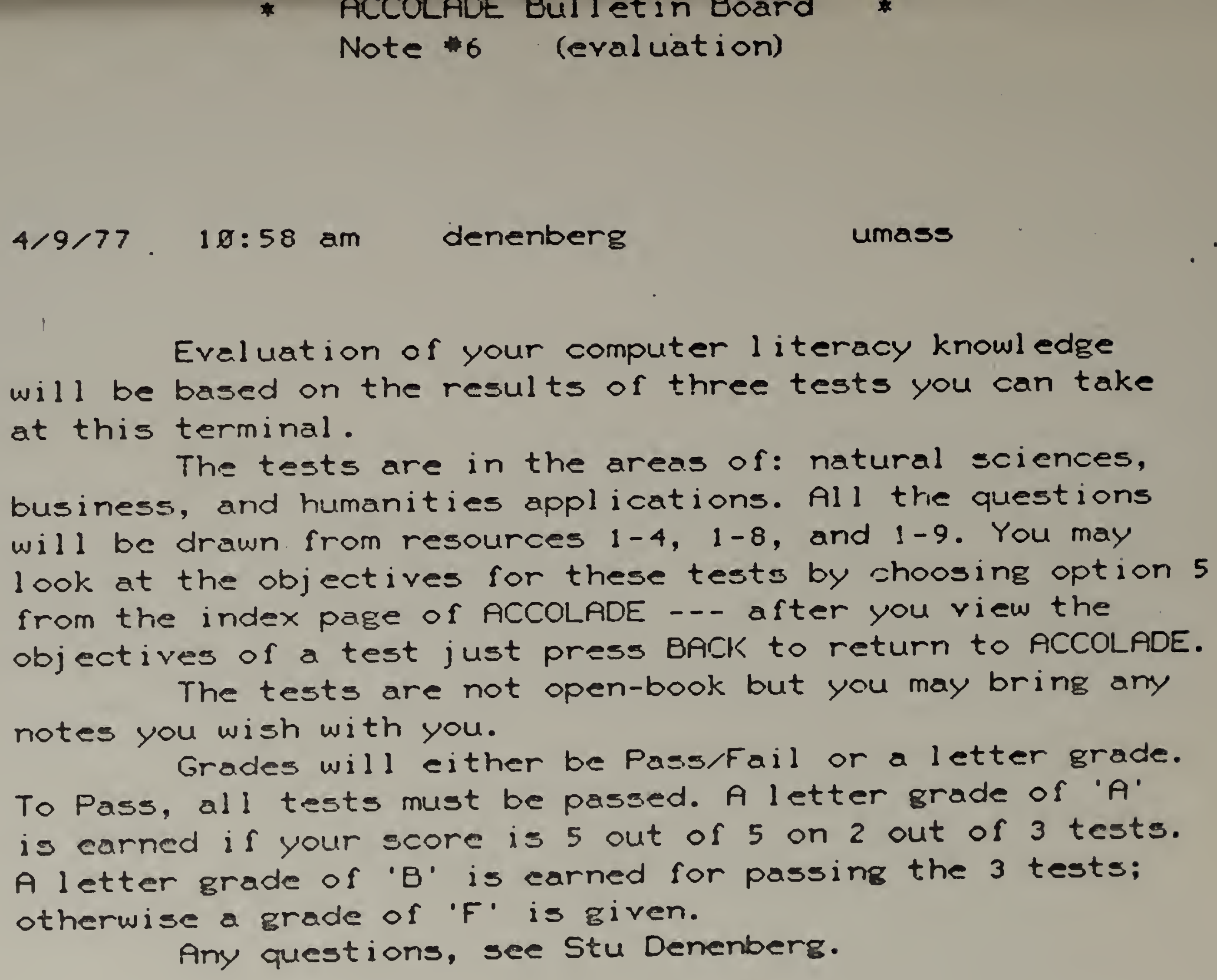
3) Bulletin Board

Previously discussed in section 6.3.3.5, this mechanism can also be a valuable source of Learner feedback to the Teacher but is less private since it can be viewed by all Learners and Teachers in the ACCOLADE System. Figure 11a shows the index page for the Bulletin Board. If a Learner or Teacher types in the number 6 then he will see the note shown in Figure 11b which describes the evaluation criteria for receiving a one credit grade for the knowledge part of the course.



```
 #   Date    Title          Resp  *   ACCOLADE Bulletin Board
 2   2/3     purpose           9
 3   2/24    survey            2
 4   3/3     Comments
 5   3/1Ø    flying*           3
 6   4/9     evaluation
 7   4/12    recall                   What note? »
 8   4/14    badnews
 9   4/19    earlybirds



                                      Press HELP   for information

                                      SHIFT LAB    to write
                                      SHIFT DATA   director options
                                      SHIFT BACK   to exit
```

Figure 11a. Bulletin Board



```
           *    ACCOLADE Bulletin Board    *
                Note #6    (evaluation)


4/9/77   10:58 am    denenberg            umass


        Evaluation of your computer literacy knowledge
will be based on the results of three tests you can take
at this terminal.
        The tests are in the areas of: natural sciences,
business, and humanities applications. All the questions
will be drawn from resources 1-4, 1-8, and 1-9. You may
look at the objectives for these tests by choosing option 5
from the index page of ACCOLADE --- after you view the
objectives of a test just press BACK to return to ACCOLADE.
        The tests are not open-book but you may bring any
notes you wish with you.
        Grades will either be Pass/Fail or a letter grade.
To Pass, all tests must be passed. A letter grade of 'A'
is earned if your score is 5 out of 5 on 2 out of 3 tests.
A letter grade of 'B' is earned for passing the 3 tests;
otherwise a grade of 'F' is given.
        Any questions, see Stu Denenberg.
```

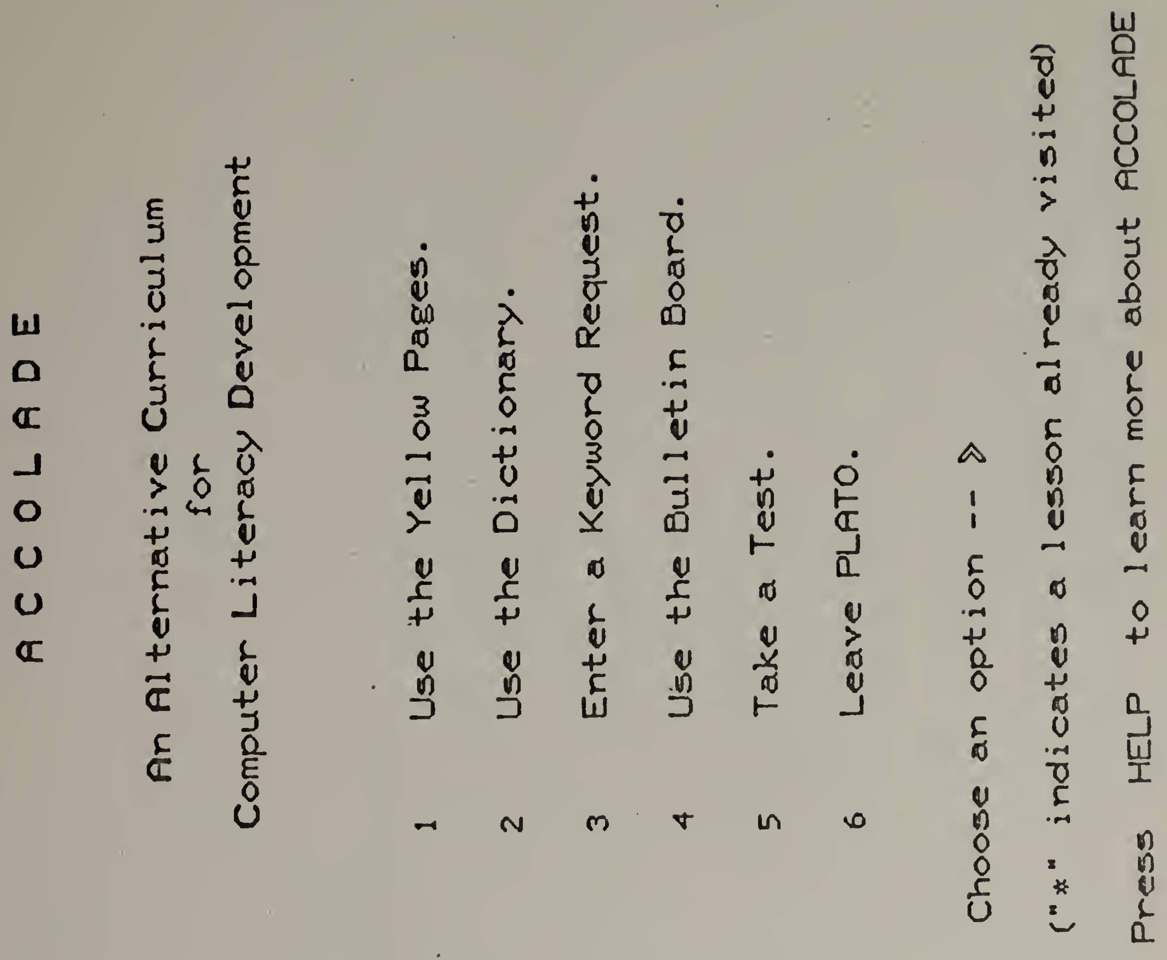
Figure 11b. Note of 6 of Bulletin Board

6.4 An Illustrative Learner Session on the Computer

This section will show a possible Learner interaction with the Computer. All of the following pages have been copied from a Plato terminal screen using a Tektronix 4632 Video Hard Copy Unit.

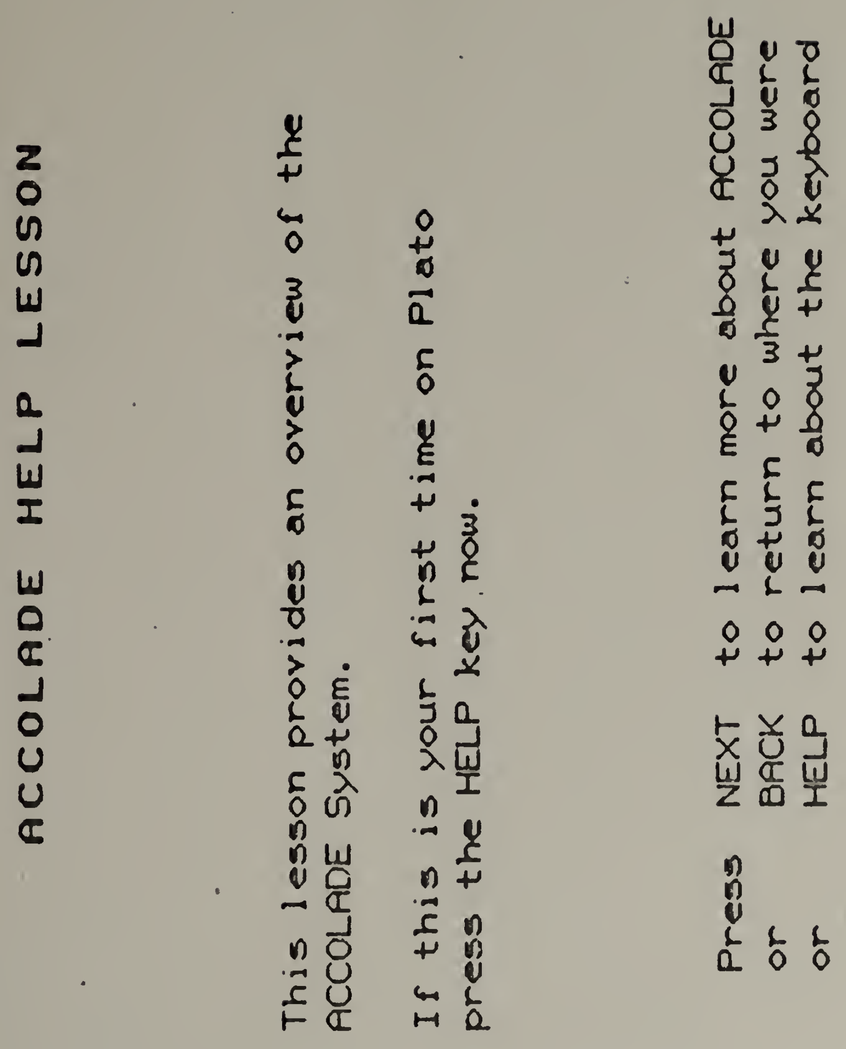
Although the copying device must make its copies along the broad side of the sheet of paper, comments describing the process of the Learner will be typed at the bottom of each page in the normal vertical position, so as to differentiate between the author's running commentary and the displays the Learner actually sees.

Of course, all the time the Learner was interacting with the Computer compponents of ACCOLADE data of the types described in section 6.3.3.6 was being gathered for later viewing and analysis by the Teacher.



```
A C C O L A D E

An Alternative Curriculum
           for
Computer Literacy Development

1    Use the Yellow Pages.
2    Use the Dictionary.
3    Enter a Keyword Request.
4    Use the Bulletin Board.
5    Take a Test.
6    Leave PLATO.

Choose an option -- »

("*" indicates a lesson already visited)

Press  HELP  to learn more about ACCOLADE
```

After the Learner signs on to Plato, he is immediately taken to the Router lesson (this page). He presses the HELP key which takes him to...



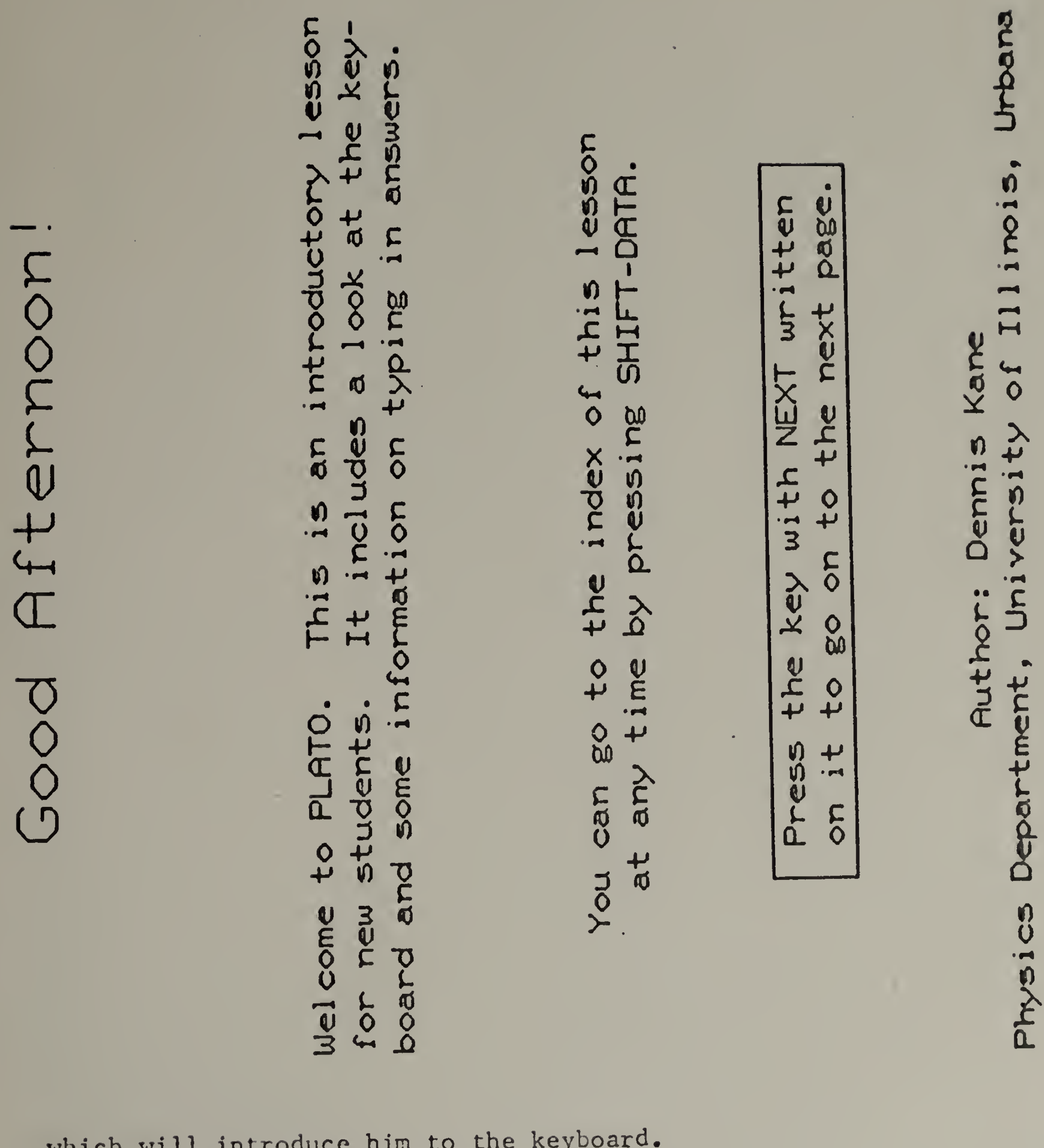
ACCOLADE HELP LESSON

This lesson provides an overview of the ACCOLADE System.

If this is your first time on Plato press the HELP key now.

Press NEXT to learn more about ACCOLADE
or BACK to return to where you were
or HELP to learn about the keyboard

lesson "ahelp", the help lesson for the Router, where an overview of ACCOLADE is presented. If this is the Learner's first time on Plato he may press HELP again and be taken to a lesson...



Good Afternoon!

Welcome to PLATO. This is an introductory lesson for new students. It includes a look at the keyboard and some information on typing in answers.

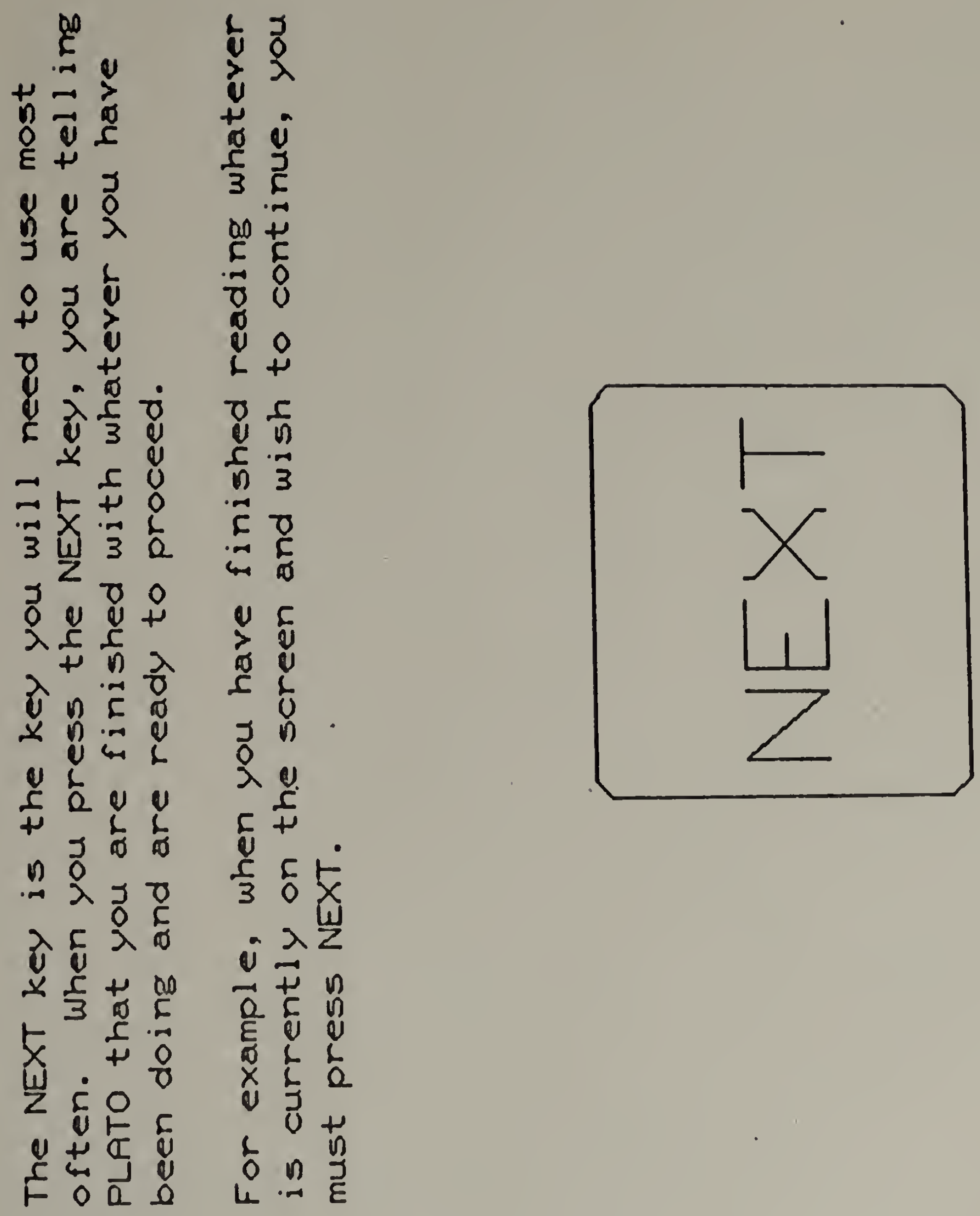
You can go to the index of this lesson at any time by pressing SHIFT-DATA.

Press the key with NEXT written on it to go on to the next page.

Author: Dennis Kane
Physics Department, University of Illinois, Urbana

which will introduce him to the keyboard.

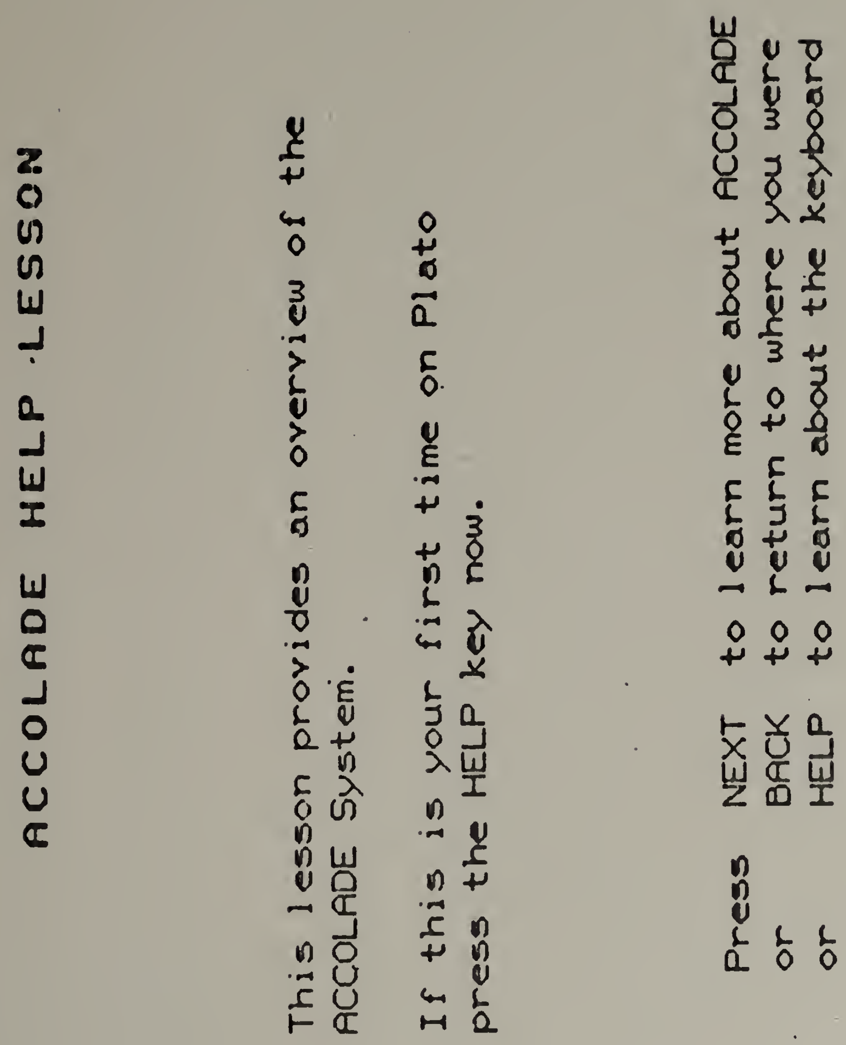
The Learner presses NEXT...



The NEXT key is the key you will need to use most often. When you press the NEXT key, you are telling PLATO that you are finished with whatever you have been doing and are ready to proceed.

For example, when you have finished reading whatever is currently on the screen and wish to continue, you must press NEXT.

and begins his introductory lesson to the Plato Keyboard. when he is finished, he presses BACK to return to...



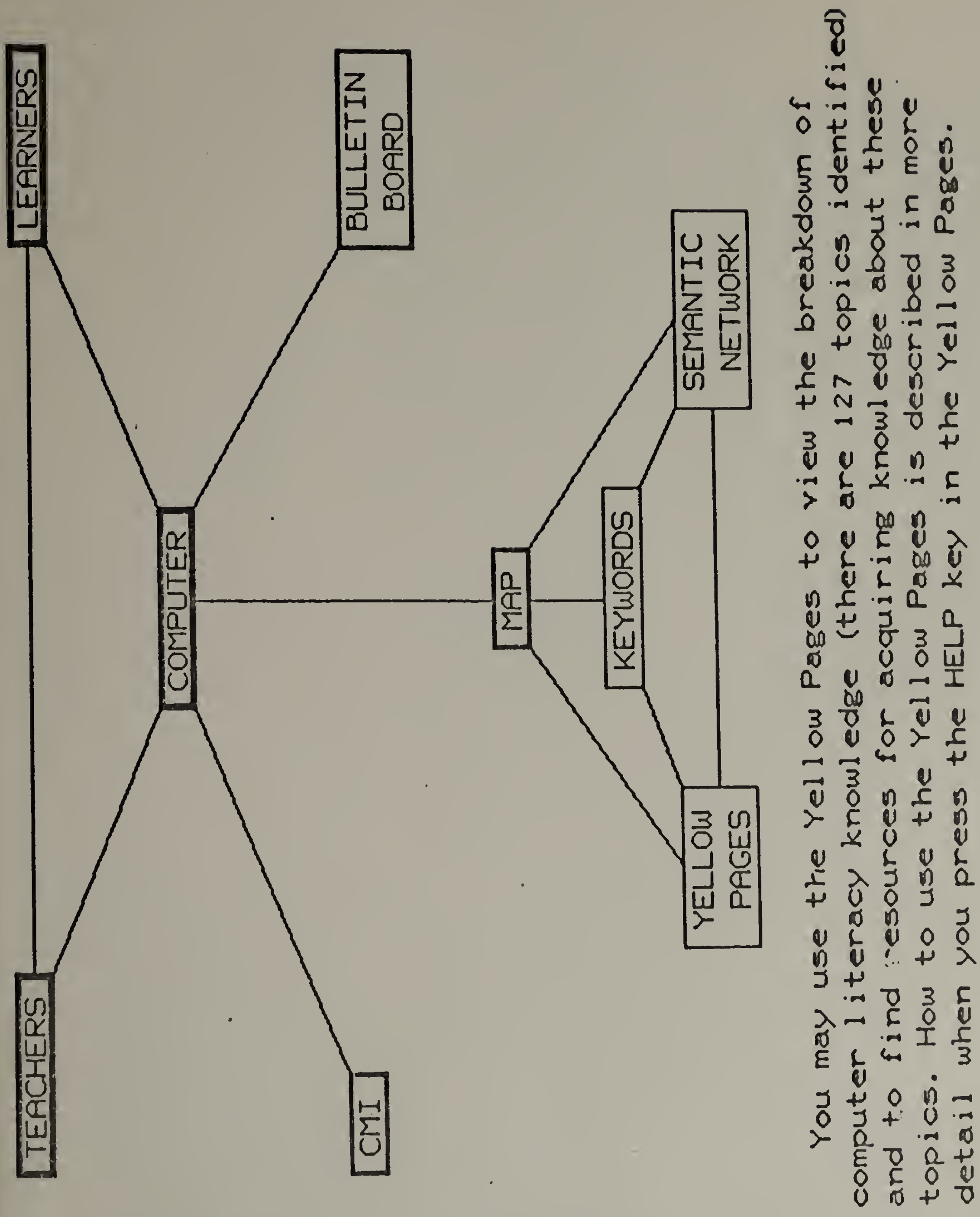
ACCOLADE HELP LESSON

This lesson provides an overview of the
ACCOLADE System.

If this is your first time on Plato
press the HELP key now.

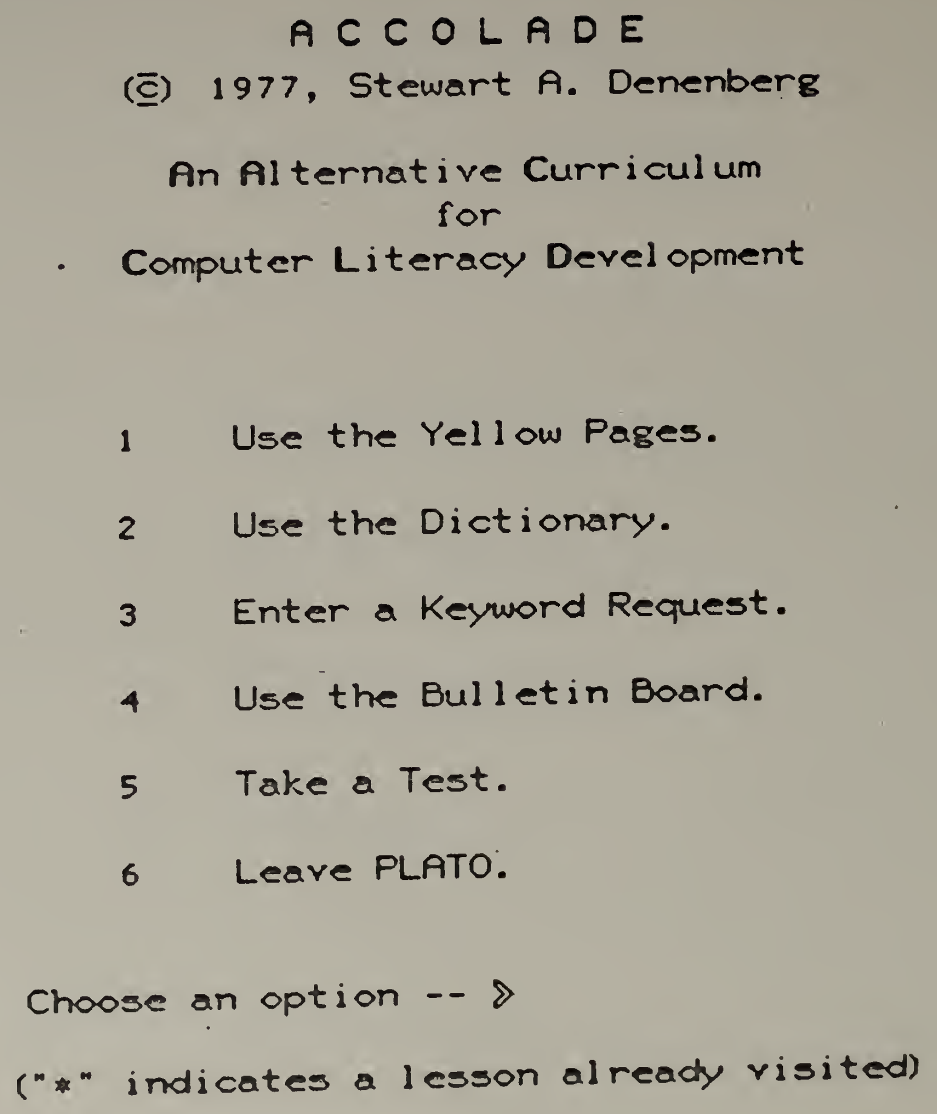
```
Press  NEXT  to learn more about ACCOLADE
or     BACK  to return to where you were
or     HELP  to learn about the keyboard
```

lesson "ahelp" once again. Continually pressing NEXT moves the Learner through this lesson until he reaches...



You may use the Yellow Pages to view the breakdown of computer literacy knowledge (there are 127 topics identified) and to find resources for acquiring knowledge about these topics. How to use the Yellow Pages is described in more detail when you press the HELP key in the Yellow Pages.

about the middle of lesson "ahelp". When he finishes this lesson, he presses BACK and is taken back to...



A C C O L A D E
(©) 1977, Stewart A. Denenberg

An Alternative Curriculum
for
Computer Literacy Development

1 Use the Yellow Pages.

2 Use the Dictionary.

3 Enter a Keyword Request.

4 Use the Bulletin Board.

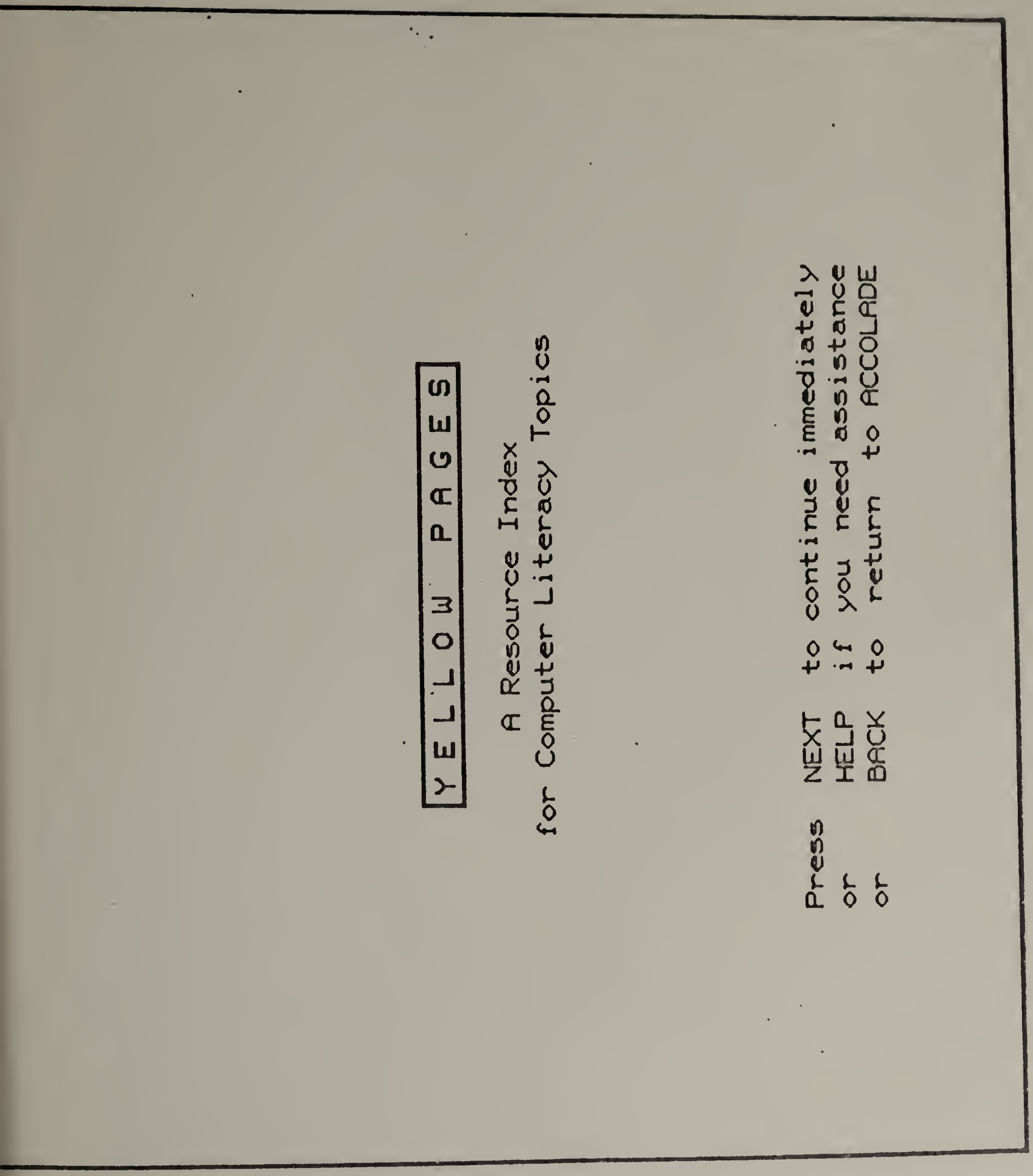
5 Take a Test.

6 Leave PLATO.

Choose an option -- ≫

("*" indicates a lesson already visited)

the Router once again where he now decides to use the Yellow Pages. After typing the number 1, he is taken to...

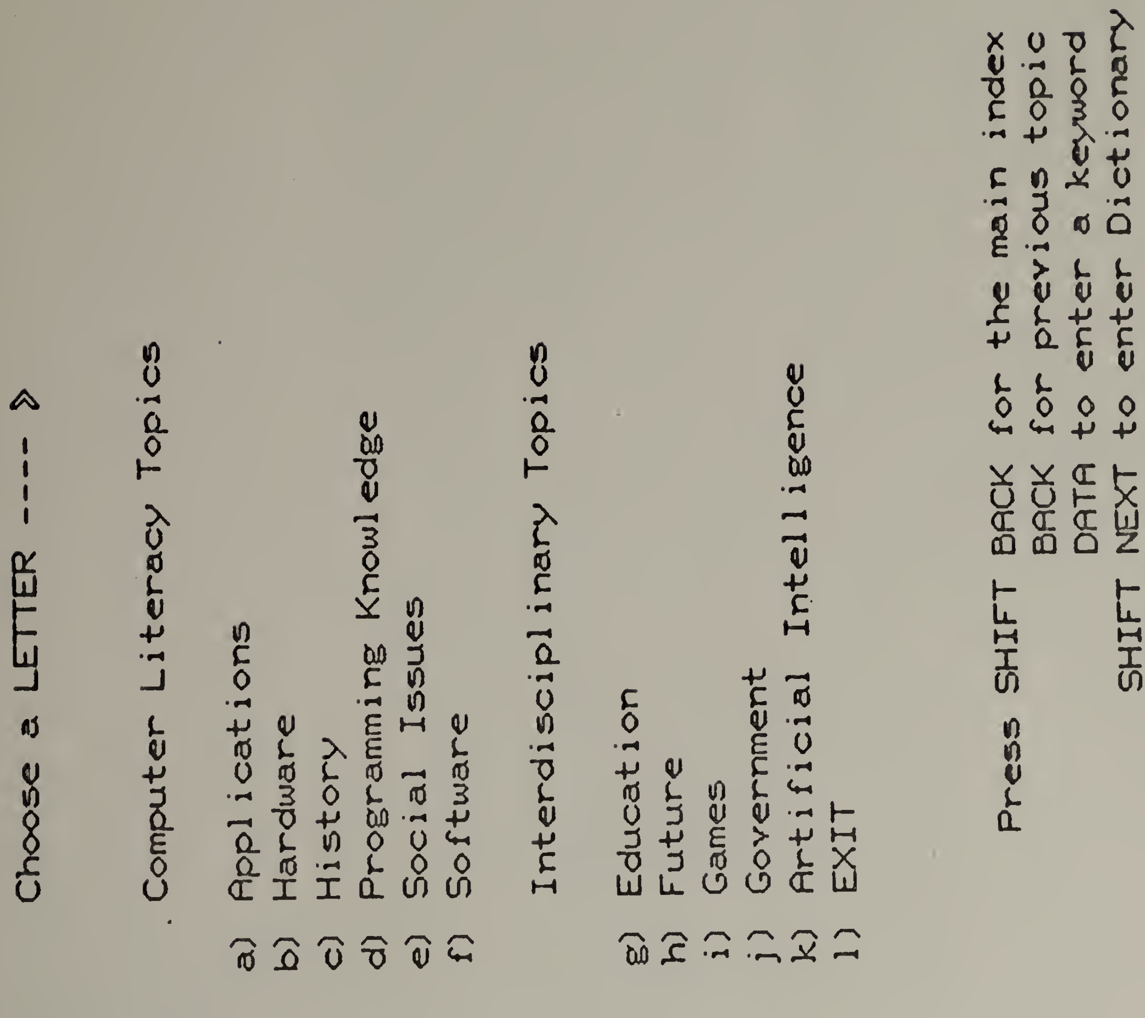


YELLOW PAGES

A Resource Index
for Computer Literacy Topics

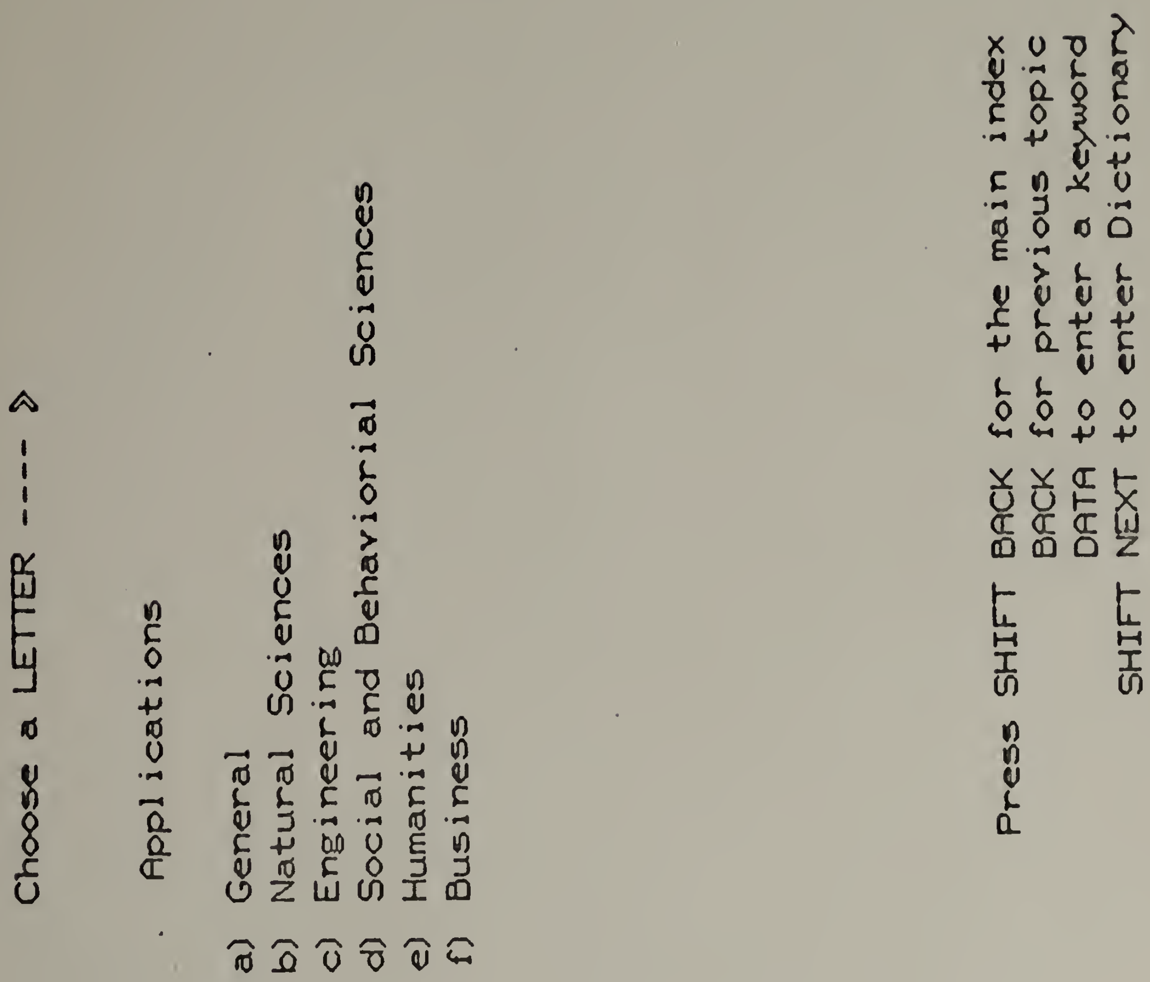
Press NEXT to continue immediately
or HELP if you need assistance
or BACK to return to ACCOLADE

the front page of the Yellow Pages. He presses NEXT ...



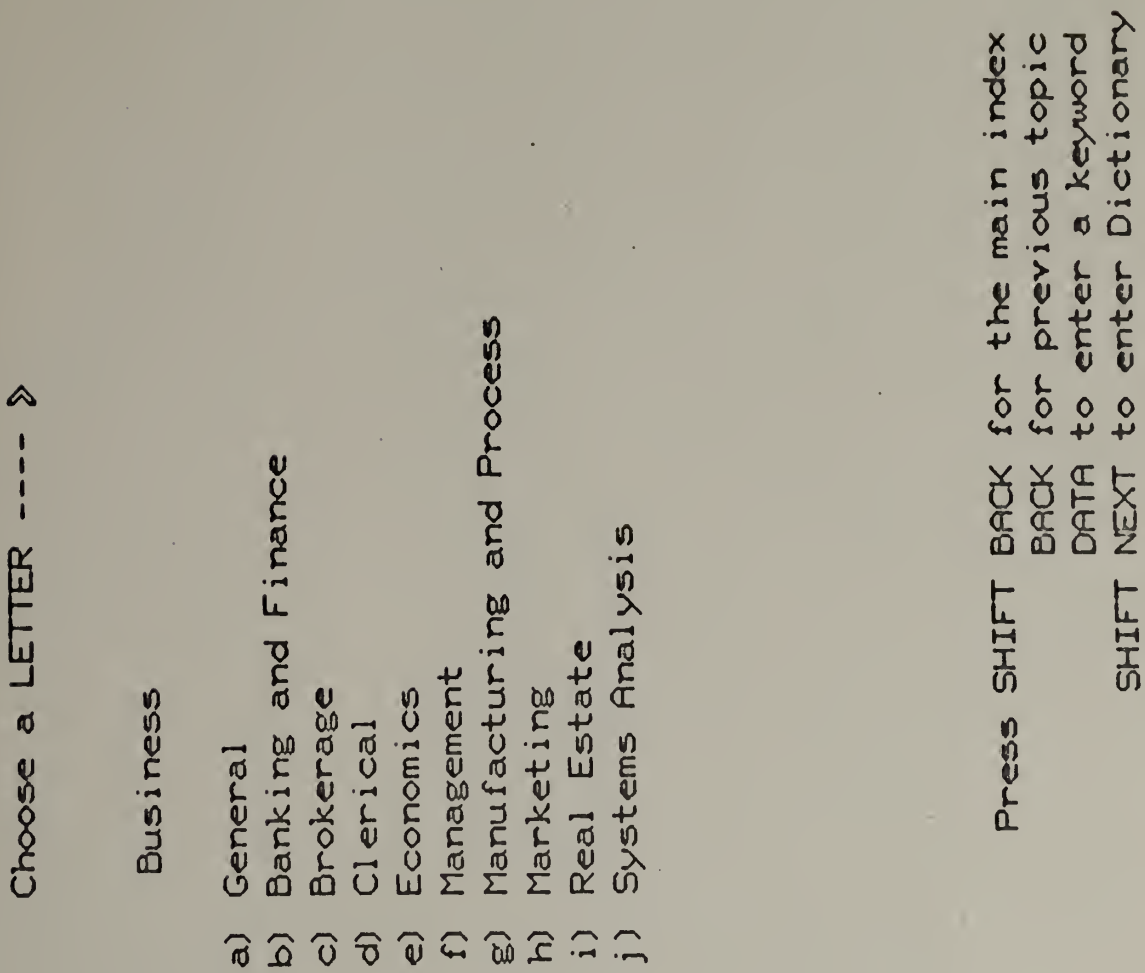
```
Choose a LETTER ---- >

     Computer Literacy Topics

a) Applications
b) Hardware
c) History
d) Programming Knowledge
e) Social Issues
f) Software

     Interdisciplinary Topics

g) Education
h) Future
i) Games
j) Government
k) Artificial Intelligence
l) EXIT



        Press SHIFT BACK for the main index
                   BACK for previous topic
                   DATA to enter a keyword
             SHIFT NEXT to enter Dictionary
```

and sees the main choice page (root node of the tree). He presses the letter'a' and is taken to...



```
Choose a LETTER ---- ≫

   Applications

a) General
b) Natural Sciences
c) Engineering
d) Social and Behavioral Sciences
e) Humanities
f) Business

                    Press SHIFT BACK for the main index
                                BACK for previous topic
                                DATA to enter a keyword
                          SHIFT NEXT to enter Dictionary
```

the topic'applications' where he is invited to make another choice. He types 'f' and is taken to...



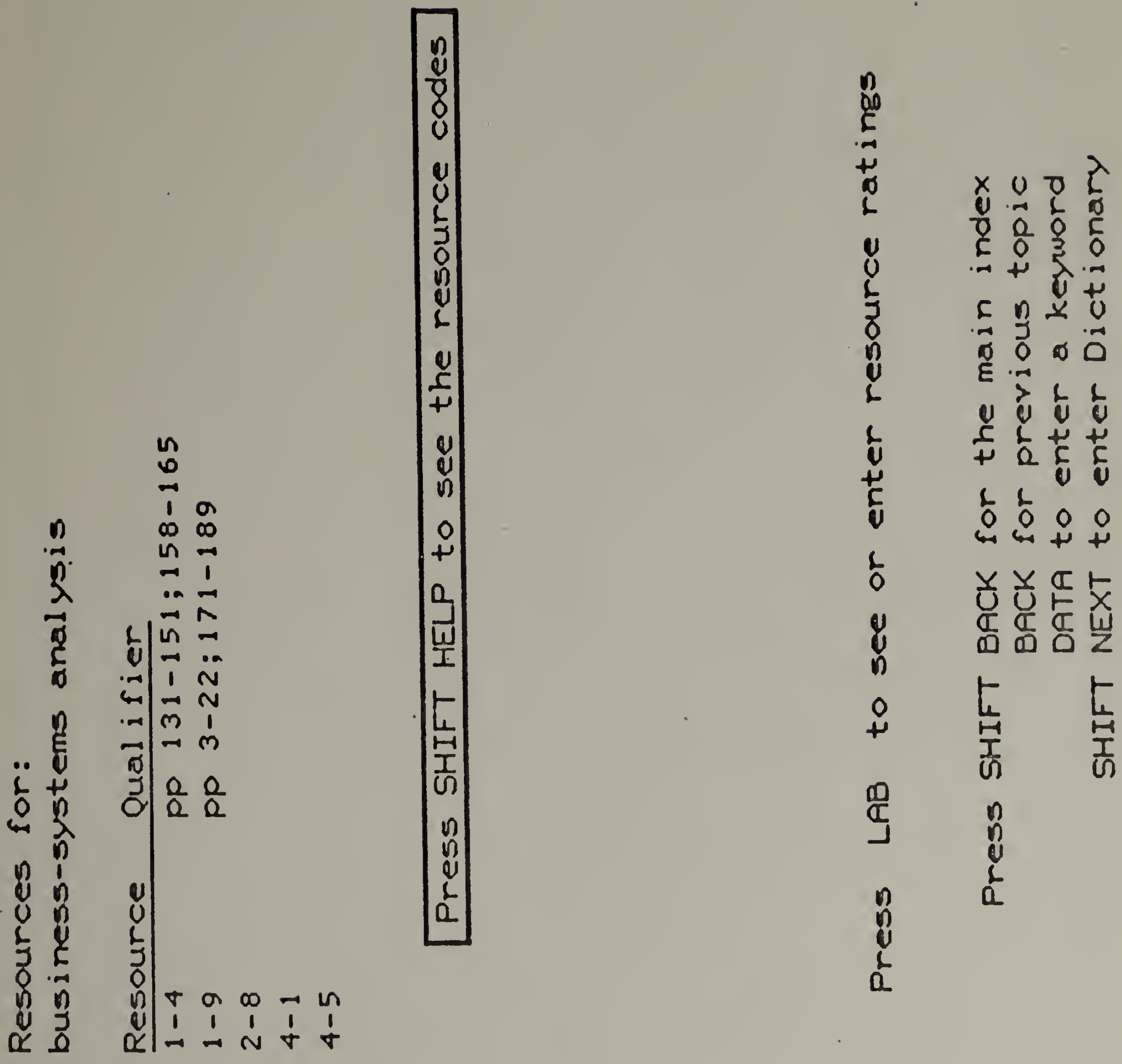
Choose a LETTER ---- >

Business

a) General
b) Banking and Finance
c) Brokerage
d) Clerical
e) Economics
f) Management
g) Manufacturing and Process
h) Marketing
i) Real Estate
j) Systems Analysis

Press SHIFT BACK for the main index
BACK for previous topic
DATA to enter a keyword
SHIFT NEXT to enter Dictionary

the topic 'business' where more choices are available. He presses 'j' and is taken to...



Resources for:
business-systems analysis

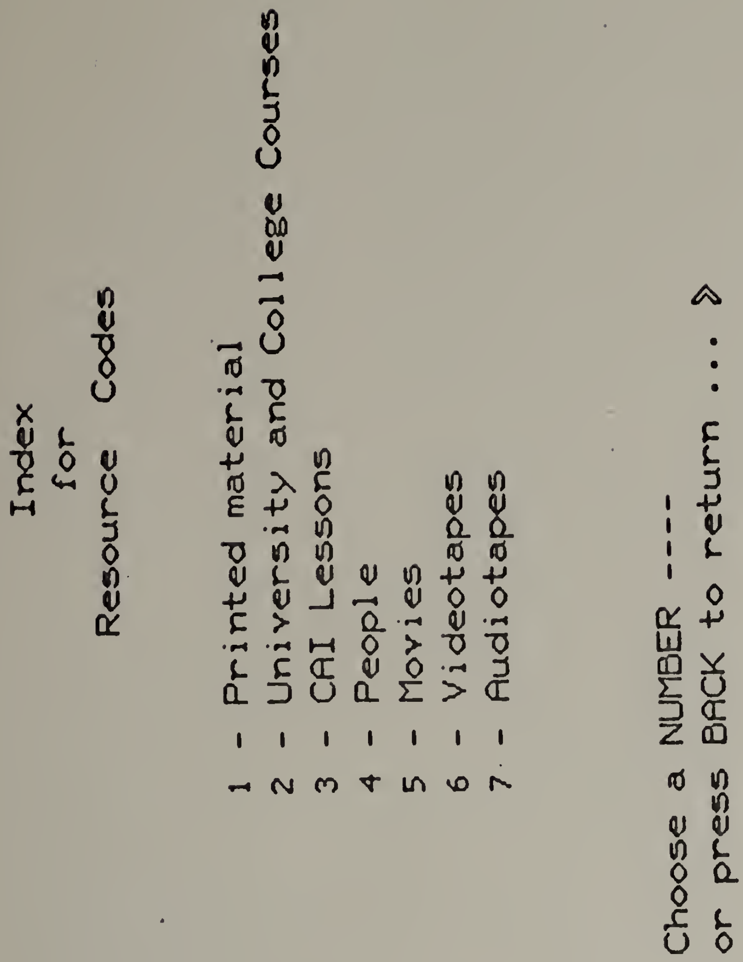
Resource	Qualifier
1-4	pp 131-151;158-165
1-9	pp 3-22;171-189
2-8	
4-1	
4-5	

Press SHIFT HELP to see the resource codes

Press LAB to see or enter resource ratings

Press SHIFT BACK for the main index
BACK for previous topic
DATA to enter a keyword
SHIFT NEXT to enter Dictionary

the topic of 'systems-analysis' which is a terminal topic (leaf node in the tree) and, as such, has resources associated with it. The resources are coded and the Learner must press SHIFT HELP to decode their meaning. He wishes to find out what resource 1-9 is, so he presses SHIFT HELP and is taken to...

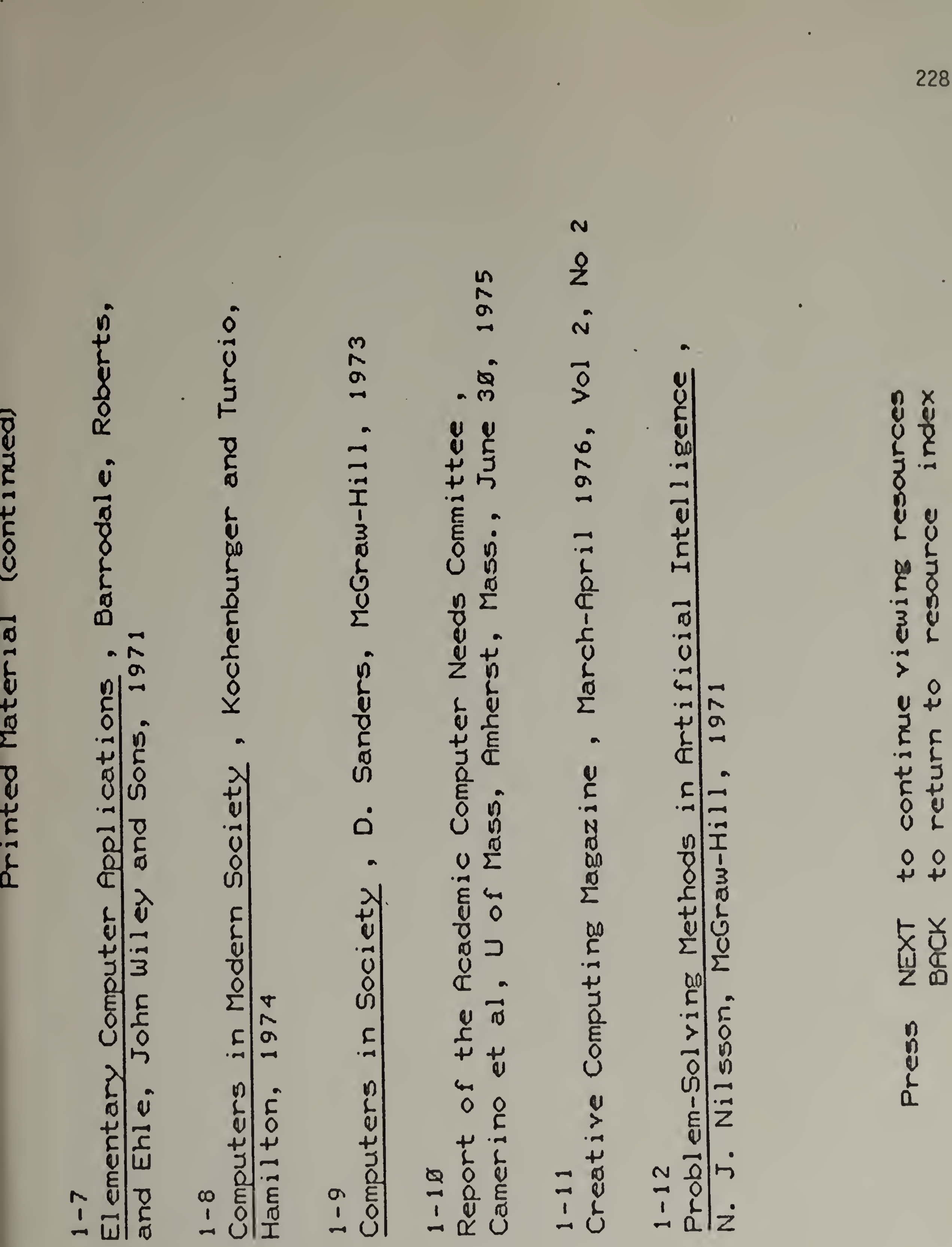


Index
for
Resource Codes

1 - Printed material
2 - University and College Courses
3 - CAI Lessons
4 - People
5 - Movies
6 - Videotapes
7 - Audiotapes

Choose a NUMBER ----
or press BACK to return ... »

an index page for decoding the resource codes. Since he is interested in resource 1-9 (the 9th resource of type 1) he types in a '1' and after moving forward one page he sees...



Printed Material (continued)

1-7
Elementary Computer Applications , Barrodale, Roberts,
and Ehle, John Wiley and Sons, 1971

1-8
Computers in Modern Society , Kochenburger and Turcio,
Hamilton, 1974

1-9
Computers in Society , D. Sanders, McGraw-Hill, 1973

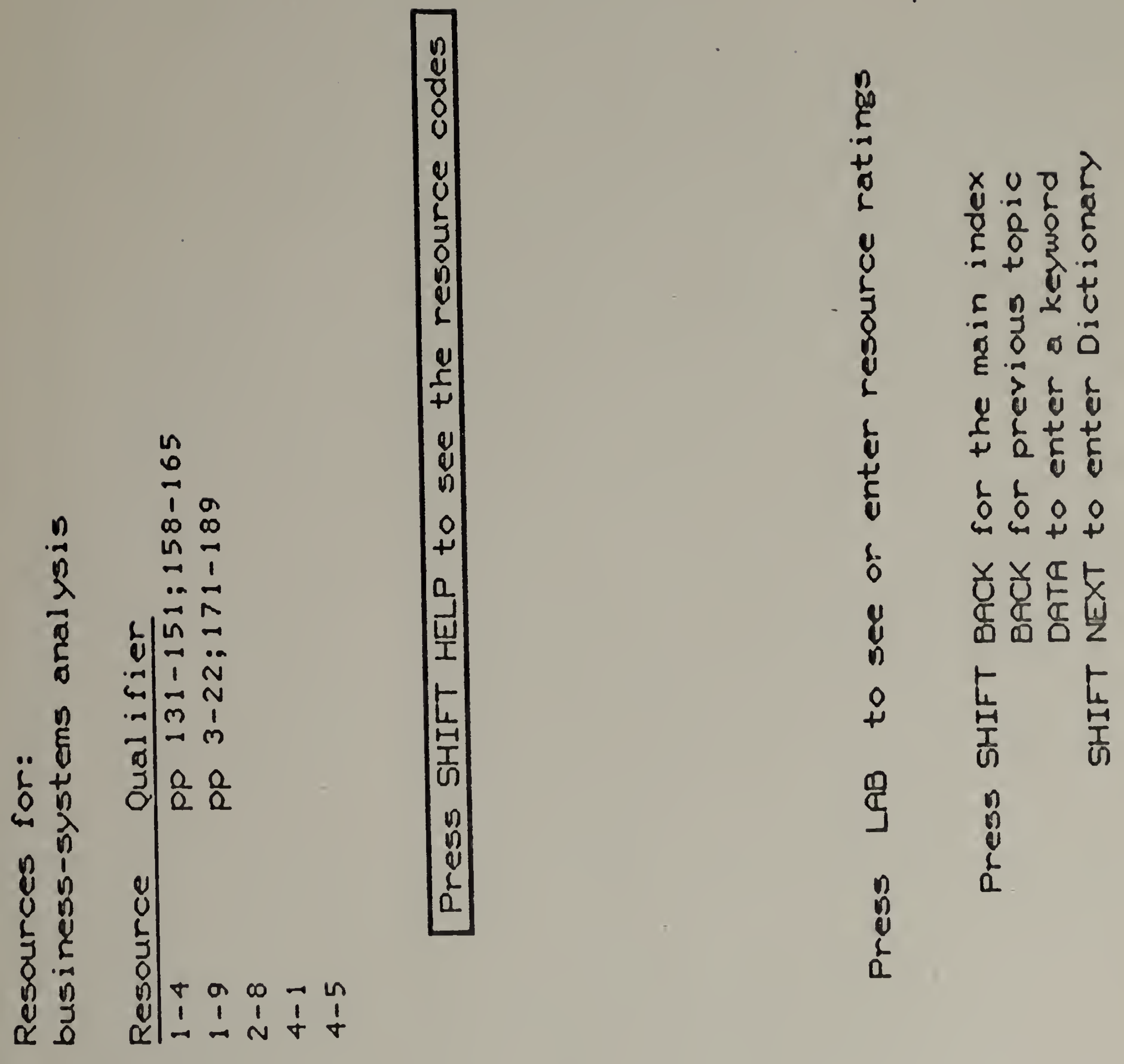
1-1Ø
Report of the Academic Computer Needs Committee ,
Camerino et al, U of Mass, Amherst, Mass., June 3Ø, 1975

1-11
Creative Computing Magazine , March-April 1976, Vol 2, No 2

1-12
Problem-Solving Methods in Artificial Intelligence ,
N. J. Nilsson, McGraw-Hill, 1971

Press NEXT to continue viewing resources
BACK to return to resource index

the title and author of resource 1-9 which is available in the same room as the Plato terminal. He presses BACK to get back to...



Resources for:
business-systems analysis

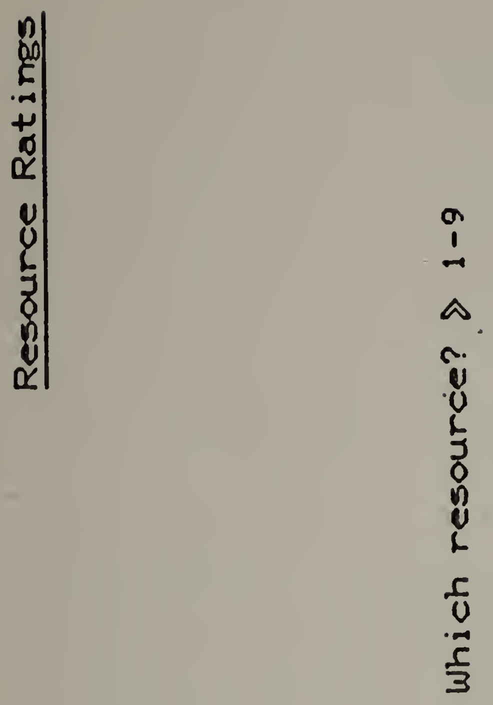
Resource	Qualifier
1-4	pp 131-151;158-165
1-9	pp 3-22;171-189
2-8	
4-1	
4-5	

Press SHIFT HELP to see the resource codes

Press LAB to see or enter resource ratings

```
Press SHIFT BACK for the main index
            BACK for previous topic
            DATA to enter a keyword
      SHIFT NEXT to enter Dictionary
```

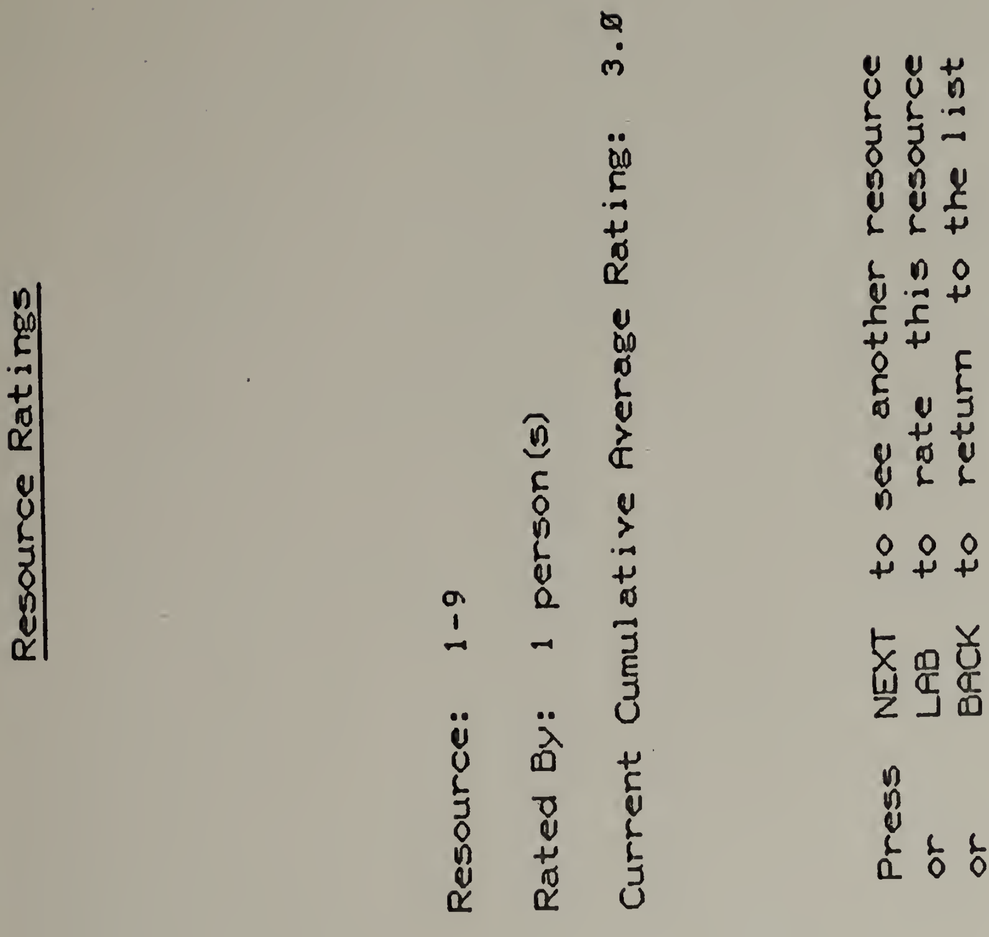
the resource page for the topic 'systems analysis' and decides to see what the peer group ratings are for resource 1-9 by pressing LAB--he is taken to...



Resource Ratings

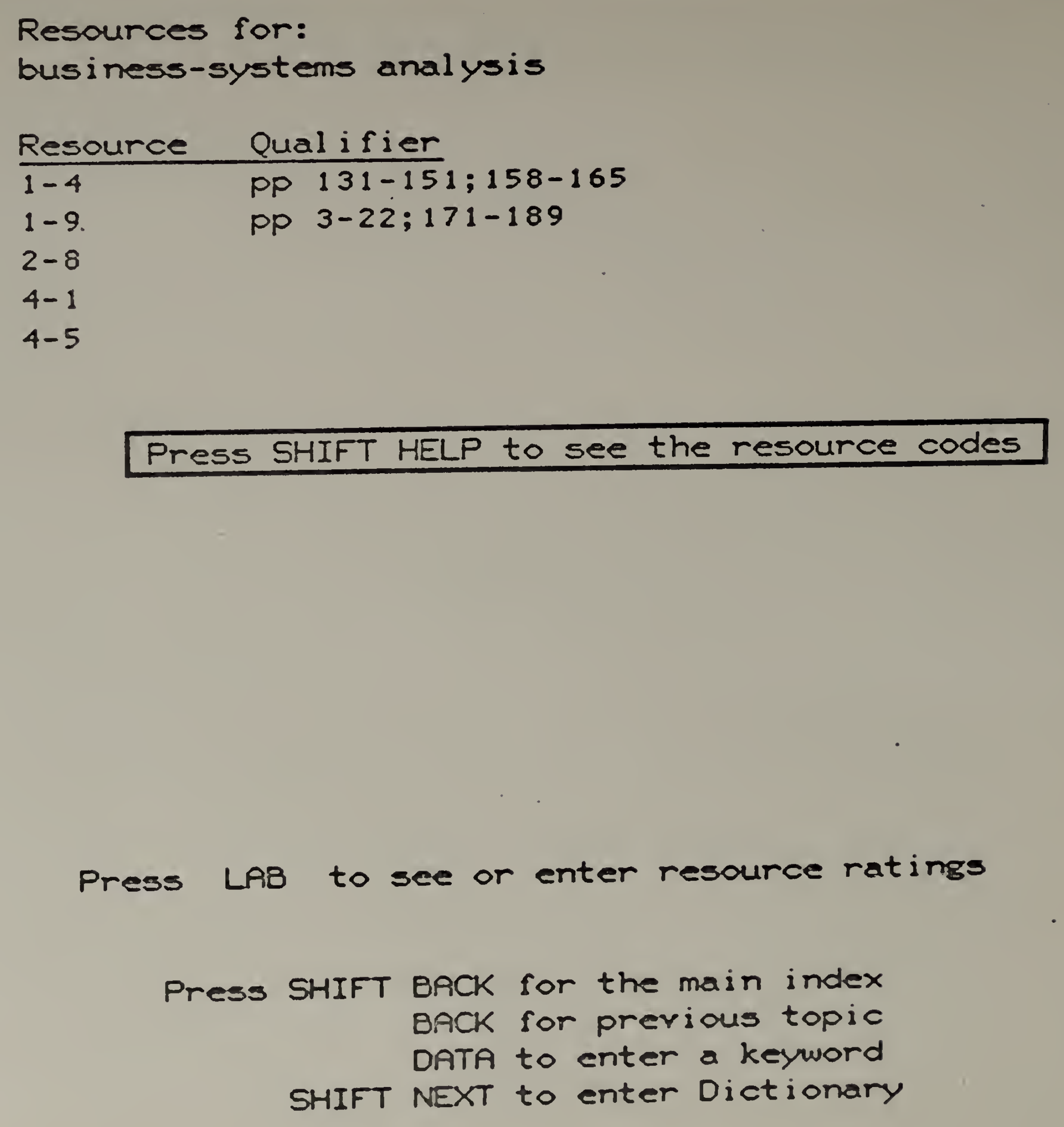
Which resource? ≫ 1-9

a page which invites him to see or rate a resource. He types in the resource (1-9) he wishes to see the rating for...



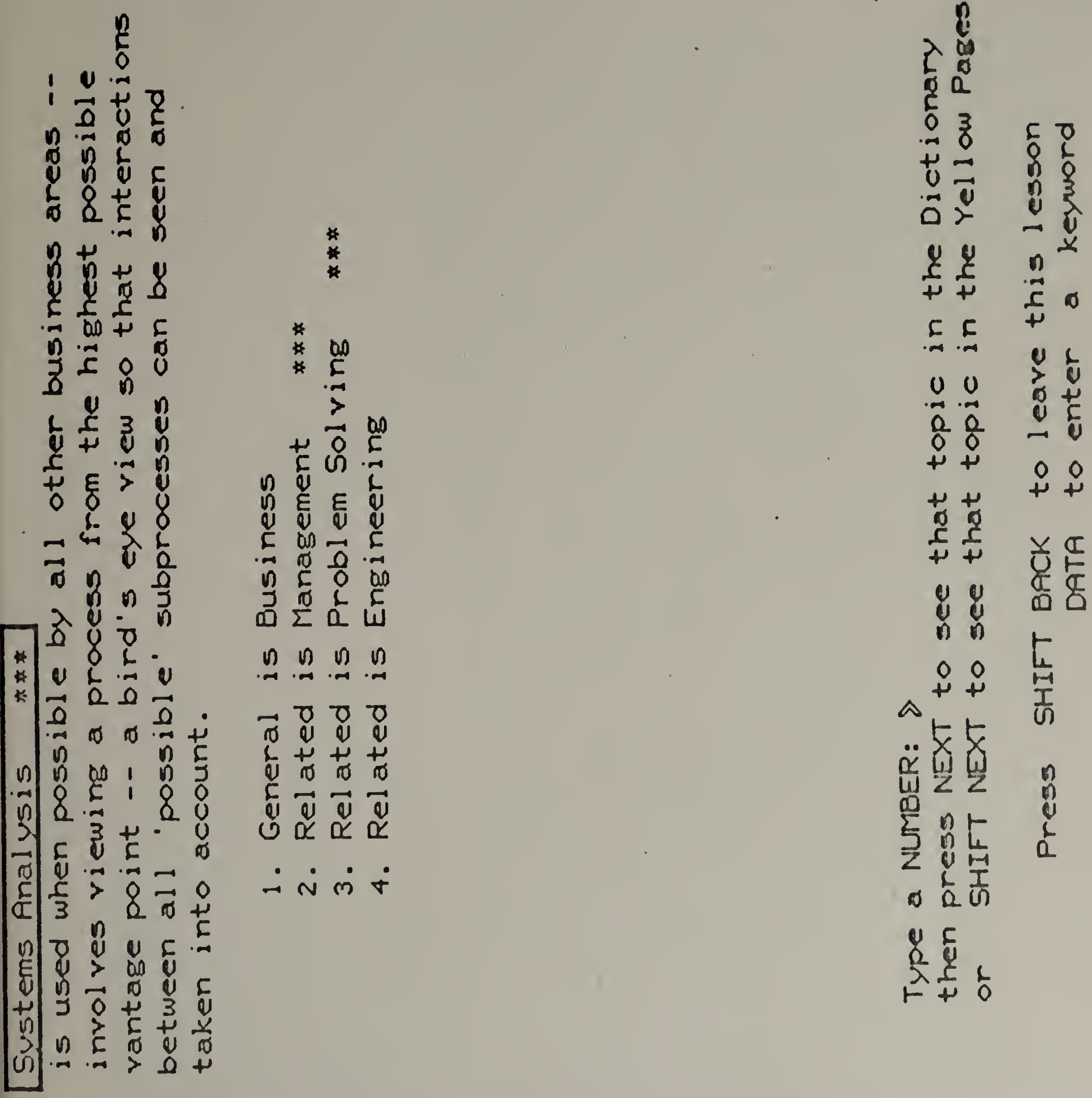
```
Resource Ratings

Resource:   1-9

Rated By:   1 person(s)

Current Cumulative Average Rating:   3.Ø

Press   NEXT   to see another resource
or      LAB    to rate this resource
or      BACK   to return to the list
```

and finds that it has only been rated by one other person and that its rating is just mediocre. He presses BACK to return to...



```
Resources for:
business-systems analysis

Resource    Qualifier
1-4         pp 131-151;158-165
1-9.        pp 3-22;171-189
2-8
4-1
4-5

  Press SHIFT HELP to see the resource codes


 Press  LAB  to see or enter resource ratings


   Press SHIFT BACK for the main index
               BACK for previous topic
               DATA to enter a keyword
         SHIFT NEXT to enter Dictionary
```

the resource page for the topic 'systems analysis' and now decides to explore further and view this topic in the Dictionary, so he presses SHIFT NEXT and is taken to...



Systems Analysis ***

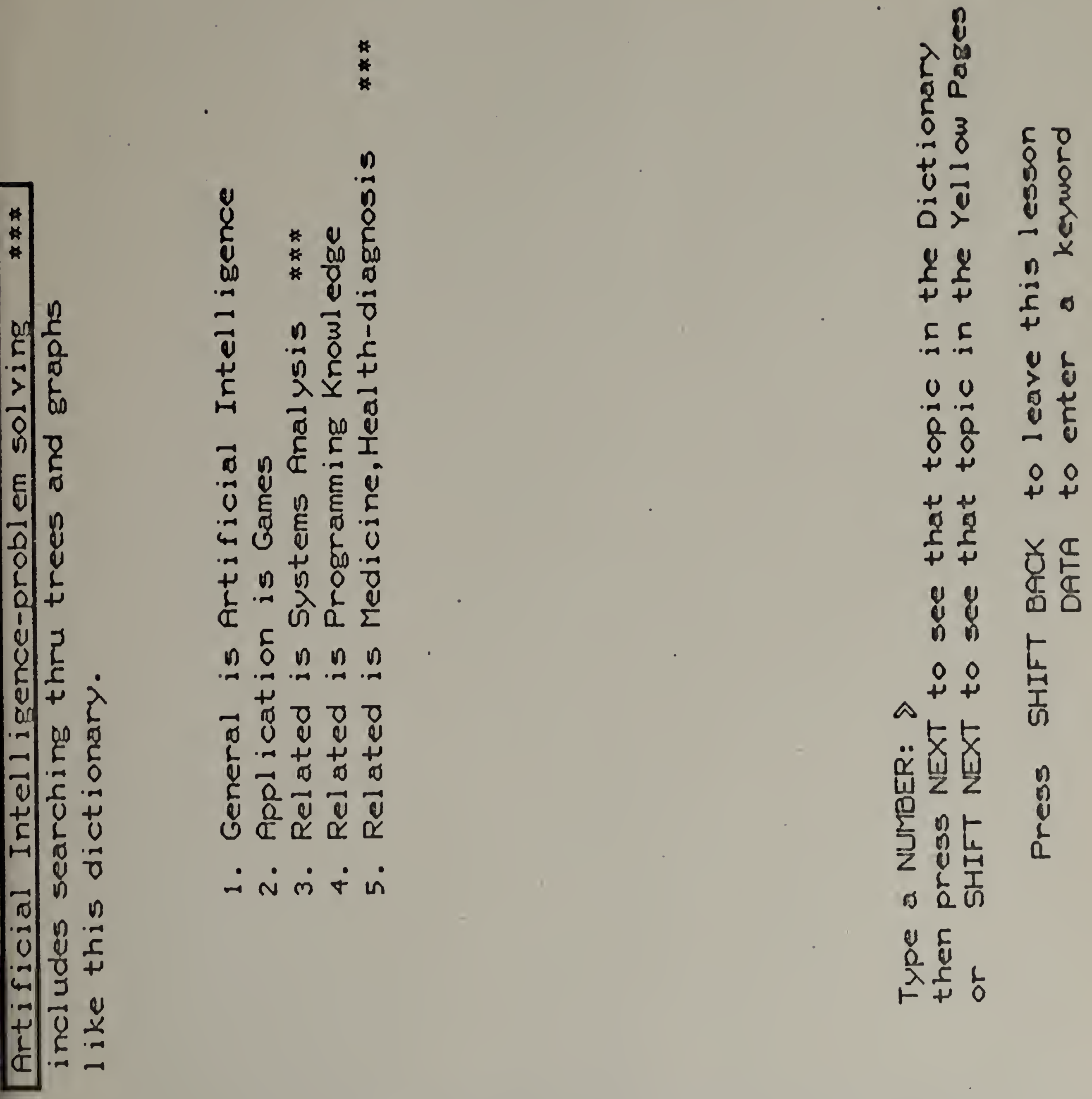
is used when possible by all other business areas -- involves viewing a process from the highest possible vantage point -- a bird's eye view so that interactions between all 'possible' subprocesses can be seen and taken into account.

1. General is Business
2. Related is Management ***
3. Related is Problem Solving ***
4. Related is Engineering

Type a NUMBER: »
then press NEXT to see that topic in the Dictionary
or SHIFT NEXT to see that topic in the Yellow Pages

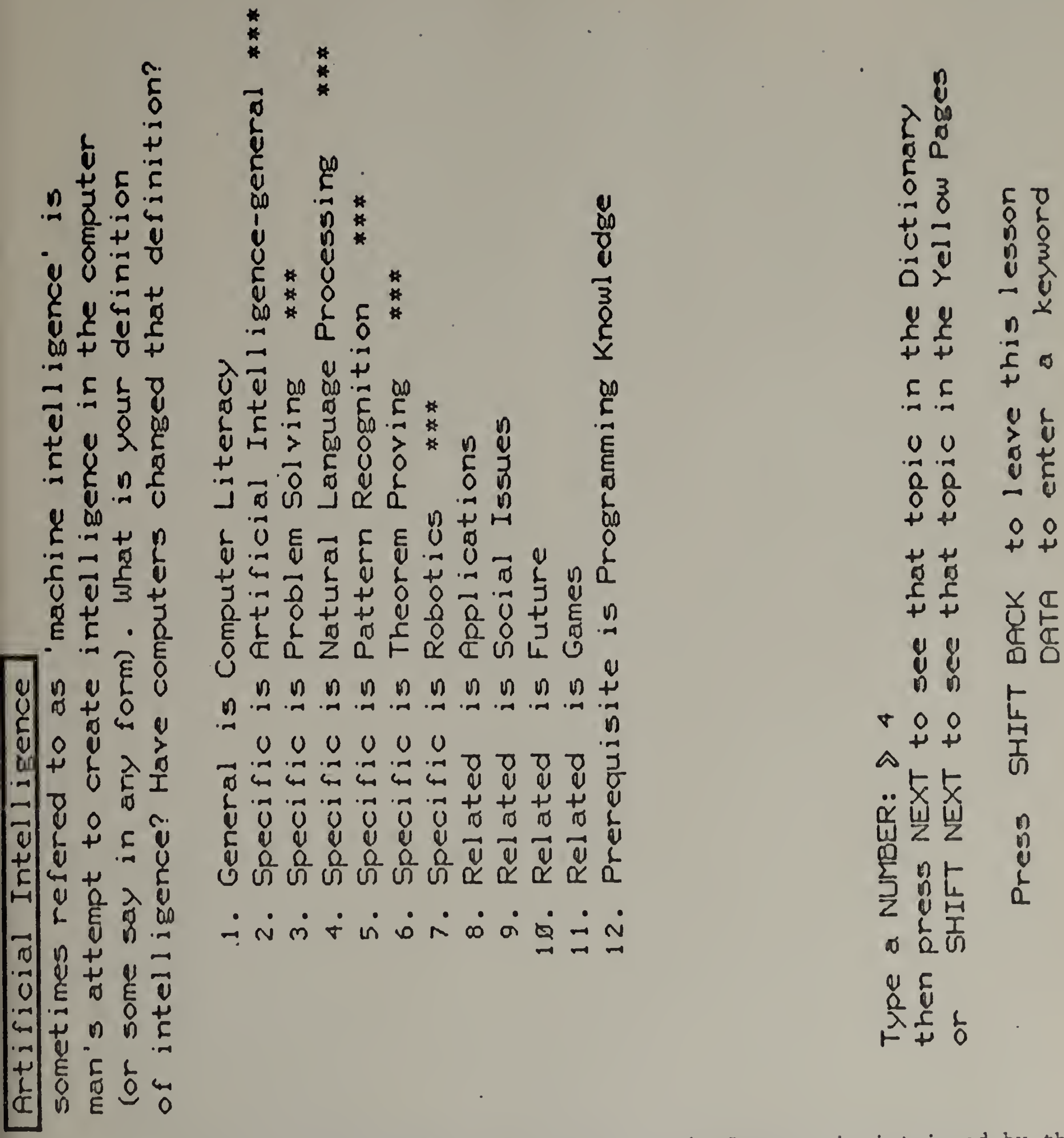
Press SHIFT BACK to leave this lesson
DATA to enter a keyword

the topic 'systems analysis' in the Dictionary. He becomes interested in seeing where the topic 'problem solving' will lead, so he types a '3' and presses NEXT and is taken to...



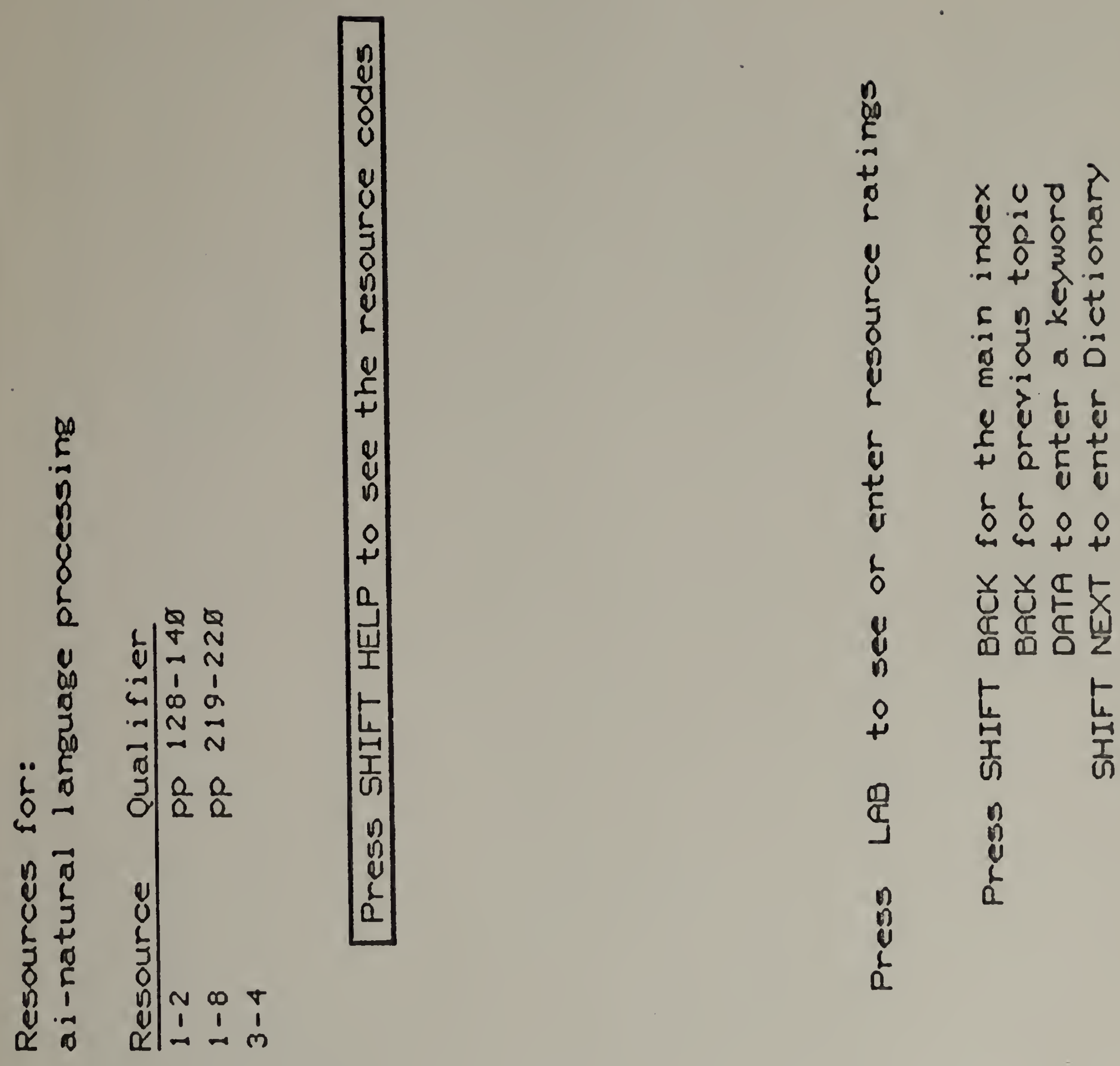
```
Artificial Intelligence-problem solving  ***
includes searching thru trees and graphs
like this dictionary.


  1. General is Artificial Intelligence
  2. Application is Games
  3. Related is Systems Analysis  ***
  4. Related is Programming Knowledge
  5. Related is Medicine,Health-diagnosis   ***




Type a NUMBER: »
then press NEXT to see that topic in the Dictionary
or   SHIFT NEXT to see that topic in the Yellow Pages

   Press  SHIFT BACK  to leave this lesson
                DATA  to enter  a  keyword
```

the topic 'problem solving' in the Dictionary (to view the space around that node in the semantic network). Wishing to see the structure around the more general topic 'artificial intelligence' he presses '1' followed by NEXT and is taken to...



```
Artificial Intelligence
sometimes refered to as 'machine intelligence' is
man's attempt to create intelligence in the computer
(or some say in any form). What is your definition
of intelligence? Have computers changed that definition?

  1.  General is Computer Literacy
  2.  Specific is Artificial Intelligence-general ***
  3.  Specific is Problem Solving   ***
  4.  Specific is Natural Language Processing    ***
  5.  Specific is Pattern Recognition   ***
  6.  Specific is Theorem Proving   ***
  7.  Specific is Robotics   ***
  8.  Related  is Applications
  9.  Related  is Social Issues
 1Ø.  Related  is Future
 11.  Related  is Games
 12.  Prerequisite is Programming Knowledge



Type a NUMBER: ≫ 4
then press NEXT to see that topic in the Dictionary
or   SHIFT NEXT to see that topic in the Yellow Pages

     Press  SHIFT BACK  to leave this lesson
                  DATA  to enter a keyword
```

the topic of 'artificial intelligence'. The Learner is intrigued by the topic of 'natural language processing' and wishes to learn more about it. He notes that there are resources for acquiring knowledge about that topic (the presence of asterisks) in the Yellow Pages so he decides to go directly there by typing '4' and SHIFT NEXT. This takes him to...



Resources for:
ai-natural language processing

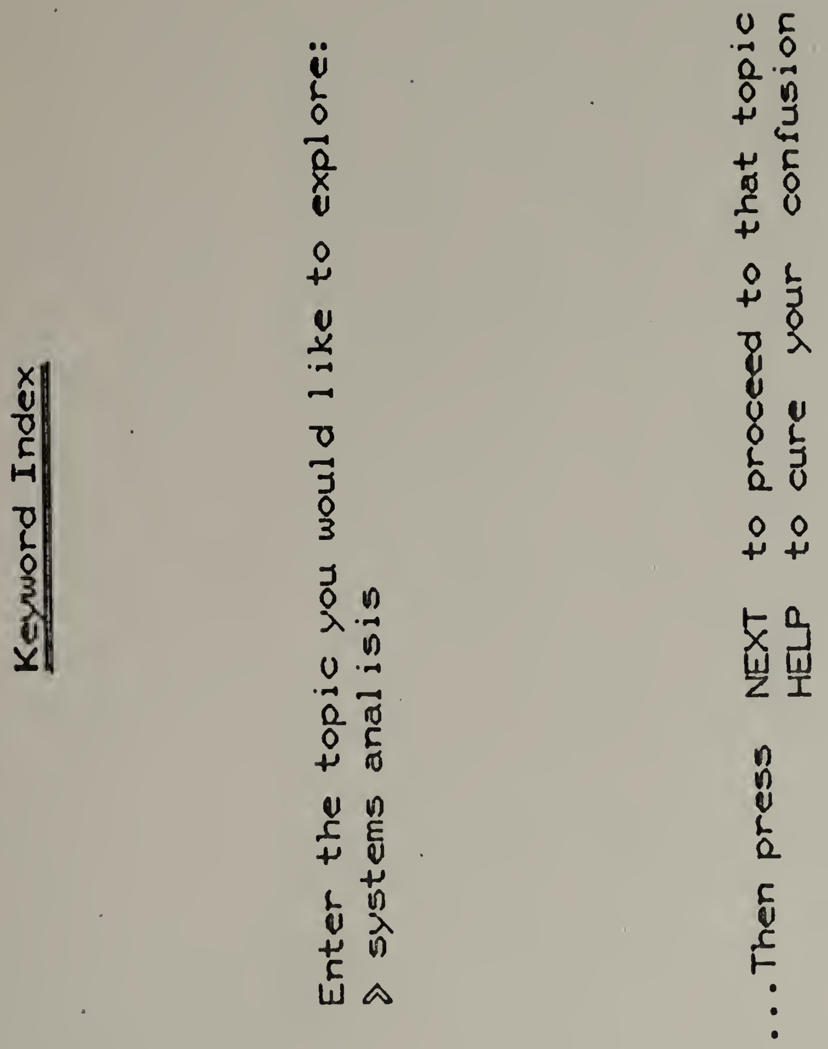
Resource	Qualifier
1-2	pp 128-14Ø
1-8	pp 219-22Ø
3-4	

Press SHIFT HELP to see the resource codes

Press LAB to see or enter resource ratings

Press SHIFT BACK for the main index
BACK for previous topic
DATA to enter a keyword
SHIFT NEXT to enter Dictionary

the resource page for the topic 'natural language processing' in the Yellow Pages. After decoding and noting the resources for this topic he wishes to return to the topic of 'systems analysis' without having to search the tree of the Yellow Pages. So he presses DATA and is taken to...

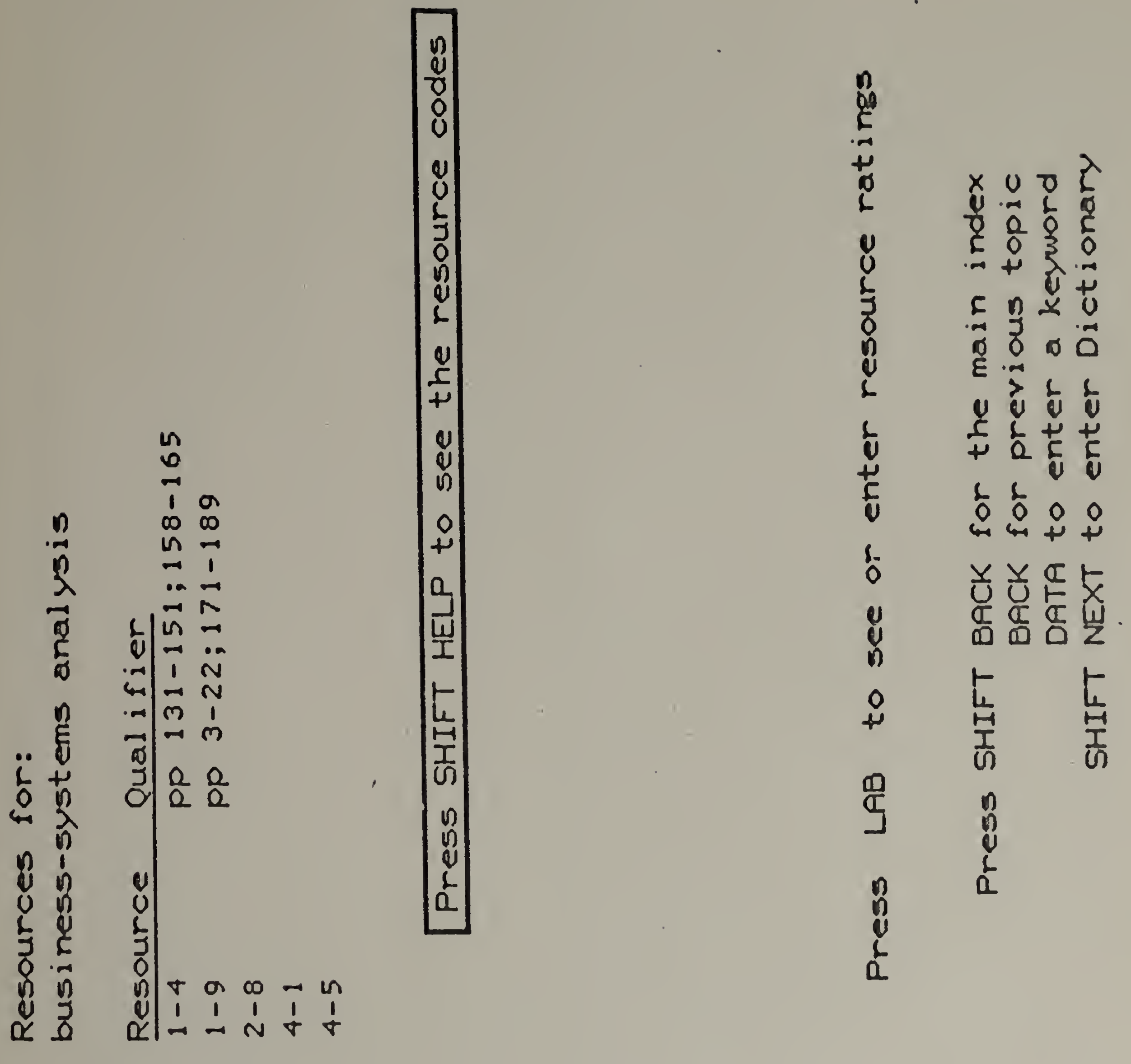


Keyword Index

Enter the topic you would like to explore:
» systems analisis

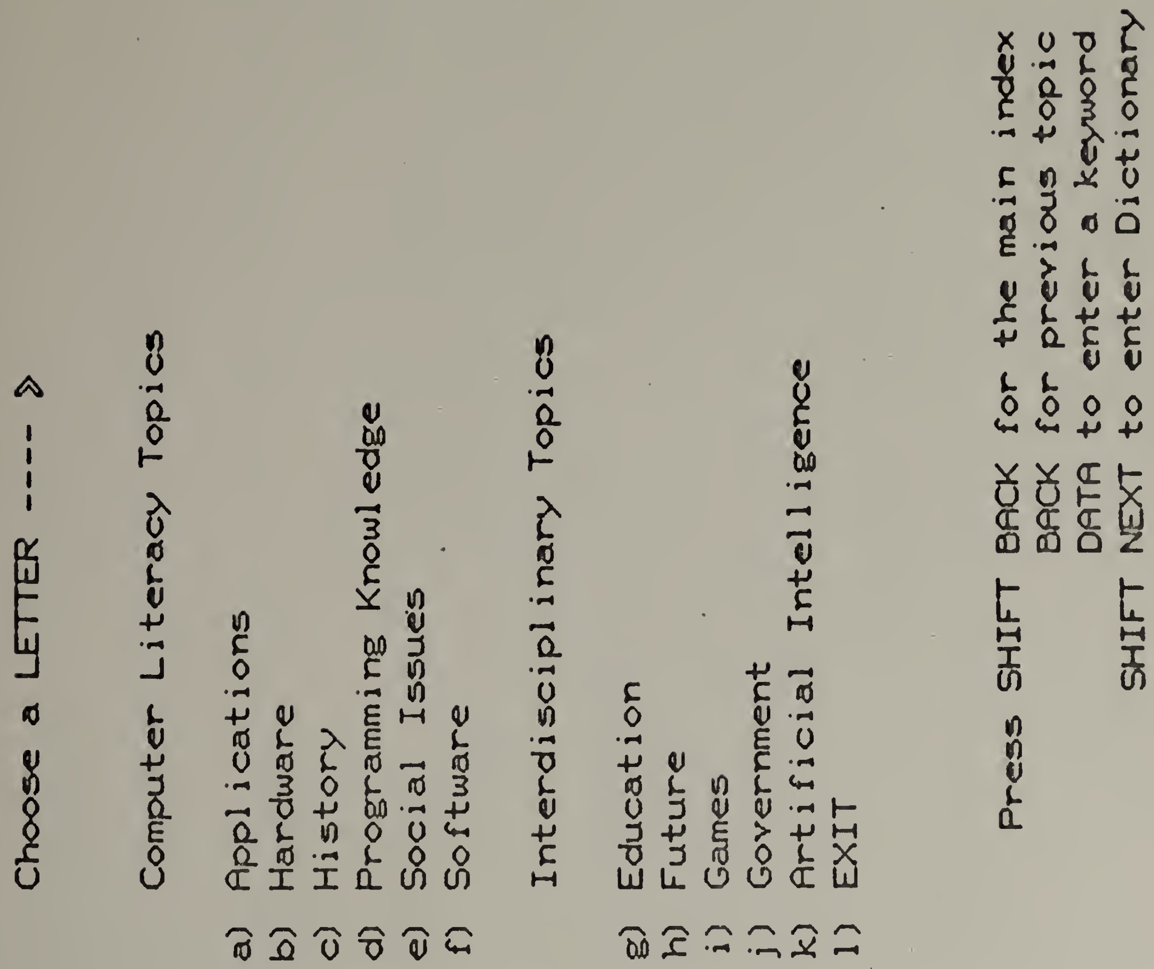
...Then press NEXT to proceed to that topic
HELP to cure your confusion

the Keyword Index lesson where he types in the topic he wishes to see (various misspellings are acceptable), presses NEXT and is taken directly to...



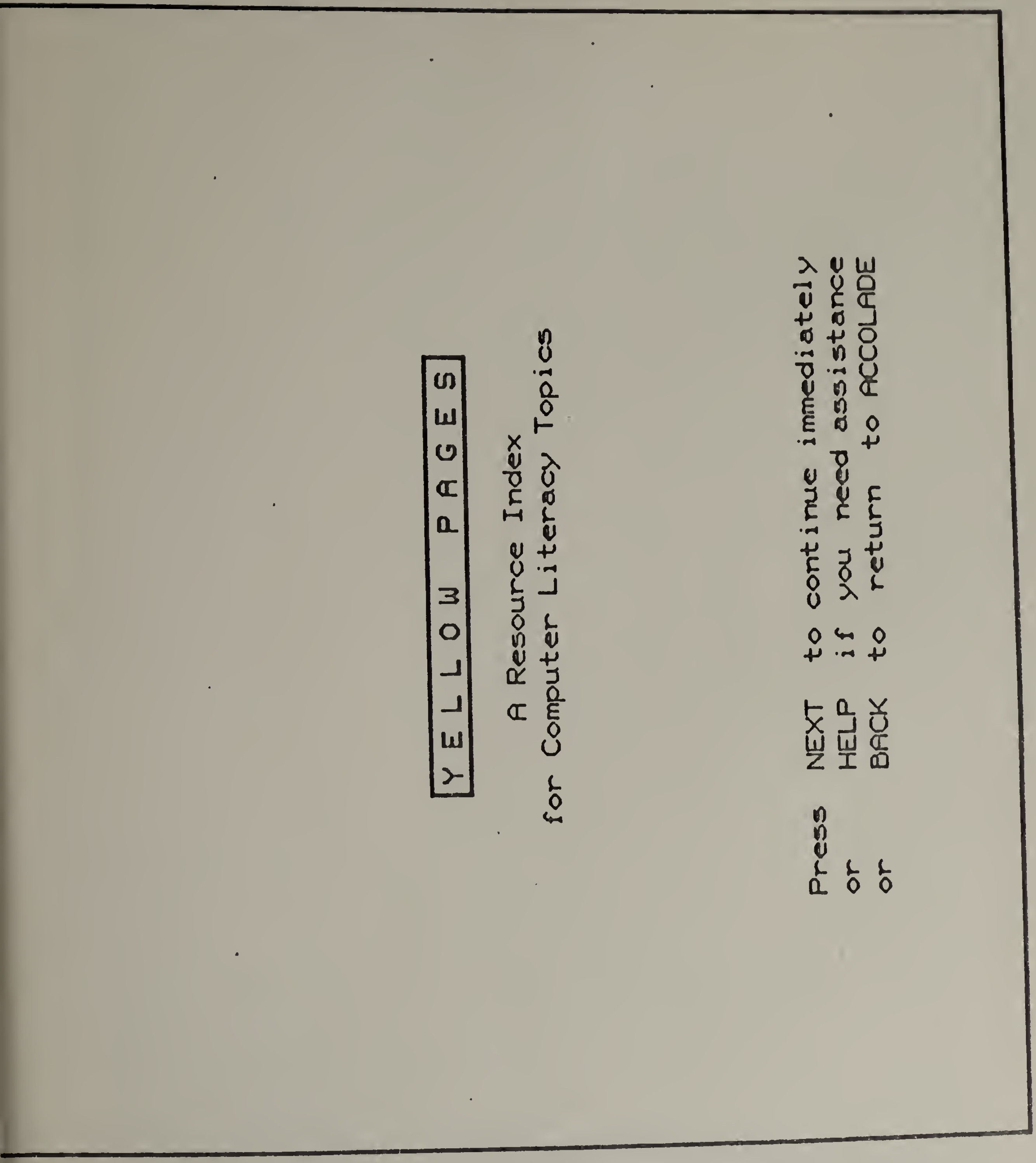
```
Resources for:
business-systems analysis

Resource   Qualifier
1-4        pp 131-151;158-165
1-9        pp 3-22;171-189
2-8
4-1
4-5

 Press SHIFT HELP to see the resource codes 

Press  LAB  to see or enter resource ratings

Press SHIFT BACK for the main index
            BACK for previous topic
            DATA to enter a keyword
      SHIFT NEXT to enter Dictionary
```

the topic 'systems analysis' in the Yellow Pages (where he was before he used Keyword Index). After reviewing the resources once again, he decides to return to the Router so he presses SHIFT BACK to return to...



```
Choose a LETTER ---- >>

     Computer Literacy Topics

a) Applications
b) Hardware
c) History
d) Programming Knowledge
e) Social Issues
f) Software

     Interdisciplinary Topics

g) Education
h) Future
i) Games
j) Government
k) Artificial Intelligence
l) EXIT

     Press SHIFT BACK for the main index
                 BACK for previous topic
                 DATA to enter a keyword
           SHIFT NEXT to enter Dictionary
```

the main choice page. He presses BACK again to return to...

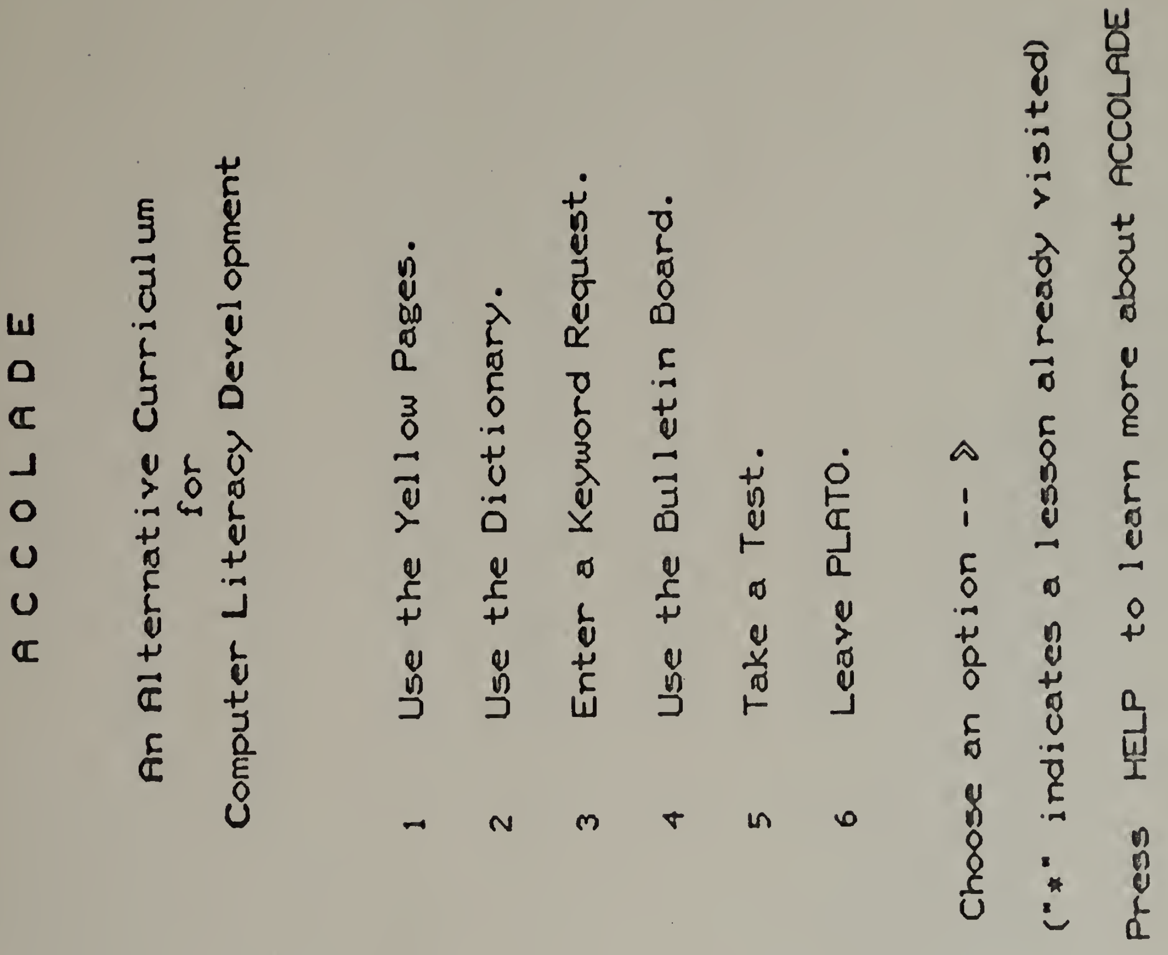


YELLOW PAGES

A Resource Index
for Computer Literacy Topics

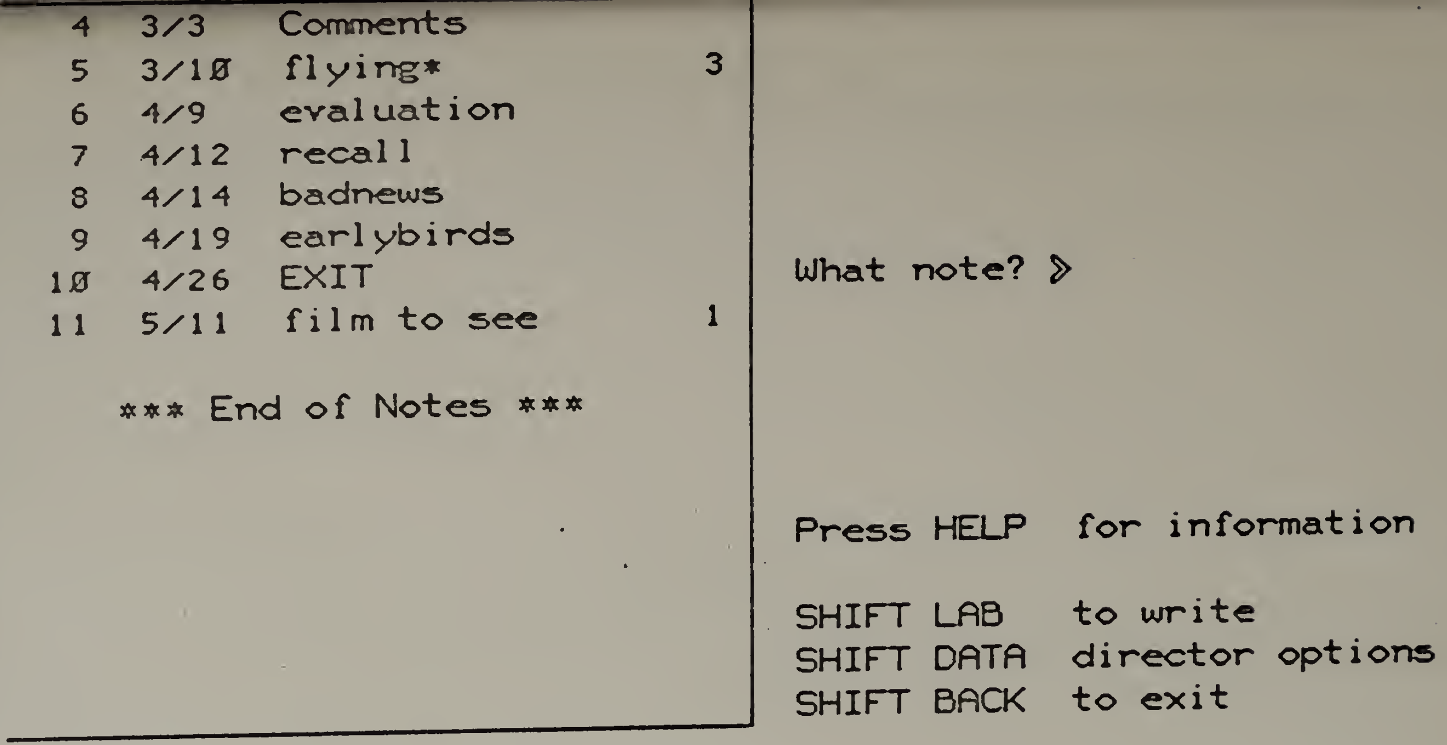
Press NEXT to continue immediately
or HELP if you need assistance
or BACK to return to ACCOLADE

the front page of the Yellow Pages and BACK once more to return to...



```
A C C O L A D E

An Alternative Curriculum
for
Computer Literacy Development

1   Use the Yellow Pages.

2   Use the Dictionary.

3   Enter a Keyword Request.

4   Use the Bulletin Board.

5   Take a Test.

6   Leave PLATO.

Choose an option -- »

("*" indicates a lesson already visited)

Press  HELP  to learn more about ACCOLADE
```

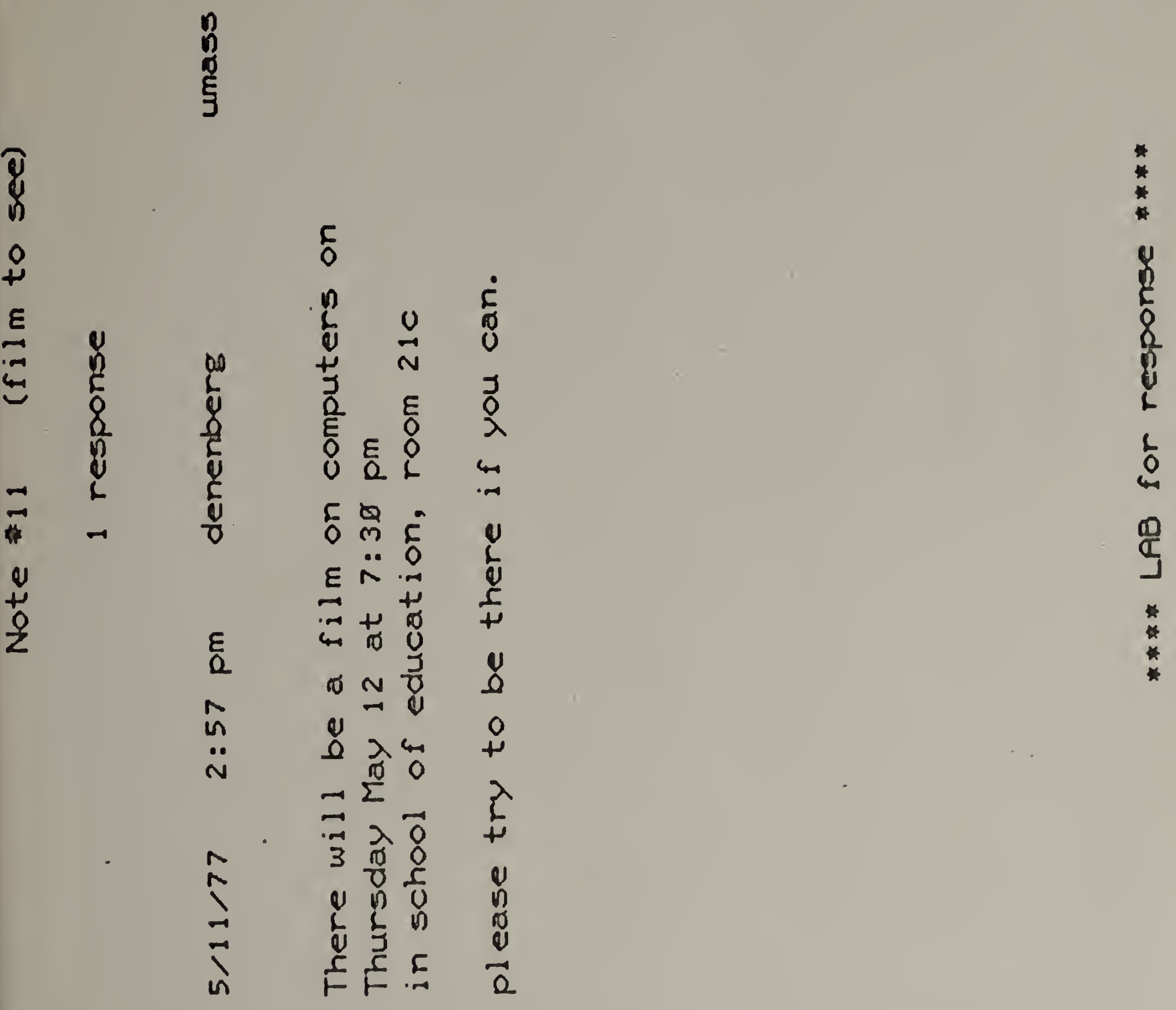
the Router lesson where he decides to look at the Bulletin Board, so, he presses '4' and is taken to...



```
 4   3/3    Comments
 5   3/1Ø   flying*                  3
 6   4/9    evaluation
 7   4/12   recall
 8   4/14   badnews
 9   4/19   earlybirds
1Ø   4/26   EXIT
11   5/11   film to see              1

      *** End of Notes ***
```

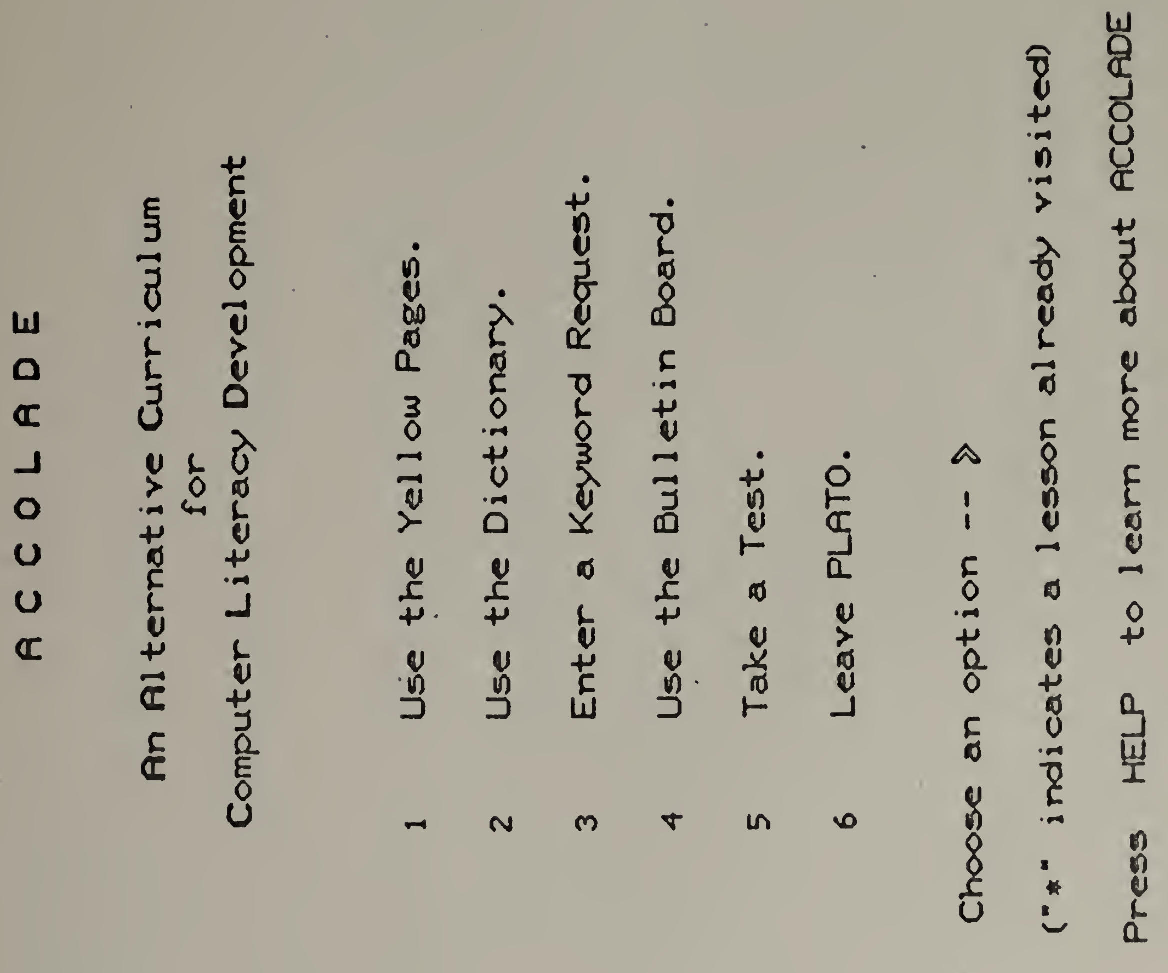
```
What note? »



Press HELP   for information

SHIFT LAB    to write
SHIFT DATA   director options
SHIFT BACK   to exit
```

the index page of the Bulletin Board. He has not yet read note 11 so he types'11' and NEXT and sees...



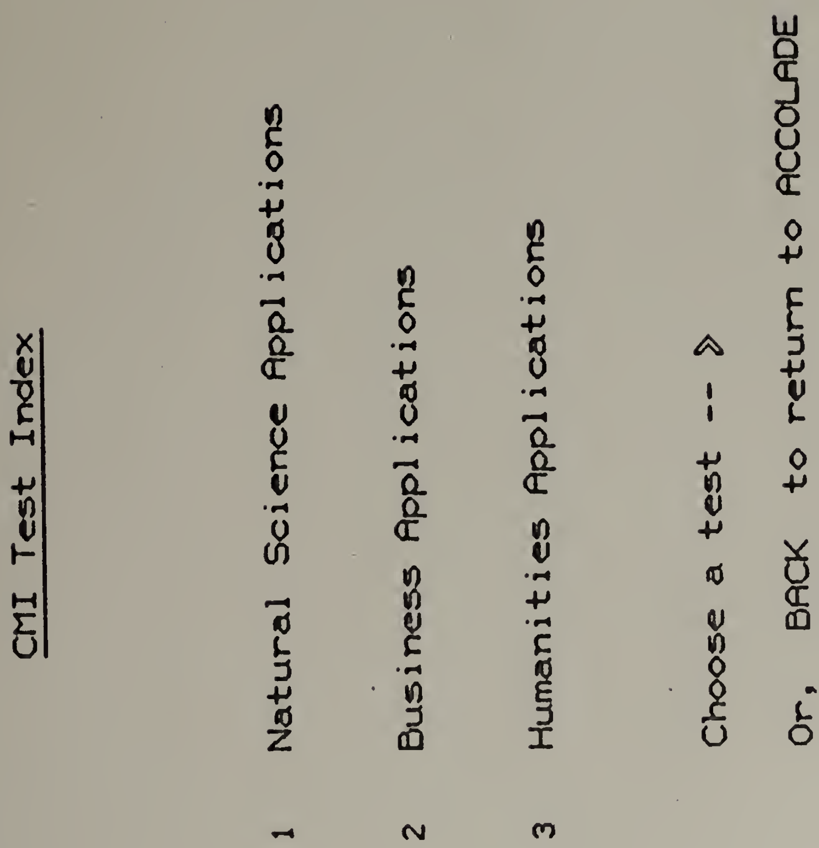
```
Note #11     (film to see)

           1 response

5/11/77    2:57 pm    denenberg         umass

There will be a film on computers on
Thursday May 12 at 7:3Ø pm
in school of education, room 21c

please try to be there if you can.

                         **** LAB for response ****
```

the contents of note 11. Pressing SHIFT BACK the Learner moves back to...



```
                ACCOLADE

      An Alternative Curriculum
                 for
 Computer Literacy Development

   1   Use the Yellow Pages.
   2   Use the Dictionary.
   3   Enter a Keyword Request.
   4   Use the Bulletin Board.
   5   Take a Test.
   6   Leave PLATO.

Choose an option -- »
("*" indicates a lesson already visited)
Press  HELP  to learn more about ACCOLADE
```

the Router lesson. The Learner wishes to take the test on Humanities Applications so he types '5' and is taken to...



CMI Test Index

1 Natural Science Applications

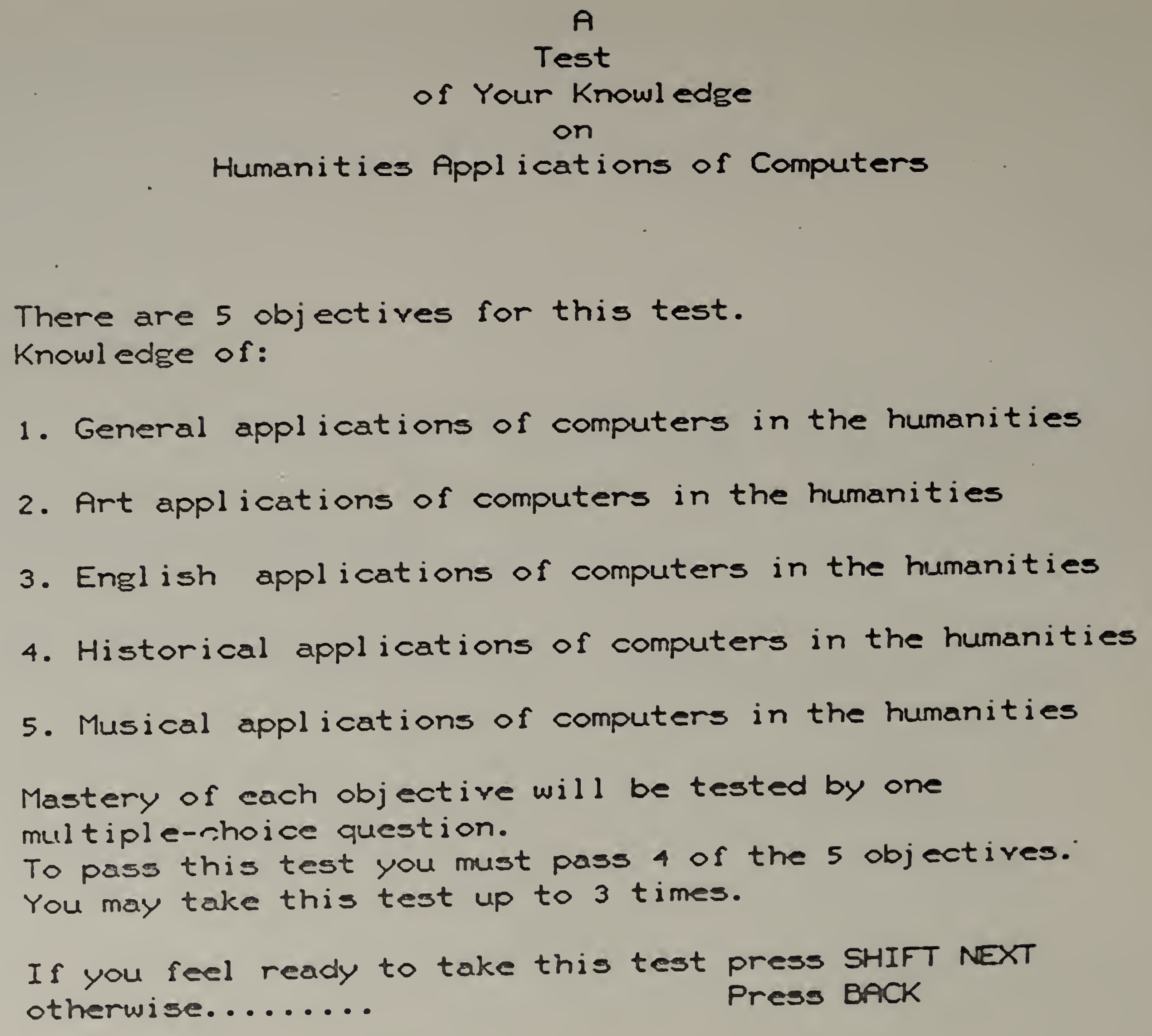
2 Business Applications

3 Humanities Applications

Choose a test -- ≫

Or, BACK to return to ACCOLADE

the CMI test index where he types a '3' and is taken to...



A
Test
of Your Knowledge
on
Humanities Applications of Computers

There are 5 objectives for this test.
Knowledge of:

1. General applications of computers in the humanities

2. Art applications of computers in the humanities

3. English applications of computers in the humanities

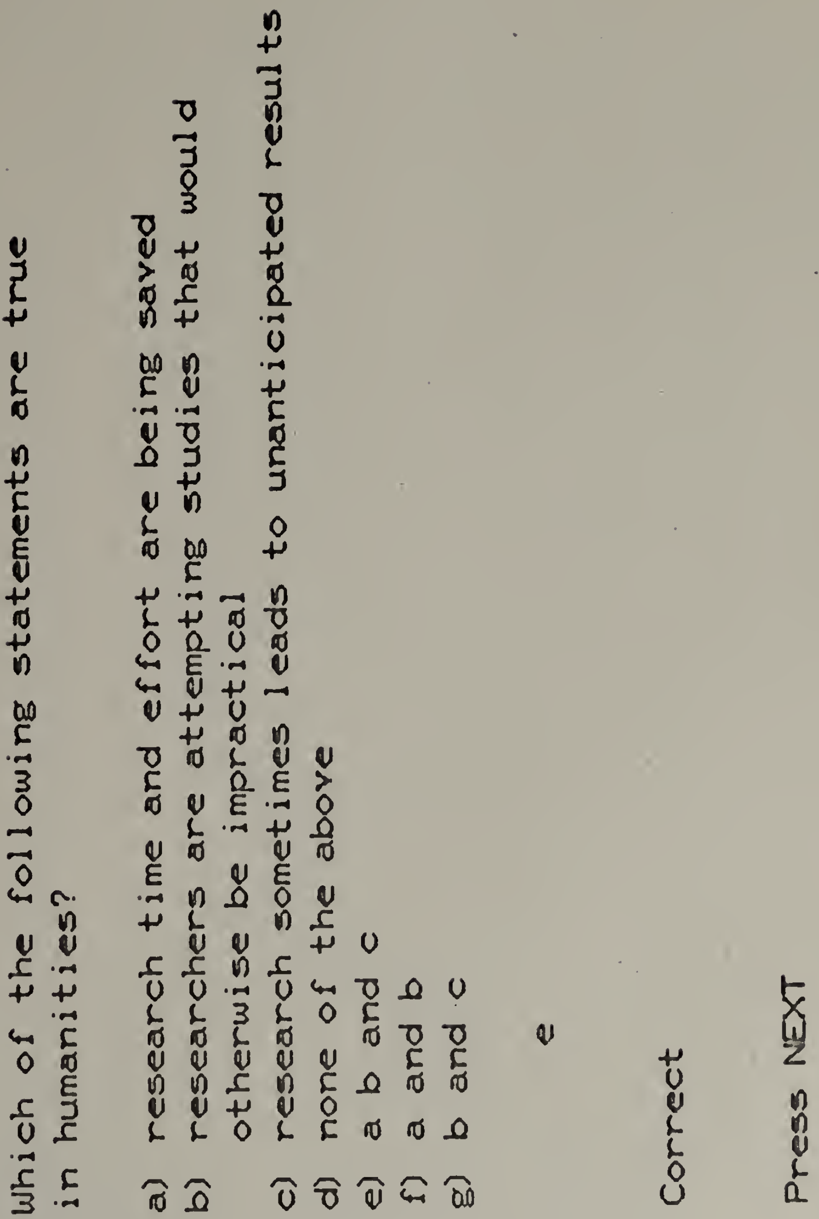
4. Historical applications of computers in the humanities

5. Musical applications of computers in the humanities

Mastery of each objective will be tested by one
multiple-choice question.
To pass this test you must pass 4 of the 5 objectives.
You may take this test up to 3 times.

If you feel ready to take this test press SHIFT NEXT
otherwise......... Press BACK

the front page of the Humanities Applications test. He presses SHIFT NEXT to initiate the test...



Which of the following statements are true
in humanities?

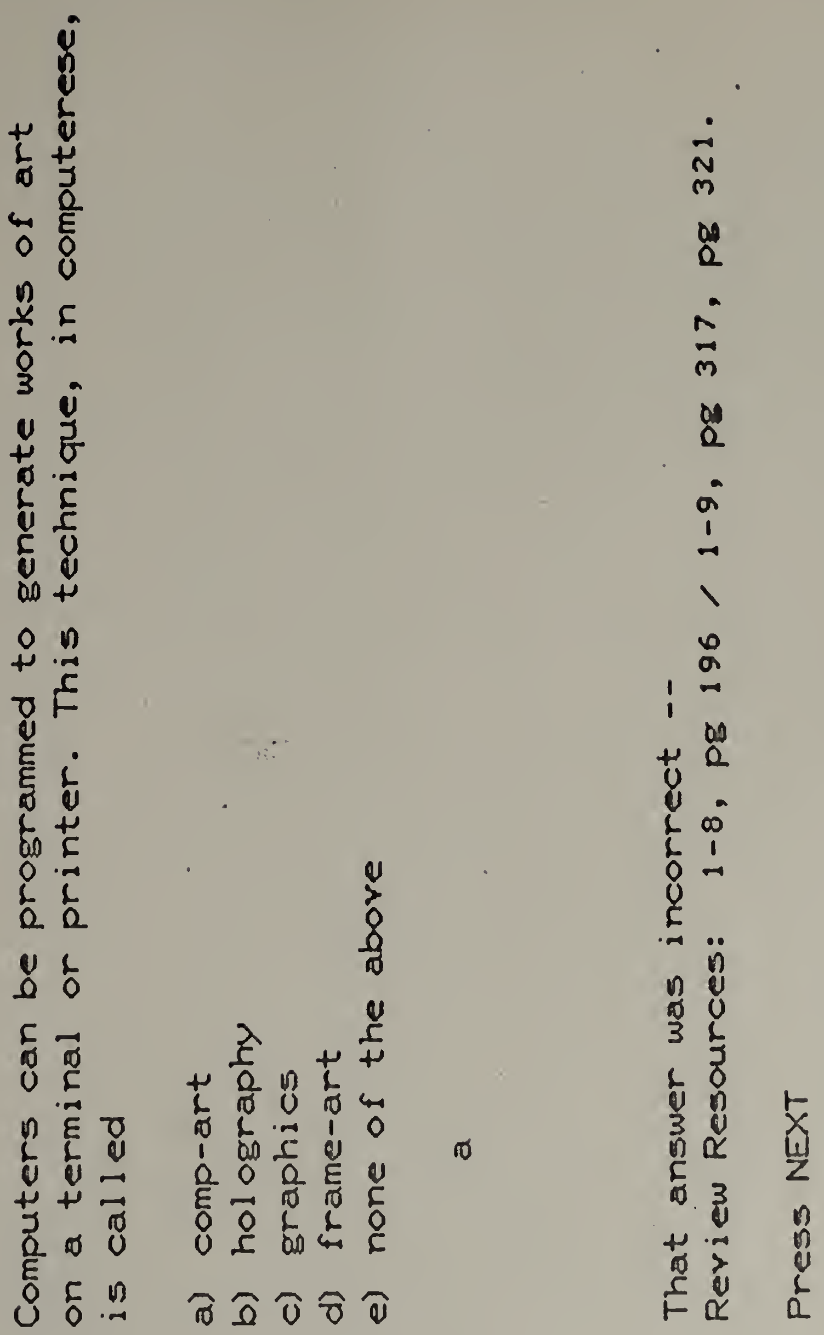
a) research time and effort are being saved
b) researchers are attempting studies that would
   otherwise be impractical
c) research sometimes leads to unanticipated results
d) none of the above
e) a b and c
f) a and b
g) b and c

e

Correct

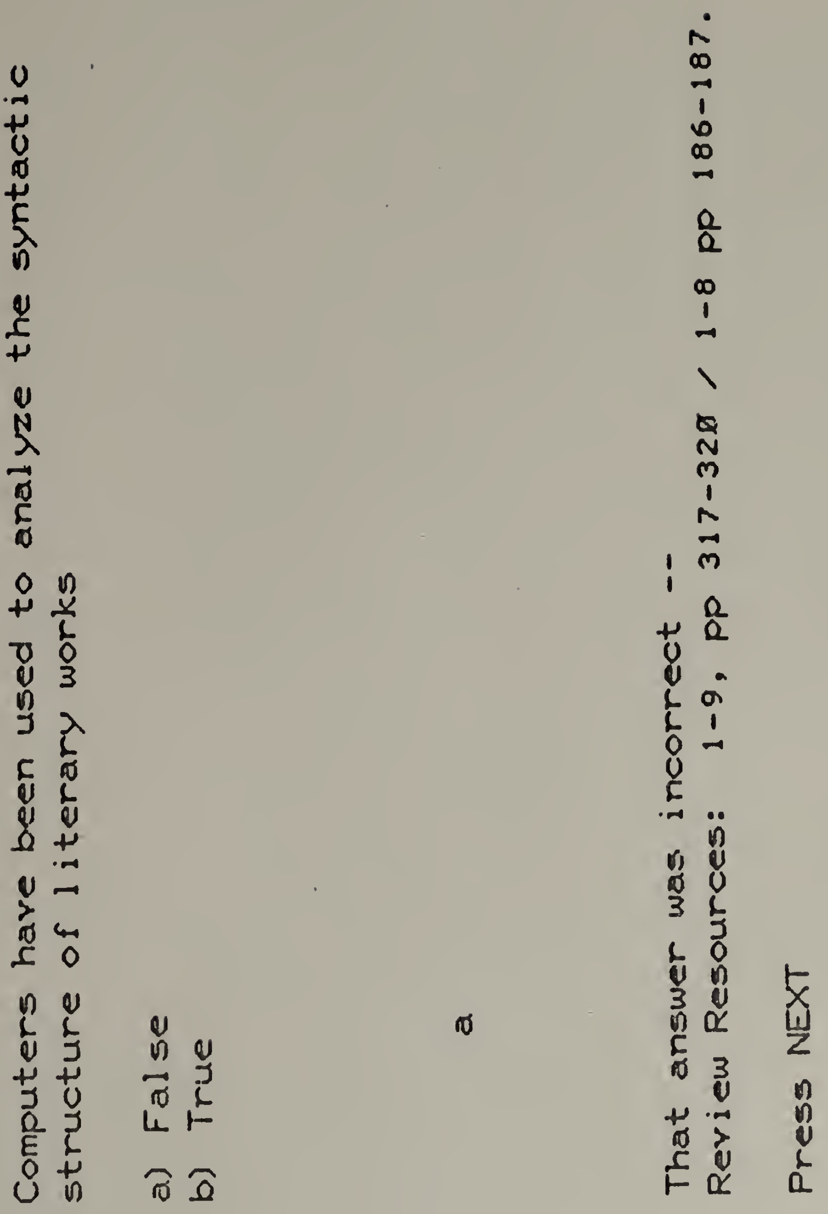
Press NEXT

and answers the first question (which measures the first objective) correctly, then presses NEXT...



```
Computers can be programmed to generate works of art
on a terminal or printer. This technique, in computerese,
is called

a) comp-art
b) holography
c) graphics
d) frame-art
e) none of the above

     a

That answer was incorrect --
Review Resources:  1-8, pg 196 / 1-9, pg 317, pg 321.

Press NEXT
```

and answers the second question incorrectly and receives the incorrect diagnosis and a prescription in terms of the resources stored in the Yellow Pages. He presses NEXT to go on to the next question...



Computers have been used to analyze the syntactic
structure of literary works

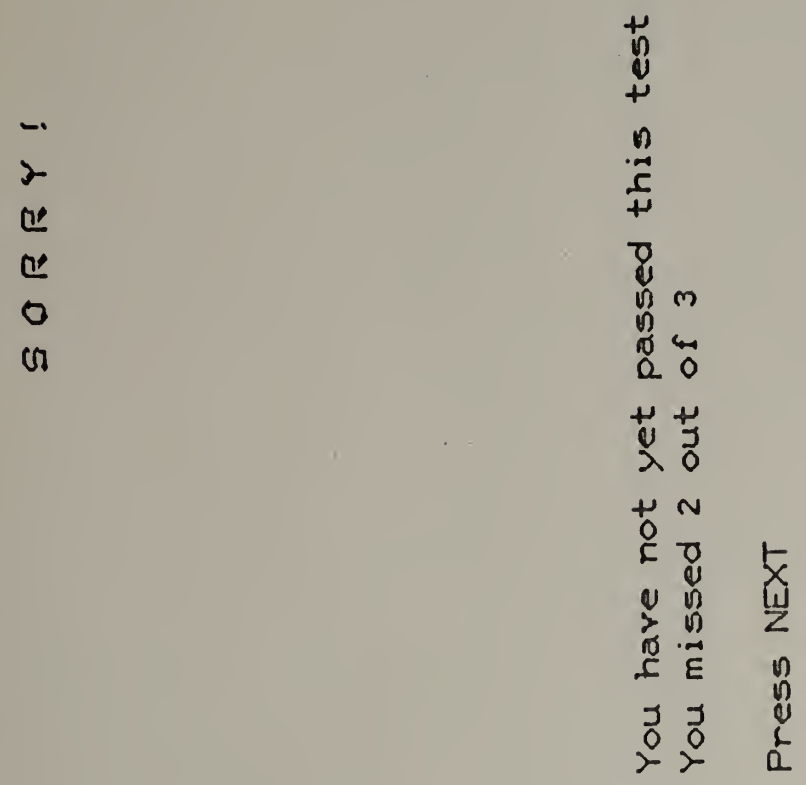
a) False
b) True

a

That answer was incorrect --
Review Resources: 1-9, pp 317-32Ø / 1-8 pp 186-187.

Press NEXT

and unfortunately answers this question incorrectly also. After receiving the prescription and pressing NEXT...

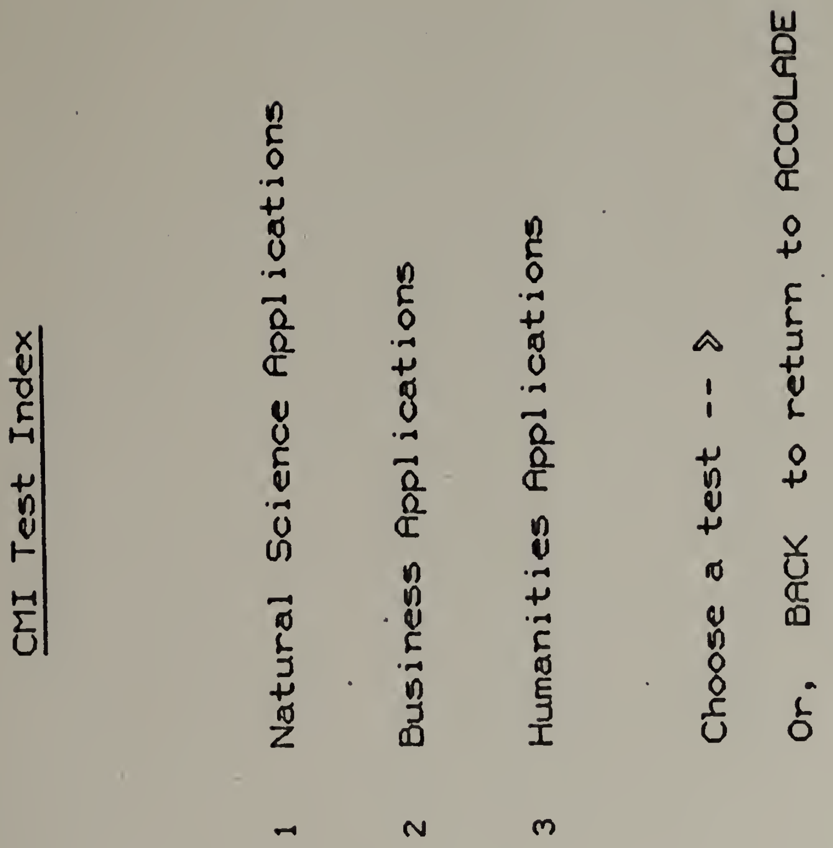


S O R R Y !

You have not yet passed this test
You missed 2 out of 3

Press NEXT

the Learner will be shunted out of the test with an explanatory message. He sighs, presses NEXT and is taken back to...



CMI Test Index

1 Natural Science Applications

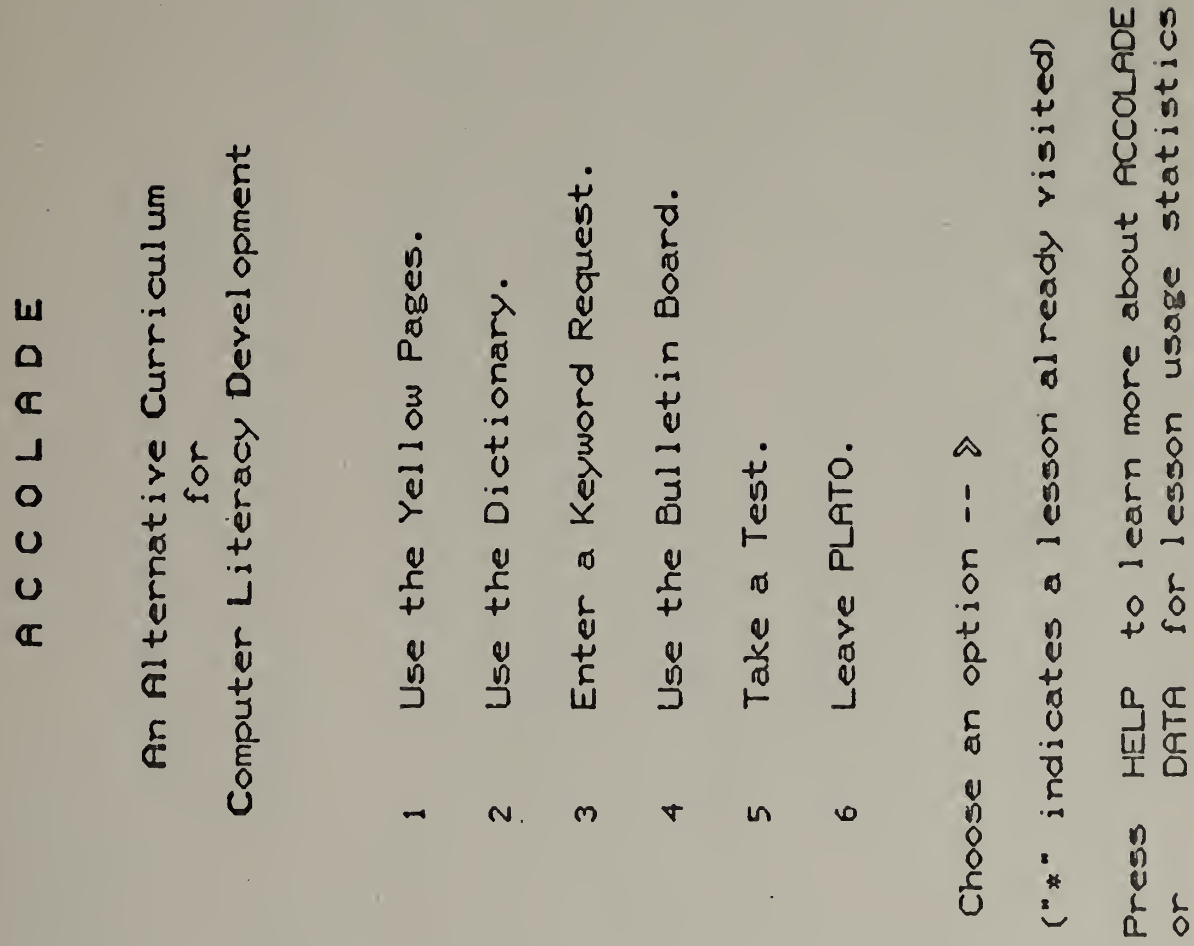
2 Business Applications

3 Humanities Applications

Choose a test -- »

Or, BACK to return to ACCOLADE

the CMI test index where he can, if he wishes, elect to re-take the same test (if he has not yet taken it three times) or to take another test, or to return to the Router. He decides not to take any more tests and presses BACK to return to...



```
                A C C O L A D E

        An Alternative Curriculum
                   for
     Computer Literacy Development


   1    Use the Yellow Pages.

   2    Use the Dictionary.

   3    Enter a Keyword Request.

   4    Use the Bulletin Board.

   5    Take a Test.

   6    Leave PLATO.


  Choose an option -- »

  ("*" indicates a lesson already visited)

  Press  HELP  to learn more about ACCOLADE
  or     DATA  for lesson usage statistics
```

the Router lesson where he decides to call it a day, presses '6' and is automatically signed off the Plato system.

6.5 Observations, Results, and Interpretations

The pilot study was recorded and evaluated using two instruments. All of the Learner interactions with the Computer and the Coordinator were observed by the author and recorded in a log book. Also, all Learners filled in a questionnaire which partially evaluated ACCOLADE, partially measured Learner attitude toward self-directed education, and allowed for comments and suggestions on the advantages and disadvantages of the system. Although the purpose of the implementation was not to formally evaluate ACCOLADE, the results gathered from the log book and the questionnaire served to provide valuable user feedback which can and in fact, has been used to improve the system.

A computer literacy knowledge test[10] and a computer literacy attitude questionnaire[11] were administered to the Learners at the beginning and end of the course. The knowledge test was based on different course content and the attitude test was based on a different course where the content was intended to shape the students attitudes. On closer examination of the goals of ACCOLADE it was decided that these tests and their results were not relevant to the pilot study and so will not be described here--however the issue of evaluation of ACCOLADE will be discussed in Chapter VII.

[10] Morsund, D., "What is Computer Literacy?", Creative Computing, Nov.-Dec., 1976, p. 55.

[11] Lucas, W., "Planned Attitude Change While Teaching Computer Literacy," Joint Bulletin (SIGCUE and SIGCSE) of ACM Conf. on Computer Sci. and Education, Feb., 1976, p. 93.

6.5.1 Log Book

Although there is some overlap, the observations recorded in the log book fall into three categories:

1) Those dealing primarily with Learners.

2) Those dealing primarily with the Computer.

3) Those dealing primarily with the Plato System.

Learners

- A common initial Learner reaction to Plato was a delighted laugh--hah! hm!?!--of surprised wonder. Perhaps a unique measure of effectiveness of the system would be to count the number of Learner laughs per hour.

- When two Learners shared a terminal, the one with more previous computer experience started to dominate. Previous computer experience seemed to help ease the Learner's first encounter with a Plato terminal.

- Without exception, all Learners enjoyed Plato games and simulations---in particular "airfight," "west," "biorhythms," "rose" and animated notes. One Learner started doing his homework on the x-y plotter in "rose." Another Learner become so engrossed in the biorhythm lesson that he purchased a book on the subject and performed his own experiment on the validity of the predictions by gathering data from friends and testing to see if there was any correlation. One person, not in the course, came in and requested some computer terminal time saying that he had become addicted to "airfight" during a Plato demonstration! Three of the Learners brought in friends to see the system.

- Most Learners were highly concerned with evaluation; they continually asked how they were to be "tested" since the tests were not programmed until the middle of the course. Before they knew the topics they were to be tested on, Learners were quite free in using the Map to search out knowledge of interest. After the test topics were identified, their search path was heavily biased towards those topics in order to "prepare for the tests." At this point, most Learners abandoned use of the Dictionary (since it could not directly help them prepare for the tests) and they concentrated on the Yellow Pages and the Keyword Index.

- It was extremely difficult for the author not to offer unsolicited advice to the Learners. Learners would tend to wander into Plato lessons beyond their capabilities, become stuck or confused with regard to Plato terminal function keys, and invent inefficient search procedures for exploring the knowledge space. The author felt a peculiarly heavy obligation to streamline the Learners' search strategies for identifying resources so that they could prepare for tests (since he had created the tests, he felt obligated to prepare them to pass them) but he was usually able to wait until the Learner asked him for help or advice.

- As a result of a discussion between the author and a Learner on search strategies, the Learner came to a realization that in the past she had been a depth-first Learner while this semester she had switched to being breadth-first and that felt much more natural and enjoyable.

- Several Learners expressed an interest in learning how to program (control) Plato. The author furnished them with some introductory Plato manuals and showed them how to use Plato lesson "aids." Only one of them actually followed through and wrote a small CAI Tutor program for drill and practice on calculating volumes of cylinders.

- No Learners showed up for the film that was shown. This was primarily due to the author not advertising it soon enough on the Bulletin Board.

- One Learner complained about the lack of Teacher-supplied "course objectives" although he had previously indicated that he preferred to "take responsibility for his own education." This suggests that a self-directed attitude and its converse may be compartimentalized in some Learners.

- Many of the Learners appreciated having the book resources available in the same room as the Plato terminal. One went so far as to indicate that she preferred having them there rather than at home--having to come in to read "makes it a priority and imposes self-discipline."

- The last week of classes was especially hectic as most of the Learners waited until then to do the reading and take the tests; this was aggravated by the Plato system going down for one day during this time. It is probably a natural inclination for people to put off unpleasant activities (tests) until the last moment; however it is clear that a self-directed course cannot compete effectively in the same environment with traditional courses that assign letter grades and give tests at times specified by the teacher. Learners are pragmatically forced to assign lower priorities to courses that allow them more flexibility and, in fact, the ACCOLADE course as a program of individualized instruction did run almost one month past the formal closing of classes allowing three Learners to complete the curriculum and receive credit for their endeavors.

Computer

Some observations and interpretations regarding the Computer component of ACCOLADE have already been made in previous sections of this chapter. Here we will extend those results based on the observations recorded in the log book.

- One Learner noted that a small disadvantage to the structure of the Yellow Pages was that he could only see one node at a time--he could not see above or below that node simultaneously and that was "disconcerting at first."

- Many Learners found the coded resources in the Yellow Pages to be initially confusing. However, the Learners who read the instructions slowly and carefully did not experience this confusion. Most Learners expected to see the resources directly on the terminal and not just be supplied with pointers or references to them. This appears to be an initial problem with many computerized information systems where references rather than the actual content is provided to the user.

- The rating function provided by the Yellow Pages was not flexible enough; only one number measuring the overall effectiveness of the resource could be provided or viewed. Additional rating parameters such as depth-of-presentation and difficulty-of-understanding would enhance the usefulness of the ratings mechanism.

- The Keyword Index was used very effectively by one Learner to search out resources for test topics after reading the objectives on the front of the CMI tests.

- Overall, not much time was spent in the Dictionary by the Learners. Possible reasons include: (1) The Dictionary was not completely programmed when most Learners started the course so the Learners were advised to not use it at first and this attitude stuck. (2) Since they could not use the Dictionary initially, the Learners concentrated on the Yellow Pages and Keyword Index parts of the Map and the more they used it, the easier it became to use and the more useful it became thus decreasing the need for the Dictionary. (3) There was no actual need to use the Dictionary to prepare for the tests. (4) One of the Learners said that "dictionary is a boring word"; perhaps the name should not be associated with an educational tool that has many unpleasant memories for many Learners. (5) A fully developed plato help sequence was not programmed for the Dictionary until the course was over; however help was always available personally from the author.

- Two of the Learners were foreign students and had difficulty understanding the meanings of some of the words on the CMI system tests.

- Although the Learners appreciated the immediate responses from the tests, the tension would mount perceptibly as the Learners typed in the answer and waited for the split-second response. All Learners breathed heavy sighs of relief whenever the Computer responded with a "correct" diagnosis.

- The author asked the Learners, insofar as possible, to think out loud when attempting to answer the CMI test questions. This not only allowed the author some insight into the Learner's thinking process, it suggested several changes to the phrasing of test questions. Many times the result of missing a question came from either too much knowledge or overthinking a rather straightforward multiple choice question. However, all

Learners passed all the tests within three tries.

- In addition to modifying the test questions as the course progressed, the observation of the Learner-Computer interaction allowed the author to improve the software engineering of various components (e.g. the Learner must press SHIFT-NEXT rather than NEXT to take a test so that he cannot accidently get into a test) and the definitions in the Dictionary (topology was not explained adequately to one Learner so it was improved right then and there). Since the system was constantly being debugged and improved while it was being pilot tested, no effort was made to properly evaluate the system; rather that is suggested in the next chapter under "Suggestions for Future Research and Development."

Plato

All of the observations here regarding the Plato system are in the nature of problems experienced by the Learners--problems that could be corrected by appropriate hardware and software.

- The Plato system would benefit from a term-HELP or a HELP key default that would, at a minimum, advise the Learners how to get out of a lesson.

- New Learners experience much confusion with two aspects of group notes files such as the Bulletin Board: (1) The distinction between how to respond to a previously written note and writing a new note. (2) The use of the BACK and SHIFT-BACK keys to move from notes to the index and out of the file.

- When browsing through the library of Plato lessons most Learners would type the lesson name and then press NEXT (rather than DATA to execute the lesson) and be confronted with a page that invites them to "inspect" the lesson by pressing the LAB key. Thinking this is what they want to do (inspect the lesson) they press LAB and begin to inspect the Tutor program--not at all what they had intended!

- During the middle of the course many problems were experienced due to limited ECS (computer memory). Learners were being bumped out of lessons and, in many cases, could not even sign onto the system as a student.

- One of the foreign students had consistent problems reading the Plato Keyboard; he confused the letter 'i' for the numeral '1' and the letter 'q' for the letter 'a.' This almost cost him the passing of one of the multiple choice tests: the correct answer was 'a' and he pressed 'q'--fortunately 'q' was not a possible answer and was ignored by the test program.

6.5.2 Learner Evaluation

After each Learner had completed the course, he was asked to fill in a questionnaire that provides a measure of the 'success' of the

ACCOLADE pilot study. Although the primary interest of the pilot study was to debug and improve the system, it is certainly of interest to receive feedback from the Learners concerning its relevance, effectiveness, convenience and enjoyment.

The questionnaire, is shown in Figure 12. Since only nine Learners filled in the questionnaire, all responses are shown in ranked order and followed by the mean; a dash indicates no response. The 22 questions generally fall into six categories: those that measure the effectiveness of the Computer components of ACCOLADE (5 through 10), those that compare ACCOLADE with a traditional computer literacy course (1 and 3, 2 and 4, 10 and 11, 18 and 19), those that rate the programming skills and behaviors (12 and 22), those that rate the design goals of ACCOLADE (13, 16, 17, 18, and 22), those that measure Learner attitude toward self-directed education (20 and 21) and those that rate the overall effectiveness of ACCOLADE (1, 2, 14, 15, and 17). The results from these categories will be interpreted followed by a brief discussion of the more common comments found on the last page of the questionnaire.

1) Effectiveness of Computer Components

Questions 5 through 10 ask the Learner to rate the suitability of the Yellow Pages, Keyword Index, Dictionary, the Map as a whole, Bulletin Board and CMI tests. Clearly the Yellow Pages is viewed as the most suitable mechanism for searching out information although the Map as a whole and the Keyword Index run very close seconds. The Dictionary is rated considerably below the Yellow Pages and the Keyword Index, perhaps for the reasons set forth in section 6.5.1 under Computer. Although me: ______________________________

udent No: ____________________ Age: ________ Sex: ______

jor: ______________________________ GPA: ____________

terests:

evious Programming Experience

No. of years: ______

Languages: ______________________________

Hardware: ______________

Applications: ___________

0 - terrible
5 - neutral or average
10 - superb

te the following items on a scale from 0 to 10		Mean
Rate the effectiveness of ACCOLADE	7, 7, 7, 8, 8, 8, 9, 10, 10	8.2
Rate your enjoyment of ACCOLADE	1, 7, 7, 8, 8, 8, 10, 10, 10	7.7
Rate the effectiveness of traditional introductory computer courses	4, 6, 8, 8, 8; 9, 9 10 –	7.8
Rate your enjoyment of traditional introductory computer courses	1, 4, 5, 6, 6, 6, 7, 10 –	5.6
Rate the Yellow pages as a suitable mechanism for searching out information	8, 8, 8, 9, 9, 9 10, 10, 10	9.0
Rate the Keywords as a suitable mechanism for searching out information	7, 7, 8, 8, 9, 10, 10, 10, 10	8.5
Rate the Dictionary as a suitable mechanism for searching out information	5, 5, 5, 7, 7, 8, 9, 10 –	7.0
Rate the MAP (combination of the Yellow Pages, Keywords, and Dictionary) as a suitable mechanism for searching out information	7, 8, 8, 9, 9 10, 10, 10 –	8.[illegible]
Rate the usefulness of the Bulletin Board part of ACCOLADE	5, 8, 8, 8, 9, 9 9 10 –	8.3
Rate the computer testing part of ACCOLADE in terms of		
a) validty	5, 5, 6, 7, 7, 8, 10, 10, 10	7.6
b) convenience	8, 9, 9, 10, 10, 10, 10, 10, 10	9.6
c) enjoyment	1, 3, 4, 7, 9, 9, 9, 10, 10	6.9
Rate traditional course testing of knowledge in terms of		
a) validity	4, 5, 5, 7, 8, 8, 10, 10 –	7.1
b) convenience	1, 1, 3, 4, 5, 5, 6, 7 –	4.0
c) enjoyment	0, [illegible] 2, 2 [illegible] –	[illegible]

Figure 12. Questionnaire

Learner Evaluation of ACCOLADE

Name:

Student No: Age: Sex:

Major: GPA:

Interests:

Previous Programming Experience

No. of years:

Languages:

Hardware:

Applications:

Rate the following items on a scale from 0 to 10	0 - terrible 5 - neutral or average 10 - superb	Mean
1) Rate the effectiveness of ACCOLADE	7,7,7,8,8,8,9,10,10	8.2
2) Rate your enjoyment of ACCOLADE	1,7,7,8,8,8,10,10,10	7.7
3) Rate the effectiveness of traditional introductory computer courses	4,6,8,8,8,9,9,10-	7.8
4) Rate your enjoyment of traditional introductory computer courses	1,4,5,6,6,6,7,10-	5.6
5) Rate the Yellow pages as a suitable mechanism for searching out information	8,8,8,9,9,9,10,10,10	9.0
6) Rate the Keywords as a suitable mechanism for searching out information	7,7,8,8,9,10,10,10,10	8.8
7) Rate the Dictionary as a suitable mechanism for searching out information	5,5,5,7,7,8,9,10-	7.0
8) Rate the MAP (combination of the Yellow Pages, Keywords, and Dictionary) as a suitable mechanism for searching out information	7,8,8,9,9,10,10,10-	8.9
9) Rate the usefulness of the Bulletin Board part of ACCOLADE	5,8,8,8,9,9,9,10-	8.3
10) Rate the computer testing part of ACCOLADE in terms of		
a) validty	5,5,6,7,7,8,10,10,10	7.6
b) convenience	8,9,9,10,10,10,10,10,10	9.6
c) enjoyment	1,3,4,7,9,9,9,10,10	6.9
11) Rate traditional course testing of knowledge in terms of		
a) validity	4,5,5,7,8,8,10,10-	7.1
b) convenience	1,1,3,4,5,5,6,7-	4.0
c) enjoyment	0,0,2,2,4,4,5,7-	3.0

Figure 12. Questionnaire

2.

		Mean
12) Rate the programming skills part of ACCOLADE (if you took that part) in terms of		
a) skills aquired	7,8-------	7.5
b) enjoyment	8,10--------	9.0
c) convenience	5,10-------	7.5
d) overall "grade"	7,9-------	8.0

Rate the extent to which you agree with the following statements (0-strongly disagree, 5-neutral, 10-strongly agree)

13) I now have a better idea of what computers can and cannot do	5,7,7,9,10, 10,10,10,10	8.7
14) ACCOLADE gave me the educational control I wanted	1,5,5,5,5,5,6,10,10	5.8
15) ACCOLADE satisfied my computer literacy needs	5,5,5,5,6,7,7,10,10	6.7
16) ACCOLADE encourages a human-computer partnership by acting as a model of that relationship	5,6,9,9,10, 10,10,10,10	8.8
17) ACCOLADE proved to be:		
a) relevant	5,6,7,8,8, 9,10,10,10	8.1
b) easy-to-use	8,10,10,10,10,10,10,10,10	9.8
c) enjoyable	5,7,10,10,10,10,10,10,10	9.1
d) effective	5,7,8,9,9,9,10,10-	8.4
18) ACCOLADE allowed me to take responsibility for effecting my own education	1,6,9,9,10,10,10,10,10	8.3
19) Traditional computer courses allow me to take responsibility for effecting my own education	0,6,7,7,7,9,9,10-	6.9
20) I want to take responsibility for effecting my own education	5,5,7,8,9,10,10,10-	8.0
21) Educationally, I prefer to be told what to do and when to do it	0,0.1,2,3,4,5,5,5	2.8
22) Of all your programming experience (including previous courses etc) rate how it has increased your:		
a) problem solving skills	5,5,6,6,7,8.9--	6.6
b) confidence	0,5,5,5,6,7,9,10-	5.9
c) autonomy	0,5,5,5,6,9,10--	5.7
d) logical thinking	5,5,6,6,7,8,8,10-	6.9
e) ability to learn	0,3,5,5,7,7,8,9-	5.5

Figure 12. Questionnaire (continued)

the Learners did not appear to actually enjoy the tests they rated their convenience extremely high (9.6), second only to the ease-of-use rating of ACCOLADE in general in question 17c (9.8).

Subject to the average rating of the Dictionary, the Computer components of ACCOLADE seemed to be quite acceptable to the Learners.

2) Comparison of ACCOLADE with Traditional Courses

Questions 1 and 3 do not indicate any substantial difference in the Learner's perception of the overall effectiveness of ACCOLADE over a traditional computer literacy course; however questions 2 and 4 indicate that the Learners enjoyed ACCOLADE more. In terms of testing, questions 10 and 11 show that the Learners considered the validity of the two types of tests about equal but were overwhelmingly on the side of ACCOLADE for convenience and enjoyment. Similarly, questions 18 and 19 show an edge towards ACCOLADE in allowing the Learner to take responsibility for effecting his own education.

ACCOLADE appears to compare quite favorably against the more traditional forms of computer literacy education, however this is to be expected since the sample of Learners is heavily biased towards Learners who were seeking alternatives to traditional classes; i.e. the reason they subscribed to ACCOLADE was primarily its alternative format.

3) Ratings of Programming Skills and Behaviors

Not much interpretation can be made on question 12 since only two Learners elected to take the programming part of ACCOLADE. It is clear however that they both enjoyed that part very much (9.0).

Since many of the Learners had taken programming courses previously they were asked to rate how it had increased the associated skills and behaviors in question 22. It seems that "problem solving skills" and "logical thinking" rate above the rest but that none of these are exceptional. This data would seem to refute the contention that computer programming can effectively teach the skills and behaviors of problem solving, logical thinking, learning, confidence and autonomy--at least in the opinion of the Learners themselves.

4) Achievement of Design Goals

The extent to which the Learners felt ACCOLADE fulfilled the goals outlined in the design section 5.2.3 are measured by questions 13, 16, 17, 18 and 22. With the exception of teaching the skills and behaviors of programming (question 22) achievement seems to be high--greater than 8 in all cases--and expecially high (9.1 and 9.8) in the 'enjoyable' and 'easy-to-use' categories. Question 17c is equivalent to question 2 and question 17d is equivalent to question 1 and these were included as a consistency check on the Learners' responses. While the effectiveness ratings (17d and 1) correlate quite closely, the enjoyment does not (17c and 2). This probably due to a '1' rating on question 2 (which was rated as a '5' on 17c by the same person!) which would considerably lower the mean value because of the small number of respondents; e.g. if the '1' were meant to be a '10,' the mean would become 8.7.

5) Learner Attitude toward Self-Directed Education

Questions 20 and 21 are phrased so that if the ratings are consistent they should be at opposite ends of the measurement scale. They did, in fact, appear consistent and indicate that, as a group, the Learners were inclined towards the self-directed mode of education.

6) Overall Effectiveness

The overall effectiveness was measured in general terms by questions 1, 2 and 17 and specific items were asked by questions 14 and 15. While the effectiveness as measured by questions 1, 2, and 17 proved to be adequate, the response to questions 14 and 15 seemed not only to be low but inconsistent. For example, the rating of 5.8 to question 14 is not compatible with the high responses to questions 18 and 20; that is, it appears inconsistent to hear that "ACCOLADE did not give me a very high degree of educational control" when one also hears "ACCOLADE allowed me a high degree of responsibility in effecting my own education" and "I want to take responsibility for effecting my own education." Similarly, the low rating of question 15 "ACCOLADE satisfied my computer literacy needs" appears to be inconsistent with the high rating of 17a "ACCOLADE proved to be relevant." Perhaps the wording of the questions 14 and 15 unduly influenced the responses. Any future questionnaire should be redesigned taking this problem into account.

7) Comments

The last page of the questionnaire asked the Learners to describe what they felt to be the advantages and disadvantages to the system as well as soliciting comments and suggestions for improvement.

a) Advantages

The advantages most mentioned centered on accessability ("Everything is right in front of you,") convenience in learning and testing ("...allows one to do as much as he wants when he wants.", ease-of-use ("fast...good control,") usefulness ("...no limit on how much can be made available to students through this system," "...it helps one find sources of information, even if he's not sure what information he needs.") and the fact that ACCOLADE provides the alternative of self-directed education ("It also leaves a person's educational decisions and goals to himself.")

b) Disadvantages

Only one disadvantage was cited more than once (twice) and this was a complaint that the resources for the computer literacy topics did not provide enough depth. The other disadvantages mentioned were: there was only one Plato terminal available at restricted times, the Plato system would go off when in use, the system is hard to grasp in the initial stages, and the system could weaken the teacher-student relationship. Mentioned as a disadvantage but obviously an accolade was the statement, "You can't have one for your own."

c) Suggestions for Improvement

Most of the suggestions for improving the system focused on improving the quality of the tests. More flexibility was suggested so that a Learner could skip questions and then go back to answer them. Also recommended were more comprehensive tests, less specifically tied to the book resources. Other suggestions included: adding more depth and breadth to the topics and resources, increasing the capabilities of the Keyword Index, providing more copies of the book resources, supplying more information directly through the computer (via CAI lessons) rather than by pointers to resources, and implementing the system on a larger scale.

d) Comments

Five of the nine learners responded with notes of praise for the system and the Teachers. They appreciated the highly individualized attention they received and two of the Learners expressed an interest in continuing their acquisition of computer literacy.

6.6 Conclusions

As a result of the implementation of ACCOLADE, the log book data, and the questionnaire results, several conclusions can be drawn; recommendations for future research and development will be discussed in the next chapter. In terms of the goals of ACCOLADE, the implementation appeared to be reasonably successful:

- It allowed the Learner the educational alternative of taking responsibility for effecting his own education.

- It was moderately responsive to Learner and Teacher needs and requirements.

- It demonstrated an alternative and reasonably effective teaching-learning system.

- It encouraged the human-computer partnership by acting as a model of that partnership.

- It identified and provided resources for the acquisition of computer literacy knowledge in the areas of applications (in depth), history, social issues, hardware and software while showing the structure of this knowledge space and helped shape in the Learners a more informed attitude about the capabilities of computers.

- It was not possible to measure if the skills and behaviors of logical thinking, problem solving, learning, confidence, and autonomy were fostered through the medium of computer programming.

- It was effective but costly--, relevant, very easy-to-use, highly enjoyable for both Teachers and Learners but very expensive if it had to be paid for by the Learners (a Plato terminal rents for about $800/month and the telephone charges were $1600/month due to the high price of a dedicated WATS line).

- It was moderately adaptive--the Keyword Index, Dictionary and content of the CMI tests were all modified during the course and as a direct result of the implementation.

The Plato version of the implementation of ACCOLADE also seemed to be propitious. The extremely fast, highly responsive, and quiet Plato terminal was an excellent communications medium between Learner and Computer. The special function keys on the keyboard of the Plato terminal made it very easy to write the computer programs that constituted the Computer part of ACCOLADE and at the same time afforded the Learners a powerful, easy-to-use, and highly controlled search tool. Perhaps furnishing Learners with cognitive tools such as the Map of ACCOLADE is one of the most important functions an educator can provide--it is in basic agreement with the thought from a poster hanging in many teachers' offices: "Give me a fish and I eat for a day, Teach me to fish and I can eat for a lifetime...".

It is also clear that the computerized knowledge part of an ACCOLADE system can, in theory, be extended to other domains of content knowledge. Any curriculum which can formulate a set of objectives, create criterion-referenced test questions which measure mastery of these objectives, and can identify the topics and relations between the topics can use the Computer implementation of ACCOLADE as a framework for delivering that knowledge. Whether or not the curriculum contains a set of skills and behaviors is content dependent and is a decision for the curriculum planner or designer.

ACCOLADE supports the contention that the function of the Teacher is to serve as a guide who gradually fades away as the Learners become self-guiding; the ultimate goal of the Teacher is to produce an autonomous Learner capable of educating himself. In the prose of Buddhist metaphor, the Teacher is like a ferryman with a boat to assist the Learner in crossing the river; however when the further shore is reached there is no need for the Learner to continue to carry the boat _and_ the ferryman on his shoulders as he continues on the path.

> "The function of education is to give the student abundant knowledge in the various fields of human endeavor and at the same time to free his mind from all tradition so that he is able to investigate, to find out, to discover."[12]

Finally, the implementation of ACCOLADE proved to be a challenging, exciting and highly rewarding learning experience for the author. The design of the system was an interesting academic exercise, but the

[12] Krishnamurti, J., _Think on These Things_, (New York: Perennial Library, 1970), p. 176.

implementation proved to be a valuable crucible where hypotheses and intuition were tested, synthesized, improved and discarded. There is much truth in the statement attributed to former president Harry Truman: "It's what you learn after you know--that really counts."

CHAPTER VII

SUGGESTIONS FOR FUTURE RESEARCH AND DEVELOPMENT

Three types of suggestions for future research and development of ACCOLADE are made: (1) short range suggestions which represent extensions to the present system and can probably be developed within one to three years, (2) long range suggestions which would require substantial modification and reconceptualization of the present system and could be developed over the next five to ten years and (3) suggestions for research on some of the educational issues raised by ACCOLADE.

7.1 Short Range Suggestions

These include features that were not developed in the implementation due to time constraints and generally represent the difference between the design and the implementation.

- Carry out a follow-up pilot study involving at least 30 Learners, four Teachers, (a Coordinator, Programming Guide, Peer Learning Group Facilitator, and Terminal Room Consultant) and three Plato terminals. This pilot study would have augmented Computer components as described in the following suggestions.

- Implement the Mailbox feature of the Communications File so that personal notes can be sent between Teacher and Learner.

- Add more depth to the Yellow Pages topics (especially in hardware, history, programming knowledge, social issues, software, education, future, games, government, and artificial intelligence).

- Begin to supply a CAI lesson for each terminal topic in the Yellow Pages and allow the Learner to branch to that lesson and back to the Yellow Pages.

- Improve the display of the resources in the Yellow Pages so that resources will be shown directly (uncoded) on a terminal topic page.

- Improve the rating feature of the Yellow Pages so that: (1) the Learner can see all at once the ratings of all resources for a terminal topic, (2) Teacher ratings, expected times of completion, and an abstract for each resource are available.

- Refine the choice of topics and relations in the Map---Perhaps attempting to determine a more optimal set of topics and relations.

- Display the Semantic Network of the Dictionary graphically using nodes and arcs labelled with relationships.

- Allow touch panel input on the Plato terminal---possibly improving the effectiveness of the Dictionary and the Yellow Pages.

- Increase the synonym base in the Keyword Index and flag occurrences of unrecognized Keywords that occur more than three times so they may be brought to the Coordinator's attention.

- Implement the Map (Yellow Pages and Keyword Index and Dictionary) as a data base system; i.e. develop text editors that will allow topics, relations and resources to be added, deleted, and modified. (This is a large task that could take several man-years of effort.)

- Add a level-of-difficulty parameter to the Map. This could be based on the Learner's view of himself: 'Beginner, Intermediate, or Experienced.'

- Provide CMI tests for each terminal topic in the Yellow Pages and allow the Learner to take the test as an option from the Yellow Pages. With a test available for each computer literacy topic, modular credits can be offered and the search path of the Learners will be biased only by their own interests.

- Allow a Learner to skip questions in a test and come back to them; allow a Learner to retake a test to improve an already passing grade.

- Add several types of CMI management reports (discussed in the design section 5.3.3.3.3 but not yet implemented):
  (1) Total Visits and Time of Visit to Computer Components by Individual Learner.
  (2) Computer Path and Visits to Specific Topics in the Yellow Pages and Dictionary by Individual Learner.
  (3) Summary and Detailed Histories of Learner Test Progress by Individual Learner. (These would be indicators of learning

style.)

7.2 Long Range Suggestions

- Develop a "paperless notes" system with ACCOLADE; that is, allow the Learner to take notes or otherwise store and retrieve messages to himself never having toleave the Plato terminal or use a pad and pencil.

- Allow the Learners to specify topics, relations and synonyms for the Map. A highly adaptive system would accept suggestions from Learners and modify itself accordingly based on frequency of requests.

- Extend the knowledge part of ACCOLADE to other curricula. There appears to be no reason why the Computer components, especially the Map, cannot be used to allow Learners to search out and view the structure of other knowledge spaces such as music, automobile mechanics, geography or even the Plato System itself.

  Consider this in view of the following scenario (enhanced from section 5.4):

- Attempt to incorporate more of the functions of the Teachers into the CMI system via artificial intelligence techniques--AI in CMI.

An AI in CMI Scenario

The initial interview can be between Learner and Computer, where the Computer engages in an interactive dialogue with the Learner gathering profile information which is then used to form an internal model of the Learner and to suggest the Initial Learning Plan.

The Computer's Learner Model, based on the profile and later Learner interactions with the Computer, suggests the depth and breadth of the Map presentation and the degree of difficulty of topics and resources, recommends resource types, the testing method, a path of topics to explore including times to be spent on each topic, and matches specific Teachers to the Learners.

Both the Learner and the Teacher interact with the Computer and continually supply information to the adaptive Computer Learner Model. The results of Teacher's Meetings are a major source of input to the Computer Learner Model. If any of the Teachers notice lack of confidence, for example, that information is supplied to the Computer Learner Model which adjusts the manner it will react in the future to that Learner. This would imply that help sequences would be expanded, and perhaps the amount of information seen at any one time would be decreased. When the Learner rates resources, asks for help, overrides the depth/breadth of presentation parameters or is exploring the knowledge space via the Map, the Computer is monitoring and adjusting its Learner Model based on these interactions.

If the Learner Model anticipates problems (a very simple one to meausre would be that a Learner is not using the Computer often enough), it notifies the Coordinator. The Model is also constantly correlating the Learner paths through the Computer components; and, when significant matches are found, this information is supplied to the Coordinator who in turn can update his mental model of the Learner.

Some of the functions of the Programming Guide can begin to be automated. As previously mentioned in section 5.2.1 research is being performed in this area and perhaps the future holds a computer program that can teach people how to write computer programs.

7.3 Educational Issues

In one sense, the educational issues and implications raised by ACCOLADE are not highly controversial. ACCOLADE is an alternative form of computer literacy education and, as such, would probably not be required by the bulk of any given student population. However, the presence of an ACCOLADE system within a traditional university does raise some interesting questions which center on: Evaluation, Content vs. Process, Authoritarian vs Self-Directed Education, Artificially Intelligent Computer Managed Instruction, Games, and Communications Files. Little attempt will be made here to answer these questions, only to identify the issues arising from these questions and to suggest directions for further educational research.

7.3.1 Evaluation

How to evaluate ACCOLADE is a difficult question. Two methods have already been suggested in the implementation (Chapter VI):

1) Using the naturalist's approach of just watching the system operate and taking careful notes in a log book resulted in many useful ideas for improving the system. In the words of the Zen Master:

> "To give your sheep or cow a large, spacious meadow is the way to control him. So it is with people: first let them do what they want, and watch them. This is the best policy. To ignore them is not good; that is the worst policy. The second worst is trying to control them. The best one is to watch them, just to watch them, without trying to control them."[1]

Future studies on ACCOLADE might include analysis of videotape sessions between Learner, Computer and Teacher. Since the log book proved effective in capturing the system interactions, perhaps a more information-rich medium such as television would yield deeper and more useful insights into the operation of the system---thus allowing more effective

[1] Suzuki, S., Zen Mind, Beginner's Mind, (New York: Weatherhill, 1976), p. 32.

evaluation.

2) The other evaluative technique was to ask the Learners to fill out a questionnaire evaluating the system and this too proved useful. It would also be interesting to have the Teachers fill out a questionnaire to see how it satisfied their needs.

However, there are other areas of evaluation such as Learner attitudes, Learner knowledge, and cost-effectiveness of the system that were not fully considered in the implementation.

Since it is not a goal of ACCOLADE to shape Learner attitudes toward computers into any specified mold, it might be considered a waste of time to design an instrument to measure Learner attitude. However, many of the myths discussed in Chapter IV are simply attitudes, so perhaps an attitude survey could be designed on this basis and be given as a pre- and post-survey to the Learners.

The knowledge gained by the Learners is, in one sense, evaluated automatically by the CMI component of ACCOLADE. If the objectives are valid and the questions measuring mastery of those objectives are also reliable and valid, then the Learners are gaining correct computer literacy knowledge. In order to compare the extent of this knowledge and the time of acquisition of the knowledge to that of a traditional computer literacy course, a formal control-group experiment is necessary and this in itself is a possible research topic.

The cost of administering computer literacy through an ACCOLADE framework could also be analyzed. Included in such a study would be the computer costs (terminal, rental and central processor time), terminal room costs, communications costs (telephone line charges), Teacher salaries, training costs, and system maintenance (programmer) costs. On the benefit side one would have to consider convenience and enjoyment by the Learners, depth, breadth, and speed of learning, future motivation (do they want to continue to learn?), tools acquired (programming and its associated skills and behaviors), and relevance to Learner and societal needs. These benefits can be compared to the costs associated with a traditional computer literacy curriculum.

Finally, comparative evaluations can be made: is ACCOLADE more effective for students with low or high grade point averages? For males or females? For engineers or artists? The only certainty seems to be that ACCOLADE will be more effective for self-directed learners than for traditional students who, in many cases, prefer to be told what to learn.

7.3.2 Content vs Process

An important pedagogical technique of ACCOLADE is to split the computer literacy curriculum into content and process where the knowledge part is content and the computer programming part represents process. The idea was to choose neither process nor content entirely as the mode of education but to use each where appropriate. Can an experiment be designed which will test the validity of separating content and process? Is the ratio of content/process Learner dependent, class-size dependent, or is there a threshold beyond which the ratio levels out to a constant and educational effectiveness remains optimal?

A sub-issue to consider here is: precisely what are the skills and behaviors that programming teaches? Are they problem solving, logical thinking, learning, confidence and autonomy or are there more, less or indeed, is this the wrong question to be asking?

7.3.3 Authoritarian vs Self-Directed Education

Several issues are raised when Teachers and Learners enter into a partnership that allows them to share the authorities and responsibilities of an educational experience.

ACCOLADE allows a mode of self-directed education that exists within a much larger and more powerful authoritarian environment. How can any self-directed curriculum fare in such a setting? It has already been mentioned that all of the Learners in the pilot study gave higher priorities to their traditional courses than they did to ACCOLADE. This seems to be a fact of life for curricula that attempt to individualize instruction and to allow the Learner entry into an educational partnership:

> "One of the most disturbing problems with PSI [individualized instruction] has to do with its greatest advantage--the self-pacing feature. Students have both the privilege and the burden of assuming major responsibility for their own learning, and some students are apparently not ready for or comfortable with that responsibility. While students give self-pacing top ranking among course features contributing to their enjoyment of learning (Nelson and Scott, 1974) and the majority claim that they do not find the burdens of self-pacing frustrating (Hoberock and others, 1974), 71 percent of the faculty respondents to a recent survey reported difficulty with student procrastination (PSI Newsletter, June 1974, p. 3)."[2]

How should self-directed courses compete with traditional authoritarian ones? It is not simply a question of eliminating traditional courses

[2] Cross, P.K., Accent on Learning, cited by Alfred Bork, Physics 3--Course Description, University of California at Irvine, p. 10.

in favor of self-directed ones but rather attempting to discover methods that will keep self-directed curricula from being smothered. Is it possible only outside of a formal educational setting such as a library or on-the-job training? Much research could be focussed here.

The above quote raises another important question: How many undergraduate college students are willing to take responsibility for effecting their own education? How many want to? If the percentage of students who prefer to be told what to do is high, what does this tell us about our past and present educational strategies? In what ways can we plan for the future?

ACCOLADE represents a mixture of authoritarian and self-directed philosophies. However the questions remain: How much guidance is best for the Learner? What is the best mix of authoritarian and self-directed education and is it Learner-dependent? If it is Learner-dependent what is the best type of measuring instrument for determining the optimum mix? Is a valid approach merely to ask the Learner?

Finally, teachers may find themselves affected by ACCOLADE. How will authoritarian teachers feel towards ACCOLADE? Will they welcome it, ignore it, or perhaps be threatened by it? An interesting study would be to offer ACCOLADE to teachers (of subjects outside computer literacy) and to measure their attitudes towards the system, cross tabulated by their inclination toward authoritarian or self-directed education.

7.3.4 Artificially Intelligent Computer Managed Instruction

This topic has been discussed in section 7.2 as a possible future development for ACCOLADE. Here we consider some possible ideas for educational research in this area.

ACCOLADE addresses the issue of 'representation of knowledge' which is currently a major research emphasis in Artificial Intelligence. Although ACCOLADE uses a semantic network representation of computer literacy knowledge, are there better ways--procedural representations for example? Would Learners react differently to different representations? How would different representations affect the cost and effectiveness of the system?

Another area of inquiry in Artificial Intelligence is problem solving using search strategies. Is there any way to measure a Learner's improvement in search techniques after being exposed to the Map of ACCOLADE? Can searching heuristics be taught to Learners that will improve the efficiency of their search or is it 'better' to allow Learners to discover their own techniques?

How will Teachers and Learners feel toward an artificially intelligent CMI system within ACCOLADE? Will Learners be threatened by computer guidance or will they feel more relaxed and secure knowing it is not another person (the teacher) that is look over their shoulder, judging and advising them? Will the Teacher-Learner partnership be strengthened or weakened by a strong artificially intelligent computer managed instruction system?

7.3.5 Games

In the initial stages of the implementation, ACCOLADE allowed the Learners to choose between using the Computer components of ACCOLADE or playing computer games on Plato. The Learners chose to spend over 75% of their time playing games. Certainly much research can be carried out in this area; specifically, the following questions could be studied:

Is this extremely high interest in games merely an entertainment--a need to defeat boredom--or is it a legitimate stage in the development of a programmer? This question could begin to be answered by utilizing the large game playing population already existing on the CDC Plato System.

Another question centers on the types of games that are most played and why they are popular. Plato offers at least three types of games to the user: user vs computer (chess, checkers, tic-tac-toe), user vs user (empire, conquest, airfight, moon war) or user with computer (simulation games such as biorythm generation and Conway's game of Life). Based on statistics kept by the Plato system, by far the most popular games seem to be user-vs-user-war-type games where the computer acts only as bookkeeper, information display generator, and referee. Why is this so? In particular, which types of Learners are attracted to which types of games? (These may be some correlation here with which types of Learners are attracted to which types of ACCOLADE Computer literacy topics.)

Most of the games on Plato are competitive (as are most games in general) but a few are also partially cooperative. "Empire," for example, requires that the user join a team at the outset of the game and the competition is then between four possible teams--the goal being conquest of the universe. Research in the theory and practice of cooperative games has been available since 1950,[3,4,5] however little research has been done where the computer enters as player. Here, it seems, is an enjoyable area where a valuable research contribution can be made.

Computer games can also provide a controlled opportunity to test the contention that learning through play is an effective educational strategy. The computer can be programmed to unobtrusively gather data and 'observe' while the game proceeds without incurring the Heisenberg effect of the data being modified by the presence of a measuring instrument. It would be of great interest to test the paradoxical Zen statement: "The best way to control people is to encourage them to be mischievous."[6]

7.3.6 Communications Files

Along with the games, the Learners were allowed access to over 300 communications files on the CDC Plato system. These "notesfiles" are all similar to the ACCOLADE Bulletin Board--in general, most anyone can read the notes and respond to them. Each notes file usually caters to a certain class of user or special interest group; for example,

[3] Lentz, T. and Cornelius, R., All Together (A Manual of Cooperative Games), Peace Research Laboratory, 6251 San Bonita, St. Louis, MO 63105, 1950.

[4] Fluegelman, A., The New Games Book, (New York: Doubleday, 1976).

[5] Brand, S., "Theory of Game-Change," The Co-Evolution Quarterly, Summer 1976, pp. 95-99.

[6] Suzuki, p. 32.

there are notes files dealing with movies, music, books, astronomy, and games, as well as general interest notes files such as "pbnotes" (public notes) and "forum." It is the experience of the author that, as Learners become more experienced users of the Plato system, they tend to spend more time in these notesfiles. Thus it is entirely possible that in a large Plato system, a user could spend all of his time just reading and writing notes! Statistics gathered on the University of Illinois Plato system based on a 311 day sample this past year indicate that out of 4000 possible users, there were an average of 1847 notes files users who wrote an average of 924.6 notes and read an average of 4830.9 notes each day.[7] Therefore, an ACCOLADE system which uses notesfiles must consider these kinds of implications.

At its highest level, the suggested research is about "understanding the nature of human dialogue via electronic media."[8] No one is quite sure yet what these notesfiles really are, let alone their implications. At a minimum, they are a teleconferencing mechanism allowing geographically remote groups of people to share ideas. At another extreme they fulfill many of the functions of a companion. Being on the CDC Plato system is like have 200 pen-pals who communicate through various notesfiles. However since the medium is electronic and a very powerful computer-supplied notes editor (composer) is available, the interaction times are considerably quicker than those

[7] Avner, A., CDC Plato System, notesfile "neher2", Note 5, Response 5, April 21, 1977.

[8] Neher, W., CDC Plato System, notesfile "neher2", Note 2, Reponse 7, April 17, 1977.

afforded by the postal service. It is perhaps due to this speed that the quality of the interaction changes in much the same way it changes between mailed letters and the telephone. On the current CDC Plato system it is possible to leave a request for a good bicycle maintenance manual in the morning and receive half a dozen responses by late afternoon. Some notes (politically or philosophically oriented) generate dozens of responses (and replies to the responses). Whether notesfiles are in the incarnation of the learning webs of Illich,[9] community bulletin boards, or electronic companions, their meaning and their future implications are ripe for research. There are already three recorded instances of marriages resulting from communications as a result of an initial contact in the Plato notesfiles.[10]

Another issue that is spawned from the notesfiles is that of censorship. Usually the computer memory for the storage of the notesfile is 'owned' by a group of 'directors' who are programmers that work for CDC. The directors usually state a policy outlining the types of notes appropriate for the file but they find it very difficult to define a censorship policy. CDC cannot allow pornography or obscenities to appear in notesfiles which are open to all users on the Plato system (which includes second-graders). CDC cannot afford to market a system that will offend its customers. On the other hand, most CDC employees are people that do not want to impose censorship, realizing it can create more problems than it alleviates. The default position has

[9] Illich, I., Deschooling Society, (New York: Harper Row, 1972).

[10] Private communication from 'luke/nyc' on the CDC Plato system. Also discussed in CDC Plato notesfile 'pbnotes,' note: 'marriages,' 7/18/77.

been to rely on peer group pressure to regulate the quality and content of the notes and, in cases where the people are acquainted with one another, this pressure seems to work adequately well. However more research needs to be done here. Is a notesfile like a "letters to the editor" column where the directors play the role of editors and can censor as they like? Is it possible to gather profile data on Plato users which can be used to screen certain notes from their view? Censorship of communications, whether in an educational or entertainment milieu, remains a flinty philosophical problem.

A final aspect to the notesfiles that relates to education is the humanizing influence they can exert on communications between people. Although it may seem at first paradoxical that such a highly technological medium as a computer notesfile can humanize communications, it does in fact happen all the time and has been noted by various Plato users. The author, for example was in the highly complex game of empire for the first time and sent out a message to all other users, "new player in game--please be gentle." A response was received almost immediately, "need some help?". After responding affirmatively, the author was intelligently instructed in the rules of the game as well as some useful strategies ("don't use your phasers, use photon torpedoes instead--much more powerful."). After many minutes of valuable instruction, the author was asked by his mentor, "how old are you?" "38," replied the author. "YIIIKES," was the response. "What's the problem?" puzzled the author. "I'm 12," replied the mentor. "No problem," said the author--although it was very difficult for the author to avoid talking 'down' and the 'mentor' from talking 'up' in the communications that followed.

The point is that since the computer communications in games as well as in notesfiles are stripped of their stereotypic cues (sex, age, race, tone of voice, physical characteristics, smells, body language) the communication becomes truly egalitarian. As a result, communications seem to be enhanced and humanized by one of man's most advanced technological tools. This humanizing influence is important and ready for future study and research.

APPENDIX A

DRAFT LIST OF GENERAL HEURISTIC CUES

from EDUC 790 M: Computer Models of Learning, H.A. Peelle

- break the problem into sub-problems
- start with a good example or a good model (if you can)
- use analogy: "think of ____ as ____"
- simplify (try a simple case)
- invent a possible solution
- look from a different perspective (e.g., orthogonal)
- ask how it works
- ask why (if it works)
- demonstrate how it works - explain your dea: "what do you mean by ___"
- ask qualitative questions, e.g., "Does it have the right idea (basically)?"
- change something and observe what else changes
- conjecture ("what happens when you ____")
- ask quantitative questions - e.g. change position but not time
  e.g. change time, but not position
- interpolate (try something inbetween)
- extrapolate
- construct a theory
- test your theory
- discover, investigate and fix bugs
- search for a counterexample
- sum up

- enumerate (all the) possibilities - (or give a procedure for doing so)
- automate a good thing
- remove part(s)
- put it back

APPENDIX B

COMPUTER LITERACY TOPICS

Following is a hierachically organized (incomplete) list of suggested topics for a computer literacy curriculum. Topics followed by an asterisk have resources associated with them in the current implementation of the Yellow Pages.

I. Applications

   A. General*
   B. Natural Sciences
      1. General*
      2. Astronomy, Space*
      3. Biology
      4. Agriculture
      5. Chemistry
      6. Earth Sciences
      7. Mathematics
         a. Numerical Analysis
            i. General*
            ii. Solution of Single Equations*
            iii. Solution of Linear Systems of Equations*
            iv. Integration*
            v. Differentation
            vi. Interpolation and Extrapolation
            vii. Matrix Operations
            viii. Ordinary and Partial Differential Equations
         b. Statics
            i. General*
            ii. Probability
            iii. Descriptive Measures*
            iv. Frequency Distributions*
            v. Regression and Correlation*
            vi. Analysis of Variance
         c. Topology
            i. General
            ii. Four Color Problem*
      8. Meteorology*
      9. Physics
         a. Neutron Diffusion*
         b. Optics*
         c. Rockets*

C. Engineering
   1. General*
   2. Aeronautical
   3. Architecture*
   4. Chemical
   5. Civil
   6. Design*
   7. Environmental Planning*
   8. Electrical
   9. Energy Systems
   10. Mechanical
   11. Numerical and Process Control*
   12. Transportation
      a. Ground*
      b. Air*

D. Social and Behavioral Sciences
   1. General*
   2. Anthropology
   3. Law
      a. Courts*
      b. Crime*
      c. Lawyers*
   4. Library*
   5. Medicine and Health
      a. General*
      b. Diagnosis*
      c. Process Control*
      d. Research*
      e. Training*
      f. Information Systems*
      g. Hospital Administration*
   6. Political Science
   7. Psychology
   8. Sociology

E. Humanities
   1. General*
   2. Art*
   3. English*
      a. Journalism*
   4. History*
   5. Music*
   6. Performing Arts*

F. Business
   1. General*
   2. Banking and Finance*
   3. Brokerage*
   4. Clerical
      a. General*
      b. Accounting*
      c. Inventory*
      d. Payroll*
      e. Sales*
      f. Scheduling*

5. Data Processing Techniques
a. Sorting
b. Matching
c. Merging
6. Economics*
7. Management*
8. Manufacturing and Process*
9. Marketing
10. Real Estate*
11. Systems Analysis*
12. Operations Research Techniques
a. Simulation
b. Linear and Nonlinear Programming

II. Hardware
A. General*
B. Analogies-Units
C. Offline Devices
1. Keypunch
2. Sorter
3. Reproducer
4. Interpreter
D. Architecture
1. Control
2. Memory
3. Arithmetic/Logic
4. Input/Output
5. Special Purpose
a. Networks
b. Minicomputers
c. Multiprocessors
6. Data Communications
7. Switching Theory/Boolean Algebra

III. History
A. General*

IV. Programming Knowledge
A. General*
B. Lanugages
C. Techniques

V. Social Issues
A. General*
B. Privacy*
C. Personal Information Services
D. Monitary Systems
E. Elections
F. Education
G. Employment
H. Liability Questions
I. Monopoly Considerations

- J. Patents and Copyrights
- K. National Development
- L. Social Power
- M. Metaphor for Man

VI. Software
- A. General*
- B. Flowcharts and Decision Tables
- C. Utilities
  - 1. Application-oriented
  - 2. Input/Output
  - 3. Debugging
  - 4. Program Maintenance
- D. Processors
  - 1. Assemblers
  - 2. Compilers and Generators
  - 3. Interpreters
- E. Supervisors
  - 1. Basic Monitors
  - 2. Multiprogramming
- F. Data Structures
  - 1. Scalars
  - 2. Arrays
    - a. Vectors
    - b. Matrices
  - 3. Strings
  - 4. Lists
  - 5. Trees
  - 6. Graphs
  - 7. Fields/Records/Files

VII. Education
- A. General*
- B. Administrative
  - 1. Grading*
  - 2. Scheduling*
  - 3. Business
    - a. Budgeting
    - b. Planning
    - c. Alumni Records
  - 4. Personnel
  - 5. Attendance
  - 6. Testing
- C. Instructional
  - 1. General*
  - 2. Computer Assisted Instruction*
  - 3. Computer Managed Instruction*
  - 4. Computer Literacy*
  - 5. Programming*
  - 6. Disciplines
    - a. Mathematics

b. Physics, Chemistry
c. Biology
d. Business
e. Humanities
D. Research*

VIII. Future
A. General*

IX. Games
A. General*
B. Bobstones*
B. Tic-tac-toe*
D. Deepspace*
E. Mastermind*
F. Grundy's Game*
G. Wargames*
H. Plato Games* (includes chess and checkers)

X. Government
A. General*
B. Employment*
C. International*
D. Military
1. General*
2. Fire Control*
3. Missiles*
4. Navigation*
5. Training*
E. Politics*
F. Welfare*
G. Space
H. Health
I. Education

XI. Artificial Intelligence
A. General*
B. Problem Solving*
C. Natural Language Processing*
D. Learning
E. Pattern Recognition*
F. Theorem Proving*
G. Robotics*

APPENDIX C

LIST OF RESOURCES

Yellow Pages

Printed Material

1-1
Digest of Papers from the Tenth IEEE Computer Society
Conference, COMPCON75, Feb 25-27, 1975,
Computer Technology to reach the People
IEEE Catalog No. 75CHØ92Ø-9C

1-2
Proceedings of the 196Ø Computer Applications Symposium,
Oct 26-27,196Ø, Computer Applications-196Ø ,
edited by B. Mittman and A. Unger,
The MacMillan Co., N.Y.,1961

1-3
Computers and the Year 2ØØØ ,
edited by Avebury et al, NCC Publications,1972

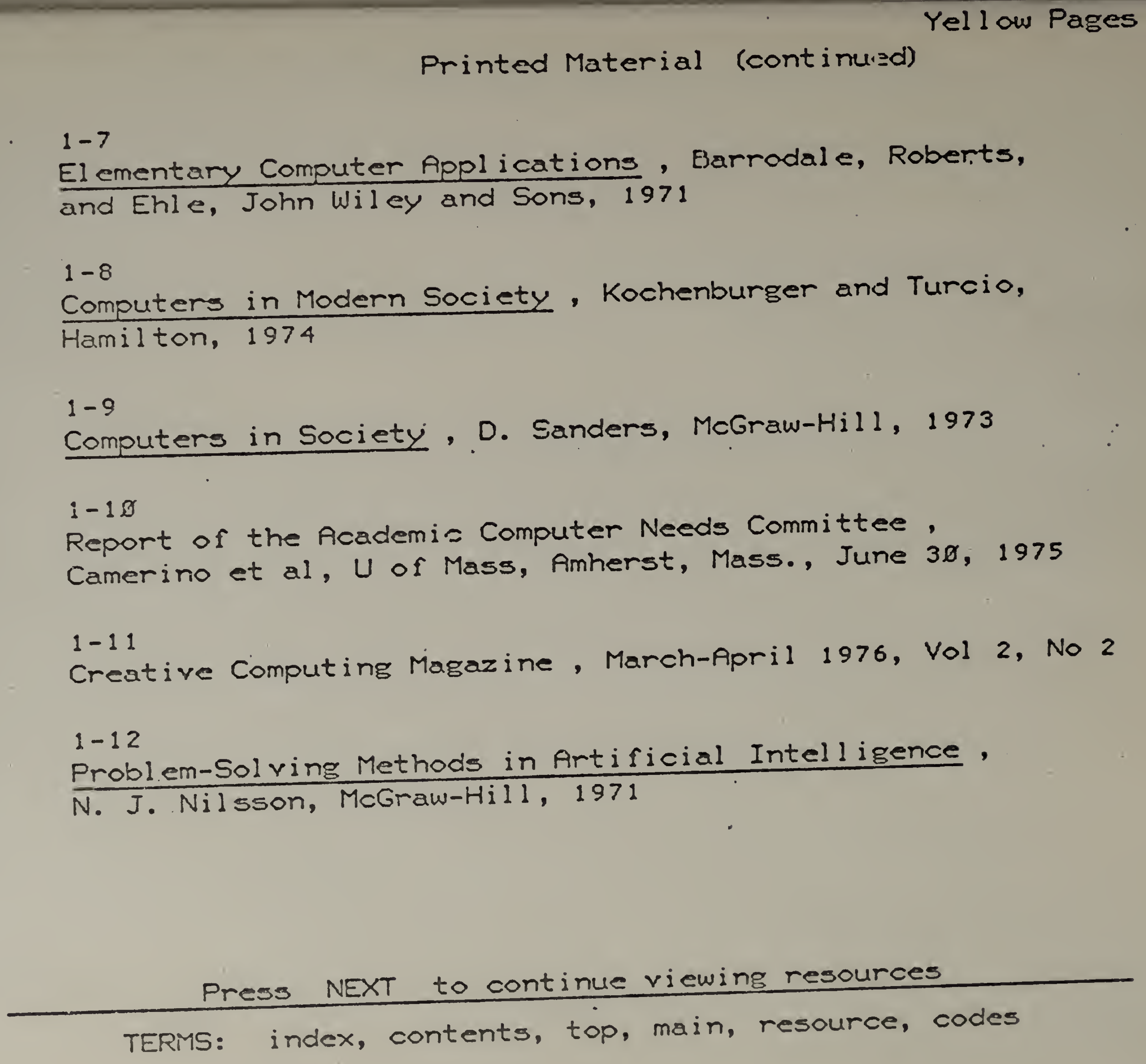
1-4
Computers and Society ,
S. Rothman and C. Mossman, SRA, 1972

1-5
Time Magazine, Sept 2Ø,1976

1-6
THE Journal,Vol 3,No 2, Feb 1976

Press NEXT to continue viewing resources

TERMS: index, contents, top, main, resource, codes



Yellow Pages

Printed Material (continued)

1-7
Elementary Computer Applications , Barrodale, Roberts,
and Ehle, John Wiley and Sons, 1971

1-8
Computers in Modern Society , Kochenburger and Turcio,
Hamilton, 1974

1-9
Computers in Society , D. Sanders, McGraw-Hill, 1973

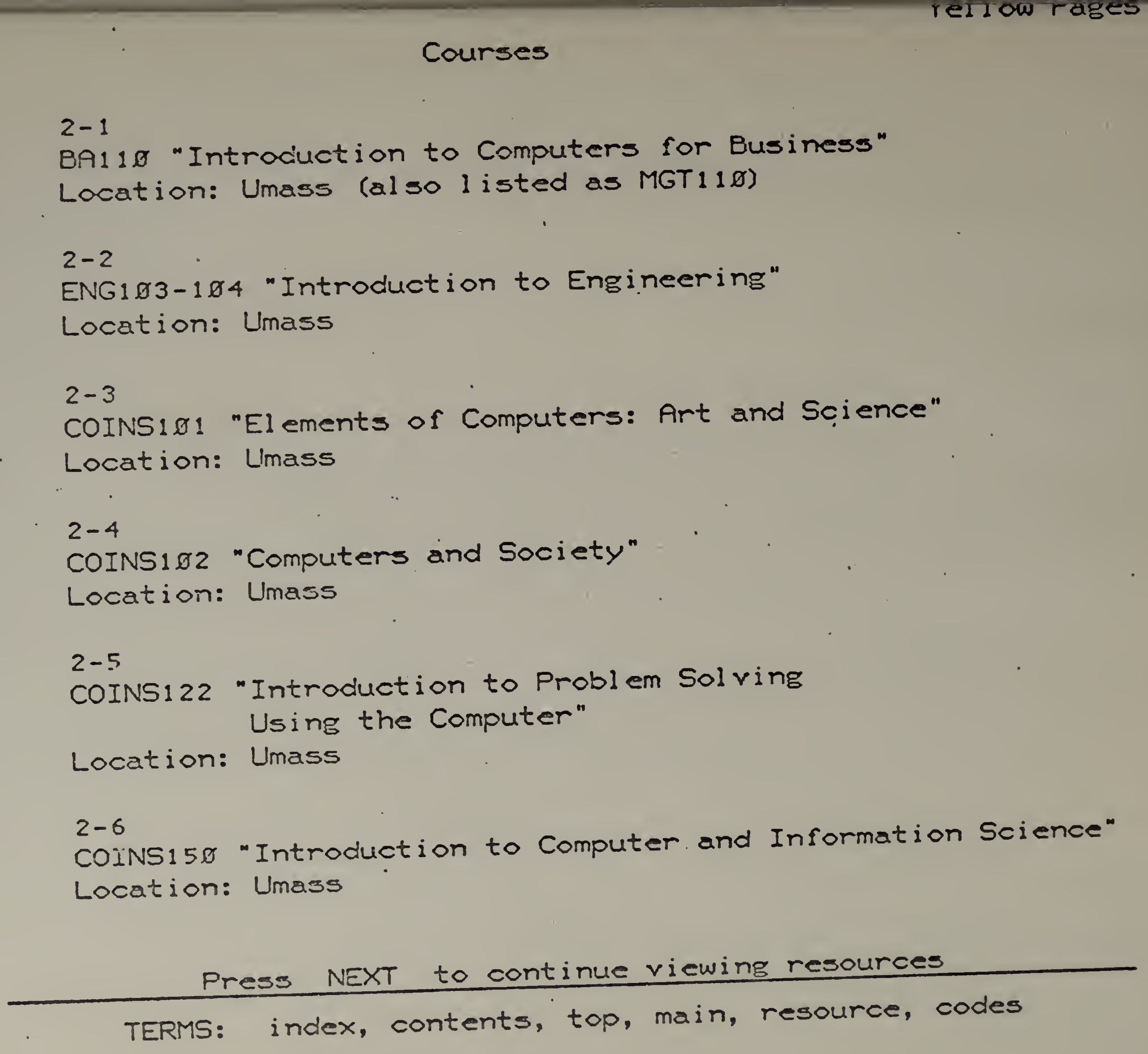
1-1Ø
Report of the Academic Computer Needs Committee ,
Camerino et al, U of Mass, Amherst, Mass., June 3Ø, 1975

1-11
Creative Computing Magazine , March-April 1976, Vol 2, No 2

1-12
Problem-Solving Methods in Artificial Intelligence ,
N. J. Nilsson, McGraw-Hill, 1971

Press NEXT to continue viewing resources

TERMS: index, contents, top, main, resource, codes



```
                                                        Yellow Pages
                         Courses

2-1
BA11Ø "Introduction to Computers for Business"
Location: Umass (also listed as MGT11Ø)

2-2
ENG1Ø3-1Ø4 "Introduction to Engineering"
Location: Umass

2-3
COINS1Ø1 "Elements of Computers: Art and Science"
Location: Umass

2-4
COINS1Ø2 "Computers and Society"
Location: Umass

2-5
COINS122 "Introduction to Problem Solving
          Using the Computer"
Location: Umass

2-6
COINS15Ø "Introduction to Computer and Information Science"
Location: Umass

     Press  NEXT  to continue viewing resources
_______________________________________________________
  TERMS:  index, contents, top, main, resource, codes
```

Courses (continued)

2-7
BA21Ø "Business Applications of Computers"
Location: Umass

2-8
BA211 "Business Information Systems"
Location: Umass

2-9
Industrial Engineering and Operations Research 256
"Data Processing and Information Handling Systems"
Location: Umass

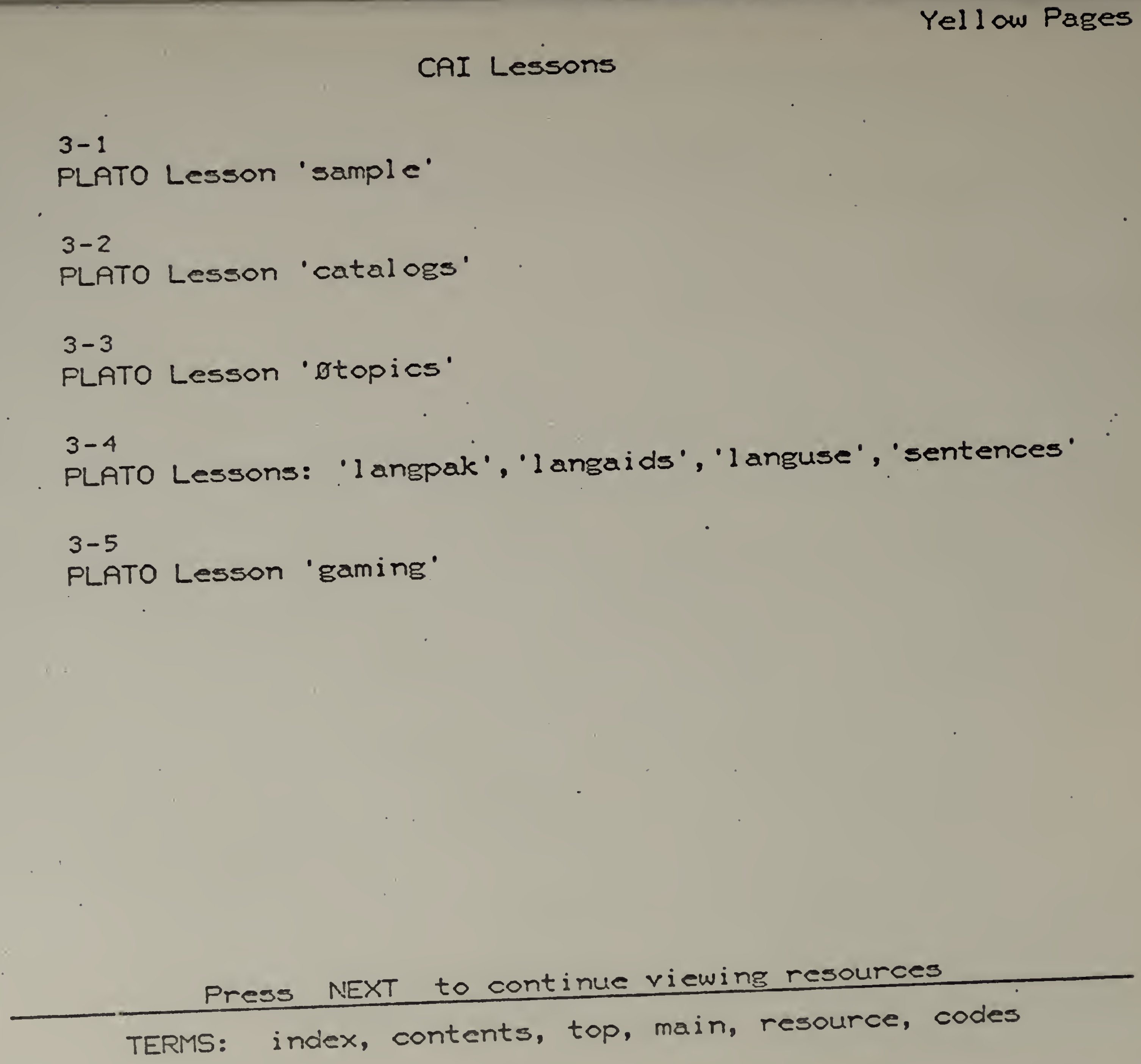
2-1Ø
LC156 "Introduction to Computers and Programming"
Location: Hampshire College

2-11
LC184 "Minds Brains and Machines"
Location: Hampshire College

2-12
121 "Computer Science I"
Location: Mount Holyoke College

Press NEXT to continue viewing resources

TERMS: index, contents, top, main, resource, codes



Yellow Pages

CAI Lessons

3-1
PLATO Lesson 'sample'

3-2
PLATO Lesson 'catalogs'

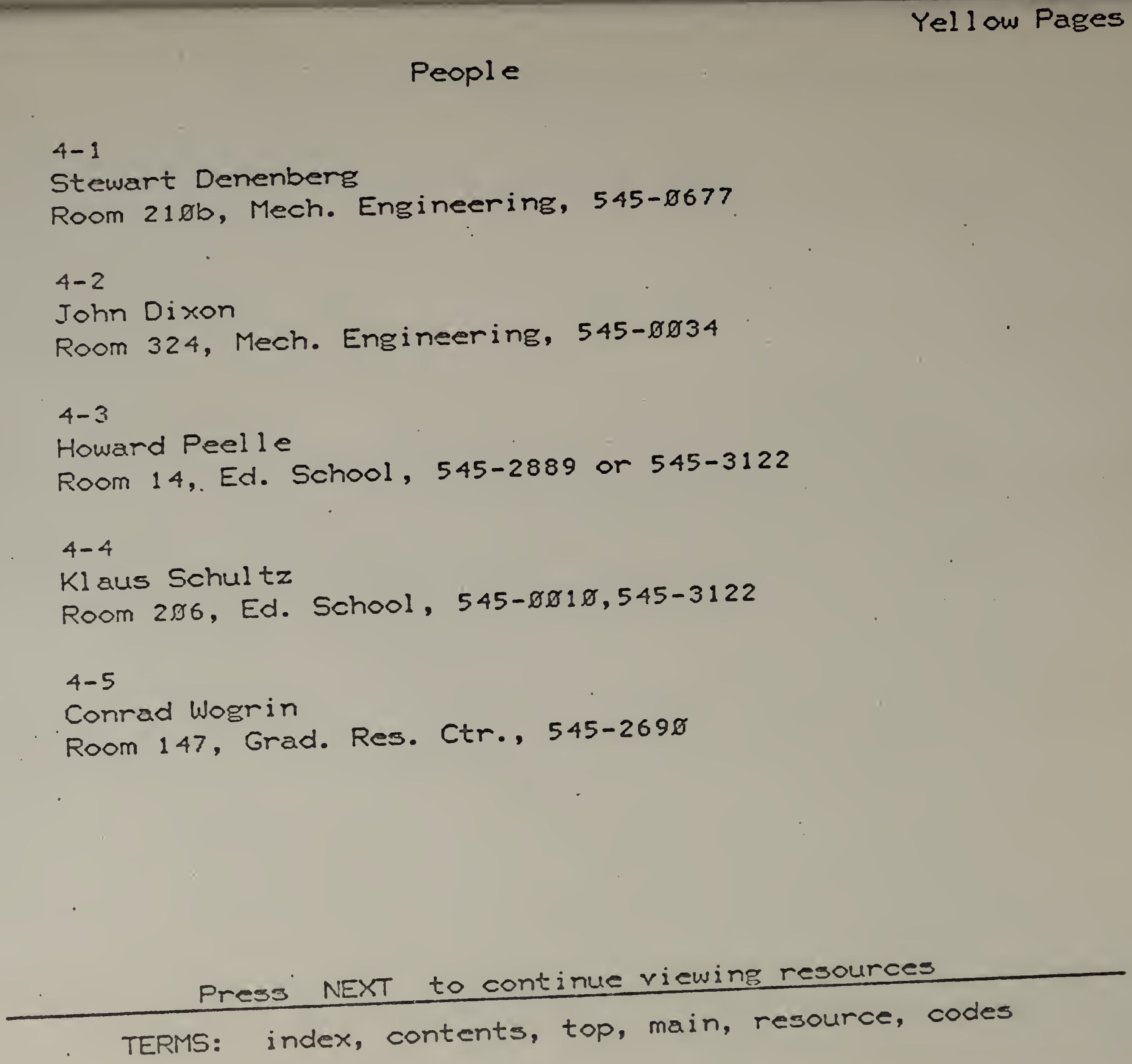
3-3
PLATO Lesson 'Øtopics'

3-4
PLATO Lessons: 'langpak','langaids','languse','sentences'

3-5
PLATO Lesson 'gaming'

Press NEXT to continue viewing resources

TERMS: index, contents, top, main, resource, codes



Yellow Pages

People

4-1
Stewart Denenberg
Room 21Øb, Mech. Engineering, 545-Ø677

4-2
John Dixon
Room 324, Mech. Engineering, 545-ØØ34

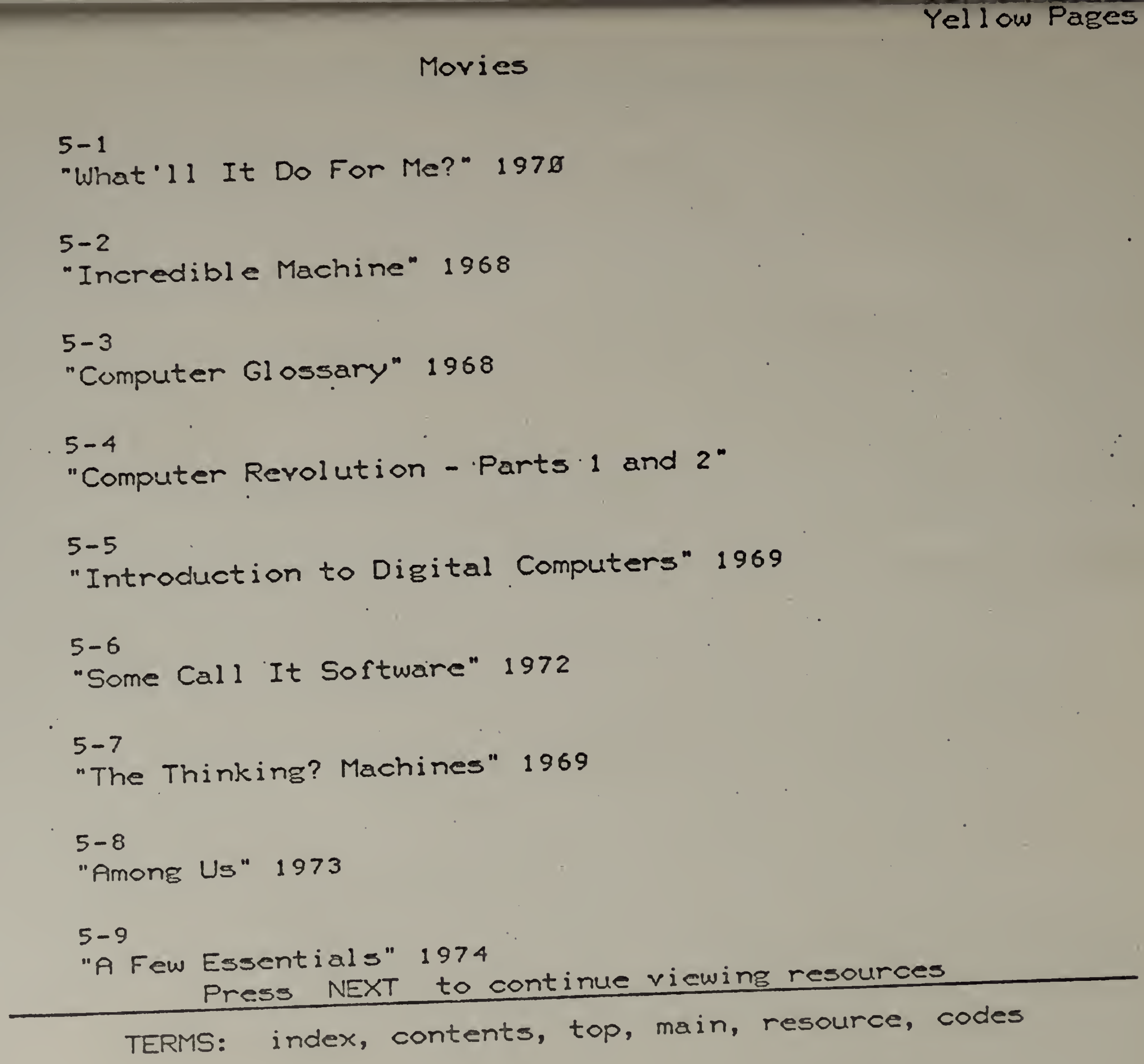
4-3
Howard Peelle
Room 14,. Ed. School, 545-2889 or 545-3122

4-4
Klaus Schultz
Room 2Ø6, Ed. School, 545-ØØ1Ø,545-3122

4-5
Conrad Wogrin
Room 147, Grad. Res. Ctr., 545-269Ø

Press NEXT to continue viewing resources

TERMS: index, contents, top, main, resource, codes



Yellow Pages

Movies

5-1
"What'll It Do For Me?" 197Ø

5-2
"Incredible Machine" 1968

5-3
"Computer Glossary" 1968

5-4
"Computer Revolution - Parts 1 and 2"

5-5
"Introduction to Digital Computers" 1969

5-6
"Some Call It Software" 1972

5-7
"The Thinking? Machines" 1969

5-8
"Among Us" 1973

5-9
"A Few Essentials" 1974

Press NEXT to continue viewing resources

TERMS: index, contents, top, main, resource, codes