UNIVERSITY OF ALBERTA

Human-Computer Enaction

By ©

Judith M. Iseke Barnes

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

DEPARTMENT OF SECONDARY EDUCATION

Edmonton, Alberta

Spring, 1994

ISBN 0-612-11151-2

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Judith M. Iseke Barnes

TITLE OF THESIS: Human-Computer Enaction

DEGREE: Doctor of Philosophy

YEAR THIS DEGREE GRANTED: 1994

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Judy Barnes
8 Maple Drive,
St. Albert, Alberta
T8N 1G5

Date: December 13, 1993

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Human-Computer Enaction** submitted by Judith M. Iseke Barnes in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dr. Heidi Kass

Dr. Thomas Kieren

Dr. Steven Hunka

Dr. Alton Olson

Dr. Margaret Iveson

Date: November 10, 1993

Dr. Lars Jansson

For Neal and Jennice.

ABSTRACT

Human-computer enaction is explored by asking: 1) how can human-computer interactions be interpreted from an enactive perspective? and 2) what are the methodological and instructional implications of an enactive stance? Descriptions are drawn from a series of studies from a research program involving two contexts. The AP context was created by students in interactions with an icon-based programming language called Authorware Professional (AP). The Chaos-theory context was created by students interacting with mathematics software for learning about chaos theory. Five chapters describe a series of studies in the two contexts. Chapter 2 describes the problems of conceptualizing and understanding icon-based programming through an interview-based account of university students use of AP. Chapter 3 discusses students perceived roles and experiences producing multimedia stories in AP in this interview-based account of computer use by high school students. Chapter 4 traces the processes of development of multimedia computer files by high-school students through a case-study approach. The focus is to understand students' development of computer concepts, patterns and styles of interaction with the computer software, and students' conceptions of the development of story. Chapter 5 describes two computer programs for exploration of ideas inherent in chaos theory and non-linear dynamics, discusses content important to the understanding of chaos theory and considers the graphics inherent in these computer programs and the values and limitations of these graphics in regard to developing understanding of the mathematical concepts. Chapter 6 examines the nature of the interactions of high-school mathematics students with chaos theory (non-linear dynamics) through explorations and manipulation of computer programs as an extension of the mathematics curriculum. Chapter 7 discusses human-computer interaction from two views of cognition, from the representationist view and the enactivist perspective. Both of the contexts, the Authorware Professional (AP) context and the chaos-theory context, are discussed from their origins in the representationist stance and then explored through the enactive view. Implications of the enactive stance for education and research are then articulated.

ACKNOWLEDGEMENTS

I would like to express my appreciation to the members of the examining committee. First I thank my supervisor, Dr. Heidi Kass, who considered my interests in providing supportive comments in regard to the research projects and who believed in me even when I faltered. This process would not have been possible without her. I am very grateful to Dr. Tom Kieren who acted as my supervisor while Heidi was away, provided the impetus for the mathematics project through his own research activities, and always provided conceptual comments and challenges to my thinking through our many interactions. Dr. Steve Hunka made his knowledge and experience available to me throughout the process of completing the masters and Ph.D.. His willingness to consult in regard to my educational and research activities is much appreciated. I wish to thank Dr. Al Olson who started me on the process of this research agenda in early courses in the Ph.D. program and provided ideas for my thinking through conversations and shared projects. Dr. Marg Iveson has shown great interest in this project and provided support for this process through her positive comments and thoughtfulness. Of course she also paid careful attention to details for which I am grateful. Dr. Lars Jansson acted as the external reader of the dissertation under great time pressure. His examination of the work and comments were both insightful and challenging. I have tried to take action upon his comments, as well as the comments of all the committee members. I am very grateful to all members of the committee whose contributions are evident in the work reported here. Thank you all for your support.

Others who have made this project possible are the students and staff at the schools in which I worked. I cannot name them here but I hope they know they are appreciated. In addition, I would like to express my thanks to the many staff and students in the Department of Secondary Education who provided an environment in which I could pursue my interests.

Finally, I wish to thank my family for the support and interest in my projects. Without their willingness to change their lives along with me I would not have been able to complete this research project nor begin the adventure which lies ahead. Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Human-computer interaction has emerged during recent years as an important field of study. Its proponents come from many areas of specialization including computing science, psychology, linguistics, philosophy, anthropology, sociology, and neurophysiology. It is often linked to cognitive science which sees as its central tenet a view of cognition as information processing. Some in cognitive science view computers as appropriate models of human thought processes . However, not all members of the cognitive science community are comfortable with this limited view. The scope of the field of cognitive science has failed to consider people interacting with people and their environment, peoples' personal history, and culture as aspects which might be worthy of inclusion in studies in this field (Norman, 1980, Putnam, 1989, Searle, 1992).

A response to this call for a broader understanding of human-computer interaction is often met with skepticism. These skeptics wish to restrict their inquiry to experimental approaches separating out problems from their surroundings and oversimplifying the contexts. Bannon and Bødker (1991, p. 230) propose that the experimental approach produces a view of a person acting in a setting as disembodied with little or no relationship to the view of a person acting in the world. In response to these limitations of experimental methods more qualitative approaches to the study of human thought and action have been proposed. But a shift in research methods is not entirely at the root of these more qualitative approaches. Underlying these more qualitative methodologies are different views of cognition than that which is generally associated with cognitive science, that is, the representationist stance.

From the representationist stance "The world is pregiven... its features can be specified prior to any cognitive activity. Then to explain the relations between this cognitive activity and a pregiven world, we hypothesize the existence of mental representations inside the cognitive system" (Varela, Thompson, Rosch, 1991, p. 135). Action in the world is based upon these cognitive representations. The test of these cognitive representations is correspondence to the appropriate features of the world.

In discussing the representationist point of view Varela, Thompson, Rosch (1991) state that "the realist naturally thinks that there is a distinction between our ideas or concepts and that which they represent, namely, the world" (p. 136) while "the idealist, on the other hand, quickly points out that we have no access to such an independent world except through our representations" (p. 137). Both stances require a search for a ground. For the realist the ground is external while the idealist searches for an internal ground. The idealist, by rejecting the idea of an independent world, is left without a sense of an outer ground.

Varela, Thompson, Rosch (1991, p. 140) describe an alternative approach to this searching for a ground internally or externally. From this perspective, called "**enaction**", the world is enacted "as a domain of distinctions that is inseparable from the structure embodied by the cognitive system". The world is not completely independent nor is it dependent. It is not purely the construct of our own thoughts or perception but is enacted by our structure. Thus it is not an independent entity 'out there' nor is it the internal construction of individuals. It is the interplay between the internal constructions and the structure.

As an example, consider two computer user's views of a word processing program. The word processor that one user uses may include features to spell check, edit, automatically generate table of contents and the ability to format text in many ways including fractions and symbols. Another user may use the same word processor as a simple place to type text. The option to incorporate fractions may not even exist for this user (although made possible within the word processing package). Each user may indeed use the same computer program but to the two users the programs are quite different. To the first user many functions are possible and thus the view of the software encompasses these functions. To the second user the view of the same software is more limited. The functions may not even exist for the second user because he/she is unaware of them. However when this second user has been shown, or has discovered, a feature such as the ability to place fractions into the text the word processor changes for this user because it includes this new function. This feature has come to exist for this user by its introduction to him/her. From an enactivist stance the computer user brings forth a world as he/she interacts with the computer and software. This world is determined by the user's structure.

However the computer environment (hardware, software, and interface) also occasions the actions of the computer user. The computer program allows for the placement of text in various formats including fractions. The user of the word processor acts within the possibilities made available by the word processing environment but limited by the user's view of the environment. This view is ever-changing as the user takes action in this environment thus the environment to the user is also changing. The environment thus provides a "sphere of behavioral possibilities" through/in which the user takes action. The user's actions coemerge with/in the environment.

From this example we see that the user determines the world by his/her actions and the actions coemerge with/in the world. Thus the user and world are said to coemerge. The user's knowledge through actions coemerges with the environment. This knowledge is not separate from the environment. These users are immersed in a technological age and find themselves in a computer environment using a word processor. Their cognition is to be judged in this context not by how well their actions match a predefined outcome in the environment but by how they coemerge with/in this world.

Statement of the problem

The initial work for this dissertation began in a representationist frame. However this stance seemed limited and so a shift toward an alternative approach called enaction was undertaken. During this shift a number of questions were asked: if we move beyond the cognitive science view of cognition which is typically representationist and take an enactive stance then what does human-computer interaction become? How might a view of human-computer interaction be enacted? In answering these questions an exploration was undertaken of examples of studies of students learning and their actions and experiences within computer-assisted environments. 'Views' of human-computer interaction were explored with the emphasis upon interaction and the nature of the human experience in the presence of computer technology. This research agenda, through its various studies, addressed the experience of computer technology, the conceptualization from an enactive stance, and the nature of human-computer interactions by assessing the computer environments and students interactions with them. Questions inherent in the studies were: is the computer environment pregiven in a program through which students work or is the computer environment coevolving with student actions as students are immersed in this environment? Computer environments are prestructured in programs to allow certain possibilities (and not to allow others). Students select from these possibilities by their actions. Does the environment depend on perceptually guided activity of the student?

Through the various chapters of this dissertation the shift toward an enactive stance is evident and the question of whether an enactive stance can be taken toward the data are replaced by questions of how this can be done. Thus the questions addressed in the final chapter of the dissertation (Chapter 7) are:

1) How can human-computer interactions be interpreted from an enactive perspective?
2) What are the methodological and instructional implications of an enactive stance?

Methodological Orientation

This dissertation describes studies from a research program involving two contexts, the Authorware-Professional[1] (AP) context and the Chaos-theory context, which attempt to address the stated research questions. These questions are quite global in nature and interpretation of these questions can be advanced from many perspectives. To interpret human-computer interaction one must examine an image of the computer environment, a view of cognition (the human element), and some sense of the interaction (and enaction) of the environment and the participant. It is necessary to examine participants interacting with/in an environment and to examine their perceptions of this environment. A methodology involving observation, interviews, audio and video recordings, and student record-making was devised in order to examine students' activities and perceptions and to provide a framework to develop these interpretations. This methodology was developed within an enactivist frame of reference which attempted to examine the codetermination of participants and their world.

A research orientation, called phenomenography, also attempts to determine the participants' experiences of their world. It aims at learning about the human experience of a phenomenon in the world. A distinction is made between first and second order phenomena. First order phenomena are in the world. When discussing first order phenomena we make statements about the world. Second order phenomena are about peoples' ideas and experiences of the world. The research studies reported in Chapters 2 and 5 also make this distinction in describing the software used in the computer environments for these studies and the ways these environments can be viewed. Performance and product of performance are described. However these initial studies miss the point that action is embodied. The task then of Chapters 3, 4, and 6 is to explore how students' actions are not separate from their histories and their structures. Thus the methods of the phenomenographic approach, looking at experiences, are implicitly present in the studies for Chapters 2 and 5 but the philosophic underpinnings are enactive. These studies are based on the assumptions that the student brings forth a world through embodied actions in the world; and, based upon the student's history and structure. The methods used, while phenomenographic in nature, take this assumption into account. Looking only at products of actions misses the nature of the action. This is not to say that products are unimportant. They are studied in this research agenda. However, they are deemed insufficient and thus Chapters 3, 4, and 6 respond to this incomplete picture by looking at students' embodied actions. Multiple data sources are thus required to generate a picture of embodied action.

Methodology

The research program for this dissertation consisted of seven studies in two contexts: 1) the Authorware Professional (AP) context and 2) the chaos theory context.

The Authorware-Professional Context. The AP research project began by asking: how does a student develop the skills and concepts to be functional in this computer context? The task for the first study was 1) to explore the icon-based computer programming environment called Authorware Professional (AP) and identify concepts and skills necessary for student use of AP and 2) to develop instructional simulations (IS) which would encourage students to develop concepts and skills through exploration of aspects of AP. Previous research regarding learning programming and in regard to developing simulations was reviewed (Lodding, 1983, Alessi, 1988, Goktepe, Ozguc, & Baray, 1989, Reigeluth, & Schwartz, 1989, Jones, 1990, Stead, 1990). This research

---

1 Authorware Professional is a produced and distributed by Authorware, Inc., Minneapolis, MN, 1992.

review was the basis for the first IS generated. The intent of this initial IS was to initiate students' processes of inquiry about the properties of icons and flowcharts. Further simulations were developed into the four instructional simulations used for the studies described in Chapters 2, 3, and 4.

The IS's developed by the investigator were provided for students to illuminate concepts involved in the use of AP. Many concepts were included in the simulations but a brief list includes the concepts of icon (collections of operations, visually stand for a set of operations), tool box (enables definition of text, lines, circles, and boxes), pull-down menus (variations upon sets of operations are chosen here), types of variations (font size and style, line thicknesses and patterns, modes and fill patterns, types of effects), creation of a file, the operation of pausing (separates two computer actions in time), and variation upon the operation of erasing (zoom-to-point, zoom-to-line, and fade-out)

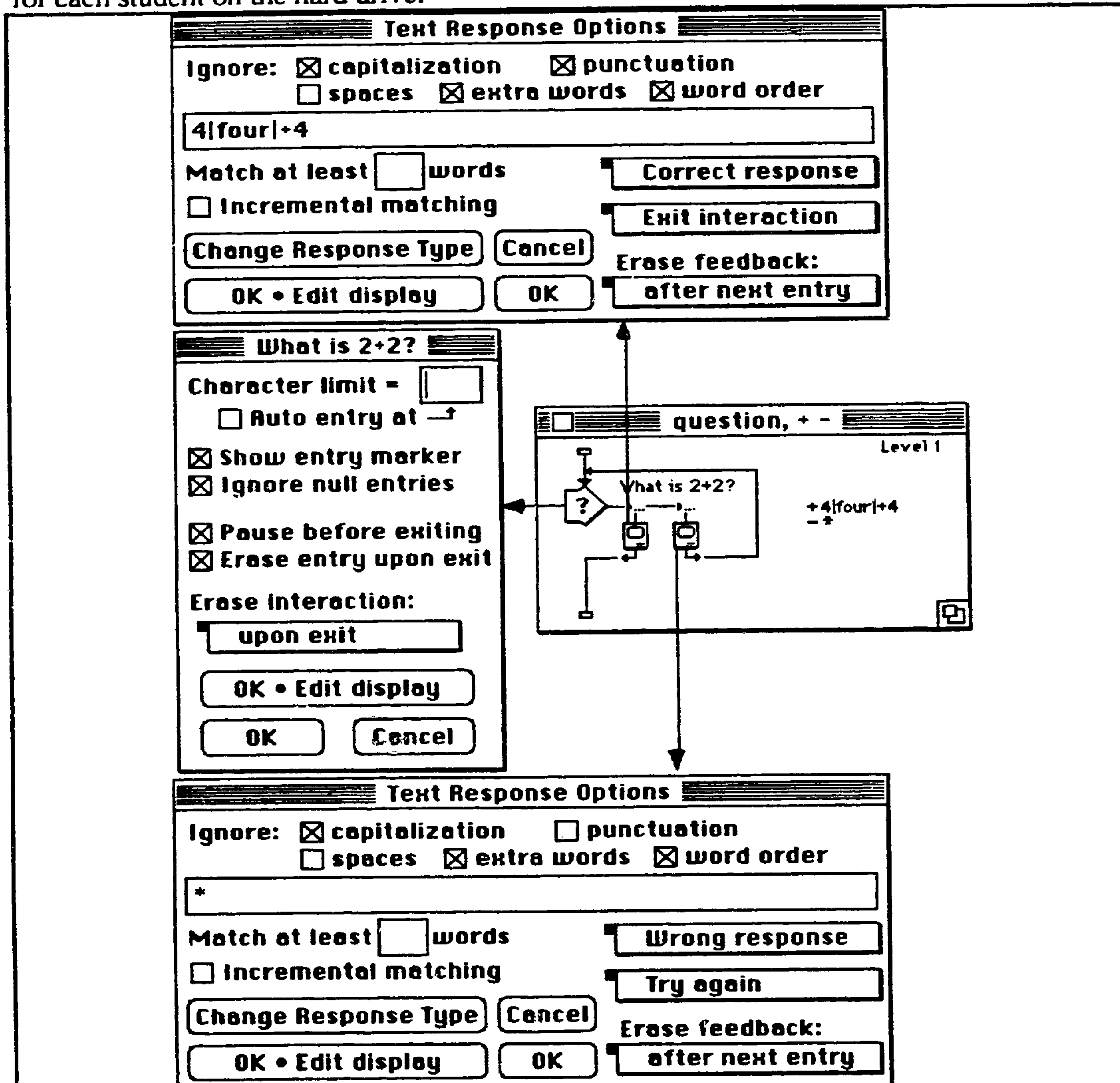
The first study in the AP context was conducted with thirteen adult students in a third-year university course (for training of teachers) called "Introduction to Computer Assisted Instruction". Its intent was to examine students' concepts of AP through responses to questions developed for use in audio taped interviews. These questions were asked during and following student use of the IS. These questions were developed in regard to: 1) demographics and prior computer experience, 2) the computer laboratory context and interaction with other students, 3) displays used in the IS, 4) concept development through the IS, 5) using the table of contents pull-down menu in the IS, 6) conceptual understanding of AP's icons, 7) conceptual understanding of flowcharts in general, 8) conceptual understanding of decision icons in flowcharts, and 9) conceptual understanding of interaction icons in flowcharts an example of which is provided in Figure 1-1. The task of this research study, detailed in Chapter 2 of this text, was to study the learning and comprehending of the concepts inherent in AP. Such an approach is phenomenographic from a second-order perspective in that it attempts to understand the experiences of phenomenon which students express in answers to interviewer questions.

By asking some of these questions of each adult student in the study interesting discussions transpired. Responses to the questions varied. In some cases the students seemed to be encouraged to think about their activities and draw from their experiences in answering the questions. The questions did not require students to restate definitions they had been taught. Rather the questions required that students consider situations in AP which they may or may not have encountered previously. Students extrapolated from their experiences or proposed possible solutions in answering questions. From these discussions it was decided that these questions would be useful as activities for students to explore. These questions were then developed into activities which students could use to assist them in exploring AP (see Appendix A). The instructional package (IP) containing the four IS's interspersed with eight activities was developed with the intent that students could use the IP to explore (and construct) concepts and skills that were implicit or explicit in the IS.

The second and third research studies were conducted with high school students who used the IP. Two existing computer processing classes (not contrived environments) became involved in the studies. These classes were flexible enough that students could agree to participate in use of the IP without undue disruption to their usual pattern of independent study of computer modules (which is customary in many business education classrooms). The use of AP had been taught to the previous year's class. In 1991-92 it was projected that moderate-sized classes would allow for this module to be taught to the new class of students. Within the computer classes students were already learning about computer applications. The computer laboratory setting for these studies provided sufficient hardware for student use. Thirty Macintosh Plus[2] computers with hard drives were connected to local area networks and four file servers. Students learned to save their

2 Macintosh Plus is a registered trade mark of Apple Computer, Inc.

programs to both the hard drives and to their own computer diskettes. AP was provided for each student on the hard drive.



a) For the flowchart above, describe the operation of the interaction icon, the response types chosen, the correct answer, and the incorrect answer(s) tested for.

Figure 1-1. Example of initial interview question used with adult students.

In the second study, details of which appear in Chapter 4 of the text, students began using the IP once they had completed the use of the word processing and computer spreadsheet units in their computer processing class. In this class students began the AP unit after 2 to 3 months of classes, 3 times per week. (In the third study students embarked upon the AP unit after completion of the word processing unit following two weeks of classes also 3 times per week.) Students worked at their own pace through the word processing and spreadsheet units. As a result they embarked on the AP unit at varying start times.

Audio recordings of interactions between students and with the researcher were made while student work was in progress and also during interviews. By transcribing these recordings immediately after they were made it was possible to place researcher observation notes into the transcripts. It soon became apparent that the observer note taking would not be sufficient for long term recording of visual references. AP makes use of a visual interface involving icons and flowcharts. As a result it was often necessary for students to point to the computer screen when discussing the IS and the activities which followed. These visual interactions were necessary for interpretations of the discussions. In response, video recordings were made of student activities and interactions with the computer.

The method of video recording underwent development in order to obtain the desired data. The video method began with recordings of the activities of the entire class, followed by recordings of a small group of students and finally focused in-depth (for about one month) on two students' interactions, one for each of the two cameras used. The video camera was placed behind the student in the position of 'peering' over her/his shoulder. By 'zooming in' on the computer screen with which the student was interacting the procedures she/he used were evident on the tape. Thus, in essence, the outward manifestations of the process of conceptual development were recorded to videotape. By viewing the videotapes and transcribing audio tapes it was possible to review each days events and determine difficulties with the IP activities. The questions and activities underwent changes as these difficulties were encountered. Students who embarked on the AP unit later in the course used the changed version of the materials. Researcher observation, student-created programs, and video and audio recordings of creative activities were used to 'capture' some of the interactions that occurred within the complex environment of the computer classroom. The computer laboratory setting was comfortable enough for students to continue learning activities despite these sometimes intrusive measures.

Following the completion of the IP students created a 'living essay' (their own autonomous creation in computer program form) based upon what they had learned and experienced in the lessons about AP and based upon their perception of the assignment and the possibilities within the computer environment within AP. Through the IP, students were encouraged to form hypotheses about manipulating and performing tasks, to test them, and to interact with the computing principles involved in a creative way. The effect of such experiences on students' manipulation capabilities and computer conceptualizations are assessed qualitatively in Chapter 4 thereby offering the potential for increased understanding of how individual learner computer concepts develop in a student-computer interactive environment.

The data from studies two and three, encompassing transcripts of conversations, descriptions of student activity and files of student work, are extensive. From this vast array of possibilities two chapters have been written. Chapter 3 offers description of high-school students' meanings of their developments of computer based 'living essays' which incorporate text, graphics, sound, animation, and movies (multimedia) in order to tell a story of the student's making. Students' reflections upon the processes of these developments and the meanings of these products to these students are explored through this chapter. Chapter 4 provides descriptions, in case study format, of two students activities in using AP to produce living essays.

The Chaos-Theory Context. Development of the materials and computer programs for use in this context occurred through a series of events that spanned eight months. Based upon initial literature searches and reading, an instructional simulation (IS) was developed which taught the concepts of iteration, computer graphing, and orbit analysis (Devaney, 1990, Falconer, 1990, Mandelbrot, 1982, Milnor, 1978). This file was demonstrated to a group of teachers/university students in a class about issues in mathematics education. Feedback was received about the content presented in the IS and the usefulness of the package. A computer program which demonstrated the process of 'Cobwebbing' was then developed. This computer program was demonstrated to the class whose 'confused' looks and general comments were noted. It was necessary to use the chalkboard to enhance the computer presentation. An extension of this computer program was created which linked the concepts of Cobwebbing (from the first computer program) to computer graphing (from the IS). This computer program became known as the Cobwebber. This program was demonstrated to the class who showed interest in the programs based upon their comments.

The Cobwebber computer program generates 'cobwebs' of iterates of functions so a visual representation of non-linear dynamics (chaotic functions) can be explored. An Excel[3] spreadsheet file can also be manipulated to explore the numerical iterates of a function and graphs of these iterates. These programs are described and analyzed in Chapter 5.

Study four (not reported in this dissertation) involved adult students in a university class entitled "Using computers to teach mathematics". These students were provided the three computer programs (the IS, the Cobwebber and the spreadsheet program). They were introduced to the input of values into the programs and encouraged to explore. The session was to take one hour. However students became involved in a variety of activities and mathematical pursuits which involved their entire class period of three hours. The following class session, students were asked to review what they had done the previous day for twenty minutes and be prepared to report their findings. Although attempts were made to encourage reporting, students did not wish to cease their explorations with the software. After one hour students were required to turn off their computers so they could return to regular class activities. No data was collected during these encounters except for researcher observations. However, based upon these early encounters these exploration programs were considered viable for research activities.

Since these programs were readily available for manipulation by university students it seemed likely that capable high school mathematics students (and mathematics teachers) could explore non-linear dynamic systems as an extension of the high school mathematics curriculum. These programs seemed to encourage the university students to inquire about mathematics, to form hypotheses of the nature of the mathematical principles and to test these hypotheses by manipulating the computer programs. It seemed that through their activities students interacted with the mathematics principles involved in a creative way and generated personal meanings of the mathematics. A method for student-recording their activities was developed in a simple chart with columns for recording input to the computer and space to make observations of findings and to note the reasons for the input entered. This software and recording method were ready to try with high-school students.

In study five a group of high school students (aged 16 or 17 years) chose to be involved in the study as an extension activity to their mathematics class which followed the provincial curriculum. Students attended six one-hour after-school sessions in six weeks in the computer laboratory. Students were self-selected by responding to a group-invitation to participate in an after-school program (on early dismissal days) conducted in the school computer laboratory. Weekly one-hour meetings of this mathematics group were held for six weeks.

This study, reported in Chapter 6, focuses upon learning and generative processes and the ways non-linear dynamics programs are used to produce the type of investigative thinking which is desired in students. In addition, this study addresses the need to understand interactions of students with non-linear dynamics computer programs and the

---

3 Excel is a spreadsheet program produced and distributed by Microsoft, Inc.

implications of these interactions for student learning and thinking. Researcher observation, student recordings of their activities, and video and audio recordings of student activities were used to 'capture' some of the interactions that occurred within the mathematical-computer context. Students responded (in writing) to researcher questions and provided their answers at the end of each meeting. These written responses were sources of information about students' activities.

Studies six and seven (not reported in this dissertation) involved university and high-school students respectively, using the three programs and making records of their activities on the recording sheet. The data from these studies is not provided as it takes new directions. The study with university students primarily discusses use of the computer to teach mathematics, and the high school students study focuses upon curriculum and its development through student activity. These studies, although related to the work presented in this dissertation, is not primarily concerned with human-computer interaction and an enactivist approach to its study. They focus instead on mathematical understanding. For this reason they are not provided here.

Questions Addressed Through Both Contexts

Several questions which elaborate the fundamental research questions were address in the studies. Among the questions addressed were (1) in what activities and interactions had the participating students been engaged with the computer as partner/recipient/stimulus? (2) what styles of interaction emerged? (3) what content (mathematical and computing) had students generated? (4) what curriculum emerged from these activities? (5) what was the interplay between the learner and the context in which they explored? and (6) what meanings did students assign to their interactions with computer programs in attaining development (mathematical and productive)? These questions, specific to each of the contexts, form the basis to address the main questions in the dissertation including the interpretation of human-computer interaction from an enactive stance and educational and research implications of such an approach.

Outline of the Dissertation

The dissertation is composed in a series of inter-related but stand alone essays. Each chapter (essay) can be read independently. They can also be read in groups. For the reader interested in programming issues and students' exploration of an icon-based programming environment then Chapters 2, 3 and 4 would be appropriate. For the reader interested in mathematical meaning making through students' explorations then reading Chapters 5 and 6 would be helpful. Readers interested in the construction of enactive environments may wish to focus upon Chapters 2 and 5. Readers who wish to understand students' experiences and activities of human-computer interaction might focus upon Chapters 4 and 6. Chapters 3 and 6 provide insight for the reader interested in students perceptions of their activities while interacting with the computer. Abstracts of each chapter (essay) are provided here.

Chapter 2 - Icon-based Programming Concepts

Abstract. The problems of conceptualizing and understanding icon-based programming are explored through an interview-based account of students' use of Authorware Professional (AP). Students interact with instructional simulations and with AP (an icon-based programming language) in developing concepts of icons, flowcharts, toolboxes, pull-down menus and AP itself. Student's concepts are explored through interviews and examples are provided of students' discussions in order to characterize concepts formed while using AP. These characterizations are thought to be applicable to the study of other icon-based programming systems. With current interest in icon-based human-computer interactions which are deemed to be more friendly, interest in icon-based programming languages is likely to increase. This study provides direction for further study into this growing field.

Chapter 3 - Students' Transformational Experiences with/in a Computer Environment

Abstract. Students' perceived roles and experiences of producing multimedia stories are discussed in this interview-based account of computer use. Five high-school students participated in interviews to determine their conceptions of producing multimedia essays in Authorware Professional (AP). During one such discussion a student explored her role as an author in developing the computer program. This idea of students role was incorporated into the format of subsequent interviews. Two in-depth discussions and three brief discussions are presented of student perceived roles and students' experiences of the development of the computer files. Students bring forth a view of the computer environment and themselves as generators of the activity. This study is aimed at understanding how a knowledge of computer multimedia as expressive environment develops. It examines modes of computer usage, technical and nontechnical experience of computer technology, and explores the transformational experience of computer technology.

Chapter 4 - Case Studies of Students' Embodied Actions with/in a Computer Environment

Abstract. The processes of development of multimedia computer programs by high school students is traced through two case studies. The focus of these studies is to understand students' development of computer concepts, patterns of interaction with the computer software, and styles of interaction and students' conceptions of the development of story. Twenty students in a second-year high-school computer applications class used an instructional package to learn about the features, functions, and development of programs in Authorware Professional (AP). Students completed the instructional package and then created a computer program referred to as a 'living essay' by the teacher and students. The product development schemes of two students were reflected in work shown on their computer screens which were recorded to videotape. The interactions with the computer, fellow students, the researcher, and the teacher are audible on the videotape. Descriptions and transcripts of these interactions are provided. The questions addressed are: What is the nature of student creations when students are immersed within this computer-enhanced environment? What strategic patterns and techniques do students use in these productions within the context of AP? What insights into computing concepts are gained from these experiences? What is the interaction between the learner and the medium? The study presents an interpretation of human-computer interactions which is articulated and expressed through this study by viewing students activities and interactions.

Chapter 5 - Simultaneous Graphics Presentations in the Study of Non-Linear Dynamics

Abstract. Two computer programs for exploration of ideas inherent in chaos theory and non-linear dynamics are described. Iteration, the logistics equation, iterate graphs, and cobweb graphs are described as content important to the understanding of

chaos theory. The chapter discusses the types of graphics inherent in these computer programs and their values and limitations in regard to developing understanding of the mathematical concepts. The features of each graphic which are considered most salient are discussed. These graphics are then considered together and the value of the simultaneous presentation of these graphics for exploration of mathematical concepts is discussed. Finally the limitations of use of these computer graphics is discussed

Chapter 6 Using Computers To Look At Non-Linear Dynamics: What Do Students Do? What Do Students Think?

Abstract. This study examines the nature of the interactions of high school mathematics students with chaos theory (non-linear dynamics) through explorations and manipulation of computer programs as an extension of the mathematics curriculum. These programs generate: (1) visual representations in the form of 'cobwebs' of iterates of functions and other graphical representations and (2) numerical iterates of functions and graphs of these iterates through a spreadsheet application program. Students are encouraged to inquire about mathematics, to form hypotheses and test them, and to creatively interact with the mathematical principles. Students' mathematical capabilities and experiences are examined offering the potential for increased understanding of how mathematical ideas develop in an interactive computer environment. The study focuses upon learning and the ways non-linear dynamics computer programs can be used to produce the type of investigative thinking which is desired in students. The ultimate aim of this study is to provide an overview of student-computer interaction which lead to student experimentation and learning.

Chapter 7 - Human-Computer Enaction: Implications for Education and Research

Human-computer interaction is discussed from two views of cognition, first from the representationist view and then from the enactivist perspective. Both of the contexts, the Authorware Professional (AP) context and the chaos theory context, are discussed from their origins in the representationist stance. The enactive view is then explored through a discussion of the two contexts. Implications of the enactive stance for education and research are then articulated.

Chapter 2

Icon-based Programming Concepts

From an instructional point of view, two major orientations to computer use by students are of interest to an instructor. One orientation provides for student activities focused upon creation of end products. The second orientation provides for the development of computer related concepts as embodied by the application being used. The orientation toward end products is predominant within training environments. In public school environments the focus is more likely the building of concepts which can form the basis of generalizations which encourage creative activity or application of ideas (transfer) to novel experiences.

Based on the orientation toward concepts, the decision of which concepts to develop within a computer environment becomes important. This decision is likely based upon the computer operating system and software being used. Which of the two would be best introduced first? For example, when introducing students to the use of a computer using a DOS operating system an understanding of the operating system is necessary in order to operate the software so the concepts of the operating system must be established first. However, using an icon based operating system which relies on pointing (for example, the operating system on the Macintosh[1] computer) may allow the operating system to become transparent and thus beginning with understanding the concepts of the software application may be possible. However, it is likely that at some point the user of the icon-based operating system will begin to consider the function of icons. This functional understanding forms the beginning of a conceptual understanding of the operating system.

With the desire for more friendly computer user interfaces (Kitajima, 1989) a heightened interest in operating systems which rely upon icons, pull-down menus and options boxes for control of operations is apparent. A natural extension of this interest is in icon-based programming systems. One form of icon-based programming includes a 'front end' which interfaces the user to an existing programming language to create statements which are then executed. This allows for icon-based interactions with the existing programming language (Gittens, 1986). Another form of icon-based programming is developed 'from scratch' which incorporates an icon-based interface for control of operations and does not produce intermediate code for execution of operations. The code created by interaction with these icons may or may not be accessible to the user of the system. An example of this second type of icon-based programming is Authorware Professional[2] (AP). The motivation for the selection of AP in this study was: availability (Rode & Poirot, 1989), icon understandability (Jones, 1990), and suitability for students and efficiency for learners (Hazen, 1987).

Authorware Professional

AP is an icon based programming language which takes advantage of a graphical interface (available on the Macintosh computer and within the Windows environment on the IBM). According to Goktepe, Ozguc and Baray (1989, p. 169)

> graphical interfaces have now been introduced that represent a significant improvement over traditional command languages as a means of realizing the full potential of a computer. As an example, iconic interfaces present command and

---

1 Macintosh is a registered trade mark of Apple Computer, Inc.

2 Authorware Professional is a produced and distributed by Authorware, Inc., Minneapolis, MN, 1992.

system information in a non-verbal manner. ... It improves user performance while reducing errors.

Lodding (1983) indicates that images can be easily recognized and learned, and thus may be better than their lexical equivalents.

To program in AP one must select icons from a menu and 'drag' them to a 'flowline'. The flowline being the path by which the execution of the icons is accomplished. The flowline and the icons together form a flowchart. The icons on the flowchart are executed one by one in the order of the flowline. Each icon 'stands for' a particular set of operations. To set the operations within the icons in AP choices must be made from the options and toolboxes provided by each icon. Icons in the flowchart can then be 'filled' with information. Text and graphic information, as well as parameters are set either during an interrupt while executing (running) the flowchart or prior to its execution. The programming code (set through parameters) and the information contained in the icons will be executed and displayed based upon the flowchart's definition and sequencing. The flowchart may be edited before, during, or after execution.

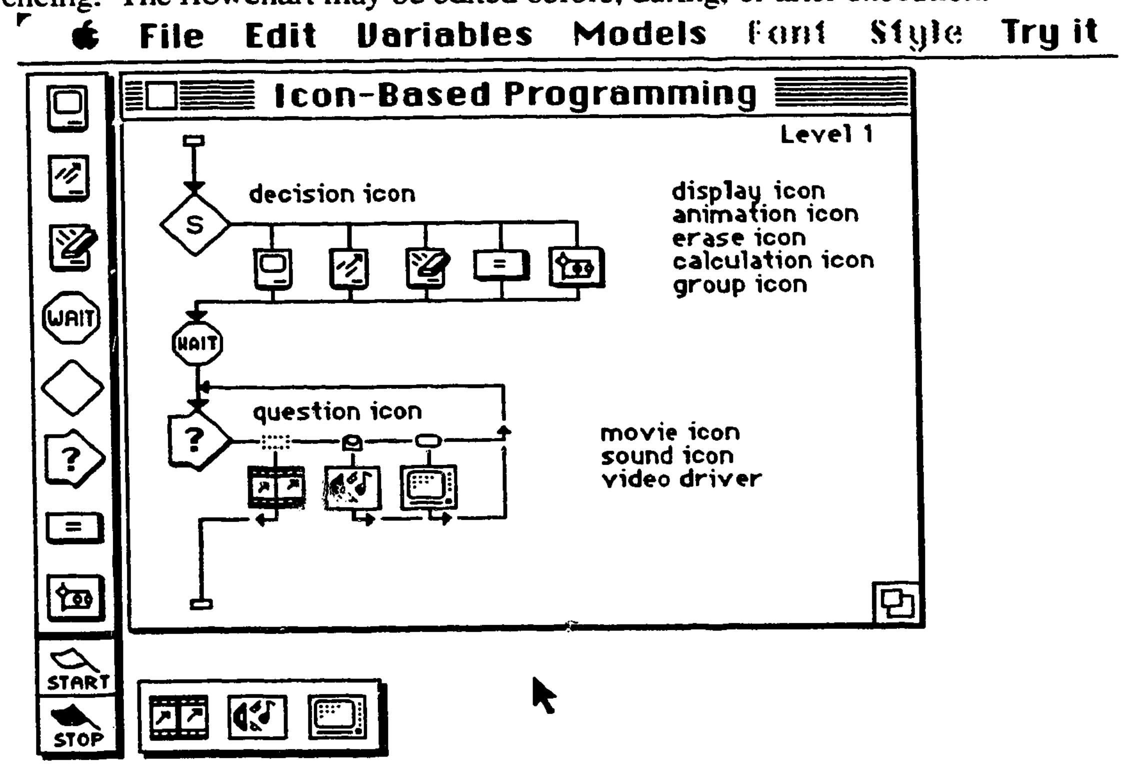


Figure 2-1. The Icons Menu in AP and an example of a File Created using AP.

An example of a program created in AP is provided in Figure 1, in which eight of the icons used in creating a program are displayed vertically in the menu on the left and three additional icons displayed horizontally at the bottom. An understanding of the icons used in AP is necessary if one is to author using AP. Townsend (1986, p. 119) suggests that icons should "clearly depict, indicate, and distinguish a program's commands and operations ... [and] should suggest and indicate a command intention rather than just duplicate or represent a particular pictorial form." AP's icons clearly distinguish between operations fulfilling this expectation of Townsend's. The icons in AP are described in the order they appear in the menu.

1) The display icon operates to create a screen display composed of objects which contain text and/or graphics.
2) The animation icon is used to move the objects within a specified display icon to another region of the screen based on a variable or a previously specified path.
3) The erase icon acts to erase the screen display of any one display icon.
4) The wait icon is used to cause a pause in activity for either a previously specified amount of time (in seconds) or until the user presses a key or clicks the mouse.
5) The decision icon is used to create looping through the icons in sequential or random order or based on a previously specified variable.
6) A question (interaction) icon is used for creating questions or interactions providing transactions with users.
7) A calculation icon is used to place variables and calculations into the flowchart.
8) The group icon is used to group a number of icons together in a sub-level of a flowchart.
9) A movie icon is for importing and playing 'movies' in AP. The 'movies' played are sequences of graphical images which are played in sequence providing animated presentations.
10) The sound icon is used to place sound into the flowchart.
11) The video driver is the icon used to control external devices like videodisc players and videotape players.

This last icon in the menu appears less intuitive than the others. McLean (1983) suggests that icons should leave little room for misinterpretation and should be meaningful. Although the meaning of AP icons appear to be intuitive this needs to be established for users of this programming language.

Jones (1990), in discussing icon usage, states that "the expectations for computer tools on the part of the teachers is that time instructing the children how to use the resource will be sufficient to overcome the complexities inherent in the interface design" (p. 307). He suggests that "a better way might be to give ... potential users of the resource a chance to demonstrate how they conceive of the programs options" (p. 307-308).

Burgess (1991, p. 24) describes difficulties in conceptual understanding and use of computerized systems:

> When a system has an explicit physical reality, as in a traditional (noncomputerized) file system, then the reality gives the user a substantive understanding of the workings and organization of the system itself. With a computer database [program], the physical reality of the data [encoded content] and its organization is masked behind layers of hardware and software where the transformation in form and perhaps content. suffice only to erect a barrier of obfuscation between the user and the data [encoded content].

The content of computer programs may be masked behind the computer programming language used. The structure, function, and organization may not be evident in usual text-based languages. However, the icons, flowchart and menus available in AP, if understood, should provide a framework through which students can produce programs and understand their execution. Norman (1983, p. 7) states that "models provide predictive and explanatory power for understanding the interaction." Mayer (1981) and Murphy and Mitchell (1986) indicate greater efficiency in computer use is possible when users have a realistic model of a computer structure.

> Metaphors have been employed for some time in computing, where end-users have required an analogical model of a system. ... Metaphor relies on attributes of a physical, external environment with which the user is familiar, being directly transferable to the objects in the computer system. (Gittens, 1986, p. 530)

AP uses display, erase and decision icons to provide familiar images of actions or objects in defining its icons. These attempt to make use of user concepts which already exist. An example of this is evident in the use of the erase icon and its image being of an eraser. "The major deficiency of metaphors is that they break down if overextended" (Gittens, 1986, p. 528).

The Research Project

The current study set out to allow subjects an opportunity to express their understanding of the icons, flowchart and interface as used by AP. In order to accomplish this task (a) concepts important to the use of AP were identified and (b) the particular concepts students developed via instructional simulations and through interaction with AP were investigated. The intent of these activities was to provide direction for research in studying user concepts developed when using icon-based programming languages.

Concepts Important to the Use of AP. As the description of AP illustrates, the icons, flowcharts, and pull-down menus in AP compose a formal system used for the definition and creation of programs. In order to program within this system it is necessary to possess some conceptual understanding of the system. Interaction within this structured environment is based upon an understanding of the functions and use of icons placed on flowlines to form flowcharts mediated by pull-down menus, toolboxes, and option choices. In addition to possessing concepts for the functions of each of these structures in AP one must understand what it is that display icons display and what it is that erase icons erase. In other words one must possess the concept of object. The flow of activity through the flowchart is also an important concept which must be understood to use AP. This flow is controlled by hidden or apparent variables which must be understood in order for programmers to control the flow of activity via these variables. Finally, when a user creates a program in AP, what has been created? A concept of program is another concept inherent in the use of AP. Other more task specific concepts and skills are helpful for users of AP but this discussion is concerned with conceptions applicable to the broader context of icon-based programming rather than task specific ones.

Methodology

In order to investigate students' concepts of AP, questions were developed to be used during audio taped interviews. These questions were asked during and following student use of the IS. The questions were as follows.

a) What is an icon in Authorware Professional?
b) What is a flowchart and what is its function?
c) What is the function of the display icon?
d) What is the function of the erase icon?
e) What is the function of a toolbox?
f) What is the function of the pull-down menus?
g) What is Authorware Professional?

During the initial phase of this project the concepts necessary to understand AP were defined (outlined above). Based on these concepts four instructional simulations (IS) were developed to introduce students to AP and to introduce the concepts. The IS were developed in a style that has been described as "like watching over the shoulder of an experienced programmer using AP and the programmer describes what is being done". These IS demonstrated techniques for creating a new courseware file in AP, for placing icons on the flowline and for the use of pull-down menus and option choices for defining operations and parameterizing of icons.

Palmiter and Elkerton (1991) state that "an animated demonstration shows the user how the interface appears as the procedure is executed and also links the input actions with the interface results. Users ... experience ... how each procedural step contributes to the overall task goal" (p. 689). This is important if students are to understand their interactions with the software and their later experiences. However, one possible drawback to this approach is the possibility that students will merely mimic the procedures without understanding them (Lefevbre & Dixon, 1986).

Subjects

Thirteen students in a third year university course (for training of teachers) called "Introduction to Computer Assisted Instruction" used the four IS during the laboratory component of this course in a laboratory using Macintosh computers connected to a local area network. During student use of the IS and following completion of the IS six of the thirteen students in the course were interviewed to determine their conceptions of AP.

Results

Students were asked to respond to a set of questions to determine their conceptual understandings of AP. When students were asked to respond to the question "What is an icon in Authorware Professional?" three ways of describing the icons were used by students. Students described an icon as:

1) "It **represents a command**."
"It's just a representation of ... computer action."
"It is the foundation of a process or a procedure."
"They are symbols and pictures that stand for various functions that they perform."
2) "The tools I would use as an author to create CAI."
"It will allow you **to write a part of a program**."
"They're steps to develop a program. ... They are there to set up your program."
3) "An icon is a tool that you can use to **execute a function**."

To students, these icons in AP symbolize, represent or provide a picture of the commands, computer actions or functions within the programs written in AP. They are tools for the creations or writing of a program or computer assisted instruction lesson. They may also serve the purpose of allowing the execution of the functions, computer actions or commands (program code) they represent. They appear then to 'stand for' a process or to be a step or part of the set up of a program for the creation and execution of computer assisted instruction (CAI).

Similar concepts are reflected in students' discussions of "What is a flowchart?" Students indicated that flowcharts served five purposes: For the 1) design, 2) representation of, or 3) execution of the program or code and to represent 4) the author's ideas, or 5) the content of the program or code.

1) Design
"To me a flowchart is a series of steps that you take to design your program."

2) Execute
A flowchart is "a path an author and student will take through the program."
"The function of the flowchart is to execute your program. Show how your program runs."
"It makes the program work."
"For instance here (points to the display icon in his flowchart) I'm going to display something, I'm going to wait (indicates wait icon), display something else and then erase it."

"Well it starts in the first icon and goes through it. It executes what I have put, what steps I have put in that icon. It goes to the next icon and completes the steps in that icon."
"How your program will run. It's the line of execution."
"The flowchart is the steps your program goes through."

3) Represents Code
"That is a series of commands and steps that the program itself is going to take, steps the course is going to take."
"A flowchart is just to show like a schemata. A view of the program. Not the particulars necessarily but an overview of it."
"Just the order of what will be displayed or where the questions go. Its just an easy representation. You look at it and you can see what point in the program you are. Its just kind of an overview of the program."
"Well rather than have me write up my program in Fortran or whatever. It's just a visual way of showing what I've done, what steps I've taken."

4) Represents the Author's Ideas
"It basically tells you, not at a real concrete level, displays what you are doing. What your processes of thought are. What your big picture is without going into detail."

5) Represents Content
"The flowchart is the steps your program goes through. ... They allowed you to put information in on those steps. They allow you to program in the icon."

The flowchart seems to be a procedure or a sequence to some students while it is a schemata or a view of a procedure to others. It functions for the design, creation and execution of the program. However, it also represents the authors ideas and schemata and may serve to represent the content and instructional strategies inherent in the program. It may represent a generalization of the instructional procedures through a schemata.

The flowchart is composed of different parts which are salient to different students. Some students see the icons as the key to the flowchart. Others see the interconnectedness of the icons as important. Still others seem to imply the importance of the flowline as the line of execution.

While discussing the question "What is the function of the display icon?" students appeared to have either 1) purposive concepts of the display icon "to give information to the learner" or a 2) functional concept of the display icon:

"It allows you to put in graphics, text, whatever you want to do onto the screen."
"[It displays] anything you want to put up on the screen."
"The display icon is the display of the screen."

The mediating effect of instruction is indicated by students' words. When one student was asked about the function of the display icon she responded by indicating that the display icon "Displays." When asked what it displayed she responded "Text". When asked if she had completed the instructional simulations (IS) she indicated she had not due to illness. Another student who had used the IS and had just begun using AP for a laboratory assignment indicated that "as far as my understanding is so far, based on these [IS] programs, it is in these icons that you put your instructions on the screen." Both students had used AP to begin the laboratory assignment but the second student had used the CAI lessons while the first had not. The second seemed to have a greater grasp of the use of the display icon compared to the student who had not used the IS. These students' responses may indicate the need to further study the kind of instruction helpful for student development of concepts.

When asked "What is the function of the erase icon?" students indicated that it "is an eraser". The real question became what does the erase icon erase? All students indicated that it erased contents of the screen. Which contents? For one it "allows the first screen to be erased so we can go on" while for another "It erases the last screen that you have displayed" and yet for a third it erases "your previous step. ... The previous item on the screen." In each of these cases students were aware that the entire screen display was composed of parts, each part created by one display icon. Only the contents of one display icon is erased by one erase icon. In fact the screen display erased is that which has been selected in the options for the erase icon. These students however seemed to be confused on this point. They assumed it either erased the objects of the first display icon in the program, of the last display icon in the program, or of the one just before the erase icon was introduced into the flowchart. The conception that the choice was automatic is actually not functional in the program. Another student expressed a more functional concept when she stated "It erases whatever display icons you select."

To one student erase icons "erase whatever you want them to erase whether the whole screen or just parts of it." This student seems aware of the fact that the screen can contain images and text from more than one display icon. The contents of this display icon can be erased separately. However only one of these can be erased with one erase icon. A competing notion that seems to confuse the issue is that one display icon may contain multiple parts. These parts must all be erased by one erase icon. Parts of the image from one display icon cannot be left on the screen.

Two competing notions of displays seem to operate for students. The first is that only one display icon can be erased by one erase icon. However multiple images may be evident within this one display icon. The second notion is that a screen may be composed of multiple images that have each been generated within one display icon. They may be the product of multiple display icons. Thus they must be erased by multiple erase icons. One student stated that "The erase is used if you put in instructions here and if you don't want it to be carried through it [the erase icon] stops it." Another way of expressing the same concept is "If this [erase] icon is not there the information would remain on the screen." Through these statements students were aware of the 'buildup' of the screen contents from many display icons. This conception of 'adding on' is common in other programming languages other than AP. To remove any contents of the screen one must provide an erase icon. Actually, as students progress further in their use of AP they will learn that other means of erasing the screen, whether the entire screen or just parts of it, are possible.

An interesting discussion was extended from the discussion of erase icons and what it is that they erase.

student:	Erase icons erase the screens that you create .... Well they erase objects is what they do.
researcher:	What is an object?
student:	An object is a display that you have created. ... A graphic.
researcher:	How is that graphic then defined? Or how do you know when something is a graphic.
student:	uh ...
researcher:	If you have a bunch of text there how do I know that that's a graphic that can be erased?
student:	The graphic is something that is done as an object and text is something that is done as text. I don't know how to explain the difference. I know that there is a difference in something that you've created as a graphic and some other stuff that's text.
researcher:	Does an erase icon only act on graphics?
student:	I don't know I haven't tried it on text. But you are not in something like a word processor here are you?
researcher:	No.

student: When you are creating new screens what you are creating is a graphic representation of something. Aren't you?
researcher: Are you?
student: Yes.
researcher: O.K.
student: If I use the text on here, like when I type something on here, it is actually not like I'm typing on a word processor. It's like something is a graphic representation so I can erase it.

This student appears to be grappling with the question of object. She seems aware that text in a word processor is different from text in AP because here it is stored as an object. She uses the term graphic for her description. This may be because she is familiar with drawing packages which indicate graphics by displaying bounds around the graphic. Text has these same bounds when typing in AP (Figure 2-2).

Text in AP appears bounded

Figure 2-2. Text in AP is bounded (when selected) by black indicators on each corner.

AP makes use of a graphical interface for creation of both graphics and text. This may also lead this student to discuss text as a graphic. This graphical interface is facilitated by a toolbox (Figure 2-3). Through this toolbox a user of AP can select objects, create text, lines (vertical, horizontal, and at 45° angles), ovals, rectangles, and free lines (oriented in any direction).

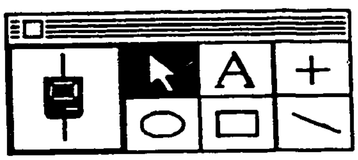


Figure 2-3. The toolbox used in AP for the creation of text, lines, ovals, and rectangles while in the display icon or the question icon.

What this student seems yet to realize is that the underlying difference between text in a word processor and text in a graphic package is the way they are encoded. In the word processor text is encoded as a string of characters with format statements. The graphics package encodes text as a series of black and white dots each maintained in the same orientation with the others. Text is defined with orientation and can be acted upon as a single entity.

"What is the function of the toolbox?" Students indicated that the purpose of this toolbox (Figure 2-3) was for the creation of a screen display. The toolbox simply provided the tools for this creation.

"It just provides you with all the tools you are going to need to create your screen: fonts, the line [tools], arrows and stuff."
"It allows you to pick which character or which process you want to activate on the screen."
"They let you do different things. If you put text in then you can use the toolbox to move it around. ... Highlighting it."

Another student simply described the toolbox from memory and the contents of the toolbox display and the purpose of each of the functions displayed:

"You can write with it. You can change the styles, patterns, etc. etc. There was the arrow, to replace the cursor back on. 'A' is for writing. What was after the 'A'? (thinks a bit) Cross lines. No. Just a line for drawing horizontal or vertical lines.

Then there is the oval, your square or rectangle, and crossbars. Oh the angle. You could draw at your own angle. That was for drawing shapes."

A student stated that the function of the toolbox "depends on what icon you are in as to what it does." With this statement the student shows awareness that different toolboxes may appear in different icons. However, this student continues with a general statement of the function of the toolbox: "It's all the things you can use and the things you can do in this icon."

Discussion of toolbox is based on the toolbox present in the display icon. It is seen as functional for the purpose of creating on the screen. It may be seen as a list of functions or a set of tools. During one discussion with the student who had explored ideas of object this student began exploring ideas of what is controlling the action: the software or the operating system. This discussion follows.

student:	The toolbox allows you to use **special features** like styles of fonts or doing edits or variable things that you want to. I guess they are tools of the **machine**. No of the **software**.
researcher:	Of Authorware [AP]?
student:	I think they are tools of the Macintosh. Just a second now. If you use fonts and styles. No I guess they are tools of Authorware [AP].
researcher:	This toolbox here (points to screen), this display toolbox is a product of ...?
student:	That would be one of Authorware [AP].
researcher:	But when you were talking about fonts and styles and things those are in **pull-down menus**. What are the function of those? Are they different from the toolbox?
student:	The fonts and styles and things like that are from the system, from the software. I'm trying to think in IBM. The fonts and styles and stuff like that are from the software that you are using.
researcher:	How about 'variables', 'models' and 'try it'?
student:	Try it is from Authorware[AP]. That's a part of the Authorware [AP] thing. The edit and the variables seem like they could be from Authorware [AP] too.
researcher:	And what do those things in pull-down menus do. Just a general idea of the things they might do.
student:	Edit allows you to do various things with the work you are doing like cut, copy, paste and well I guess you call it an editing tool. Variables allow you to change the things that you have done or present different views that you have done. Don't they? Just wait (she pulls down the variables menu). Oh yeah (points to the display options in this menu). These variables would be part of the software. I guess they would be part of the program showing functions.

This discussion shows a student attempting to understand the difference in components, options and control of the operating system in the machine and the operation of the application. She is also comparing the differences between the operating systems in the Macintosh environment and IBM environment with which she is familiar. In contrast to the findings of Mayes, Draper, McGregor and Oatley (1988) this student was able to recall the names of the menu items that she used and then able to think about the implication of how these functions were provided by software or the operating system. This indicates that this user has some problems in deciding what is provided by the operating system and what is provided by AP. She seems to be attempting to understand conceptually the interaction of AP with the operating system. Although she has not yet fully conceptualized this relationship she is making the attempt.

"What is the function of the pull-down menus?" Students responded that they "give you options", "more choices" or "more functions". Another responded "They are added in functions." All of these seem to define a similar concept: the pull-down menus are a set of additional options, choices or functions the purpose of which is to define icons and make them operate in different ways. However, as to when pull-down menus operate and on what they operate, two students responses are noteworthy:

researcher: What do they operate on?
student1: Whatever you are working on. ... If there is nothing on the screen and there is nothing to change then the menu will be grey. **If there is something that you can change or an option that you can do then it will be black.**

researcher: With the toolbox you told me that it was specific to the kind of icon you were in. How about the pull-down menus?
student2: Well obviously there are some functions, **the ones that are highlighted are the ones you can use in that particular mode.** There are some like lines, effects, fills that you can use in your display icons.

These students are responding to the pull-down menus question and have obviously noticed that the menus are options that are not always available. The options in the pull-down menus are black when available and grey when they are not available. However, students did not indicate why some options were available at some times while others were not. This means that these options are specific to particular tasks and context but what these contexts are to these students is not clear.

On the question of "What is Authorware Professional?" the students responded that it "is a program", "a software package" or a "software program". They also all indicated their perceived function of this program. The students were consistent in their descriptions of AP as being for "creation". As to what it is that AP creates students responded that it was used "to make courseware" for "creating programs" or for "creating your own tutorial or computer assisted instruction".

Discussion

Students' interviews were based on their conceptions of AP. These conceptions were based on the IS and students experiences interacting with AP. From the student interviews it was clear that some students had developed understanding of the structures and functions in AP. However, not all had developed appropriate concepts. Students had also begun to build a logical structure in which concepts of icon are related to flowchart, to pull-down menu and to toolbox. The concept of object was related to both display and erase icons. Kember (1991) indicates that students not only need an understanding of important concepts but they need to be able to relate one concept to another. Students had begun to build a structure of related concepts.

To master the use of AP requires the utilization of conceptual structures that are complex (perhaps as complex as those required for natural language) but fortunately having finite boundaries. These boundaries makes the task of instruction about the use of AP a manageable one. The question becomes one of how to encourage novice users of AP to build appropriate concepts. Is the making of mere statements of these concepts enough to encourage students? Is the use of a simulation to explain and demonstrate these concepts sufficient? Or is experience with the environment and it's structures necessary to develop these concepts? What is the sequence by which these various concepts develop? Is there a particular pattern of events and interactions which will allow students to develop concepts in an efficient sequence? What is the sequence of concepts of which students should have experiences in order that these concepts will be built?

Another issue that must be considered is students past experiences in using other programming languages, operating systems, and user interfaces and other types of computer experience which may affect students use of AP. Another variable which may affect this experience is age. For the students in this study this was not an issue. However, when considering school students these variables may be important. AP is not aimed at school students but new icon-based languages may be aimed at this audience. If schools wish to take advantage of these languages then age and experience variables should be considered in studying their use.

An issue in conducting this type of research is the language used to talk about icon-based programming. Each of the icons has a name. The name implies a meaning but does the meaning of the name fully convey the meaning of the icon or does the icon's meaning go beyond the words? For example, the fifth icon in the menu (see Figure 2-1) is an icon in AP which is called a decision icon. With experience this icon becomes associated with loop structures which are controlled by user or system variables. Does the name 'decision' fully convey this meaning? Is it appropriate to introduce and make use of the language in the AP manuals or would language developed by the users and shared within user groups be more appropriate to their understanding of the concepts? (Jones, 1990)

These many questions flow from this study of students using AP. Many more have arisen during later work with this same material. Further study into the entire question of icon-based programming must be undertaken before such systems can be used to encourage the development of generalized concepts rather than mere production tasks. If generalized concepts can be developed within students then these concepts may be applicable to the ever-changing operating systems and programming languages available. It is hoped that this study will provide some framework and direction for further research.

Commentary

Chapter 2 has explored student conceptions of the interface provided through the Authorware Professional (AP) environment. Through this study questions have emerged: In encouraging students to build concepts is experience with the computer environment containing AP helpful? How could be achieved? Given experience with the environment do students merely build static concepts or do their concepts change given further experiences?

This study has surveyed students' statements about the AP environment and provided insight into their perceptions of the features of AP including flowchart, pull-down menus, toolboxes, and icons. Students' statements provided insight into their opinions. If, however, we wish to explore their attitudes we need to consider the behavioral component. In this instance it is not only important to know what it is students say about the erase icon but how they use and treat the erase icon thus providing more insight into their perceptions. Classifying students' answers to questions provided a very incomplete picture. Students should NOT be expected to employ any single interpretation of programming objects, flow-of-activity, or programs. Even (or especially) programming experts may vary their point-of-view according to the task at hand. Thus, to get a more reasonable picture of students perceptions, more must be done than to ask questions. Students should be watched to see them using the system and performing a small variety of tasks. At least then the observer can have some sense of what students do with the icons given the task at hand. By observing students over a longer period of time it may be possible to get a sense of changing perspective. Even better, by providing students with a setting in which they vary the task at hand and observing how they interpret it a sense of how they perceive the task and setting is possible.

From this question-and-answer interview-study it was possible to conclude that students developed different concepts while engaged in learning from the same instructional simulations and while completing the same assignments. What was the reason for these differing views? It appeared that students' actions within the computer environment and students' histories may provide explanations. Students' actions and history need further study. Thus this study was not enactive as it did not consider computer work as a function of students' embodied actions. To do this would require consideration not only of students' perceptions but also their activities in attaining these perceptions.

Student perceptions of computing tasks, whether teacher or student defined, are important to consider. If students believe the task is to mimic an example shown to them then their actions and perceptions may be limited to a set predefined by the example. Students placed in situations which occasion their acting in particular ways may act accordingly. The students response to the situation may help us see what are the students' perceptions of the situations. Chapters 3 and 4 respond to this need to explore students' perceptions by providing insight into students' perceptions of their development of a 'living essay' in Chapter 3 and two case studies of student developments of living essays in Chapter 4.

Chapter 3

Students' Transformational Experiences with/in a Computer Environment

Computerized authoring tools have placed users into expressive environments through which generative productions and creative expressions of user ideas are possible. Users interpret their interactions and experiences within the computer-enhanced environments. By taking action in these environments it is possible for users to generate multimedia productions expressing their ideas. Through these generative actions students bring forth a view of the computer environment and themselves as generators of the activity.

This chapter offers a description of high-school students' meanings of their developments of computer based 'living essays' which incorporate text, graphics, sound, animation, and movies (multimedia) in order to tell a story of the students' making. Students' reflections upon the processes of these developments and the meanings of these products to these students are explored through this chapter. What are students' experiences of computer technology when they are immersed in computer multimedia environments over which they have some control? The present study is aimed at understanding how a knowledge of computer multimedia as expressive environment develops. This chapter does not attempt to describe how students develop these computer products as this is left as a task for another paper. Rather the focus is the students' experiences of the development of these products as they relate to the students' perceptions of the task and the emergence of the products based upon these changing perceptions.

Students within the Setting

Sixteen grade eleven students (16 or 17 years of age) in a large urban high school in a large urban center in western Canada participated in the study. Students had completed a previous computer processing class which emphasized applications during their tenth year of schooling. This previous computer class had used IBM[1] computers, so this class, meeting three times per week, was the first introduction to the Macintosh[2] computer for these students. The computer processing class curriculum outlined that students were to first be introduced to a word processor and then begin using a graphic package (Superpaint[3]) and an icon-based programming language (Authorware Professional). During the first two weeks of class (meeting three times per week, one-hour each class) students were introduced to the Macintosh and used a word processor to complete a typing assignment. The researcher was an observer for some of these classes. Following these activities students used Authorware Professional (AP) for several months (time dependent upon the students).

In an introductory activity for the AP unit of study students were asked to open up an existing graphic image and make drawing changes to it. They were asked to save their work. On the subsequent day students were asked to open AP and 'load in' their graphic image following the teacher's specific verbal instructions. Some students were successful at the first try but others had 'lost' their graphics and needed help in finding another to load or in opening their existing graphic file. The teacher became busy helping these few students who needed extra assistance. The researcher responded to the situation by providing specific directions to the other students in order that they could animate their particular graphics. This computerized animation sequence moved the graphic image

1 The IBM is a product of International Business Machines.
2 The Macintosh is a product of Apple Canada.
3 Superpaint is a product of Aldus, Inc.

across the screen when the program was executed. By the end of their second computing class each student had a functioning graphic animation.

How could students accomplish such a complicated computing task so quickly? AP is an icon-based programming language which takes advantage of a graphical interface available on the Macintosh computer (and also within the Windows environment on the IBM). To program in AP one must select icons from a menu and 'drag' them to a 'flowline', the flowline being the path by which the execution of the icons is accomplished. In this case students 'dragged in' two icons: 1) a display icon to contain the graphic and 2) an animation icon to move the graphic and placed them on the flowline. The flowline and the icons together form a flowchart. The icons on the flowchart are executed one by one in the order of the flowline. For these students their graphic containing display icon was placed first so the graphic would appear followed by the animation (placed second) which moved the graphic. Each icon 'stands for' a particular set of operations. To set the operations within the icons in AP, choices must be made from the options and toolboxes provided by each icon. Students set the options for the animation icon by moving the graphic across the screen. This activity told the icon which graphic to move and how long it should take to move it. Students explored changing the time setting so that the graphic would be moved in different number of seconds (typically 3, 5 and 10 seconds). Icons in the flowchart can be 'filled' with text and graphic information, as well as parameter settings either during an interrupt while executing (running) the flowchart or prior to its execution. For these students the information was set prior to execution. The programming code (set through parameters) and the information contained in the icons is executed and displayed based upon the flowchart's definition and sequencing. The flowchart may be edited before, during, or after execution.

Research Procedures

Two students' interactions with the computer were recorded to videotape when they began using AP. A video camera was placed behind each student. It was positioned to peer over the student's shoulder in order to enable a clear view of the computer screen. In front of each student was a microphone which recorded the student's voice and voices of those who interacted with him/her as well as any sounds which were the output of the computer. Students' responses to their own work could be heard in words and voices expressing pleasure and frustration. Often the researcher and classmates could be heard interacting with the students on the tapes. Questions each student asked and feedback received could be distinguished from the sounds in the computer laboratory. A visual sense of the students' work and progress could be gained by watching the videotapes. Each students' work was recorded over a three month period during which they completed assignments for class. This involved collecting 25 to 35 hours of videotape for each student. Annotated transcripts of these videotapes served as "data" for this study. Additional sources of data were provided by researcher notes, students computer programs, and transcripts of audio-taped researcher/student interactions. The main source of data presented in this chapter were audio-taped conversations and interviews between the researcher and students.

Class assignments involved creating computer programs using AP. These programs were created using icons (for which the computer code was defined and) filled with text and graphics, collected together into a flowchart which was executed as the program. A series of assignments and instructional materials were used to provide students an opportunity to learn AP but were also open ended enough to encourage students to create their own productions within the confines of the assignments. For example students were taught how one might use a question icon and attached answers to produce a multiple choice question. The students were then asked to use the question (interaction) icon and attach appropriate answer icons in order to produce an interaction which allowed users to click on a predefined region of the screen and the users choice would be acted upon. No specification of the format or content of the interaction was provided. Students in this context were required to set these parameters for themselves.

Anne - 'A director of a TV show'

Anne was one of the students whose activities were recorded to videotape. Anne used the multiple choice format to provide a series of questions embedded inside questions. She produced a format by which users made choices and based upon these choices they chose a particular path through her series of questions. This format had not been specified in the assignments but Anne had constructed this program as her assignment. She was deemed to be articulate and task-oriented by the teacher, thus her involvement in the study was requested by the researcher.

Anne was one of the students whose graphic was not available during the introductory activity. Anne had drawn her graphic and had saved it to the hard drive on her computer. However, when she returned the next day she found that someone had removed her graphic from the hard drive. She had not made a copy of this graphic on her diskette so she did not have a graphic. To begin working on the animation sequence she drew a simple graphic in AP. The flowchart for the sequence consisted of a display icon containing a graphic followed by an animation icon which had parameters set to move the graphic across the screen. In Anne's case her flowchart had two display icons each containing graphics and an animation icon. This was because her sequence involved a person entering a restaurant. First the restaurant appeared. Then the person appeared next to the restaurant. The animation moved the person from beside the restaurant to the doorway of the restaurant and then through it. Anne laughed about the simplistic graphics she had created and the simple animation but nonetheless seemed proud of her accomplishment as she showed it to another student sitting next to her.

Anne used the instructional materials provided by the researcher and then completed assignments for class. For Anne's final few assignments, instead of creating each assignment as a separate computer program she incorporated the new programming sequences into the existing program. As a result the activities of animation, drag object, and graphics import all were incorporated into one program. This program eventually emerged as Anne's major course assignment which had been termed 'living essay' by Anne's teacher. Part way through the creation of the living essay Anne was asked if she would talk about her activities. She agreed to respond to researcher questions. This conversation was held in a quiet conference room near the school office. Near the end of this rather free-flow conversation about Anne's work in AP she was asked to describe her work. She provided a description which the researcher found interesting, if not telling. From this discussion a new research direction and research question emerged.

The researcher asked "If you were to tell somebody else in another class what you are doing and what kind of things you are creating what would you tell them?"

Anne responded "I already tell my family all that."

"Oh, and what do you tell them?"

"I tell them that we are learning how to use this program that will enable use to draw. It allows us to do lots of things. You can make--we can make a test on the computer and it can like mark it itself by using the question icon. You can make cartoons on it -- 'cause you can. That's what you can really do is make a lot of cartoons and stuff on it. But you can also do calculations and things like that. But what I tell them I'm doing is making--animating things--and allowing to put things in and other things. It's sort of like a video game in a way I guess."

"For you as the creator it's like a video game or for someone else using your stuff it's like a video game."

"Oh both!! Because I like to go back and do it myself. But for other people that go into it it would be like a video game."

"How is it like a video game?"

"Well in the aspect that **I** [emphatically] am able to choose what I want to do and I can take things and put them wherever I want to put them. In a video game you can move your man just the same as I can move the fish or the bear [in my story]. Things like that."

"I never thought of it being like a video game. That's a really nice way of describing it isn't it. I never thought of it that way."

From this early conversation with Anne a question emerged for the researcher. Anne views her creation (program) like a video game. What is this program like to the student? Who is the student as the creator? If this program being created is like a video game to Anne then perhaps she is acting as a programmer or an artistic designer as she creates her program? The researcher reflected upon the conversation with Anne and decided that these questions should be pursued.

In a later conversation with Anne she indicated that she had shown her program to fellow classmates. She said that she enjoyed getting the reactions of these students since their responses told her "if her expectations were where they should be." She stated that "I found out whether I should change it or what I should do." From Anne's preliminary investigations (perhaps even formative evaluations) of her living essay with other students she learned that sometimes animations have to "move quicker because you don't really want people to lose interest in it, especially when there's more." Anne's living essay created a story of a bear living in Yellowstone Park which endured many trials and tribulations of bear life. In one of her sequences Bob, portrayed as a cruel hunter, shot at the bear.

Anne described her program; "It was just like watching a show on TV except there were no words or sounds. I could have done that but I didn't have enough time."

She was asked by the researcher "So when you were creating this living essay was it like making a story or a show on TV?"

"Yeah and I was the director." When asked to elaborate on being a director she described her activities and her response. "Well, it is good because I'm working by myself so I'm making the changes by myself. I'm creating the ideas. If there was a person helping me we might not have agreed on certain things and in that way it might not have turned out. I mean if there's two people you have to please both of them but this way it's just me - that way if I get reaction from other people then I can change it myself to what way it should be. I don't know if that makes too much sense.

"I'm not sure what you're saying."

"Well, by doing it by myself, right, I was able to put my ideas on it - it sounds kind of selfish but with two people, like me and Susan, she might have had some different ideas, that way I probably....suppose we used the bear and she wanted it to go to Yellowstone Park, maybe she wanted him [the bear] chained up somewhere, that way I wouldn't have been able to add those things in like feeding the fish and bringing in all the other characters that I did."

"Uh huh, so you had control?"

"Yeah."

"Okay, so you were able to put in your own story rather than somebody else's story?"

"Yeah, like if I would have done something on the Olympics [as originally outlined in the description of the assignment], like, I would just be repeating news but I wanted to be creative and doing this bear, that was original."

For Anne the living essay is a video game for those who view it, including herself. She has assessed its quality by showing it to other students. Despite this assessment by others she feels ownership of her work and pride in her accomplishments.

Carson - 'A programmer creating a video game program'

Carson was one of Anne's classmates. It was apparent from his first drawings in the graphics program that he had an 'artistic flair'. During the introductory activities for the unit on AP Carson drew a realistic looking rocket and animated it across the screen as it was taking off. He had positioned this initial graphic on the screen so that it appeared to be resting on the bottom of the screen. When he animated the graphic a fiery flame extended from its base as it moved upward. In addition to being in the computer processing class which used applications, Carson was enrolled in the computer programming class at the grade eleven level. In this programming class Carson had demonstrated to his teacher a developing programming ability. His teacher recommended him as a capable student whose visual work might be interesting to observe.

In his computer processing class, as one of his in class assignments, Carson had created a multiple choice question which asked which video game system was the best. Four possible choices were listed; "Genesis"[4] was considered the right answer. When asked about this Carson stated that he had this system at home so he made this the correct answer. Carson was asked to describe his current program, another of the in class assignments, in which he was to create a program which asked the user of the program to drag an object to a prespecified location. The object to be moved and the location of the final movement were up to Carson to create. He described his program: "You are supposed to drag the prince (the hero) up the mountain to save the princess. And then a dragon comes out and this guy is supposed to kill the dragon and they live happily ever afterwards." This project was to become Carson's living essay. Carson's description of his project and his interest in video games led the researcher to ask if the program he was creating was similar to the electronic games he played. Carson responded that "I got some of my ideas like the knight and the mountain move from a game I played." However he said that the story of the knight and the princess was mainly his own creation. In Carson's game many graphics were provided. These graphics faded in and out, sometimes appearing to interact with each other because of the fading. He was asked if these procedures were also present in games he had seen. Carson responded "I don't think so. I think it just looks nicer if I put it in fade in and stuff instead of just nothing."

Carson had been observed demonstrating his projects to friends from outside this class during lunch hour. "I ask them how to draw these pictures, how to do shadings, and stuff like that. More like drawings and things." It seems that he uses these friends as advisors for his graphics generation. When asked what kinds of things he was making in AP Carson responded that "Authorware [Professional] is too slow for making good games. Average games yeah -- it's good. Just basic games." He was asked if the story he was making could be a game? He responded "yeah." He described the object of this game: "To save the princess and kill the dragon." Since Carson had not yet created this part of the program he was asked to describe it: "Dragging the prince out of nowhere and then the princess tells the guy that the dragon's weakness is the neck or something so then I'll get the guy to throw, to drag, get user to drag the sword to the neck of the dragon and he dies." Carson was asked what he would tell his family about his school project. He stated "I would say making stories, drawing pictures, and stuff. I will probably print out hard copies so I can show them what it looks like."

Where did the ideas for this living essay come from? "From games I played and from stories I read. Fantasy novels. There is always a hero saving a country, saving a princess, fighting off evil, and all that stuff." From where did the idea of the dragon emerge? "Well I think it's from a game I played. Well my whole story is based on that game I think." He described the game he had seen as one of pure commands and no action on the part of the user. He said that he "liked the story." But is Carson's story like this game? "No. ... The style is totally different. ... My story is pretty straightforward I think. You just have them drag the person up the mountain and drag the sword into the dragon's mouth. I think even a child could do it." He describes his story as suitable for use by

---

4 Genesis is a video game system produced and distributed by Sega Systems, Irwin Toys, Inc., Toronto.

children "if there were more things to do. Mine just has two things to do and the rest are just like reading the screen for what's going on."

It seems that Carson believes in the need for interaction as an important aspect of computer design. This is an advanced understanding for someone so new to the design of computer programs. But Carson complained about the style of his living essay, describing it as repetitive. To eliminate these problems he would "learn some new techniques first so I can pick from a wider variety of techniques. If I am to do a new one I would put, make it look more animated like those games I have." He would incorporate "smoother animations instead of just zooming in and zooming out, zooming in. I want to make it look like it actually moves. Like first I move my left foot and then right foot, left foot, right foot. If it [the character in the story] attacks then it approaches instead of just dragging the sword into the mouth [of the villain]. Something like that." Carson referred to his productive efforts as a story.

Carson was asked if he saw himself as an author in writing this story?

"No."

"You weren't being an author when you were writing this story?"

"I didn't feel like being an author when I was doing it."

"What would you call it? It wasn't like being an author. What was it like?"

"I feel like a programmer."

"A programmer. So what you are really creating then --"

"Was a program. ... That's why I like Authorware [Professional]. I don't mind spreadsheets and stuff in 20a [computer processing applications course]. I just like Authorware [Professional] because I find it's like the closest to being a program [and] Pascal. Spreadsheets are totally different from programming. It's just doing assignments."

"Well you were just doing assignments in Authorware [Professional] weren't you?"

"But you feel like being a programmer. I like being a programmer."

"So it's the way you feel when doing Authorware [Professional] that's different than when you are doing a spreadsheet?"

"Uh huh."

"So the feeling when you are doing a spreadsheet is?"

"I just want to complete this assignment, get good marks on it so I can get good marks on the report card."

"And that's it."

"Yeah."

"But in Authorware [Professional] it's different?"

"Yeah it's different."

"How is it different? If it's not just that what is it?"

"Like in spreadsheets and stuff you are more restricted. Like if they should give you this assignment, like this inventory and stuff you are supposed to type this into a spreadsheet. Like that's very restricted. You can't do much of it. Can't do more with it. In spreadsheet you get to do more."

"In Authorware [Professional]?"

"Yeah, I mean in Authorware [Professional] you can do whatever you want."

"Even though the assignments are set for you?"

"See like the living essay is not very set. You can do anything you want."

"What type of things can you create in Authorware [Professional]?"

"A living essay ... a movie if you have the time to do it ... and a non-fiction story. ... I think it's just something of your own creation. Something you made all by yourself."

Further Perceptions of the Living Essay and Its Creation

Based upon these earlier interactions with Anne and Carson it was decided that it would be interesting to explore what other students thought about the process of creating a living essay. The researcher had often acted in a teacher role during classes. As a result the researcher frequently interacted with students in regard to their assignments. Sufficient trust in the relationship between researcher and students had been developed to enable the researcher to request interviews with three students as they embarked upon the creation of living essays. Each of these three students was asked a number of questions about the living essay and the creation of it. Excerpts were drawn from transcripts of the discussion which enhance the view of living essays provided by the previous students. Other text from the transcripts was omitted to save space.

Steve - 'A designer of his own program about the space shuttle'

Steve had just begun creating his living essay after completing the class assignments. He was asked "What is that you will build through the living essay? What will you create?" He described his ideas for the living essay: "It's going to be all about the Kennedy Space Center and Cape Canaveral - it will have a picture of the space shuttle and the different cargo bays and in a large area it will have intimate little key notes about it - you know, kind of interesting stuff that people might want to remember about the space shuttle."

"Where did your ideas for this living essay come from?"

"I'm just thinking about what I can do that is kind of interesting because I have a book at home all about Kennedy Space Center and that is where I am getting all my information and it's all up-to-date - it was printed last year. I like it - it's really interesting stuff, so I thought I may as well do something that I like, maybe put that little more effort into it."

"Why choose this topic? And what will the living essay be like?"

"it's fun - ask people about what they want to learn about - like, trying to set it up so you can ask the person if they want to see a space shuttle or information about it"

"Are you an author when you start doing this?"

"I guess you could be....but an author doesn't necessarily mean you're writing a book, a novel, or a short story. I'm designing my own program, I guess you could say, so I consider myself an author.

"Are you a programmer?"

"In a way, yes, not as if I'm setting up my own program because I have things laid out for me, like, see my question icon automatically comes up with what kind of question it is, click/touch, drag, text so I don't have to set up everything by myself - there's a basis already there.

"So it makes it less of a programming task because of those basics?"

"Yeah."

"Are you like a director of a film?"

"Yeah, basically, because I decide what is going in there. I decide what I want, what I want to cut, what I want to add to it."

"We've talked about programmer, director, author. Can you think of any other things that there might be in this process?"

"Designer. Designing my own program in a sense creator, basically. I guess that's about it."

Tracy - 'A teacher creating a story'

"What is that you will build through the living essay? What will you create? "

"A storyline. ... Basically a story using pictures and words. ... I was going to do a modern version of Goldilocks and the three bears. ... It's called Tracy and the Three Babes. ... I had to modernize it - I'm going to follow basically Goldilocks story line but it will be more revised. You will have options of where to go - if you want her to go into the house or go for a walk - so you can choose which story."

"When a user chooses where to go in the story what kinds of choices can she make?"

"Probably what happens to Tracy - yeah, what events she wants to take place and the ending of the story."

"Where are your ideas for the living essay coming from?"

"The ideas? Probably from my other programs - all the other drawings and that plus. I don't know. just everything that I have learned from class ... like the click and drag."

"What will it be like to create this thing - will it be like you being an author writing a story?"

"Yes. I think so. ... Like a children's story because you're doing your own illustrations and you can do them any way you want - you don't have someone telling you - you don't have a publisher telling you. It's like your own - you're using your own imagination.

"Would it be like directing a film and you are the director?"

"Yeah. because you're telling the computer....yeah, you're using sound sometimes and the pictures might be moving so you're directing it where to move."

"Is it like you are creating a program and you are a programmer?"

"Yes. I would think so. Well. you have to put the commands into the computer and you have put the choices into the program so that when the people go all they have to do is move the object wherever and the computer knows where to go next so you have to program that all in."

"So it is like programming so in this you are like a programmer?"

"Uh huh."

"You're like an author?"

"Yeah."

"You are like a director. Anything else - can you think of anything else you might be in this process?"

"You're kind of like a teacher because you're teaching the people how to use it. You're giving them an idea of what you can actually do on a computer."

"When you write this program, it's a teaching program?"

"Yes. I think I could help people to read if there are words to it. I think you're teaching people about Authorware [Professional] because you're letting them know that there is such a program out there."

"What kinds of things about Authorware [Professional]?"

"Well. hmmm. the different options you can do - flowcharts, all the icons in there, that you are able to do all these different choices, and that you can move things on the screen without the whole screen moving, you can change stuff, things you can't do on a regular computer program."

Nadine - 'A camera person recording an interactive shopping sequence'

Nadine has not yet begun to create her living essay but has begun to think about it. She describes a "mall idea - a person walks into the mall and then they can walk into a shop. ... Jerseys or skirts or whatever, whatever the store is called." And what would happen in the store? "The cashier would ask if she could help you. ... It will be a big project. It's going to take some time and it's going to take some revision -- just like any other assignment."

"Will it be like writing a story? Is that what you will be doing?"

"I don't know. I don't know if I will put it in a story format or that a person will just walk in and it will start talking "Can I help you? Would you like to see this tennis racquet or this shirt?"

"Do you think it will be like directing a film in a mall? Would you be more of a director?"

"Yeah, except I wouldn't be in first person. ... It would be like ... videotaping a movie, like, the person who holds the camera, an on-looker. ... You get to choose icons."

"Is creating a living essay -- is that like writing a program?"

"Yeah, definitely. Except you're writing a program from a program, based on instruction, so like it's not all from scratch. You're writing your own program - it's like using your own skills."

"You said it was based on instruction. How is it based on instruction?"

"You know, like the stuff that we learned in class, like it is based on the booklet. based on the whole program itself, like Authorware Professional" and the computer assisted instruction provided to students in class.

"So you're not sure if it will be like writing a story?"

"It's going to be more interesting [than just writing a story] ...because you have to add the pictures and like, whether I rate a story or not, it has to be based upon a story, like you have to have an idea and it's kinda like a story, you know, you have to have a main idea and then you go on to the characters, the scene, ... You're doing a whole program not just doing parts of the program -- you're like taking all the information and you're putting it all together."

Interpreting the Interviews

It seems that several important aspects emerge from these discussions with students. These include a sense that the environment of AP through which students developed living essays occasioned the use of the visual sense. This was evidenced for the researcher in the need to record the computer screen through which students worked. In listening to early audio-taped of students' talk the researcher could make little sense of their discussion. However, when the video-tapes were viewed the discussions in which students engaged were understandable. Also from the video-tapes it was noted that students often made pointing references to the computer screen in referring to their computer programs. They were making visual references.

Students extensively involved themselves in interacting with the images they generated and perceived. They took action upon these images by modifying them and animating them across the computer screen. The process of image development was dynamic. Often images they developed were modified, redrawn or rejected.

Students' perceived themselves as active in the process of developing the living essays through their development efforts. They were authors, directors, programmers, designers, teachers, and camera persons. Each of these is a dynamic active profession through which expression of creative efforts is enabled. Students see themselves in these active roles. What view of the computer emerges from these roles?

Modes of Computer Usage

Eaton and Olson (1986) define two modes of computer usage as "computer as subject [of study] and computer as instructional tool" (p. 342). In the first case computers may be viewed as the subject for discussion with interests in hardware, software and general and specific use applications being the focus. The students in this study were enrolled in a computer processing class in which they were introduced to the Macintosh and the local area network connecting the computers. Students learned how to save the products to hard drive and diskette. They learned to navigate through the computer networks to acquire graphics in order to complete their assignments. In the initial activities of this course students were introduced to the computer as the subject of study. However this learner usage was not predominant through the course of student use of the computer.

The second mode of computer usage is as a tool to assist in acquiring concepts within a content area as Eaton and Olson (1986) indicate. In this study, students used an instructional package to acquire concepts in the use of AP. The instructional package was composed of instructional simulations (IS) and a booklet of activities. The IS were developed in a style that has been described as "like watching over the shoulder of an experienced programmer using AP and the programmer describes what is being done". An effort was made to include a conceptual framework within the IS. The focus of these efforts was to encourage students to think of the concepts they were learning rather than to focus on the skills alone (although skills were also taught). As an example, consider the concept of icon. Although students were introduced to the specific use of each type of icon and the parameters set within each, it was deemed important that students have a concept of what an icon, in general, is used for, that is it stands for the commands executed when the student's program is run. So the instructional simulations could be seen as instructional tools.

To encourage students to become more actively involved in developing concepts activities were interspersed between the IS. This allowed students to refine and practice skills and to develop concepts through actual experience with AP. The student was less passive than if asked merely to read and answer questions. However this student involvement was still externally directed and actions were intended to be those defined by the assignments. The activities in which students engaged could be interpreted as extensions of this tool usage, however this did not seem to be the case for students creating living essays (students' own autonomous creation in computer program form). These were based upon what students had learned and experienced in the instructional simulations about AP and based upon their perception of the open-ended assignment and the possibilities within the AP computer environment. Students acted upon the AP environment and acted within it. The environment responded in predictable ways based upon its structure. Students explored this structure through their creative efforts in developing their living essays.

How did these students understand their living essays? In Anne's discussion of her story she has begun to see her living essay as both a video game and a television program. When Carson was asked what kinds of things he was making in AP, he responded, "Just basic games" with the object of his game "to save the princess and kill the dragon." For Tracy the living essay was "A storyline. ... Basically a story using pictures and words", whereas, Nadine's project seemed to be about structure rather than story. It had a very different flavor than some of the others as she intended to make it an interactive system which would simulate some shopping activities. Steve's essay was to inform others about information in regard to the space shuttle and Kennedy Center. Although the assignment for each student was conveyed to them by the same teacher discussion and the completion of the assignment was to be based upon the same instructional package each student had a very different conception of the assignment. It was a television program, a video game, a story, an information package and an interactive sequence.

How did these students understand their experiences of developing living essays? When students developed their living essays they seemed to view the computer and its usage as something beyond subject of study or instructional tool. This use of the computer was as an expressive environment through which they could develop and express their ideas through creative actions. They did not make use of the computer in a way that can be characterized in a simple mode. Anne emphasized a sense of autonomy and personal involvement. She also had the freedom to incorporate events and characters in her story without having to consider others. Carson also expresses this theme of creative action in his statement: "In Authorware [Professional] you can do whatever you want." Steve describes the process of developing his living essay: "I'm designing my own program ... I decide what is going in there. I decide what I want, what I want to cut, what I want to add to it." Tracy expresses ownership and creativity: "It's like your own - you're using your own imagination." Although each students' definition of the project and activities differed the resulting experience was of active and creative involvement.

We are continuously immersed in this network of interactions, the results of which depend on history. Effective action leads to effective action: it is the cognitive circle that characterizes our becoming, as an expression of our manner of being autonomous living systems. (Maturana & Varela, 1987, p. 241)

Technical and Nontechnical Experience of Computer Technology

Ihde (1979, p. 55) describes the "technical" experience of technology in which "one is relating directly *through* a machine to something in the world, or one is relating *to* a machine as something directly within my attention within the world." When one executes one of the stories that students have created the experience of the story is through the computer. Students also create their stories through the computer. The computer mediates the experience of story creation. For example, Anne began a story about a bear. But the image of the bear existed in a computer program prior to her decision to write a story of a bear. This computer image of the bear became the image of her bear in her story. The second technical experience of technology is relating to a machine as a quasi-other. There does not appear to be any evidence that the computer or its software was a quasi-other to which students related. Ihde (1979, p. 55) might have expected students to respond to the computer as a quasi-other when he states "a good amount of the *technical* experience of computer technology is of the second type."

Ihde (1979, p.56) distinguishes the "nontechnical" experience of technology as evident through "computer technology ... [which] is somehow both 'behind the scene' and yet also 'active'." The computer software in which students created their stories was often active behind the scene providing students options and features from which they might choose and allowing students actions of typing, drawing, capturing sounds, etc. but limiting possibilities to those admissible by the interface and the software features. Thus true animation was not possible for Carson although he would have liked this to be possible.

Ihde (1979, p. 63) states that "the technical user enters into a 'dialogue' with the computer ... with a sense of both control (I set the program) and limit (within the limits of the computer's restricted language capacities)." Students have certainly experienced the control of their programs but have been restricted to the possibilities made available to them by the interface and the software. Nadine stated that she was creating a program while within a program. She also emphasized the technical skills she would incorporate in producing this program.

For Steve, as a designer, it is important to "ask people about what they want to learn about -- like, trying to set it up so you can ask the person if they want to see a space shuttle or information about it." He would build the interaction to provide users with control on sequencing and choice. Tracy echoed this sentiment while describing her Goldilocks story: "You will have options of where to go -- if you want her to go into the house or go for a walk -- so you can choose which story." Users can also choose events for characters and an ending to the story, assumedly from a set of possible choices: 1) "what happens to Tracy", 2) "what events she wants to take place" and 3) "the ending of the story." Anne's statement "in a video game you can move your man just the same as I can move the fish or the bear [in my story]" implies that motion and interaction are both possible and important aspects she incorporated into her story while using AP.

Carson emphasized the need for user interaction in order for a program to be successful. He felt his product was insufficient in this regard. He described his story as suitable for use by children "if there were more things to do. Mine just has two things to do and the rest are just like reading the screen for what's going on." To improve his program Carson would improve his technical skills, incorporate more graphics, and incorporate animated sequences which would simulate action on the part of characters. Many graphics were incorporated into Carson's work. He used fading and zooming effects to make these graphics appear and disappear, often creating the illusion of one fading into another. In regard to these graphics he stated: "I think it just looks nicer if I put it in fade in and stuff instead of just nothing." Carson seems concerned here with the aesthetic appeal of his program, another feature of both software and interface design.

Donald Norman states, in an interview with Rheingold (1990, p. 8), that "the usual concerns of interface designers--creating more legible type, designing better scroll bars, integrating color and sound and voice--are all important considerations. But they are secondary. Improving the way people can use computers to think and communicate, observe and decide, calculate and simulate, debate and design--these are primary." Norman's list of computer uses seems to imply that a change in the quality of computer uses must occur. Norman (1990, p 216) states that "an interface is an obstacle: it stands between a person and the system being used. ... How can anything be optimal if it is in the way, if it stands between the person and what needs to be done?" In Norman's example the computer is an intermediary between the user and the task. It separates them. This obstacle must be overcome in order to be productive. Perhaps the implication is that the development of skill and knowledge is necessary in order to succeed. And perhaps expert or knowing assistance is necessary to succeed. But was the interface involving icons and flowchart a barrier to these students? An effort was made to introduce students to the interface and provide them opportunities to interact with it in order to understand it. The interface may have seemed to be more of a window for action than a barrier. The task of producing the living essay for Steve was like being a programmer using a library of things already created: "I have things laid out for me, like, see my question icon automatically comes up with what kind of question it is: click/touch, drag, text, so I don't have to set up everything by myself - there's a basis already there." Although Steve may be unaware of it, programmers have libraries of programming tools to draw upon. It is not an uncommon practice among programmers to use these libraries to make the task of program design easier. Steve seems to have a sense of this in his statements. Have the icons and flowcharts restricted his efforts? This doesn't appear to be the case.

The Transformational Experience of Computer Technology

In a discussion of "the existential import of computer technology", Ihde (1979, p. 53) states that "the experiences of technology is non-neutral, it transforms experience." But in what ways does it transform experience? Anne's work was not only based upon creative action but it was her own ORIGINAL work. "If I would have done something on the Olympics [as originally outlined in the description of the assignment] ... I would just be repeating news but I wanted to be creative and doing this bear, that was original." Steve's essay was to inform others about information in regard to a topic of his personal interest: "I like it - it's really interesting stuff, so I thought I may as well do something that I like, maybe put that little more effort into it." Both students had the opportunity to make a choice to do something that each enjoyed.

For these students their views of themselves as they develop these computer products seem to influence their developments. Carson stated that "I didn't feel like being an author when I was doing it. ... I feel like a programmer. I like being a programmer." In producing his living essay his creation: "was a program. ... That's why I like Authorware [Professional]. ... I just like Authorware [Professional] because I find it's like the closest to being a program [and] Pascal."

As a designer of his own program Steve is an author. "An author doesn't necessarily mean you're writing a book, a novel, or a short story. I'm designing my own program, I guess you could say, so I consider myself an author." Steve also described himself in the process as the "designer designing my own program, in a sense [a] creator."

Tracy is a programmer because she must structure the program and make decisions of choices to provide in her program and of interactions to build for user involvement. "You have to put the commands into the computer and you have put the choices into the program so that when the people go all they have to do is move the object wherever and the computer knows where to go next so you have to program that all in." She also sees herself as a teacher. "You're kind of like a teacher because you're teaching the people how to use it. You're giving them an idea of what you can actually do on a computer. ... I think you're teaching people about Authorware [Professional] because you're letting them know that there is such a program out there." She also sees the possibilities of being in a teaching role if she develops her program for this purpose. "I think I could help people to read if there are words to it."

As the designer, Nadine is like the camera person recording a movie. She intended to build options through which to move through the various sequences. "It would be like ... videotaping a movie, like, the person who holds the camera, an on-looker."

Anne sees herself as the director of the TV show that she is creating. Anne stated that her program was like watching a television show and she "was the director." She then described her sense of creative control of this project which may have lead her to make this statement.

Ihde (1979, p. 56) describes the experience of technology as an "*amplification-reduction-transformation*" through which "technologies brings with it a simultaneous amplification of certain possibilities of experience while at the same time reducing others". He continues "computer technology in its capacities is both selective out of the range of analogues of human experience possibilities, and is amplificatory and reductive within that selection." The computer interaction inclines particular types of activities through selection and then amplifies and reduces these possibilities through interaction. So in the case of these students they had the opportunity to express themselves through the computer environment in which they worked. Expressive action was both selected and amplified by the computer experience. A sense of self as author, director, creator, teacher, camera person, and programmer developed through these interactions.

Students had experiences which were of a very different quality than those that had traditionally been thought to exist in computer environments. Ihde (1979, p. 59) states that "the computer in its ordinary usage selects and amplifies our calculational, deductive, factoral and functional analytic experiences" and produces a "reduction of speech and language." These students were not engaged in the "ordinary usage" as Ihde suggests. These students had an amplification of speech and language through the experience of producing a living essay. They did not have an increase in calculational experiences nor were their activities typically deductive in nature. The potential for calculation and deductive actions existed in AP but these students did not engage in these activities. These students' activities thus were contrary to the "ordinary usage" notions stated by Ihde. (However it should be noted that Ihde's book was published in 1979 and what was considered "ordinary" at that time may well not be evident in most computer settings in 1993. The quote should be interpreted in this light.)

Students had the opportunity to take action in a computer environment and express themselves through the use of image, text, sound, and action. Did these possibilities exist for these students previously? Have they merely emerged here in amplified form? Or are these different possibilities which have emerged through students' interactions with the computer? Prior to these activities students had the ability to incorporate hand drawn images into text. But the incorporation of this graphic means of expression into a students' story telling was not likely possible. The characterization of all computing interactions as merely amplification and reduction of existing possibilities is contested. In the production of multimedia students have possibilities that were not formerly open to them.

Commentary

Chapter 3 described the computer as a subject of study or as an instructional tool. However these two modes of computer usage did not seem to account for student responses to the AP environment. As a result another mode of computer usage was proposed, that is, the computer as an expressive environment through which students develop and express ideas through creative actions. Students' experiences seemed therefore to be beyond these initial modes of usage.

Students' experiences of the AP environment were discussed in regard to Ihde (1979). He distinguished both "technical" and "nontechnical" experiences of technology. From a technical standpoint one relates either 1) *through* a machine (computer) to the world, or 2) *to* a machine (computer) in the world. The 'nontechnical" experience of technology is evident when the computer is active behind the scenes. Another important characterization which Ihde notes is that technology, the computer being one example, is not neutral in its effects upon users of the technology. Ihde proposes that there is an "amplification-reduction-transformation" through technology in which some types of human activity is amplified while other are reduced. The AP environment with its graphical interface occasioned students to act in particular ways (using the mouse interface device to make changes to their programs) while reducing their likelihood of functioning in other ways (inputting text to take action in the AP environment).

Chapter 3 described and discussed student perceptions of the activities in which they engaged while developing a living essay. It seems that several important aspects emerged from this discourse. These include a sense that the environment of AP through which students developed living essays occasioned the use of the visual sense. Students were extensively involved in interacting with the images they generated and perceived through a process of development which was dynamic. Students perceived themselves as active in this process through their development efforts. They were authors, directors, programmers, designers, teachers, and camera persons. Each is an active and dynamic profession through which expression of creative efforts is enabled. Students saw themselves in these active roles.

What activities were students involved in through which they developed such perceptions of their activities, themselves in the activities and the computer? Chapter 4 addresses this question through the description of two students interacting with AP in developing their living essays. Provided are rather lengthy cases (in case study format) of two high school students who used AP in a computer applications class. The approaches and methods they used and developed are described in the next chapter.

Chapter 4

Case Studies of Students' Embodied Actions with/in a Computer Environment

Technological advances have placed high powered production and creation tools in the hands of novices. This allows for the rapid development of technical expertise and makes possible opportunities for expression of ideas and creative productions through interactions and experiences within a computer-enhanced environment. By learning to control and manipulate the computer environment a student is free to invent within this environment, explore it, and generate concepts about it. The activities, interactions, and conceptual frameworks planned and explored through this study were intended to provide experiences which would allow students to 'bring forth' a view of computing, and hopefully to see themselves as creators of experiences and generators of meaning in their world.

This study examines the nature of the interaction of high school computer-studies students with the branch of computing studies called icon-based programming. An icon-based programming language called Authorware Professional (AP) was used by students in this study. Within this context it's important to ask: What is the nature of student creations when immersed within this computer-enhanced environment? What strategic patterns and techniques do students use in these productions within the context of AP? What insight into computing concepts is gained from these experiences? What is the interaction between the learner and the medium?

The research problem addressed through this study is the interpretation of human-computer interactions. These interactions are different for each student-participant dependent upon their experiences and their interaction with the medium. This interpretation is articulated and expressed through this study by viewing students activities and interactions.

Procedures

To address this research agenda an intensive case study methodology was developed which incorporated video techniques for recording the activities of two high-school students (aged 16 years) while they interacted with the computer software and classmates. Seventeen grade eleven students were enrolled in a computer processing class focused upon the use of computer applications. Three applications were to be used including a word processor, a spreadsheet program, a graphics generation package and an icon-based programming language (AP). In this study students were introduced to the Macintosh computer, completed assignments using the word processor and then used a spreadsheet program for class assignments. Students had worked at their own pace in completing these assignments so they began the unit using AP at different times.

Materials and Activities

An instructional package (IP) containing four instructional simulations (IS) interspersed with eight activities (see Appendix) was provided by the researcher for student use to explore (and construct) concepts and skills for using AP and general computing concepts. Following the completion of the IP students created a multimedia computer file containing graphics, text, animation, interaction and sound. Students drew upon their experiences with AP and the lessons learned from the IP. Possibilities within the computer environment enframed their work. Students incorporated story lines and characters of their own creation into the productions. The teacher in the classroom called the productions 'living essays'. This term may have lead students to incorporate story lines and characters.

Through the IP students were encouraged to form hypotheses about manipulating and performing tasks, to test them, and to interact with the computing principles involved in a creative way. During the study all students used the four IS as part of their regular class progression. After completing the first IS entitled "**1. Introduction**" students completed **Activity 1** which allowed students to create a program based on what they had learned. Students were directed to create screen displays of objects, to erase screens they selected and to incorporate pauses in the program. Fade or zoom effects were also to be incorporated into the displays and erases. The second IS entitled "**2. Introduction to the icons**" introduced students to icons and particularly to the method of creating loop structures in AP. After this lesson students were provided **Activity 2** in the form of worksheets which displayed 'screen images' similar to those they had just seen. They were asked to describe (and in some cases draw) the dynamic flow of activity within the displayed loop structure programs. **Activity 3** described a program (loop structure) students were to build and a list of parameter settings students were to set. Following the setting of the parameters, students were to predict the outcome of running the program and then run the simple program and record the results. Students then systematically changed the parameters, made predictions, ran the program and recorded the results. In essence this activity was designed to provide strategies for designing, predicting, and testing programs. Students were encouraged to relate the answers in **Activity 2** to the results of **Activity 3**.

Following **Activity 3** students studied the third IS entitled "**3. Multiple Choice**" which taught the creation of a complex interaction and error checking when creating a computerized multiple choice question in which 'clicking' upon the correct answer or entering the answer from the keyboard were both tested. The multiple choice question was used as the first interaction type because students possessed a concept of a non-computerized multiple choice question and this conception would be helpful to the understanding of the computerized version. **Activity 4** was the creation of a computerized multiple choice question. Students were encouraged to create their own multiple choice question and to design error checking to suit the content of their question.

The fourth IS entitled "**4. Drag Object**" taught the creation of a drag-object interaction. This involved the creation (or importing) of a graphic which became the object to be moved. A second graphic was created (or imported) to stand for the destination to which the first graphic was to be moved. **Activity 5** involved students creating a drag-object interaction. For example, one student imported graphics of a bear and a salmon. When the program was run the user was to feed the bear the salmon by dragging the salmon to the bears mouth. Another student created a question in which musical notes were to be 'dragged' to an image of a parabolic dish. When the notes were 'received' by the dish music was heard. If the notes were misplaced (not placed on the parabolic dish) an error message was displayed and another opportunity to place the notes was provided.

The directions for creating a program for **Activity 6** were modeled after the IS. However, instead of seeing the step-by-step dynamic creation of a program by a programmer, static screen images of the program were displayed indicating the various steps to be accomplished by the programmer. The conceptual framework was provided by asking guiding questions through which students were to build their own frameworks. Students were instructed in the creation of a text-answer response type of interaction. In **Activity 7** students created programs to ask questions and test for correct, incorrect, and indistinguishable text-based answers.

During **Activity 8** students returned to the five programs they had created in the earlier activities: i) displays, erases and pauses, ii) loop structure, iii) multiple choice question, iv) drag object question, v) text answer question. These five programs were then copied and placed into one AP program. With the completion of **Activity 8** students had integrated small interconnected pieces into a complete and relatively complicated program. A flowchart is provided showing the steps which students took in completing the IP (Figure 4-1).

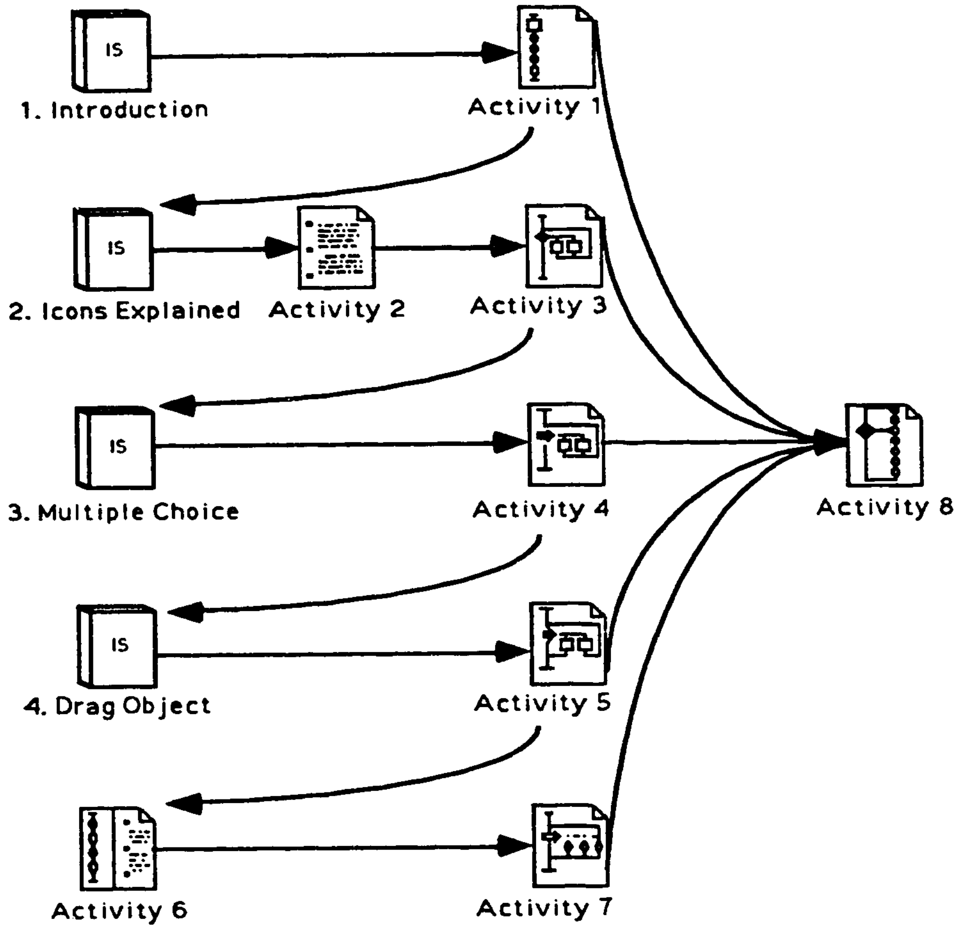


Figure 4-1. Summary Flowchart of the Instructional Package.

Each of the IS are shown in Figure 4-1 by book-like icons indicating their instructional nature. The activities, which required students to create programs, are represented by flowcharts. Activity 1 is a simple linear flowchart indicating the nature of the flowchart students created. Activity 2 required the answering of questions so is a picture of a loose-leaf page with answers on it. Each of the other activities involved the creation of a program with some path (loop) structures so these flowcharts are more complex in appearance. The icon for Activity 6 uses both a flowchart and a loose-leaf page to indicate that both programming and answering of questions was required of students. Activities 1, 3, 4, 5, and 7 are shown leading to Activity 8 as it incorporated the products of these earlier activities.

Following the completion of the IP students embarked on the creation of a complete program on a topic of their own choosing. As students embarked upon the creation of their multimedia productions the only stipulation was that they were to use the concepts they had learned and incorporate graphics, sounds, and video sequences into their programs. Greater definition of the multimedia productions was provided by the teacher as students progressed.

Student Cases

Student progress was recorded using two video-cameras with attached microphones. The initial recording was of the activities of the entire class. The second recording was of a small group of students in the class. Finally each video-camera was focused in-depth (for about one month) on one student's computer screen. The cameras were placed behind one student in the position of 'peering' over her/his shoulder. By 'zooming in' on the computer screen with which the student was interacting the procedures used were evident on the tape. Student interactions with classmates, teacher and researcher are audible on the videotaped recording. These interactions are presented as part of the cases. Each students' work was recorded over a three week period during which they developed living essays (multimedia stories) for class. This involved collecting 6 to 8 hours of videotape for each student. Annotated transcripts of these videotapes served as the main source of "data" in this study. Additional sources of data were provided by researcher notes, students computer programs, and transcripts of audio-taped interviews and researcher/student interactions.

Susan

Day 1. Susan had begun her multimedia production by opening a new computer program file. She placed a display icon on the screen and into it she typed the initial lines of her essay:

> **Once upon a time there was a little mouse named Cheeser. As you can imagine he adored cheese. As a result of his love of cheese he grew very fat.**

Susan drew a simple diagram of a mouse but then erased it. Susan asked the researcher "Would it be better to write your story and draw your pictures [at the same time] or would it be better to write your story and then draw your pictures later?"

"Well maybe you want to write some of the story first and then draw pictures and figure out where you want them in the story and then write more of the story."

"Is it okay to have just three sentences on each page or is that wrong? I don't know what she [the teacher] is asking me to do. I'm just guessing."

"Well, um. What do you think is best? Do you want a whole screen full of text?"

"No."

"Probably not. Is three sentences too much? Too little?"

"I don't know --"

"I think maybe what you want to put three sentences and then try it. And then maybe you want to then explore six sentences and see if that's too much or not."

"Okay."

"You are sort of building as you go. Once you get under way I think the answers to those questions will become clear."

At the computer next to Susan (on her left) a student named Logan had discovered a sound of someone vomiting. Susan said "Hey I need that sound for my mouse. He ate too much."

Susan decided she wanted to begin with a picture of her mouse so she saved her file with one display icon and the introduction to her story in it and then went to Superpaint to draw. In Superpaint Susan began drawing a mouse by dragging first a round body and head, then adding a tail and then adding pointed ears. She drew various versions of this mouse, erasing the previous ones, experimenting with what she wanted the mouse to look like and also experimenting with various drawing tools available. She saved her mouse in Superpaint before leaving class (Figure 4-2).

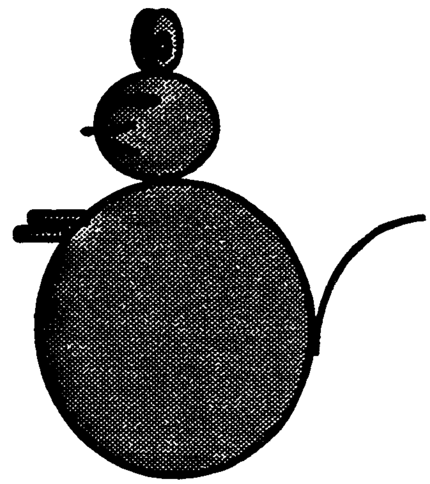


Figure 4-2. Susan's Cheeser the Mouse as she began her multimedia production.

Day 2. When Susan entered class she turned on her computer and opened Superpaint. She copied her mouse and placed it in the Scrapbook. She then opened her AP file and copied her mouse into her file. The mouse was very big. Susan stated "My mouse is too big." The researcher responded "I thought it might be. Maybe you should save this and then go back to Superpaint and we can shrink it." The file was saved, AP closed and Superpaint was opened. Susan said to Logan (the student next to her) "I don't know how to do this. Logan how do you shrink it? Do you know? Logan said "No." With the researcher's verbal guidance Susan selected her mouse, reduced its size, copied it to the Scrapbook and saved the mouse image file. She then returned to her AP file, opened the display icon with her introductory text, placed the new mouse, and deleted the oversized mouse which was still on the screen. She then added more text, made minor editing changes to the words she had typed, and then saved the file.

Susan then returned to Superpaint. She copied the image of cheese she had drawn in Superpaint and placed this in the scrapbook. She then returned to her AP file where she attempted to paste the cheese image into her display icon but the computer software responded with an error message about not enough memory. Susan asked the computer, "What do you mean?" Frustration was evident in her voice. Susan again attempted to paste the cheese into her picture but again received the error message. She waited for assistance and then demonstrated the error to the researcher. The difficulty was that the computer being used had very limited memory capacity. The computer file was saved in hopes that this would reduce the memory requirements and enable the paste action. This did not solve the problem. An attempt was made to paste the image into another display icon which was empty. This was unsuccessful. AP was closed and Superpaint reopened where the cheese was reduced in size to match the mouse. The entire mouse and cheese were grouped, copied and pasted into the Scrapbook. A copy of the small cheese was also placed in the Scrapbook in case it was needed. AP was reopened. The small cheese could not be pasted. The original mouse was cleared and a copy of the new mouse with the cheese was copied. The mouse appeared but the cheese did not.

Susan asked "Could I just draw cheese?" The researcher responded "Yes. Square cheese. [AP has very limited drawing options: ovals, rectangles, and lines.]" Susan said "But cheese is not square." Logan, the student sitting next to Susan said "Block cheese is." Susan stated "I'll change it to Pizza and draw that." Susan drew a triangular piece of pizza and added circles and squares as toppings. She then returned to the icons and dragged in another display icon. Into this icon she typed in more text:

Cheeser got so incredibly fat that his doctor ORDERED him to quit eatting [sic] so much cheese. (Please help Cheeser with his diet and feed him the apple.)

Susan then copied another image of her mouse from the scrapbook and added this to the second display. She drew a table with an apple on it. These display icons were then moved into a question sequence of the move object type. (At this point the camera has been moved so Susan's actions are no longer clearly visible thus are not described here.) Susan saved her file at the end of this class.

Day 3. As Susan began she opened her AP file. In this file were the first two display icons she had placed in the flowchart followed by a question created using an interaction icon and display icons as feedback. She ran the program but the second display appeared on top of the first. She placed an erase icon between them and ran the program again. This time the first display was erased before the second display appeared. However the first display could not be read as it quickly appeared and was erased. Susan placed a wait icon before the erase icon so in execution the first display would appear and then wait until the user pressed the return key before proceeding.

Susan then opened the interaction icon and set the two move object areas. (These areas are used in an interaction when an object on the screen is to be moved to a particular region of the screen.) The one move object area she stretched to be the size of the full screen, the other small one she placed next to the mouse. She then ran the program again stating "Okay I have to see what happens." The first display appeared and was erased. The second display appeared with text, the image of the mouse and an image of a table. Then a third display appeared with the image of an apple which was positioned to sit on the table. The interaction icon indicator opened and the move object areas appeared. AP indicated that the interaction icon had no text or graphic content and opened this icon automatically to enable its inclusion. Susan closed the icon and jumped to icons. She paused without acting and then opened the third display icon with the apple in it and closed it. She then opened the second icon with the mouse, table and text. Susan then called for assistance.

The researcher approached. "What's happening?"

"I don't know. I'll just run it and show you." The file was executed as earlier with the final step being the appearance of the interaction icon indicator.

"Ah huh. See what comes up" pointing to the interaction icon indicator. "The question [interaction] icon comes up. That means it is empty. There isn't anything in that question [interaction] to tell people what to do. You need to put something in."

Susan jumped to the icons and opened the interaction icon. She set this icon's options. "Don't I have to have my apple in here."

"You already have it in there."

"I know but I don't know where anything is. I'm so confused." Susan executed the program once more and again was confronted with the interaction icon indicator. She said "those shouldn't be there" pointing to the move object areas.

"Those won't be once you fill in the question."

"Where do I put that?"

"Where do you want to give directions on the screen?"

"Right here" pointing to the bottom of the screen. "I'll erase this" pointing to the text already on the screen as a result of the second icon.

"Where is that? In what icon is that located?"

"I don't know."

"You have [the images and text of] a bunch of different icons all on the screen at the same time. So I am not sure where they are in the icons."

Susan jumped to the icons.

The researcher asked "Where are the directions? In which icon?"

Susan opened the second icon in which there was text, the table and the mouse.

The researcher directed "Cut out the text."

"Why would I want to cut it off?"

"To put it in the question [interaction] icon."

Susan cut the text from the second display icon and opened the interaction icon into which she pasted the text directions. Now she executed the file again. This time the interaction icon did not open but the directions to move the apple to the mouse simply appeared. The correct response display icon now opened because it did not contain text or graphic images. Susan closed this icon and continued with the execution of the program. It finished and Susan executed it again but this time moving the apple away from the mouse which was an incorrect move. Despite the fact that Susan had created a response icon to respond to incorrect moves this move of the apple to the wrong location was not detected. Susan then checked the options set for the interaction icon but made no changes. She opened the display icons to provide feedback for the incorrect and correct answers and entered text. She then executed the program again with the incorrect movement of the apple. However the incorrect feedback again did not appear. Susan opened this icon to check its contents. The feedback was still present. She checked the options set for this feedback and then spoke to the researcher.

"The apple's not snapping back." The option set was to have the apple return to the surface of the table if it was not moved into the mouse's grasp. Susan executed the program again moving the apple to the wrong region. "See if I go like that it's supposed to snap back." Susan opened the feedback and checked its contents. She opened the options setting for this feedback and changed the option first to snap to center with no effect and then back to put back. These changes seemed to have no effect. She asked for assistance. Susan stated as she executed the program. "The correct one works but not the wrong one"

The researcher responded "we are still in the question. You haven't told that wrong response that it's the apple that's being moved."

"So what do I do?"

"What we'll do is we'll keep our finger on the shift key so the screen is not erased and we'll jump to icons." Susan performed these actions. "And now we'll open the arrow part of the move object. Yeah" as Susan pointed to this icon and selected it. "Notice at the top" of the options box which appears it says "Drag object to the target position." Susan moved the apple and the move object indicator box changed position. "Now you need to change the size of the box" so that the entire screen is the wrong answer region. Susan stretched the box to the edges of the screen. "Click okay and we'll try it." Susan clicked on OK. "You have to tell it which object is being moved or it doesn't know it otherwise." Now Susan executed the file and it ran as desired. When the apple was not placed in the mouse's paws it was moved automatically back to the table and the message appeared **Please put the apple in the mouse's paws.** When the apple was placed in the correct position feedback appeared affirming the move. Susan then added another feedback loop for times when the wrong move was made three times. She then removed this response.

Susan now added a sound icon to the end of her flowchart so that when the correct answer was entered it would play. But she positioned it after the interaction icon which meant it played after the question had disappeared. She changed her correct answer feedback to a group icon and then placed the sound icon inside it. Now the sound would play as the text affirmation of correct action appeared. She executed this sequence and it worked. She then placed two erase icons at the end of her file to erase the image of the mouse and apple. The question was finished.

Susan then began a new phase. She added a display icon to her flowchart and began entering text

Cheeser was doing very well on his diet and seemed to be losing weight. All his friends were very proud of him, so they gave him presents.

She added a copy of the image of the mouse but then deleted the mouse image and saved her AP file. She opened Superpaint and drew three gift-wrapped packages next to Cheeser the Mouse (Figure 4-3). Notice that Cheeser is much thinner in this image. She grouped these images, saved them and copied them to the scrapbook. She then reopened her AP file and placed this image into her new display icon. She placed an erase icon after this display icon to erase it. She executed the program, saved it and shut off her computer as class was over.

As Susan began working the following day she executed her program to see its functioning. She then dragged a wait icon into her flowchart between the image of the mouse and presents (Figure 4-3) and the erase icon to erase this image. She executed the program again to see its functioning with the changes. This time a pause occurred before the image was erased. She then jumped to the icons again and placed a display icon at the very beginning of her flowchart. She created a title page within this display icon which read **Cheeser the fat mouse.** Various text fonts, sizes, and positions were attempted before she found an acceptable format. She then added an erase icon to follow the title page and set it to erase the title page. She then added a wait icon after the erase icon. She executed the program. It displayed the title page and erased it by fading it out slowly then paused with an empty page. She executed the program again. This sequence was repeated but she let the program run until it displayed the beginning of her story. She then jumped to icons and deleted her wait icon. She executed the program again.

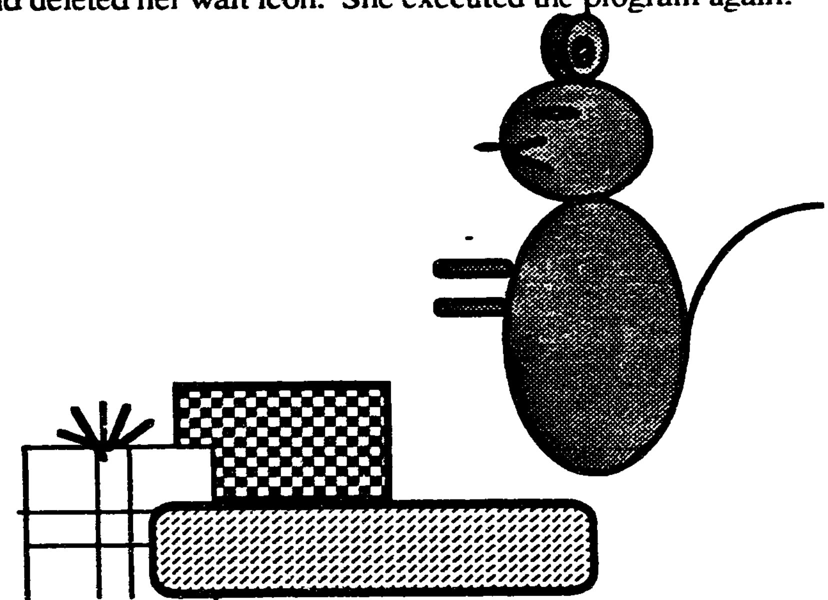


Figure 4-3. Susan's Cheeser the Mouse receiving gifts.

Now Susan noticed Logan next to her had begun working with graphics. Susan asked him, "Where do I import them from?" She jumped to icons and saved her program. She quit AP. On the desktop was the icon for her file called fat mouse which contained the first image of the mouse she had drawn. She double-clicked on this icon to open it and the image of her fat mouse appeared in Superpaint. She closed Superpaint and her fat mouse image. On the desktop were various files containing applications.

In regard to Logan's screen Susan asked "How do I get there?" She opened a file folder named Superpaint and in it was the icon for this application. She asked Logan "Which Superpaint do I go to?"

Logan said "That one."

Susan opened Superpaint by double-clicking on the icon. She then chose to open from the pull-down menu. She returned to a listing of files on the hard drive which she could open. She then asked Logan "Now what do I do?"

"Now go to drive" an option button which appears on the screen but which was inactive at this time. "Oh you can't. Well go to Macpaint" a file on the hard drive which had some graphics in it.

Susan selected the Macpaint option but the drive button was still inactive.

"Oh you don't have another drive on here." In order to find the graphics one must connect to the file server's hard drive through the chooser under the apple menu.

At this point the researcher happened by and Susan asked "How do I import the graphics?"

The researcher responded with the question "Where are your graphics?"

"I don't know."

"What graphics are you using?"

"The ones that everyone else is playing with I guess."

"Have you gone to the MacB [file server] to look at the graphics?"

"No that's what I'm trying to find."

"Go up to the chooser and connect to MacB."

Susan used the chooser to connect to the file server named MacB. Once this was accomplished Susan could use the open choice in Superpaint and select the MacB drive and pictures were available. She began looking at different images from those provided. She looked at an old farm truck and a shack to which she tried making some editing changes with the paint brush. She then closed Superpaint. She asked the teacher "Do you have to import graphics for it [the living essay]?" She was told that it was required. She reopened Superpaint.

The researcher asked, "Did you find what you want?"

Susan responded, "No. I'll have to look again 'cause there's nothing I want in here. I have to find something -- Nothing to do with my story." Susan found a large list of images. She commented "But none of these go into my story. She [the teacher] said I have to put one of these in my story. None of them belong there. Well maybe I'll use the church. But it doesn't have anything to do with my story at all."

The researcher asked "Is there anything else that might fit?"

Logan piped in "How about Cheeser [the mouse and main character] goes into a car?"

Susan asked "Why?"

The researcher asked "Well why do you have to import these? There have to be more graphics."

Susan opened an image of a car and said "well let me see maybe I could fit him in a car."

The researcher said "Maybe behind the wheel or something. That's a pretty big car too."

"Can we make it smaller?"

Logan said "Yeah you can."

The researcher suggested "You will loose a lot of the detail but you can."

Susan asked "How?"

The researcher said "You are in paint so circle it." In other words, choose the image you wish to shrink.

Susan selected the car image and then used scale selection to reduce the image of the car to various sizes. She then asked "Can you put it sideways?"

Logan answered "Yeah."

Susan began making selections of Flip Horizontal and Flip Vertical and saw the car upside down and a mirror image of it's previous image. She then selected rotate and the car turned on its side to which Logan laughed and Susan said, "That is not what I wanted to do." She selected Undo and the car returned to its previous orientation. She then closed the file. She opened other images and then opened a goat. She shrunk this image to 50%. She said, "He [Cheeser] can have a friend names Mr. Mountain Goat or something." Susan copied the mountain goat image into the scrapbook and closed Superpaint. At this point the class was over.

Day 4. As Susan returned to class the following day she opened her AP program. Now Susan noticed that the icons in her flowchart extended to the bottom of the screen so there was no longer any room to add another icon. She asked "I have to get down here though" pointing to the below the bottom of the screen.

The researcher asked "It's getting too big right?"

"Right."

"Select the title page and the other icons" pointing to the initial list of icons "and group them."

Susan selected the initial list of icons and chose the group option from the pull-down menu.

"Remember when we were doing this" making a program "we did something like this" dragging in a decision icon "and then we dragged in these" dragging in the group icon. The researcher then put things back the way they were and suggested that Susan "Try it".

Susan dragged her decision icon onto the flowchart and placed her group icon next to it. She set the group icon options to erase before next selection and then executed the program. She returned to the icons and tried to select the next 4 icons but the group command was inactive. Susan had selected an interaction icon as part of her selection of 4 icons but had not selected the display icon answers to the interaction so the group command was inactive. Susan was not aware of this as she tried it again. So she placed another display icon at the end of her flowchart and opened it. She selected the text tool but the cursor was very large. She checked the style of the text and found it to be very large. She closed the icon and opened another in which she had placed text. She checked the style of text here and then returned to her new icon and made the same selections. She then entered new text for her story.

> **Cheeser finished opening his gifts. All his friends went home, so he decided to take a better look at his presents. He got a new book from his friend the rabbit, a new pair of mittens from his friend the elephant and a new cook book from his friend the robin.**

She opened the scrapbook to find an image and flipped through all the contents of the scrapbook. She found the image of a wine bottle and balloons. She then returned to her image of the mouse. She closed the scrapbook. She opened up her icon containing the mouse-with-presents image which she selected and ungrouped. She copied the mouse image into the scrapbook. She opened an icon and pasted the mouse into her icon. Because the image had been ungrouped when she attempted to move the icon around it the image was changed rather than moved. Susan exclaimed "Oh no." But she deleted it and placed a new copy of the mouse image into her icon. She then drew an object in the mouse's paws but deleted it. She then asked Logan "Do you remember where the program opened a book?"

Logan responded, "Yeah that's under movie."

"How do I get there?"

"Go to icon. Drag that thing" pointing to the movie icon "onto the line."

"Oh yeah." Susan dragged a display icon to the line but it disappeared off bottom of the screen. Susan placed a group icon in the middle of the icons, opened it, and placed the movie icon into it. She chose the image of an opening book as her movie and moved it so it would appear to be in the mouse's paws. She executed her program which ran properly but the book page just kept turning. Susan opened the icon for the movie and changed the options to wait until done and asked "How do you get this to end?"

Logan asked her to execute her program and said "You've been doing all this. Holy smokes" as he watched the execution of the file. Later he said "Cool!"

Susan then said "It's the next one." The program sequence in which the book was turning appeared but the mouse did not appear. Susan asked "Where's the rest?"

Logan selected the pause option and dragged the image of the book around on the screen.

"No I want it on a different page."

"What do you mean on a different page?"

Susan used the jump to icons command and opened the group icon containing the movie icon. She ungrouped it. But then she said "There is another thing there" pointing to the bottom of the screen. "There is another one of these" pointing to the display icon in the menu of icons. "I've lost it."

Logan asked "Is this the third page?"

"It's the fourth page."

Logan pointed to two icons appearing higher in the flowchart and said "If you group these two. You have to start grouping."

"I did but I lost one of them."

"No it's still on there" it just can't be seen.

The researcher was watching this interaction. She showed Susan that she could group the icons by selecting them. Susan began to select the four icons as she had previously. The researcher suggested that she had to capture all the icons to the right of the interaction icon.

Susan said "Oh that's why it didn't work." This time the group option was available and Susan was able to group these images. Susan said, "Oh", and pointed to the screen on which now appeared her fourth page icon.

The researcher suggested 1) that Susan drag her new group icon up to the decision icon so that it would contain the first and second pages, 2) that "the fourth [page] isn't lost it's just we couldn't see it", and 3) that Susan drag her third and fourth pages also to the decision icon.

Susan executed these suggestions and then they set the options for each of the pages to erase before next selection and to pause. Susan then executed the program as she said, "I didn't hardly do anything this whole class."

As far as adding new pages to her story Susan did not accomplish too much. But she completely reorganized and restructured her program. This is a major programming activity but did not accomplish major changes in the story, so this may have appeared to be "hardly doing anything."

Now Susan executed the program. After the first page the return button appeared on an empty screen. Susan stated, "I don't want that there. Before it just went on to the next page." She pushed the return key and the program ended to which she responded, "Oh it's over?" Susan used the jump to icons selection and returned to view her icons.

The researcher pointed to the icon for the first page and asked Susan "What would make it go to the first icon and quit?"

Susan opened the decision icon and changed the options within this icon to until all selected. She then executed the program again. At the end of each page was an extra Return key to which Susan responded, "I'll have to get rid of those." When she got to the end of the program the book just kept turning. She said "It will never end now." She jumped to the icons and saved the program as the class was over.

Day 5. Susan began this class by executing her file. She ran the file until the first return key appeared. She chose show current icon which selected the first group icon. She opened the options for this icon and changed the option so it would no longer wait after the execution. She changed the setting for the next group icon as well as it also had the problem. She then executed the program. It no longer paused with an empty screen. She then chose the display icon for page three and grouped it. She then set the options to erase before next selection and to pause. She then executed the program.

Susan next added another display icon after the loop structure. She typed text into the display.

> **All of a sudden Cheeser heard a knock on the door. Quickly he hurried to the door and to his surprise it was his cousin, Gary goat.**

She then added the image of the mountain goat from the scrapbook. She then added the image of the mouse to the right of the mountain goat. Next to the mountain goat's mouth Susan added a circle and in the circle she typed text (Figure 4-4). This made it appear that these words were coming from the mountain goat in typical cartoon character style she had created a 'voice bubble'. She then added an erase icon to erase the mountain goat. She then grouped these icons together and added them to the loop structure as the fifth page.

Figure 4-4. Susan's mountain goat Gary.

Susan added another display icon to the end of her flowchart and into it placed text.

> **Cheeser and his cousin sat down and talked for awhile. Gary had told Cheeser how good he looked now that he had lost weight. Cheeser started to feel really good about himself. Gary realized it was now getting late so he decided to leave.**

She copied the image of the mountain goat from the scrapbook and added this to the display and drew a door on the screen using the rectangle and circle tools. The door hid the goat's head as if the goat was exiting by the door. Susan then added the mouse to the display using the scrapbook. However the mouse was not facing the correct direction. He was facing away from the goat. Susan simply dragged the paws, nose and mouth to the other side of the oval mouse body and the mouse now faced the other direction. She then added a 'voice bubble' next to the mouse's mouth. Behind the mouse Susan drew a clock which read 10 o'clock. She closed the display icon and returned to the icons. She placed the display icon on the end of the loop structure naming it Page 6.

Susan placed another display icon below the loop structure in her flowchart. She typed text into this icon.

> **Cheeser was so tired that he cleaned the kitchen the[n] went straight to bed.**

She then drew a picture of a mouse lying in bed beneath a window which was colored black and had a round white circle, presumably a moon, in it. Next to the mouse's mouth she added the letters zzz. She asked Logan, "Does this look like he's snoring with the z's?" Logan and Bob (sitting to Logan's left) each answered "Yeah." Bob asked, "He's snoring with the z's?" Susan said, "You know what I mean." Bob said, "Yeah." Above the image of the mouse Susan drew a clock which read one o'clock. She then closed the icon. She placed this icon into the loop structure as Page 7.

Susan executed the file but when it got to the book turning the pages it would not proceed. Susan jumped to the icons, found the movie icon in which the book was animated (Page 4), and opened and closed the options window without making changes. She placed 2 erase icons after the movie icon, one to erase the book and the other to erase the mouse image. She executed the program but these changes had not solved the problem. She reopened the group icon and deleted these erase icons.

Susan continued to solve the problem of the book turning. She opened the Page 5 group icon, circled around the icons with the screen pointer and closed it. She also surveyed Page 6 in the same manner. She then opened the Page 4 group icon and the display icon which preceded the movie icon, closed the display and opened the movie icon. She moved the options box down so she could see the screen and then moved it back up. Using the screen pointer, Susan surveyed the options available for the movie icon finally selecting the OK button and returned to the icons..

Susan again dragged in two erase icons and placed them after the movie setting them to erase the mouse image and the book animation. Susan then executed the program again. When it got to the book and would not continue Susan said "I have to get rid of this stupid page and it won't go away!!"

The researcher asked "Did you put in an eraser?"

"I did. I put in two!" and jumped to the icons and pointed to the two erase icons with the screen pointer.

The researcher asked Susan to open the erase icons and step-by-step these icons were set to erase the mouse and the book.

"You mean that's all I have to do?" sounding like a non-believer. "Is there some way I can jump to the middle of my story?" Apparently Susan was tired of having to run the entire story to check something at the end.

"You have these flags. The start flag can be placed where you want it to begin."

"I didn't know what they were for. Cool!" Susan placed a flag in her program just before Page 4 in the loop structure. She then executed the program as before. It began it's execution from the beginning once again. Susan responded "No" but continued to execute the program. At this point the camera moved so the screen was no longer visible. As the image returned to Susan's screen Page 6 was open. Susan closed this and opened Page 1 and the first display within it. She then opened the second display in it. She then closed Page 1.

Susan renamed the group icon Page 1. But then did not want the new name so pressed the delete key. This resulted in the removal of the icon. Susan was concerned. The researcher suggested that she save the file but rename it before saving it so her previous copy of the file was not lost. Susan did this resulting in two copies of the file. The researcher asked Susan to open an old copy of the file and copy Page 1 from this file. Then the new copy of the file was opened and the Page 1 group icon was placed into the file. This new file was saved as it was the end of class.

Day 6. The teacher at the beginning of this next class explained to students what the living essay was to be. "You need to have some kind of a story. If you want to use an essay that you'd use in Social Studies or English you can do that or you can make up your own story."

A student asked "In Authorware [Professional]?"

"You'd do it in the word processor first."

"You have to have pictures too, eh?"

"The story is one part. Then you need two types of graphics: one that you have made yourself and one you can import from another source. So you need two types: one that you have created and one that you have imported. You need some type of animation. You know the drag object and you need some sound. That's basically it. You need some type of word processing story. Oh, and you need some type of multiple choice--some type of choice in it. I think that's it."

Susan listened as she opened her file. She executed the file. The first page appeared and was cleared and then a return key prompter appeared on the empty screen. Susan returned to the icons and changed the options for each of the pages to not pause. She then executed the file again. Now Page 3 did not pause before erasing the page. Susan said "Everything is screwed up." Susan proceeded to check Page 1 opening the display icons presumably to determine the contents of these icons. She added a descriptor to the name of this icon so she could remember it's contents. She then opened Page 2 and added to the name of this one. Then she checked Page 3 and opened the display icon. It was the page which had quickly appeared on the screen and then been erased without a pause. Susan closed the icon and placed a wait icon after the display icon. This would cause a pause. She closed the group icon called Page 3, added to the name and then opened Page 4. She opened and then closed the display icon. She added to the name of the group icon. At this point an electronic alarm sounded in the computer room in which Susan was working. She said, "Is that me? If it is I think I'll die." However it was only a nearby student's watch alarm sounding. Susan continued opening each of the group icons, checking their contents and then renaming them by adding a descriptor to each. To Page 5, which contained the image of the goat she added the word "goat" so the name was now Page 5 goat. However, when Susan opened Page 6 she encountered only an erase icon which she opened. She paused her activity here. She then closed the erase icon and labeled the group icon Page 6 erase goat. She continued to check and rename each of the icons. She then executed the program.

Susan had not fixed the problem of the book that kept turning its pages. She jumped to icon and checked the options in the decision icon. These must have seemed acceptable because she did not change them. She executed the file one more time and again the book just kept turning. She opened Page 3 and looked at the interaction icon. She closed this icon. She opened Page 2 and closed it. She then opened Page 4. She opened the display icon here. She closed it. She opened the wait icon and changed the options to Keypress but not Mouseclick. She executed the file again. Susan asked the researcher, "Could you tell me what I'm doing wrong because I can't get it to erase what I did last time." She executed the file. Now one page paused asking for a mouse click but did not accept one. Susan returned to the wait icon that had created it and changed the options to also accept a return and show prompt. She then opened the display icon that asked for the mouse click and removed this request. She then executed the file again. This section now operated appropriately but the next one paused without showing a prompt for the pressing of return. Susan opened the appropriate wait icon and changed the options to show prompt. She executed the program again. Now the return prompt appeared but the mouse click was not able to make the program proceed. Susan said, "Now this is where my problem is." She jumped to icons, opened the wait icon and changed the option to accept Mouse click. Now when executed this part of the program was acceptable.

Susan now returned to the section of the program with the book continually turning. She jumped to the icons and opened the movie icon options. This time she changed the options from being set to play repeatedly to the option for number of times to be executed and she entered 3 times and then set this option by pressing the OK button. She executed the file again. Now the book turned three pages and the program moved on. Each of the following three pages appeared and disappeared without pause. Susan saved her file. She then opened the movie icon options for the display of the book and asked "Could you put this to play until the mouse is clicked?"

The researcher responded "Yeah. Choose Until condition is true and you need a variable."

"Variables? Where do I find that?"

"Under the pull-down menu called variables. You'd better Show variables because I'm not sure where to find it."

Susan chose this menu-option. A list of variables appeared and the one called MouseDown was pasted into the variable space in the movie icon options. The file was executed again. The book continued to play repeatedly as it had done before. The setting of the variable was not working. Susan went back to the movie options and set it back to play 5 times. Now she executed the file again.

"Was that too fast?"

"Yeah because they have to be able to read [the text on] the screen."

The researcher suggested that Susan use the options for erase and pause provided by the group icon. Susan set these options and now the book appeared, the pages were turned 5 times and then a pause waited for the user to press the return key. Once Susan had pressed the return key the program moved on to execute her final three pages but each one as yet had no pauses. The researcher suggested that Susan jump to icons and save the file. Susan did this.

Now Susan began to try to solve the problem of her last three icon not pausing. She opened the first of the three icons. She closed it, returning to the icons. She then selected it and grouped it into a group icon. She placed a wait icon after the display icon. She changed the settings in this wait icon so it no longer would accept a keypress. Then she executed the file. This page now paused but the last two pages did not pause. She found these icons, grouped each, and added a wait icon to each, changing the options to not allow keypress. She then executed the file. Each page now executed with a pause.

Susan jumped to the icons and placed a new display icon on the end of the loop structure. She then opened the display icon before it and copied the image of the mouse lying in bed to the scrapbook. She then reopened her new display icon and placed the graphics into this icon. She changed the time on the clock from one o'clock to three o'clock. She lowered the image of the moon in the sky. She removed the "**z z z**" next to the mouse. She place a text bubble next to the mouse's mouth and entered the text **I'm hungary !!** [sic] into it. She then entered further text of her story below the picture.

> **Cheeer** [sic] **woke up at three o'clock in the morning and decided he was very hungary** [sic]. **Cheeser got out of bed and hurried to the kitchen.**

She then jumped to the icons and grouped this display icon. Into the group icon she placed a wait icon and then renamed the group icon Page 9.

Susan added another display icon and labeled it Page 10. Into this icon Susan added more text for her story

> **All Cheeser could find to eat was a carrot and he was sick of eating carrots!! All he could think about was the block of cheese he hid in the cupboard before he started the diet.**

She added the image of the mouse she had created earlier and stored in the scrapbook. She drew a rectangle in the paws of the mouse and added a dot pattern to the rectangle . In the background behind the mouse Susan drew lines and circles creating a row of cupboard doors behind the mouse. She then jumped to the icons, added a wait icon after the display icon and executed the program. She then opened a display icon that had occurred earlier and edited the text, changing the story. She returned to the icons and executed the program.

Susan jumped to the icons and added another display icon to the end of her loop structure. She named this icon Page 11. She opened this icon and added text to her story.

> **Cheeser could not help himself, so he sat down and ate the huge block of cheese. Cheeser was so full when he was done, that he went straight to bed.**

She then jumped to icons and saved the program as the class was over.

Day 7. Susan began the next class by running her file in AP. She then jumped to icons and opened the one with the image of the mouse in bed under the window with the moon visible. She appeared to check the text by moving her mouse over it. She then closed the icon without changing it. She then opened another image of the mouse in bed but this time the "z z z" was rising from his mouth. She closed the icon. She then moved through the icons on her decision icon until reaching the last one. She opened it and the image of the mouse in front of the cupboard doors was visible. She chose one of the lines forming the image and issued the command send to back. She then closed the icon. She then reopened this icon and pointed to the image of the mouse. She then closed the icon. She moved the pointer to the upper left hand corner of the screen as she waited for the icon to close as it takes a while when multiple images are involved. When the icon had closed she had the pointer in position to click on the close window box which appears in the upper left hand corner. She appeared to have anticipated that this would appear and moved to the location in anticipation of her next action.

Susan now opened the display icon at the end of the list of icons. It only contained the text **Cheeser could not help himself** --. She pointed to the text appearing to survey it. She then opened the scrapbook, selected the image of the mouse sleeping in bed and copied it. She pasted it into the icon. However, when it appeared on the screen it was positioned so only the upper right part of it was visible. The rest of the image was off the screen to the lower left. She attempted to select it by encircling it with the pointer. However she was only able to move the window in this way. She then attempted to circle the entire image but could not. Susan could be seen pointing to the screen with her finger indicating her difficulty. The researcher also pointed with a finger and suggested the use of the select all command. Susan used this command and then 'deselected' (at the researchers suggestion) the text and the window which did not need to be moved. Susan was then able to move the image. She then moved individual parts refining the image. She then edited the text and closed the icon. She grouped it into a group icon and then added a wait icon after the display icon. She then closed the group icon.

Susan placed another display icon at the end of her list of icons on the decision icon. She saved the file and closed it. She then reopened it. Susan placed a sound icon onto the flowchart after requesting assistance in using this icon. Upon opening this icon she was directed by the software interface to choose a sound. She moved through the hard drive to find the necessary sounds. She selected various sounds and leaned close to the computer to listen. She pressed the OK button and returned to the icons. Susan used the chooser to connect to the network and the control panel to increase the volume of sound. She then selected sounds from the network server and played them. She then pressed OK thus selecting the sound of vomiting from the choices provided.

Susan opened the display icon already at the end of the decision icon. She opened it and typed text into it.

> **Good thing Cheeser made it to the bathroom in time !!! From that day on Cheeser never cheated on his diet again.**

She then drew two overlapping squares in the icon. She used the line tool to draw lines connecting the squares into a cube. She then selected the two squares individually and selected a pattern for these sides. She then selected the lines and chose a pattern. The lines became dashed but the cube was not filled. (This apparently was not a useful strategy because the cube was not an object so could not be 'filled' with pattern.) Susan selected the entire set of pieces and deleted them (apparently realizing the problem with her strategy).

Susan drew two ovals overlapping each other. She repositioned these and sent the larger of the two to the back. She then added a square but then deleted all three pieces. She began again with two ovals, one smaller than the other. She positioned the smaller one below the larger one. She then connected the ovals with two lines extending from the larger oval to the small one creating a basket shape (Figure 4-5). She then pulled down each of the pull down menus until she got to the one named Style. She changed the font size to nine point and closed the icon.

Susan reopened the display icon with the basket image. She resized the image several times. She then opened the scrapbook and selected an image of the mouse which she copied into the display icon. She repositioned the mouse. She then attempted to select the entire basket and to reposition it. However each time she selected it only one piece would be moved. She finally selected the entire basket and grouped it into one group of objects. Now she repositioned the entire basket to look like it was sitting in the mouse's paws (Figure 4-5). She then changed the mouse's facial features (perhaps in order to change his expression). She then jumped to icons and ran the program.

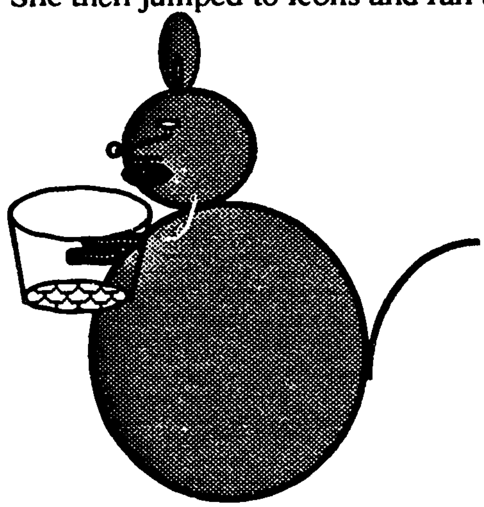


Figure 4-5. Cheeser the mouse needs a basket.

Susan stopped the running of the program by jumping to icons. She placed a flag before the display icon she had just constructed. She ran the program from flag this time seeing the image of the mouse while hearing the sound. She opened the icon and made changes to the basket position and attempted to apply a pattern to the group of objects. The software would allow her to suggest these changes but the image on the screen remained unchanged by them. Susan ungrouped the basket selected only the bottom oval and patterned it. She then added further text to her existing text entry and then jumped to icons.

He also changed his name to Chuck because he would never eat cheese again!

Susan now began the multiple choice question she was required to place in her living essay by positioned an interaction icon below the decision icon. She opened it and appeared. It seems she had expected it to appear. Susan then jumped to icon and opened the interaction icon. The options to be set for this icon appeared. Susan moved the pointer over these choices appearing to survey them. She then deselected the pause before exiting option and then selected OK to accept these changes. She attached another display icon as answer to the question. She then ran the file. She then jumped to icons and saved the file as class was over.

(a) (b) (c)

Figure 4-6. Answers to Susan's multiple choice question.

Day 8. As Susan began this next class she selected the options to connect her computer to the network server. She then opened the hard drive and surveyed her files. She then opened her working file and ran it. But now her move object did not work. She selected show current icon and opened the interaction icon and looked at the icons. She said, "Now it's not working and every other time it was." Then to Logan she said "Holy cow are you ever fast at that."

Logan responded, "What?"

"Your story."

"I know."

"It's taking me forever." Then Susan said to the researcher, "What is this supposed to be because before it was working and now it's not."

The researcher responded, "Okay what's the problem?"

"I'm taking the apple into his hands, to his paws, and this just comes up on the screen." She pointed to the interaction icon options which were open.

The researcher asked if Susan wanted a pause before exiting.

Susan said she did and selected this option. She then opened the display within the interaction and the two click/touch areas (hot spots) were visible. These appeared fine to the researcher and so the icon was closed. Susan ran the program.

"Did it work?"

"This time it did. Oh the return key. Maybe I had erased it before. I don't know what I did. Thank you."

Logan was having difficulty because he wanted to flip his image so it faced the opposite direction. The researcher suggested that was only possible in Superpaint. Susan said she thought she had used an approach by which she chose the image, selected the "black mark" on the edge of the image and dragged this mark across the page. She thought this had enabled her to reverse the image. Logan tried this but was unsuccessful. The researcher suggested that possibly Susan had used this technique within Superpaint and Susan agreed she might have.

resized these areas each to enframe one of the answers and then she closed the icon. She then opened the first answer (apple) and looked at the options to be set.

The researcher said, "Okay. Options. Is it correct?" as Susan pointed to the not judged choice and changed it to correct response. "Okay it exits. Sure" as Susan selected the exit interaction choice. "After next entry, before next entry. When do you want to erase the feedback? You know you tell them they're correct or you tell them they're wrong."

Susan answered, "Before next entry."

"But that will likely make it just flash on [the screen]"

"Okay don't erase."

"Don't erase means it will never be erased."

"We'll just leave it there." as she returned the pointer to the choice after next entry.

"That's usually a safe choice."

Susan selected OK.

While pointing to the screen image of an apple the researcher stated "Now this is the feedback that is going to come up when they get it right. Is this what you want?"

"No. I have to move it."

"Well where should that be?"

"Out of there."

"Well why don't you Select all and Group it and then Cut it out of here."

Susan performed these tasks. She then opened the question and placed the image in the questions.

The researcher asked, "Where do you want it?"

"Right down there." Susan moved the question to the top of the screen.

"Okay. Remember this is the hot box, the click/touch area, which they [users] will touch so put it [the click/touch area] over the apple." Susan resized the box and moved it. She continued by moving the second click/touch area into position next to the first.

"Do I put my cheese in here?"

"All parts of the question need to be in there."

Susan opened the second icon, selected the cheese and text, grouped these together and cut them out. She then opened the question and pasted the cheese into this icon. She then repositioned the cheese and text next to the apple and resized the click/touch area to match the size of the cheese and text. She then closed the interaction icon. She opened the second response which was incorrect and set this option. She then opened the third icon, selected all parts of the image, grouped these together, and cut the pizza image. She opened the interaction icon and pasted the image into this icon. She repositioned it and resized the click/touch area to fit the pizza and text. She closed the interaction icon.

Susan placed a flag before the interaction icon in the flowchart and then ran the file from the flag. She clicked on the apple, the correct answer, and a display icon opened up ready for input. Susan exclaimed, "No!" Susan jumped to icons and opened the pizza answer. She set the answer option to incorrect response and pressed OK. Susan closed the icon and then asked, "Where do I put my feedback?" She then said, "Logan do you know if I put my feedback in the quest ... No I don't" as she opened the answer for the apple [correct answer]. She resized the text style and then proceeded to enter feedback **Correct. You read carefully!!!! The mouse would eat an apple.** She repositioned this text, changed its size, and closed the icon.

Susan opened the second display icon (used for the answer cheese). She entered **No the mouse loves cheese, but the apple is better for him. Click on the apple.** Editing, resizing and repositioning occurred before closing this icon and opening the final display icon used for the pizza answer. She entered and edited the text **No, pizza is not the best for the mouse when he is on a diet.**
The correct answer is the apple. She closed this icon. She then ran the file from the flag so only the question was executed. When she clicked on the apple [correct answer] the feedback she had entered flashed on the screen and disappeared without time to read it. Susan jumped to icons and opened the display icon for the apple. She repositioned the text at the top of the screen. She then opened the next two icons in turn repositioning the text at the top of the screen. However she did not change the options. She ran from flag this time selecting the cheese. The feedback appeared and stayed on the screen until she clicked on the pizza at which point the feedback for the cheese was replaced by that for the pizza. She then clicked on the apple and the feedback appeared and then disappeared immediately. She jumped to icons and opened the display icon for the apple again. The options did not appear. She slightly repositioned the text and jumped to icons. She grouped this display icon and opened the new group icon. Into this icon she placed a sound icon. She then selected sounds and the option to play it 3 times. She ran the whole file from the beginning but after two screens appeared she selected run from flag. She first chose the cheese, then the pizza and then the apple. The feedback appeared while a harp played a note and then the feedback was erased. She jumped to icons.

Susan asked her teacher "Mrs. A this initial project -- what do we need included in it ?"

Mrs. A responded, "Word processing, a story, the animation, a click and drag, a multiple choice. Graphics - import one and make up one."

"Okay. You saw my story the other day. It's pretty well finished then eh?"

"I thought so."

"And I just did a multiple choice today. Okay."

Susan placed a display icon at the end of the flowchart and named it the end. She opened it and selected text styles and then typed THE END in large letters. She then drew a happy face on the screen. But drawing the smiling mouth was a problem. The researcher guided Susan to draw a circle and then place an invisible square over it which blocked the top of the circle from view. The two were then grouped and moved into the happy face (Figure 4-6). She closed the icon and grouped it. Into this group icon she placed a sound icon. She opened it to set the options but then closed it without doing so.

Susan tried to save the file but the message appeared that indicated she had too many icons to be saved in the student version of the AP program. She said "Oh oh."

Bob looked on and responded, "Oh oh. Oh oh."

Susan asked, "Does that mean I just lost it all?"

Logan answered, "I don't know".

Another student asked, "Did it crash?"

Logan answered, "No. It just says she can't save."

Susan then read the error message aloud. "The file cannot be saved because the demonstration version allows you to have a maximum of 50 icons."

Mrs. A suggested, "You need the full blown version."

Susan asked, "So what do I do? I'll cry if it erases."

"Get out of it."

"All I can do is press continue though. But what if it erases I'll cry." There were no other options so Susan pressed continue. She then used the chooser and connected to the network server. "What do I do now. I just keep doing everything wrong."

Logan suggested, "Get out of there."

Now Susan proceeded, with Logan's help, to find the sounds that the others had been using which were stored on the file server. She set the options for sound, turned up the volume in the control panel and then pressed play. She laughed at the sounds, playing them more than once to hear them fully. But she said, "I don't want them I want Daffy Duck." Susan opened the files and found the required sound. "That's what I want." She played it but it was too fast so she reset the 100% speed to 50% and replayed it. She then selected another sound and played it exclaiming, "It's so cute." She pressed OK.

Now Susan executed the file. She said to the researcher, "I'm finished. I just have to correct my spelling and stuff." She moved through each 'page' pointing to the text as she read along. She corrected spelling and edited text. On one page she modified the graphic as it was a bit distorted. She then added an erase icon after the interaction icon and a wait icon. She then attempted to save but could not as the error message recurred. She then tried the interaction icon followed by the end message. However the screen was not cleared in between. Susan said, "No. I'm going to make this thing work if it kills me. I want to finish it today." She then proceeded but as her final sound (an exit message) was played she said, "Oh shut up", in a frustrated voice but then laughed. She then moved to save the file.

After Susan pressed save, the error message about not being allowed to save again appeared. She asked the researcher to help.

The researcher said "Here's what we are going to do. We select these icons and cut them."

Susan selected the multiple choice question and cut these icons.

"Now they [the icons] are in the clipboard." The option to show clipboard was used to make this evident. "Now we are going to save." The researcher pressed save and the file was saved without these last icons. "Now we are going to open a new one [file]."

Susan opened a new file. "What do you want to call this new one?" Susan named the file M.C. (for multiple choice), placed the remaining icons into it and saved it leaving it for the next day as the class was now over.

Day 9. At the beginning of this class Susan opened her main file and executed it. She then opened the file containing the multiple choice question and made changes. It still did not clear the screen. She persisted in adding more erase icons even though automatic setting for this purpose existed. She eventually found the setting for erasing the question, using these settings to solve the problem. She saved these changes.

Susan went back to the main file and executed it. She then closed the file and returned to the desktop. The researcher now arrived and installed the full-blown copy of AP on the hard drive. Now the two parts of the file could be put together in the full-blown package and saved. Each of the two files was opened and converted to the format for the full-blown package. The file named M.C. was grouped and copied into the first file. In this way the two were joined and Susan's project was complete. She copied the file to disk and handed in the disk to her teacher. Susan then began the next unit of study.

Logan

Day 1. Logan was a classmate of Susan's who sat next to her on her left. He had begun his living essay by opening a file in AP. It contained an interaction icon which was used to create a question in the computer file when executed. An icon attached to the interaction icon provided feedback to the response(entry) and served to test a response (entry) through its matching features. Attached to the interaction icon in Logan's file was a display icon. The display icon was a text answer type which matched text entries. Logan opened the options box for this feedback. Options from pop-up menus were available and automatically set to 1) not judged as correct or incorrect, 2) exit interaction (rather than return to the question), and 3) to erase feedback after next entry (rather than before next entry, upon exit or don't erase). When the interaction was initiated and a text answer matched feedback would be provided by this display icon. The three parts of the interaction, 1) question, 2) answer and 3) feedback, could be erased or left on the screen depending upon the options set. In Logan's interaction the question and answer were set to automatically erase when the interaction was executed. The feedback option was set to erase after next entry and exit interaction. The feedback, when within the interaction, would only be erased after the next entry. However when the interaction was exited the feedback would be erased unless the options was set to don't erase.

Logan began to systematically explore the options provided for erasure and exiting beginning with those for feedback. He changed the exit interaction to try again. This meant that when the answer was matched the interaction would not exit but instead return to the question and repeat it allowing another response to be entered. Logan changed the setting to continue. This option meant that when the entry was matched the interaction would continue to try to match other feedback before returning to the question and accepting other answers. Logan mistakenly typed the question **Would you like to see my show?** into the display icon as feedback.

Logan attached a second display icon to the interaction icon. Three options appeared preset to not judged, continue and erase feedback after next entry. He changed the options to exit interaction and erase feedback upon exit. He then opened the display window but closed it again. He then reopened it and typed the text **On With The Show !!!!!!!!!** at the top of the screen. He then closed the display window. He renamed this second display icon Yes|Y|yes|y and renamed the first display icon No|N|no|n. These names also defined the entry characters which would be matched.

Logan then moved the first display icon and placed it behind the second one so the first answer tested was yes and the second was no. He opened the second icon, deleted the text within it and typed **Oh, well you are going to see my show whether you like it or not.** Logan executed the file and the interaction icon was opened automatically. Logan entered the question **Would you like to see my show?** and executed the file. He entered the answer y and the feedback **On With The Show !!!!!!!!!!** appeared with the pause indicator (image of a return key). Logan pressed return and the screen cleared of question, answer and feedback. He jumped to icons and opened the options for the yes answer. He changed the option to erase feedback after next entry. He pressed OK and reexecuted. Once again when y was entered the feedback appeared on the screen and when the return key was pressed the question, answer, and feedback were cleared.

Logan returned to the icons and opened the options for the yes answer but closed them again. He executed the file. Logan spoke to the student next to him "How do you erase the question?"

Bob responded "Which question?"

"Okay watch." He executed the file and entered the yes answer to which the feedback appeared so both the question and feedback were on the screen. He said, "See. How do I erase this now?" while pointing to the question.

"Press an erase icon after it. In the decision [interaction] icon, or whatever, put erase after exit."

Logan opened the options on the yes answer and changed the erase feedback option to the choice upon exit. He reexecuted the file with a yes answer. It ran as before. Logan stated, "It didn't work." He reopened the options and changed the choice to before next entry. It executed as before. Next Logan changed the option to after next entry and reexecuted. Again no change occurred. (Note that when the options are set to exit the interaction the other options before next entry, after next entry, and upon exit all act the same erasing when the interaction is exited.) Now Logan changed the option back to upon exit after some hesitation. Logan had changed all the options to erase feedback finding they acted the same. This was because the other option was set to exit the interaction. Although Logan's explorations were in regard to the feedback he asked "Can I get this question to erase?" He finally changed the option to Don't erase and reexecuted. This time the feedback was not erased but the question was erased as the interaction was exited. It seems Logan wanted the question erased while the feedback remained on the screen because he was satisfied with this result.

Logan next opened the options for the second display icon (no answer) and changed the option to don't erase. He reexecuted with the yes answer. The feedback appeared and the question was erased. He returned to the icons and moved the second display icon No|N|no|n back to the position of being first. He opened this display and closed it. He then opened the interaction icon. Its options appeared and then he closed this icon.

Logan reexecuted with the yes answer. The feedback appeared and after he pressed return the question was erased but the feedback and answer remained on the screen. He reexecuted with the no answer. Feedback appeared. Logan returned to the icons, opened this display icons options and changed the setting from Continuous to Try again. He reexecuted the file with the no answer, jumped to the icons, opened the options for the yes answer and changed them from Don't erase to Upon exit. He also changed the options for the no answer in the same way. He reexecuted with the no answer. The feedback appeared, Logan pressed return and the question and feedback both disappeared but the entry no remained on the screen. Logan jumped to icons and opened the options on the interaction icon. He set the option to erase interaction After each entry. Upon reexecution the question was erased when an entry was made.

Logan then said, "How do you get the answer [entry] to erase?"

The researcher responded (inappropriately to solve Logan's problem) by stating that "There is an erase feedback After next entry, Before next entry, Upon exit."

Logan asked, "Where is that?" The researcher stated that it was in the feedback options. Logan accessed the options which he had just explored and changed the option to erase feedback After next entry. Logan executed the file with the no answer. He said, "It doesn't erase you stupid [to the computer]." He jumped to icons and changed this option again to before next entry. "It's not working." Logan reexecuted and asked, "How do I get my answer [entry] erased? See it's still there."

The researcher responded "Open the question [interaction] icon" which Logan did. "Erase entry Upon exit. That [answer] is your entry."

Logan said, "Oh" as he set this option and pressed OK. He ran the file again with the satisfied statement, "It works."

Now Logan appeared to be ready to move on. He placed a display icon below the interaction icon, opened it and typed:

The Unfortunate Camper
Caught in the Wrong Place at the Wrong Time

He used various font styles. He then said, "All right let's import some graphics here now." He closed the icon and stated, "Actually let's run it first" and executed the file. The question appeared to which Logan answered yes. The title page then appeared to which he said, "Is there any good ...". He chose jump to icons. Logan was distracted by Bob's (the student next to him) watch alarm. They discussed this. Logan returned to his project. He opened the title page display icon and said, "How do I get it to ... Oh yeah", as he pulled down the edit menu and selected effects. He then chose zoom from point from the list of options. Very small text appeared at the center of the screen and gradually transformed to large text seeming to emerge from the center of the screen. He set the options to make the text immovable and executed the file. He answered yes to the question, saw the feedback and then the display icon 'zoomed in'.

Logan next stated, "I'd better get some music here" as he jumped to the icons. He placed a sound icon on the flowchart above the display icon and opened it. He said "I wonder if there is a barking sound?" as he moved through the network server files to find lists of available sounds. He selected a sound and then increased the volume control from the control panel. He then played the sound Macpuke (the sound of vomiting) and said "Yeah that's great," and pressed OK. He then said, "Oh yeah I have to get it so it will run" and repositioned the sound icon after the title page display icon. He opened the options for the sound icon and set the option to concurrent (to make it play while the next icon was executed). He then moved the sound icon back above the display icon. He executed the file answering the question, yes. The sound played as the title page display appeared on the screen. He said, "That's good you know I mean the sound, the title." He executed the file twice, laughing at his sound choice. "All right I'm going to change that sound." He returned to the sound icon, selected the sound "Yabadabado" and then executed the file. Logan said "Perfect. All right."

Next Logan placed an animation (movie) icon after the sound and display icon. Its options appeared. He pressed Cancel and deleted this icon. He placed an erase icon in its place and set it to zoom out the title page. He executed stating "Yup it works." Logan then placed a wait icon between the display icon and the erase icon. He opened the options but made no changes. He ran the file again.

Logan added a display icon to the bottom of the flowchart, named it and other icons above it. He ran the file again and the display icon at the end of the flowchart opened as it was empty. Logan saved the file and closed AP.

Logan opened Superpaint. He asked, "Where did you find the bear picture? Mary where did you find the bear picture?" (Students sat at a row of computers in the order Mary, Bob, Logan, Susan)

"In pictures."

"No kidding but where?" as he moved through the file. "Oh there it is I thought it said beaver." He opened the image of a bear and copied and pasted the copy back into the same file. Logan asked of the researcher "Mrs. Barnes how do you put this in the clipboard?" He raised his hand to get her attention and asked again "How do you copy onto the clipboard?"

"Just choose Copy."

"Oh okay." He copied the image, quit Superpaint and returned to his own file. "Now what do I do? Oh I have to open this up" as he opened the display icon and selected Paste. The bear (Figure 4-7) appeared on the screen to which Logan responded, "All right. Smokey." He then began to speak the words he was typing "**Smokey the bear does not like campers.**"

Bob suggested "We don't need to hear that when you type it."

Logan returned to the icons and dragged a group icon onto the flowchart trying to place it on top of the display icon. It moved above the display. He then tried to place it on top of the display icon again and it moved below the display icon. He deleted it. He placed another display icon at the bottom of the flowchart and asked, "Are there any [image of] human beings in this thing [file server]?" He then saved the file.

Bob asked, "Why don't you group some of those? You could group the whole thing."

Logan dragged a group icon onto the flowchart just below the interaction he created at the beginning of the day. He opened this icon and its flowchart appears. "I don't want to group them all."

"Just select the ones you want to group."

Logan selected each of the icons below the interaction icon and each in turn became black. He then selected one icon and dragged it onto the group icon's flowchart. Only the one icon moved. He moved it back to its original position. "It didn't work." Logan then selected four of the icons (including the interaction icon) at once by clicking on them while holding down the shift key and they all turned black. He asked Bob, "How do I make them group."

"Just say group."

"It didn't work." Logan waited and then asked "Mrs. Barnes I'm first. How do I move all of this into the group icon."

"Select all of it and then under edit say group."

Logan selected them all again and then opened the edit menu and chose the group option and uttered, "Oh." (This time it grouped because all icons in the interaction sequence were selected.)

Bob stated "Logan is an idiot." Logan removed the group icon he had added before and then paused to play with the video camera which was recording his activities.

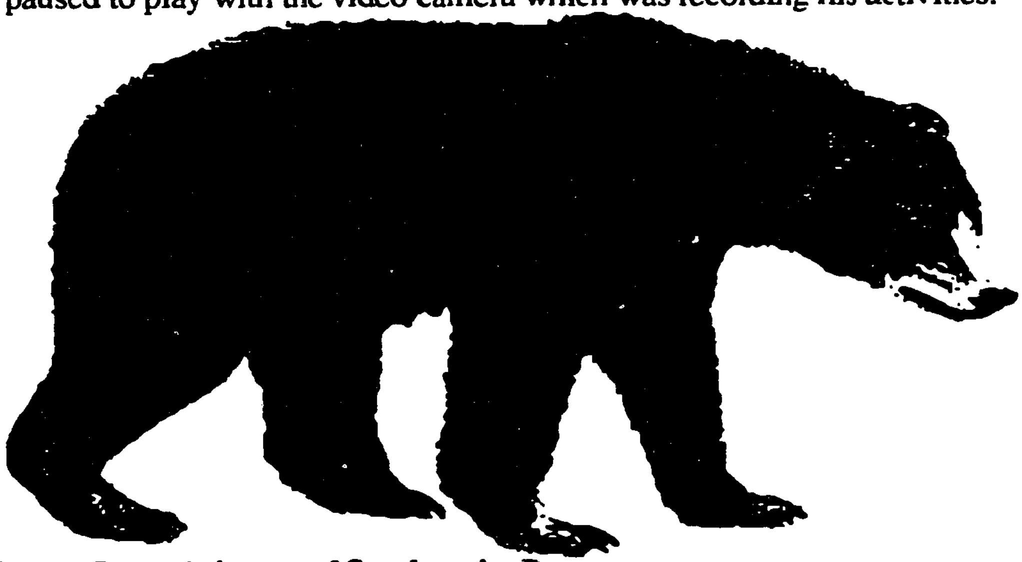


Figure 4-7. Logan's image of Smokey the Bear.

Logan returned to his file and executed it. At the end of the execution the display icon containing the bear image appeared. He jumped to icons and added another display icon asking "Are there any human pictures?"

The researcher responded, "I think there is a snow boarder."

"No, just a normal guy."

"I don't think so."

Logan opened the display icon stating, "Let's draw." He created a face and said "There's Bob"

Bob responded "I'll kill you".

Logan deleted the drawing of the face and saved and closed the file. He then opened Superpaint and asked, "Where did you get that tent from? Mary."

She responded, "In there."

"Okay." He searched the list of files opening a number of them, looking at the images and closing each. He then found an image of a car which he copied. He closed Superpaint and opened his AP project file. He asked the researcher "What are you going to say about us in your report?" but got no response.

Logan opened the display icon at the end of his flowchart and 'pasted' the car image into the icon (Figure 4-8). He typed text under the image of the car as he spoke "**Joe Blow from Idaho went camping one day where Smokey the bear lived.**" He continued to type speaking inaudibly **He was unaware of Smokey's decision to hate campers.** Logan asked Bob, "How do you do that? Do you just move your mouse around?" He then asked, "What do you call it when you hate something ... I don't care for this." He edited the text.

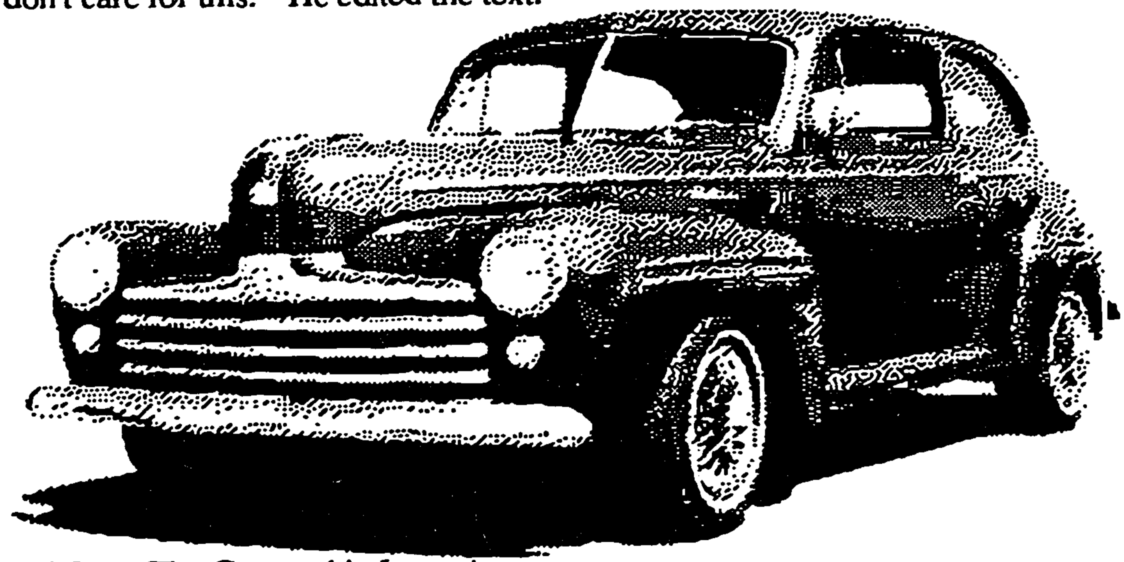


Figure 4-8. The Car used in Logan's story.

Logan placed an animation icon on the flowchart and opened it. He moved the car off the screen to make it appear as if it was being driven. He set the time option so it would move slowly. He then added an erase icon after the animation icon which he set to erase the car image. Next Logan ran the file. The question appeared, was answered and erased. The title page appeared and was erased with the 'zoom out' effect. Then the bear appeared and a display icon opened to be filled. Logan paused and then closed this icon. He asked, "What is this for?" He jumped to the icons and named the icon car. He then opened and closed icons stating "I've got to get this title page to work." He opened an icon and an image of the bear appeared to which he responded, "there's Smokey the bear. Mrs. Barnes I want to make Smokey the bear disappear." He then opened the erase icon after this display icon, set the option for Fade out stating "I like fade out."

Bob commented "the bear disintegrated."

"I'm going to run this." Logan executed the file. He called out "Mrs. Barnes it's not working." He grouped together the display of the bear and the erase icon. He then grouped together his final three icons. He named each of the icons. He executed the file again. "How do you get this thing to wait? Oh yeah a wait button." He opened a group icon and placed a wait icon in before the erase icon. He executed the file again and then stopped it by jumping to the icons. He placed a start flag before the group icons and executed from the flag. "Mrs. Barnes I'm like totally lost on this."

"Why what's happening?"

"Well like --" He ran the program. When the final sequence of the car and text appeared these images began to move across the screen but were erased. Then they reappeared and were erased with the Zoom to point effect. Logan jumped to icons.

"Okay here's the car" pointing to the display icon named car "and here's the animation" pointing to the second icon "and here's the erase. There isn't a pause."

Logan dragged a wait icon between display icon and the animation. He opened this icon and changed the options to allow a keypress and a pause for only 5 seconds.

The researcher asked "Why are you putting five seconds? What if it takes me 10 seconds to read it?"

Logan removed the seconds designation. "What's this?" He pointed to the option show prompt.

"You have to have something to click upon. That is the return key."

Logan said, "Oh okay" and selected this option. He executed the file. Now a return key appeared on the screen and the program waited until return was pressed before proceeding. Once pressed the car began to move but was erased. "It's supposed to go to the corner and then it's supposed to be erased. Like it did it for me once but now I don't know what I did." He jumped to icons.

"Okay two questions. Do you want the car and the text to move or only the car?"

"Well just the car."

"If it's only the car you want to move it must be in an icon separate from everything else. Is it?"

"No." He opened the icon and the car and text appeared in the same icon.

"They are all in one. How could you solve this?"

"Copy this icon and delete whichever."

"and delete the parts you don't need."

Logan copied the entire icon and 'pasted' it onto the flowchart. Now he had two of them. "Which should be first Mrs. Barnes?"

"Well which should go first the text or the car?"

"The text I suppose."

"Well now here's a thought. To display a car is complex and takes time, a little bit but there is a slight pause. The text is quick because it's small. Does that influence your decision?"

"Yes." He opened the first icon and both the car and text appeared and both were selected. "Mrs. Barnes how do you do this?" because both parts were selected. He tried to use the ungroup command but it was not available (gray in color). From the tools on the screen he chose the arrow tool then the text tool and back to the arrow tool. He then used the arrow and clicked on the text. He deleted it. He then closed this icon and opened the second one. In this icon he chose the car and deleted it. He then executed the file from the flag. Now the car appeared followed by the text and the return key. The car was erased and then reappeared and then was erased with the zoom to point effect.

Bob stated, "Our class is over in thirty seconds." Logan saved the file and shut down the computer.

<u>Day 2</u>. As the class began Logan turned on the computer, started AP, opened up his file and executed it. He laughed when the Yabadabado sound was heard. The car and text appeared. Then the car moved and was erased. The car reappeared and was erased with the Zoom to point effect. Logan jumped to icons and reexecuted. When the image of the bear appeared followed by a pause he chose show current icon. He looked at the icon, opened it and then closed it. He moved the flag to just before the group icon with the car. He chose to run from flag and then to show current icon when the execution was complete. He opened the icon containing the text and closed it. He then opened the icon containing the car. He re-labeled these icons to indicate their contents. He then removed the erase and animation icons. He re-executed and then returned to the icons removed the wait icon and

placed an animation icon after the two display icons. He set the icon to move the car to the left corner like it was being driven off the screen. He executed the animation icon and then changed the number of seconds and executed it again. He then jumped to icons and executed from the flag. The car sequence now executed properly.

Logan uttered, "Is there any going away music in that file?" He jumped to icons and placed a wait icon before the animation. He executed the file. The text and car appeared. The return key was pressed. The car moved off to the left of the screen. Logan pointed to the car and said, "Now how do I get this to disappear?" He jumped to icons. He put an erase icon after the animation and asked the researcher, "Would it be here?"

"I don't know. What do you want it to do?"

Logan said "Well okay" and ran the file from flag. The sequence was executed ending with the options for the erase icon appearing and waiting to be set. Logan said "Well the car is moving. Its like well -- This guy is going. Well it's symbolizing this guy is going camping." While pointing to the car he asked "Do I click on this to be erased?" Without an answer he clicked on the car and set the effect option to Zoom to a line. The car disappeared with effect.

The researcher asked, "Is that it?"

"Yeah."

"And now you are at the end of the program."

Logan reexecuted from flag. "Awesome. Now I have to get some music to put to this." He jumped to icons and opened the group icon containing the bear sequence. Into this sequence of icons he placed a sound icon, selected the sound of rain falling, set its options and typed OK. He then renamed the icon and executed the file. Before it could run he selects Run from flag. The sound was heard during the bear sequence not the car sequence. He reexecuted asking the researcher "Why isn't the sound working?"

"What sound?"

Logan jumped to icons. "Okay like here he [the bear] fades out" pointing to the erase icon. "Do I want the sound before he fades out?" pointing to the sound icon placed before the erase icon. "Like how do I get it so that when he fades out the sound goes too? Should the sound be after or before?"

"Before. But then play it concurrently." In other words choose the option concurrent.

"It is" as he opened the sound icon and showed it was set to concurrent. He closed the sound options and executed but the sound was not heard. "It's not working."

"I was thinking it is hitting the stop flag and not playing it."

Logan said, "Oh" as he removed the stop flag. He executed from flag. The sound was heard as the image faded.

"Now here's a suggestion. You've got a whole bunch of pieces just like in that last activity where you put them all together on a loop structure. Do you remember that activity?"

"Yeah in groups."

"Well you've got all of yours grouped already. How about putting them on a loop structure?"

"Don't I have a loop structure already?"

"They don't build to the right they build down. You pull in one of those diamond shaped things remember [the decision icon.]" Logan placed a decision icon near the top of the flowchart below the introductory question. "And you pull all the pieces onto the loop structure." Logan attached the group icons to the decision icon. He opened the options for the first group icon. "You want to erase before next selection" as Logan set this option. "And do you want a pause?"

"No pause." He de-selected this option and pressed OK. He set these same options for the next two group icons. He then opened the options for the decision icon and set them to sequential and don't repeat. He said, "I guess I've got to" as he opened the erase icon after the car. He then opened the icon containing the image of the bear. He was distracted by Bob.

"That's so cool. The bear, Smokey, kills Joe Blow and then he tries to take the car and gets killed."

Logan executed the file. Only the first of the group icons was executed and the program ended. "Why did the stupid thing stop?" He opened the decision icon and at the options he asked "What does sequential mean?" He changed the option from Don't repeat to Until all selected. He ran again and the other group icons were also executed. He said, "No sound" when the bear image appeared. Logan opened the sound icon and checked the options. "What does perpetual mean? I don't want it to play forever." He did not change the option but instead closed the options to return to the icons. He executed the file again. The Yabadabado sound was heard.

Susan asked, "How do you put sound?"

"You have to put a sound icon in there. You drag it out." As the program ran it arrived at the bear sequence which did not play the sound. "It's not working." Logan continued to interact with Susan and then reexecuted the file. He instructed Susan about where to find the sounds on the network. He then selected Run from flag. He asked the researcher "Why isn't the sound working again. I had it working before."

"It is being either erased before it can play or it is being erased as it is being played."

"Yeah but it erases but it doesn't play the sounds."

"Put a pause in after the erase and set it not to have a prompt but to just give you some time." Logan placed a wait icon and set the options by deselecting the show prompt option and entering 3 seconds in the space provided. He reexecuted and the rain sound played. The problem was solved.

Next Logan dragged in another group icon. He renamed this icon and saved the file. He then closed AP and opened Superpaint. In Superpaint Logan opened a graphic of a tent. He cleaned up the graphic by removing the background. He copied the tent image and then returned to AP and his project file. He opened the file and then opened the group icon on the end of his decision icon. He placed a display icon into the group icon and then pasted the tent graphic onto the screen (Figure 4-9).

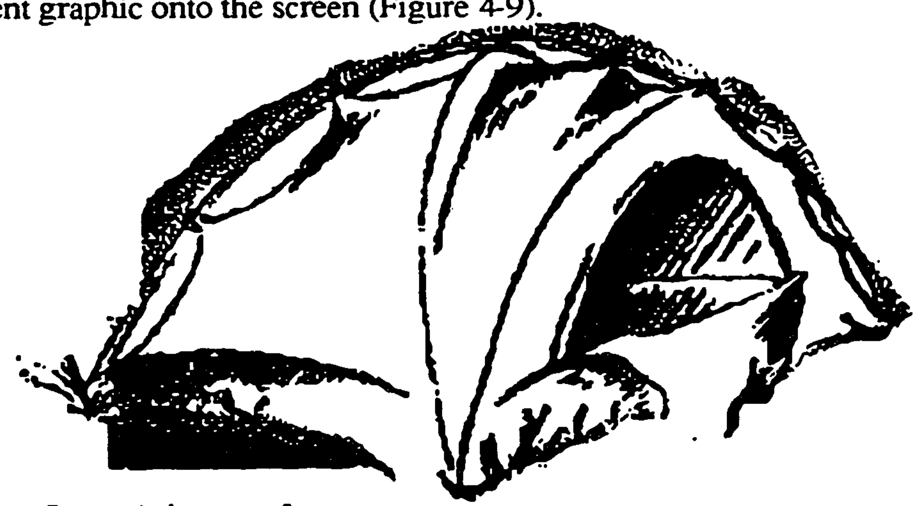


Figure 4-9. Logan's image of a tent.

Logan then added text to the icon contents. He edited the text as he proceeded.

The day was very hard for poor Joe. His boat sank while he was fishing and he almost drowned. Lucky for him there were boaters around, but he [had] no change of clothes so he had to walk around naked for the rest of the day.

Logan then closed, copied and 'pasted in' the copy of the icon. He opened the second of the two identical icons and selected and cut the tent image from the icon. He then closed the second display icon. He renamed the first icon tent, opened it and cut the text from it. He then closed this icon. He then renamed the second display icon text. He then added another display icon after the first two. He placed a flag before the tent icon and ran from flag. The tent and text appeared followed by the empty icon opening to be filled. Logan closed it and his file began to run from the beginning. This was because Logan had run in the middle of a loop structure. Logan jumped to icons and placed an erase icon after the first two display icons (tent and text). He then opened the erase icon to set its options. What appeared on the screen was the image of the first display in the loop structure not the image of the tent or its associated text. Logan said, "No" and closed the erase icon options and returned to the icons. He then opened the text icon and closed it. He then opened the erase icon again and set it to erase the text with zoom to point effect. He then said, "Okay" expectedly and ran from flag. The tent appeared followed by the text which appeared and then quickly disappeared. The empty display icon then opened. Logan spoke to another student during the execution of this sequence so perhaps his attention was distracted from its running. He ran from flag again. He then jumped to icons and placed a wait icon before the erase icon. He opened this icon and set options to pause and wait for keypress. He ran from flag and a return prompt appeared on the lower right side of the screen. Only half of the prompt was visible as the lower half was below the screen. Logan said, "How do you move the prompt?" He then used the screen pointer to pick it up and move it fully onto the screen. He said, "Oh" as he managed this action. He then ran from flag again. The tent and text appeared with a return prompt on the screen. Logan said to Bob, "Here read this. Do you think this is good?" as he executed the file. The empty display icon opened. Logan said "That's it."

Susan asked "Don't you want to erase your tent too or no?"

"Not yet."

"Mine is ten pages long already."

Logan then placed text into the empty display icon.

After his clothes were dry, Joe thought he would cook supper. Unfortunately he forgot something to cook, so he settled for some food that the other campers had left over.

Logan interacted with Bob again but this time not in regard to his project or school work. Logan then continued to type the text. He commented on his story and said, "This is pretty gruesome. Well actually it isn't." He ran the file, then ran from flag in the first frame. He then ran from flag again. He jumped to icons and opened the final display icon containing text. He renamed it and ran from flag again. It did not pause before proceeding to run the first display in the decision icon. "Mrs. Barnes, how come it goes to here?" while pointing to the screen.

"Are you running from flag?"

"Yeah."

"Well that's why. It shows you the inside of the group icons and then goes back to the beginning."

"Okay. I can't run it from the flag."

"You could put the flag before the decision icon and then it would work."

"I was thinking about this. This question is really the first part" pointing to the question icon and its answers which appeared at the top of the flowchart before the decision icon (loop structure). "Could this be placed in a group icon and placed at the beginning of the loop structure?"

"I guess so." Logan selected the question and answers and grouped them using the group command. He then moved the group icon into the loop structure.

"Have you changed the options here?" as Logan opened the options for the decision icon. The options were set to sequential and until all selected.

"If it was set to don't repeat what would it do?"

"It would only run once."

"Okay."

Logan returned to the icons and ran the file. "I want to try it from the beginning." At the bear sequence the bear and text appeared and began to be erased. Logan said "It wasn't supposed to go that fast."

"Maybe you need to put a pause in." Logan jumped to icons and opened the group icon containing the bear sequence. He placed a wait icon into the sequence after the sound icon. He ran the file again and this time a return prompt (return key) appeared which Logan pressed.

"I must have put my pause in the wrong spot or something." He jumped to icons. "It started playing the ... no. I can't remember." He placed a flag and ran from flag. "See it started playing the rain"

"And then it hit the pause so it displayed the return [prompt]."

"And then it does it." Logan jumped to the icons and picked up the wait icon and moved it off the flowline and said "So this pause has to go where. That's the thing."

"It has to go after Smokey [name of the icon containing the bear graphic] so they have time to read. Right?"

"Uh huh" as he moved the wait icon up before the sound icon. He ran the file again and said "There we go. It works. Thank you." He returned to the icons.

Logan was distracted from his programming by banter with other students. He opened the first group icon which contained the interaction icon (with the question Do you want to see my show?). He closed this. He then executed the file but jumped to icons at the question. He then opened the no response to the question. He changed the text in this display. He closed the icon and executed the file. He said, "Isn't this awesome. Smokey the bear ..." He continued running the file. He then said, "Does somebody want to read my story. It's interesting." He ran the file again. As it ended he said, "That's all. I better save." He jumped to the icons and saved the file and then closed AP and shut down the computer.

Bob said "Logan there's still 10 minutes."

"Oh I thought there was five."

The researcher asked Logan, "Could you start it back up again. I need to get a copy of it."

"A copy of what?"

"Your file." Logan opened AP again and his file.

"Where do you want me to copy this? In Authorware [Professional]?"

"Yeah. Oh no at the desktop." Logan closed the file and returned to the desktop. He copied his file to a diskette and shut down the computer.

Day 3. On the following day Logan opened his file in AP and 'pasted in' an image of the bear but the bear was now facing the opposite direction. He had obviously been in Superpaint and had reversed direction of the bear image and then copied it before opening his file. He then executed the file. The tent sequence ran but at the end of it was now the addition of another picture of the tent overlaying the first and then the bear appeared. Logan jumped to icons and placed an erase icon before the second tent image. He then placed a flag and ran from flag. But the flag had been placed too late to execute the sequence and set options for the erase. So he jumped to icons and moved the flag to the beginning of the sequence. This time as he ran he set the erase icon options but then canceled these settings instead of pressing OK. He then returned to the icons and removed the display icon containing the second tent image. He ran the sequence again 3 times. On the third sequence he set the erase icon options. He then ran again. He then returned to the icons and removed a display icon. He ran the sequence again. He started to set the erase options but then again canceled them. He jumped to icons and added another group icon to the loop structure. He ran from flag. He then jumped to icons and ran again. He jumped to icons and opened the display icon at the end of the flowchart. The bear image appeared. Logan selected the bear and repositioned it on the screen. He then added text speaking the words as he typed: **The same night Smokey was hungry so he went searching for food.**

Bob spoke to Logan and Logan responded in regard to another student named Tanya. Logan continued to type and speak, "**Along his way he saw Joe's tent. So he decided to investigate.**" Logan then set the effect option for this display but then canceled it. He then jumped to icons and closed the group icon. He opened the new group icon at the end of the loop structure but closed it and reopened the previous one saying "This is cool. I like this." He then opened the last (empty) group icon again. He placed a display icon and opened it. He 'pasted in' the image of the bear. He then jumped to icons and opened a previous group icon. He then reopened the first icon in the bear sequence which contained the image of the tent. He copied the image of the tent and closed this icon. He opened first the group icon at the end of the loop structure and then the display icon within it. Into this icon he placed the image of the bear and an image of the tent. He then jumped to icons.

Susan (sitting at Logan's right) was having difficulty saving her file. Logan said "Oh oh."

Bob asked "Did it crash?"

Logan responded "No. She just can't save her file." Logan opened the tent and bear sequence in the second last group icon. He opened an icon with the bear and then closed it. He then opened the last group icon and the display icon in it. He added the text **He investigated** to the display of the tent and bear.

Logan then gave Susan directions about how to navigate the network, attaching to the MacB file server in order to find sounds which she could place in her program. He helped her select sounds and play them.

Logan asked, "Mrs. Barnes I need help. How do I do this click and drag thing?"

She responded "Click and drag? Click and drag what?"

"I want the person to click on the bear and drag him to the tent" as he pointed to the screen containing the text, the image of the bear and the tent image.

"Oh that's a drag object question."

"Yeah."

"How do you make a drag object question?"

"I don't know. That's why I'm asking you."

Susan suggested, "You just made some."

The researcher responded, "But that was two or three weeks ago."

Logan agreed, "Yes", and jumped to the icons and pointed the mouse pointer to the display icon provided in the list of icons. He asked the researcher "In the display icon? Yes? No?".

"No. That's not a question [interaction] icon." Logan dragged in an interaction icon. "Well where is your bear?"

Logan responded, "in here," as he pointed to the display icon. Logan then opened the clipboard displaying its content was the image of the tent.

"Where is the tent?"

"They [text, images of the tent and bear] are all in one [display icon]."

"Okay. They have to be in separate icons remember." Logan copied the display icon and pasted a copy of it.

"Should the bear come first or the tent?"

"You put the bear and tent up first before the question." Logan moved the two display icons up.

"And in the question you tell people what they are going to do." Logan then opened the first display icon and removed the tent and most of the text. He then opened the second display icon and removed the text and bear. He then labeled the icons bear and tent respectively.

"Which is going to be moved the bear or the tent?"

"The bear."

"Well in the icon with the bear there is text. The text will also be moved. Is that what you want?"

"No."

"Where do you want the text?"

"Before"

"Before or ?"

"After."

"Or it could go in with the tent couldn't it?"

"Yeah." Logan removed the text and bear from the icon accidentally. He selected Undo and the objects reappeared. He then selected only the text and deleted it. Logan returned to the icons and opened the icon with the tent image. He typed **He investigated.** into this screen. He returned to the icons and opened the interaction icon. Into this icon he typed **Click on bear and move into the tent** as he spoke these words. "Okay now do I go like --"

"Now when the bear moves what type of response [feedback] do you want?"

"If they don't put it in the tent it will snap back and say put the bear in the tent."

"Okay so it's going to be text."

"Yeah."

"So it's going to be a display [icon]."

"Yeah."

"So that's the wrong answer. How about the right answer?" Logan dragged in a display icon and placed it next to the interaction icon. He set the options to text answer type.

"Could you open that back up. Okay what kind of answer? Change response type" as Logan opened the options for the answer and chose change response type. "Are people going to type in text?"

"Oh no. Click/touch area."

The researcher listed the answers available: "Are they going to touch an area; are they going to pull down a menu; are they going to press a key; are they going to move an object?"

Logan responded "Move object" and selected this option. An options window for this answer type appeared.

"Okay it says drag object to the target position except the objects aren't on the screen yet. Maybe while we are here" and pointed to the options to be set. "You just told me this was a wrong answer and it's going to snap back." Logan set these options and returned to the icons. He dragged in another display icon and was asked by the researcher, "so what is your wrong response going to be?" Logan renamed the two icons correct and wrong. "Do you remember that thing we did about order of correct and wrong answers?"

Logan answered, "If you go try again it goes back. So the right one would have to be first." Logan moved the correct answer to be first and opened the options. He set the icon options to correct response, exit interaction and then said "I don't get this thing about erase feedback."

"The feedback is the text or whatever you display after they get it right. When do you want that erased? Before they drag it again, after they drag it again, or."

"For the right response?"

"For the right response the question has already been answered so what would be logical?"

"Upon exit." He set this option, jumped to icons and opened the wrong answer icon.

"When its the wrong response they are going to try again. So when do you want it erased?"

"Before next entry." He set this option.

"Okay we'll try it."

Logan jumped to icons and opened the icon containing the image of the bear. "So what do I do" as he highlighted show current icon.

"Jump to icons."

Logan opened the icon for correct response rather than the options for this icon. He closed the options and opened the interaction icon. "No. Where are the options for the right answer." Logan opened these options and the screen images of the tent and bear appeared. Logan began to move the tent.

"Are you moving the tent?" (He had earlier stated that he did not want to animate the tent. He wanted to animate the bear.)

"Yeah."

"So move this [the options box for the animation icon] out of the way so you can move it [the image of the bear] to the correct spot." Logan moved the options box out of the way. "So grab the bear and move him to where you want him to go." Logan moved the bear to a position with his nose in the tent. "So whenever the bear is dragged over this little region [area to be matched] it will be right [correct]. Is that right?"

"Yeah."

"So now move this [the options box] back and say OK."

Logan asked, "But don't I want to make this a correct response?" as he selected the OK choice accepting the options set. He returned to the icons and pointed to the second display icon on the question icon stating, "Okay now wrong response."

"You do the same thing."

"Oh okay." He moved to point at the display icons placed before the interaction icon which contained the bear and the tent. "Do I have to open the bear and the tent?"

"Just the bear would be okay."

Logan opened the bear and closed it and then opened the wrong answer options stating, "wrong response," as he set this option. Logan pointed to the box indicating the active region "Do I have to make it across the whole screen?"

"Yes"

Logan made the box frame the screen by first placing it in the upper left corner and then stretching it to fill the screen.

"One of the things you have to do is tell it that it is the bear that you are moving so you have to click on the bear."

Logan did this and the indicator box moved to be centered on the bear but no longer filled the screen.

"Now the two are snapped together. Move the bear so the hot box fills the screen."

Logan moved the bear repeatedly down and to the left moving the indicator simultaneously. After several attempts Logan stated, "Ah that's good enough"

"Lets just stretch the box to reach to the corner."

Logan stretched the box and pressed OK.

"Okay run it. Oh run from flag I guess."

Logan placed a flag before the sequence and executed from the flag. When the question appeared Logan moved the bear to the correct position. The display icon for the correct response feedback automatically opened awaiting content. Logan immediately selected the text tool and typed **You have done well.** Logan opened the wrong response feedback icon speaking the typed text "**Please put Smokey's nose in the tent.**" Logan closed this icon and returned to icons. He then opened the display icon containing the tent. He repositioned the text and returned to icons. He then saved the file and quit AP. The class was over.

Day 4. As class began Logan executed his file from the flag while in AP. When he moved the bear to a position not near the tent (incorrect answer) the feedback, **Please put Smokey's nose in the tent.**, flashed and disappeared. Logan opened the options for the incorrect response and changed the erase feedback to After next entry. He executed the file again and moved the bear again to an incorrect position. The feedback appeared and remained on the screen. He then moved the bear to the correct position but the incorrect response still appeared.

Logan stated, "It's not working anymore. It worked once and then nothing. It wouldn't work for me again." Logan opened the icon containing the bear. He closed it and opened the tent. He then opened the interaction icon. "Is the whole bear supposed to go in this spot [the tent opening] or just his nose?"

The researcher responded, "No. The program tests the center of the entire bear not just his nose."

"Oh, so I can't just test his nose?"

"No you test the entire object but you can test where his nose is."

"How do I do that?"

"You have to open the correct answer icon or [rather] the options for that." Logan opened the options. "Now put the bear exactly where you want it." Logan moved the bear to the correct position. The researcher pointed to the indicator box on the screen. "This box [click/touch area] is very big and its going to test the entire area. When you move the bear as long as some part of the bear touches the box it is going to be a match."

"Oh, okay."

"You probably want to shrink the box [click/touch area] a little bit and then click on the bear." Logan changed the box to a smaller size and clicked on the bear but the tent was under the bear so it was selected instead. Logan chose the bear by clicking on an area of the bear which was not over the tent. Logan reduced the box size again and then moved the bear to have its nose in the tent. "What you are really doing is testing the center of the bear" as I pointed to the indicator box.

"Oh yeah." He then clicked OK and executed from the flag.

"You can test how off your answer can be by moving the bear somewhat close but not exactly." Logan tried this. The bear was not close enough to match "So how close do you have to get?"

"Pretty close."

"Is that what you want?"

"Yes."

"So it's really the size of that hot button [click/touch area] that matters."

"Uh huh." Logan executed the entire file. He watched the program as he verbally interacted with other students around him. He got to a point that the story on the screen had been cleared but the tent was still on the screen with a return key prompt at the right of the screen. He selected show current icon and a wait icon was highlighted. He then opened the icon which followed it which was more text and the image of the bear." Logan then opened the icon with the bear moving to tent sequence. He then closed the icon and returned to the tent icon. Logan executed the file again. He continued to interact with another student in regard to an upcoming school dance. He executed the file again getting to the point that text appeared without an image. He cut the text and reexecuted. Now this same icon was empty, open and waiting for content. He showed this icon. Logan now closed the file and did not save his changes. He then reopened the file. He opened the tent icon and then the incorrect answer for the move bear sequence. He then opened the bear and then the tent. He then opened the interaction icon. He attempted to select the hot button but could not. He jumped to icons and opened the interaction icon again. He then moved the hot button but could not move the bear. Logan then executed from flag. Moving the bear to the tent opening matched the correct response. He then executed the entire file. The tent with the return key appeared again as before. He placed an erase icon and then opened it and set the options to erase the tent. Logan placed a flag and executed the file from flag. This time it executed properly except that the file did not pause at the last entry in the story. Logan moved this last icon to the next group icon which was the group icon containing the question. He executed the file but it did not pause or erase. He placed wait and erase icons after this display. Now it properly executed from flag. He jumped to icons stating, "I'm tired."

Logan stated, "I'm starting a new thing now." He placed a new group icon and added a display icon to it. He then went back to his previous group icon and opened the image of the tent. He was interrupted by a student near to him who wanted to remove a pause. He suggests that she, "show current icons. No not jump [to icons]." He then returned to his own work, copied the tent image and returned to his new display icon. He bumped the video camera so paused to reset the camera. At this point he quit for the day.

Day 5. As this class began Logan had executed his file in AP. He opened an image of the bear and copied it. He opened his new display icon and placed a copy of the bear image next to the tent image. He placed the bear with his nose in the tent. Logan read as he typed the text below the images "**Smokey the bear growled. Due to Joe's weak heart he died.** Well that's it." Logan jumped to icons and placed a wait icon. He then placed an erase icon and set it to erase the display icon. Logan then edited the text in the display icon so it read: **Smokey the Bear was pleased at what he had found. Joe was not pleased. Smokey the bear growled at Joe and due to Joe's weak heart he had a heart attack and died.**

Logan jumped to icons and showed the clipboard which contained the tent, bear and text. He then placed another display icon in which he placed these items. He deleted the text and jumped to icons. He then opened the original icon with the bear, tent and text. He closed it again. He spoke to another student to whom he suggested, "Run this" but the student did not. Logan deleted the icon with only the bear and tent images. He placed another display icon and added the tent, bear and text. He deleted the text. From the original icon Logan removed the text and then reset the erase icon to erase the text. He then opened his second text based display icon and moved the text. He executed the file. It did not pause so the final text could be read. He placed a wait icon at the end of the sequence, added another display icon, and executed the file.

Logan's teacher watched the program. She asked about the answers which were tested (Yes|Y|yes|y and No|N|no|n) in the text answer question early in the execution. At the request of the executing file the teacher moved the bear to the tent but couldn't achieve a match. Logan paused the execution, moved the tent to the proper position and then allowed the execution to proceed. This time the teacher achieved a match. When she got to the end of the file and a display icon opened. Logan said "I have to finish that." Logan typed the text **Since Smokey was hungry he thought he'd have lunch. Since**

our great country believes in censorship the author has to ask ? into the icon as the teacher left him to help Bob.

Logan then placed an interaction icon into his file and added a display icon as the first feedback to the interaction. Logan then opened the interaction icon and typed: **Do You want to see the remains of Joe's Body?** He then asked, "How do you make a correct response any answer? How do you make a wrong any answer? Logan opened the display icon options and set the text answer to * which is used to accept any response provided. He then executed this list of icons. He placed an erase icon at the end of the list. He executed the file again and used the pause option from the pull-down menu to stop execution to move the question to the bottom of the screen and then to proceed. He answered the question. The feedback icon opened and Logan typed **Sorry, the scene is too horrific to let anybody see.** Logan tried to place a wait icon but the flowchart was too short so the wait was dropped to another flowline. He deleted it. He then stretched the size of the screen window so the flowline was longer. He then placed the wait. He executed this group icon.

Logan attached another group icon to the decision icon and added a display icon into the group. He asked, "How do you make a cross? I can't make one." Logan then used the box drawing tools and drew two rectangles, one vertical and one horizontal. He then selected both and changed the lines to darker. He then selected different modes but canceled his selections. "Mrs. Barnes how do I get rid of this part of the line" pointing to the places where the rectangles intersected across the middle of the cross. Logan clarified that he wanted the image to look like one image of a cross rather than two rectangles.

"Draw another box exactly the same size as those ones [indicating the box formed by the intersection of the rectangles]. You are going to overlay it you see." Logan drew a rectangle. "Now say [choose options] lines and select white [option]. It doesn't quite cover it [the lines] so stretch it [the rectangle] a little bit." Logan chose lines and then selected the white color. The rectangle became white almost covering the black lines. He stretched the lines a little so the black ones were no longer visible.

"Yeah. That's it." Logan then chose text, changed the font size and then typed **R.I.P** on the cross. But the text, being encased in a white region, overlaid the cross and hid some of it. "It's covering."

"What do you have to do?"

"Make them smaller" as he changed the font size.

"Or make them transparent." Logan selected modes and selected transparent. He increased the font size. Logan then added more text to the cross (Figure 4-10). He selected the text and changed its mode to transparent.

"There you go. There is my illustration. Mrs. A is that good enough for my illustration? Is there any death music in here?" He then executed the file. "I can't really use sound for mine. Oh that's all folks." Logan placed a sound icon at the end of his file and selected the sound: "Ta ta for now." But then said "Naw". He then selected the sound "That's all folks" and said "Yeah. That's what I want." He then closed all open windows but the main file flowchart. "Well where is my question?"

Bob asked, "Are you done?"

"I have to put sounds in first. Oh I forgot I have to use a puking sound." He executed the file. "That was awesome." He then opened the sound icon and changed the setting to 50% speed of execution. He then changed the pause to last longer, executed the file again, and saved the file. But the error command which did not allow saving more than 50 icons appeared on the screen. "Oh great" in a frustrated voice.

The researcher responded, "All is not lost. Don't panic. How much have you added today?"

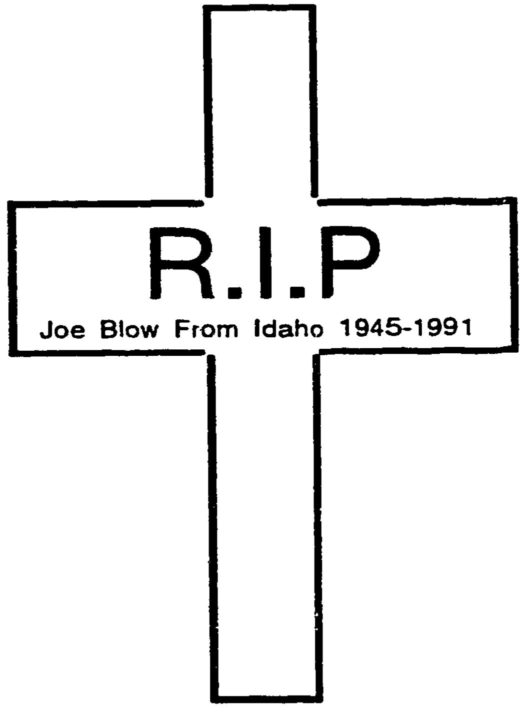


Figure 4-10. Logan's cross with text commemorating Joe Blow's passing.

Logan opened the last group icon saying "That" and then opens the second last group icon and said "That too."

"These two we are going to cut to the clipboard." Logan selected the command cut. "Let's undo that and see if we can cut only one of them [and still save]." Logan selected Undo and tried to save but again the error message appeared. The two icons were again cut and then the file was saved. A new file was then opened and the two group icons were placed in it. He saved this new file.

"Are you finished then?"

"Well basically."

"Just some cleaning up to do?"

"Yeah"

"Next day we'll put them together." Logan quit AP as the class ended.

Day 6. As class began the computer was on. Logan opened his folder containing his files. Logan's two files for his final project were visible. He opened the second one and executed it. But because the group icons were no longer attached to a decision icon the automatic erase and pause were not present. Logan added a pause and erase to the first group icon. He executed the file. He then added a display icon to his list of icons. He copied the bear image from the scrapbook and placed it into the display icon. Logan added text **After having Joe for supper Smokey drove back to the cave to have a sleep. Unfortunately for him he never woke up due to a hunter shooting him and using him in his house for a bear rug.** Logan then set the zoom from line effect for this display. He opened the icon with the image of the cross and copied its contents. He then added another display icon and placed the cross image into it. He changed the inscription on the cross to **Smokey Bear-A legend in his own time**. Because the text was now longer it covered the cross. Logan attempted to use fills and lines to change this but could not find modes. He then shortened the text by removing the word own. Logan jumped to icons and placed an erase icon in the file which he set to erase the first display with a fade out effect. Logan executed the file again. Some images overlaid each other so Logan placed an erase icon in the list to erase the first cross image. He asked the researcher, "Okay now how do I get all my project in one?"

"You've saved all of it?" Logan saved the file. "You've error checked all of it and it all works?"

"I think it does."

"What I would do is error check both parts to be sure they are perfect and then I'll help you put them together."

"Okay." Logan executed the file. He then added another display icon after the first cross image. He added text to this icon. But now this text did not erase when he executed the file so he added another erase icon and set it to erase the text. He then executed the file again. Logan added a sound icon to the file, selected a sound and executed the file. "Let's run the stupid thing." Logan added text to his file "**The hunter was mutilated by Smokey's mother** ..." using the names of two classmates Bob and Mary in the final text.

After these entries Logan was finally finished. The full-blown AP package was loaded onto Logan's machine. He opened each of the two files and copied the contents of the second file into the first one. He then had one large file. Logan then made a few more changes so that the two executed properly when in one file. He then saved the file. He also packaged the file so it would 'stand alone' without AP. He made a copy of this stand alone packaged file to send to his aunt in Seattle so she could see his work.

Findings and Outcomes

These two cases have provided information for considering the nature of students creations, the patterns and techniques they have used, and insight into the interaction between the student and the computer environment. Students' activities and interactions were different for each of the two students and varied through the course of their interactions with the AP environment.

As Susan began her multimedia production she was already talking about it as a story. She was planning the format of the story with text and graphics. She began with an image of the main character of her story, Cheeser the Mouse, and a storyline which was about a mouse eating cheese. However, when she attempted to draw she found it impossible to create the shapes she wanted in AP. Although she had the options to use an outside drawing tool, Superpaint, she found it difficult to use. She changed the story to accommodate these drawing restrictions and made her story about a mouse eating pizza. Through this example the software enframed Susan's story and production of it by its restricted drawing capacity.

The next challenge Susan set for herself was the creation of a move-object interaction. She stated that she did not "know where anything is. I'm so confused." A common difficulty in any programming task is to know the extent to which a the program code has been executed when in the midst of it. Another difficulty in any kind of computer code is to know which code has produced particular results upon execution. AP is an icon-based programming language with icons rather than text-based programming (written line-by-line with syntax and semantics). One reason for using icons is to alleviate these difficulties. For Susan it may have reduced but not eliminated the problems. AP uses a flowchart on which to place icons. This flowchart implies the execution pattern by its structure. However one must be able to adequately interpret the flowchart to extrapolate the execution pattern. Susan seemed not to have mastered this interpretation task although she was still able to create a file. In a traditional text-based programming task Susan likely could not have proceeded if she did not understand the structure. In this case she required assistance before proceeding as well. Is this icon and flowchart interface really an advantage over text-based interfaces as is often claimed? It is a question worthy of further study.

Susan often executed her file in order to determine if it was free of errors. Upon this execution it was often found that it did not run as expected. In such cases the file was often reexecuted, sometimes more than once, in order to determine the location and nature of the difficulty. It seems that Susan's understanding of her file was procedural in nature as she needed to see it reexecuted to interpret it. In AP the program always executed although not necessarily as planned. Thus there was immediate feedback to programming efforts and the potential for questioning of one's activities. The task for the student then became determining where her/his interpretation of how this would execute differed from the actually execution pattern. Features within AP which assisted Susan in the search were the commands show current icon, pause, and jump to icons which she used more extensively when difficulties arose. These assisted in locating the icons in question and allowing the investigation of options set and flow paths defined.

On day 3 Susan had incorporated a movie which played repeatedly without end. Susan wanted this movie to play a finite number of times. She first tried to solve the problem by placing erase icons after the movie icon. They would not execute because the movie would not end. She then surveyed other icons which followed the movie icon to see their contents and structure but she was unable to see any changes she could make. She then surveyed the movie icon options but made no changes. She seemed unsure as to which of these (if any) might affect the desired results. She then again placed erase icons following the movie icon. Finally she asked the researcher for assistance and erase icons were suggested. Thus far four approaches to solving this problem had been attempted.

At this point Susan paused from solving the problem directly at hand and asked the researcher if it was possible to execute only the portion of the file which she was attempting to error check. The idea of using flags was suggested. Susan then was able to make use of these flags in attempting to solve the problem of the continually turning book. However, Susan took another side venture renaming each of the icons so their contents were clear without opening them. This would also assist Susan in error checking her work as she would easily be able to locate items in the flowchart. However in this renaming process Susan destroyed an icon and had to find a way to recover it. She created another problem to be repaired before returning to her initial problem.

Finally Susan returned to the turning book problem. She checked the options for the decision icon to see if they could be affecting the way her program executed (or did not in this case). She also checked options on the erase icons and changed these options. Finally she opened the options on the movie icon and set them to play the movie only 3 times. This was a partial solution. However Susan wanted the book to turn until the user pressed the button on the mouse input device. She acquired the researcher's assistance but the approach suggested was not successful. Susan accepted the partial solution of playing the book 5 times and providing a return key for users to press when they had finished reading.

In this problem solving sequence seven strategies were attempted to solve the problem: 1) place erase icons, 2) survey icons following the movie, 3) survey movie icon options, 4) ask the researcher, 5) check decision icon options, 6) check erase icon options, and 7) set movie icon options. These strategies were enacted through moving about in the environment and checking options of various types or through trying a strategy and finding it was not successful. Consulting the researcher is considered another form of interaction within Susan's environment and was used as a strategy by both Susan and Logan. In solving the problem other sub-problems were solved which assisted Susan in better addressing the main problem. She solved the problem of how to run only a portion of her file and renamed her icons thus assisting her to better handle error-checking. Susan created a problem when she accidentally eliminated one of her icons. She had to recover this icon before she could proceed. Solving sub-problems, or other problems that emerged as a result of her actions, was not only evident in this problem solving sequence but in others as well. The creation and elimination of problems was a continual process for Susan as she created her living essay.

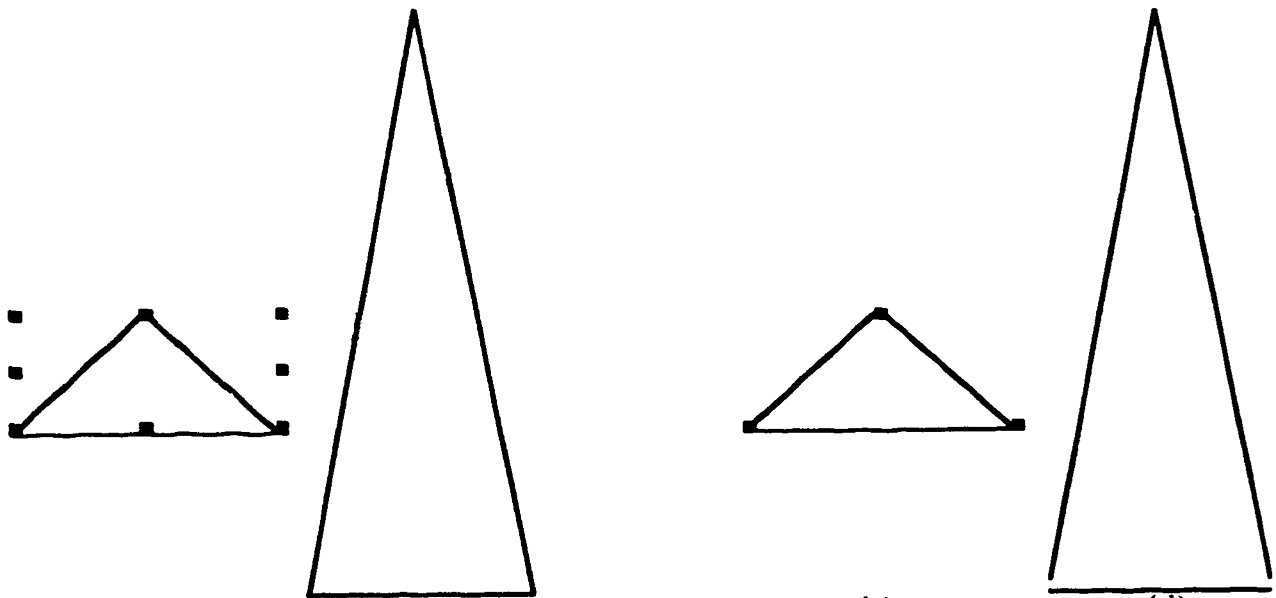
Susan created images to accompany the story text of her living essay. She was able to do this using drawing tools in AP and the pencil tool in Superpaint. These tools simulated the use of a pencil or pen in drawing but also allowed for the drawing of specific images like rectangles and ovals without the addition of extra drawing apparatus like a compass or straight edge. This meant she did not have to enter code in order to accomplish her tasks. However this also meant that to draw objects (other than rectangles and ovals) like triangles in AP they had to be constructed from lines. Susan tried to fill the triangle with a pattern. Although the software would accept the attempt to fill it would not execute this command. Susan then grouped the lines and attempted to fill the triangle again but was unsuccessful. Susan did not realize that the construction of three separate lines was not an object and thus could not be filled. However, her attempt to group the three separate objects into one showed some beginnings of this idea.

The action of grouping of three objects did not create one object but only a collection of objects. However the software indicated an object by placing small black markers on the periphery in a rectangular arrangement around an object. When one grouped objects into a collection the same black markers were used to indicate a collection of objects rather than a single object. The interface was to be questioned. A collection of objects could be acted upon and moved as one entity. However this did not indicate an object. It was clear that Susan realized after her first efforts that she needed to have one entity in order to act upon it. However it seems she assumed that grouping would create an object which could be filled. This she discovered was not the case. The ambiguity of the interface was a problem for her. Using the same indicators for groups of objects and single objects was confusing.

Susan drew an oval to represent pizza. She drew small ovals within it to represent toppings on the pizza. She changed the size and shape of the pizza and toppings together. She also changed only the outer oval in one action. Through these actions one infers that Susan had begun to struggle with the difference between an object and a group of objects. In Figure 4-11a a triangle is drawn as a single object. The lines in this triangle cannot be edited individually and cannot be deleted independently of the other lines. When this object is stretched Figure 4-11b results. If a triangle were created using three separate lines as in Figure 4-11c it could be grouped into a collection of objects. It would then look like Figure 4-11a. However, when this collection of three lines is stretched in a similar manner as the triangular object (Figure 4-11b) the result appears in Figure 4-11d. Notice how the bottom line has become disconnected from the other two lines. The triangle is no longer intact due to this change in orientation of lines to one another. This is a simple example with only three lines. This effect can be far more serious when the initial graphic construction is more complicated.

Susan had used the drawing tools in AP to create a rectangle as an object and fill it. Using the rectangle tool enabled her to act upon that object as a single entity allowing her the possibility to fill this object with a pattern. Clearly AP can identify the region bounded by the rectangle and thus can fill it. When Susan tried to create a triangle and fill it no such tool was available. She created three separate lines and grouped them not unlike the example in Figure 4-11. However she discovered that when she attempted to fill the group of objects she could not. The group of objects failed to act like a single object even though the screen indicator of a single object and a grouping of objects appeared to be the same.

The screen interface breaks down because the same indicator is used for more than one meaning. Users of computer interfaces like AP's need to understand that (a) an object maintains orientation of its components and (b) actions must occur in regard to the entire object and cannot occur in regard to components of the object. A grouping of objects on the other hand may or may not maintain its orientation when edited and individual objects in the grouping can be acted upon if ungrouping is done first.



(a) (b) (c) (d)

Figure 4-11. (a) Triangle created as a single object,
(b) Triangle (a) after 'stretching'
(c) Triangle created with 3 single lines,
(d) Triangle (c) after 'stretching' as in (b).

Why is it important for users of graphic interfaces and software to be aware of the distinctions between object and grouping? For any particular task the use of an object may be more beneficial than a group of objects or vise versa. An understanding of this distinction can assist students in being more functional in generation of images and the use of them once created. Mastery of the tools requires the acquisition of concepts compatible with the tools. Susan demonstrated a lack of compatible concepts and the difficulties this can create. Also, by understanding the tools the products of the tools can be understood. Is the output of a graphics package one object or a series of parts grouped together? Anyone who has ever been involved with desktop publishing or transfer of a graphic from one platform to another will likely have experienced the frustration of the destruction of graphics because the new platform would not maintain the orientation of a grouping of objects. However when converted to a single entity the object can often be transferred with success.

Susan experienced this frustration when she attempted to transport her mouse and cheese. She could not seem to move both pieces. Even when the two appeared to be in the clipboard she could not manage it. One difficulty may have been that she used an older version of Superpaint (which may have created this problem). Another difficulty seemed to be the limited memory capacity of the computer through which she attempted these actions. A third difficulty was the format of the graphics images. The particular format she used kept the pieces of the image as individual objects. She grouped them together in order to move them. In retrospect, perhaps creating one object rather than a grouping of objects would have solved the problem.

Susan's actions of first creating a graphic object and acting upon it demonstrated the feature of object as one entity. When she created three objects and grouped them they appeared to be a single entity so she assumed she could act on them as one object. But despite her action of grouping they were still three objects, each created individually, rather than one object. Susan then demonstrated that changes of size, shape, and location of a collection of objects can be performed in her example of drawing pizza. But notice that there was no attempt to group these objects but only a selection of a number of distinct objects.

Through these explorations Susan has explored actions upon individual objects, a collection of objects and grouping of objects. Some actions, like filling a geometric object with a pattern, can only be applied to objects. Such can be applied to an individual object or a collection of objects (like filling a collection of 3 circles each with the same pattern). But the action of filling cannot be applied to fill the space between three distinct objects as was the case in this student's actions. So the action upon objects and collections of objects is consistent but it is the generalizing of these actions to groups of objects as if they were one object which is problematic.

Susan not only drew her own graphics but she connected to a file server and the contents of a hard drive in order to access public domain graphics as well as sounds. The process of connecting to these peripheral devices was confusing to her, as it was initially to Logan. The process of navigating information sources is an important skill in a computer environment. The sharing of information on-line can be facilitated by network connections between hard drives. However the information sharing between Bob, Logan and Susan was person-to-person, between Bob and Logan (in regard to finding images and sounds on the hard drives) and then between Logan and Susan (so she could also access these files). Communication within a computer environment can occur between users about the computers and network, or can occur through the computers in regard to information stored on the network. The first usage sees the computer as the subject of study while the other usage makes use of the computer as the source or medium through which information is acquired and shared. The computer is a destination for the storing of information. The computer is also an environment through which information is generated as in the case of these students stories within computer files. But, what is this information in this case?

Susan had been writing her story and creating her computer file in the process. She reached the point that she could no longer add to the flowchart as it was full of icons. She needed to restructure her flowchart in order to continue. This task took her most of a class to accomplish. She commented, "I didn't hardly do anything this whole class." During this class she had not created new "pages" of text and graphics in her file so she had created little new information. On the other hand, as far as generating new computer code, she had changed the structure of the file significantly and thus had generated or varied code extensively. But to her this was hardly doing anything. It seems that for Susan the primary task is to create story. Creation of computer code and structure within the computer file is secondary and only the means through which the primary task is accomplished. 'Information' for Susan is story extensions.

As Susan proceeded with her production efforts her style of interaction with the software seemed to change. Early in her production she would decide what was to be done and then struggle with the accomplishment of the task. She would have to figure out how to generate a graphic, create an interaction, or display-pause-erase sequence. However later in her production efforts the task of generating seemed simplified. She added a group icon, placed a display icon within it, and added text and graphics to it. Finally, she added wait and erase icons and the sequence was complete. She then produced another sequence. She seemed freed at this point from the struggle with the software (medium) and able to almost automatically deal with it. She was then able to concentrate on her characters and the development of story.

What is the process that enabled this change? It seems that Susan had begun to see a pattern of activity which she could use in creating in AP. Although she initially had difficulty with AP, through her activities she became able to function within the interface. She seemed to have found a pattern of interaction which was functional. Her pattern of activity seemed to have been generated through actions from a background of non-patterned activity. Varela, Thompson, and Rosch (1991) describe olfaction (the sense of smell) as an example of the relation between perception and action. This description seems to propose a method by which Susan's initial activities emerged into an organized pattern of activity.

> Over many years of research, Walter Freeman has managed to insert an array of electrodes into the olfactory bulb of a rabbit so that a small portion of the global activity can be measured while the animal behaves freely. He found that there is no clear pattern of global activity in the bulb unless the animal is exposed to one specific odor several times. Furthermore, such emergent patterns of activity seem to be created out of a background of incoherent or chaotic activity into a coherent attractor. ... Smell is not a passive mapping of external features but a creative form of enacting significance on the basis of the animal's embodied history. (p. 175)

In Susan's case, she began interacting with AP based upon some of the information she had seen in the instructional simulations (IS's). This initial exposure to the features of AP provided her with a framework for acting. However, her actions were not determined by these IS's but were based upon her perceptions of the IS and her perceptions of AP. She began to take action in AP and experienced an inconsistency between the output of the program and the output she expected. (The movie kept on running). She devised strategies to solve the problem. One of these was looking at the options in the movie icon. However she did not initially see how to change these to solve her problem. In fact she had to return to checking options repeatedly before she finally began to change the options and thus take action to solve the problem. Initially these options were not possible solutions to Susan but later they became the place to solve her problem. From the "background of incoherence" came a "coherent attractor." Varela, Thompson, and Rosch (1991, p. 176) suggest that "cognitive structures emerge from the kinds of recurrent sensorimotor patterns that enable action to be perceptually guided." Susan demonstrates perceptually guided action through repeated attempts at solving this problem she has posed for herself.

> Piaget laid out a program that he called *genetic epistemology*. ... The [newborn] child begins with only her sensorimotor system, and Piaget wishes to understand how sensorimotor intelligence evolves into the child's conception of an external world with permanent objects located in space and time and into the child's conception of herself as both an object among other objects and as an internal mind. Within Piaget's system, the newborn infant is neither an objectivist nor an idealist; she has only her own activity, and even the simplest act of recognition of an object can be understood only in terms of her own activity. Out of this, she must construct the entire edifice of the phenomenal world with its laws and logic. This is a clear example in which cognitive structures are shown to emerge from recurrent patterns (in Piaget's language, "circular reactions") of sensorimotor activity. (Varela, Thompson, and Rosch, 1991, p. 176)

Susan has begun to demonstrate coherence in her activities with AP. From her increasingly coherent activities we can infer that her cognitive structures are changing through her actions. Her initial image of the movie icon may have been to simply play a movie. That the movie needed to be stopped was not an issue. Perhaps it was assumed that the movie would stop. Or perhaps the issue of stopping was not evident to Susan. However when she ran the program and the movie failed to stop the issue became quite evident to her. The issue of stopping a movie was now part of her cognitive structure.

Another instance of Susan's perceptually guided action was evident when she found that three display icons had the same difficulty. When the program was executed the display icons did not pause before being erased. The story created was satisfactory but the execution of it was not and did not allow the reading of the story. Susan grouped the first display icon into a group icon, added a wait icon to it in the position after the display. She executed the file. The text in the display now could be read. She followed the same strategy for the last two display icons, grouping and adding a wait icon to each. Susan recognized that the same problem existed for each icon. She devised one solution which was repeated for each. In this case she had to step away from concentrating upon the story to solve a story execution problem. She was then able to return to the creation of her story.

In this example of Susan's activity it seems that her story development was punctuated by AP production skills in her later efforts. She had found a means of file production that allowed her AP strategies to become automated and thus moved into the background of her concentration. She was then able to concentrate upon the story and characters which were in the foreground. Another example of this automating of production occurred when Susan was working with the multi-part image of the mouse in bed sleeping. The icon containing the image took a while to close. While Susan waited the few seconds she moved her screen pointer to the upper left corner of the screen in anticipation of the close window indicators appearance and her subsequent selection of it. The software and interface no longer stood between Susan and her story, separating her from it. However it did not facilitate the production of the story either. The interface simply became the space in which the story was created. Susan, through her actions and history of interaction with the computer software, brought forth a world which was occasioned by the sphere of behavioral possibilities. The computer software, by its design and structure, made particular possibilities available to Susan. She selected from these and acted in accordance with her selections.

Susan created a multiple choice question about the best food to eat to lose weight. This question and answer sequence used an interaction icon. Although it may be termed an interaction it is not the sort of full communication one would expect between two persons. Through the software's functions certain types of interactions (example: question with text answers, drag-object questions, and click/touch answers among many others) are facilitated. These software functions encourage the use of such interactions. However there are many types of interactions which are possible between two parties which go far beyond the mere asking of questions and receiving of answers. The lack of functions to facilitate these additional types of interactions eliminate or reduce the likelihood of these types of interactions.

When one interacts with a computer one is interacting via a computer interface. The user of the computer has the one face. Is the computer supposed to possess the other face? Occasionally throughout their interactions Susan and Logan did talk to the computer as a pseudo-other but these instances were rare and usually when the file did not execute as desired causing frustration for the student or when a sound or image appeared that was pleasing. It seems that the computer 'possessed a face' when students expressed emotion when interacting with the computer. However these instances were infrequent.

The interface, however, is an important part of interacting with a computer. The user makes sense of the computer via the interface. The interface structure determines the actions which are possible and so occasions students actions within the context of the computer environment. Logan, for example, began his production of the living essay by interacting with the interface and trying to determine the effect of setting the options. These actions were occasioned by the structure of AP and Logan's interaction with it.

Susan looked to her teacher in determining whether she had completed the task as the teacher defined it. The requirement of a picture imported from the hard-drive was a problem. It seemed that these images did not 'fit' into Susan's story. She had to construct another sequence in her story to meet this requirement. When Susan felt that her story was completed she checked with the teacher to determine that she had met all requirements of the structure of the story. The teacher confirmed that Susan had the required functioning components to meet the requirements. These requirements were simply to construct a story which contained graphics, sound, a multiple choice question, an animation or movie and a text story or essay. Some elements of story, namely character and plot, were included by students on their own initiative. Students determined that these elements were necessary for a story to be told.

The subject of Susan's story is one of a mouse who needs to lose weight. When he is successful he is rewarded with presents and "feels really good". With all the controversy about adolescent female body image and lack of self-esteem in mass-media today it is interesting that Susan chose this topic. Susan was an attractive 16 year old at the time of this study seeming to be confident and self-assured. Did her story reveal something of herself that was not evident to the observer? Perhaps this story revealed something of Susan's history which she brings forth in the writing of her story. "We exist in the present; past and future are manners of being now." (Maturana & Varela, 1987, p. 241)

Logan's story had subjects of nudity, death, and revenge. Perhaps Logan's experience with mass-media also had influenced this work and his history had become evident through it. Logan's story may have been a bit of a 'soap-opera' with the unusual characters of bear and hunter. As Logan began this production his attention was focused not upon the story or characters but upon the options to be set to erase screen displays. He explored the options to erase the feedback to a question by systematically changing the option, executing the file and then changing it again. Logan seemed to discover that when feedback icons were set to exit the interaction the options to erase feedback before next entry, after next entry and upon exit all functioned to erase the feedback upon exiting the question. Only the don't erase option functioned differently keeping the feedback from being erased upon exit. However Logan stated later that he did not understand the erase feedback features. It seems strange that if only two functions are possible that all four are available for selection. Perhaps the interface is again a problem confusing the user in production of an interaction in AP.

Logan continued to explore options for erasure of the question, the feedback, as well as the answers entered. Through his interactions Logan explored the ways this software interface 'worked' to accomplish the tasks of erasing screen displays when involved in interactions. Early in these explorations it was unclear whether Logan distinguished between the three types of entries in an interaction which were the question itself, the answers entered, and the feedback to the answers. However he was able to find a solution to his difficulty by interacting with the software and exploring different possibilities.

Logan generated this experience for himself based upon a perceived difficulty he was having. Perhaps these activities address the question: How do students create experiences for themselves from which they may learn and construct understandings of the computer-enhanced environment in which they are immersed? What is the nature of these interactions? In Logan's case he expected the execution of the file to have a different outcome than what occurred. His response to this difference was to explore systematically until he determined a solution which produced an execution compatible with his expectations. However, from his later comments, it was likely that he did not understand why this new compatibility had occurred. A computer teacher could create activities to provide opportunities for students to experience circumstances for the systematic exploration of computing environments. However such activities alone may not enable students to understand the circumstances and generate functional patterns of behavior. Perhaps the opportunity for students to discuss these difficulties and their explorations with other students or the teacher would enable students to better understand their findings. Through such an approach students would be able to use their history of interaction (findings) as a basis for interactions in novel circumstances which students might recognize as having structures similar, but not identical, to those already experienced.

An example of a student discussion which may have assisted student understanding of procedure occurred in regard to the grouping of icons. Logan wished to group a number of icons together. Bob suggested he do this and further suggested to Logan a method to achieve this task. "Just select the ones you want to group." But Logan was unable to group the icons on his first attempt. He then selected all the icons he wanted to group. Bob stated that he should just select the group command, however, Logan did not trust that this would work now as it had not in the last try (because he had not selected them appropriately). The researcher made the same suggestions that Bob had made "select all of it and then under edit say group." This time however Logan tried using the group command and was successful in grouping the icons. It was unfortunate that the researcher was present as the 'expert' to confirm Bob's suggestion because had Logan tried this suggested procedure directly he would have confirmed it for himself through his actions.

Logan's difficulty in the above examples can be characterized by a difference in student perception of what should happen and what he experienced to happen. This led Logan to try different strategies in achieving compatible results. Strategies are frequently generated and explored through students' actions. When the student interacts with other students or teachers and executes the file it assists in confirmation and generation of strategies. By devising strategies to solve their own problems, students can enact a history of activity. Through these actions students enact a domain of possibilities which enables them to solve their own difficulties.

When Logan wished to create a move-object interaction he was clear about what it was he wanted to do. The bear would be dragged to the tent and the bear's nose would be placed into the opening of the tent flaps. This motion was to simulate Smokey the Bear's looking for food. To accomplish this task required the production of a complex series of procedures resulting in a complete question. Logan wanted the researcher's assistance with this task. It appears that the nature of the assistance was to 'chunk' this complex procedure into manageable pieces which he knew how to accomplish. Perhaps the instructional simulation should have used this approach in teaching the creation of a drag-object question thus facilitating Logan's remembrance or re-conceptualization of the procedures.

In viewing the interaction of the researcher and Logan in regard to the creation of the drag-object interaction the researcher would ask questions. Logan would answer the questions in words or he would use the screen pointer to point to an icon, open options, or perform an action. Logan would sometimes begin to ask a question or make a statement and finish it with a pointing or selection action. Thus his actions with the screen pointer became gestures used in the conversation.

In the conversations a language developed among participants and with the researcher through interactions to describe procedures and parts of AP. The names of options were referred to like objects. For example, Logan stated, "I like fade out", in regard to the option set to erase the bear image. He later stated, "If you go try again it goes back." Meaning if one selected the option try again it would result in the question being asked again. When reading these accounts it may be difficult to decipher the context specific language of these interactions. The language developed used the names of options as names of objects. This language usage was not formalized but rather was grounded in the interactions within the context.

Logan interacted with classmates, the teacher, and the researcher in getting feedback about his file. On a number of occasions he asked for feedback about his production. It seems he had several different purposes to these requests. In some cases he wished to demonstrate his project of which he was proud. For example he said, "Does somebody want to read my story. It's interesting." Logan also wished to determine if the project was effective. He said to Bob, "Here read this. Do you think this is good?" In another case he asked a student to "Run this" but the student did not. The teacher ran it instead but encountered some difficulty with the execution. Logan 'fixed' these difficulties by pausing the execution and then continuing.

When Logan began to write his story he chose the character of Smokey the bear using an image of a bear from the public domain images available on the hard drive. He also chose images of a car and a tent in designing the plot of the story. It seems that the availability of these images mediated his choices and the subsequent story. Logan wanted to have a character go camping. He had a tent image to assist him in illustrating this story. To create the action of 'going' Logan chose a car to stand for Joe Blow and animated the car off the screen. The conflict between man and nature was described but from the bear's point of view. The bear was initially victorious in this conflict but succumbed to the dominance of man. However, Smokey's mother took revenge on the hunter and mutilated him perhaps demonstrating the power of nature. The final results were two crosses, one for man (Joe Blow) and one for nature (Smokey).

Conclusions

An extended case study approach to the data allowed for the development of an in-depth understanding of the students' computing concepts and an examination of some of the activities which brought each student to her/his level of competence. Views of how students conceptualize and construct a task in the AP environment were revealed in the analysis of recordings of the students' projects. Student competencies in use of skills expanded and developed throughout the final project phase.

The focus of student attention during the development of the living essay was at times the creation of a story or essay. Students seemed immersed in the characters and story. The development proceeded with little regard to the computer medium in which the development took place. At other times there seemed to be more concern with the development of skills and concepts of the medium in which the production was occurring. The work was skillful or exploratory in nature as students sought to understand the medium. The story and characters were secondary to the goal of understanding the medium.

As Susan began her living essay she typed text introducing her main character and his conflict. She paused to wonder about structuring her story and then proceeded to create images and text of her story. Logan, on the other hand, embarked upon the production of the living essay by examining features of AP. He created an interaction, asking the user of his file if they wanted to proceed. He was immediately embroiled in concerns about the structure and function of AP. Susan also encountered difficulties with her production (in regard to Superpaint) but was able to resolve these difficulties with the researcher's assistance and then she proceeded. In the face of the limitations of the environment in which she worked (drawing limitations of AP and the difficulty importing graphics from Superpaint) she brought forth a different story. Logan proceeded to build his story after receiving assistance from the researcher. He introduced the story conflict and then concerned himself with effects options in AP and the introduction of sound into his story. These initial activities of these two students indicate a difference in their orientation as they embarked upon the production of the living essay. Susan's story seemed to be in the foreground and her main focus. Problems with the software were background which occasionally emerged and forced her to pay attention to the computer elements. The story seemed to be the setting through which she encountered the medium. Logan's activities indicate that the medium (AP) in which he created the story was prominent as he embarked on the living essay production. The setting seemed to be prominent with the story being developed through the medium.

The graphics selection or generation also seemed to emphasize the prominence of either medium or story. In Logan's case he selected images which were available on the file server. He built a story around the characters in these graphics. Logan only drew graphics because one was required for completion of the assignment. He drew the simple cross graphic and copied it so he wouldn't have to draw it again. Susan generated graphics to provide images for her characters. She redrew these graphics to illustrate the developing story. She sometimes drew these graphics from scratch, even though they might have been copied because she had difficulty with the copying procedure. Susan only selected graphics from those available on the file server because they were required of her for completion of the assignment. She accomodated this requirement by altering the story and adding another character. In Susan's case she generated a story from her history and structure enacting the medium through the story. Logan, immersed within the medium generated a story enacting the sphere of behavioral possibilities available through the medium based upon his structure and history of interactions (some within the medium).

A distinction in how students think and produce in the computerized environment of AP can be made in comparing the processes used early in the development and later in productive efforts. In later development efforts both students began by placing text or graphics on the screen. The file was executed. Based upon this viewing (and possibly as an extension of it) the next step was created. The file was reexecuted and the process of development continued. This has been named the iterative model as it involves the reenacting of a simple cycle, but the cycle never is exactly the same and never produces the same results. A model of this development is provided in Figure 4-12. Note that the outer activities encompass the inner activities. In this model the student-author views past efforts and then builds upon them in an iterative way by moving further from the starting point but incorporating the starting point as the work proceeds.

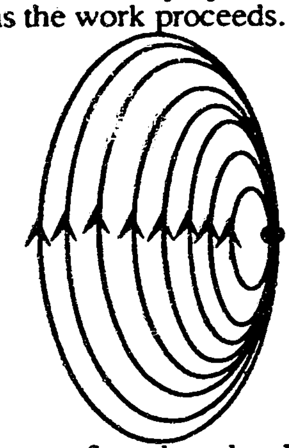


Figure 4-12. An interactive pattern of product development.

During early development sequence, particularly for Logan, something interesting was created and saved (his initial question). Another seemingly independent product was created (the title page with zoom effect). A further seemingly unrelated sequence was created (first story sequence). Upon return to these sequences, editing occurred. Each one may have been based on ideas that appear to be generated from information in the hard drive, graphics, sounds, or other students. The sequences were grouped together into group icons. The title page and story beginning icons were placed upon a decision icon. However, the initial question asked remained outside the decision icon unconnected to the rest of the production for a while. Eventually this question was grouped and added to the decision icon but only at the researchers suggestion. A model of this developmental style appears in Figure 4-13.

In Figure 4-13 we see a number of different developments each revolving around an idea (shown by the •). Some of these developments are related (indicated by the dashed arrows) but one is shown here that is apparently unrelated to the others and is isolated from the others. Logan was observed working on a program sequence which he did not link to the others but worked on in isolation from the others. These others were opened, closed, reopened, and seemed cognitively joined by the action of working on all of them simultaneously but the one program sequence was not incorporated, thus, it is shown in Figure 4-13 as isolated from the others. This style of development may be thought of as a parallel processing model of development as it appears to be work on more than one project or idea simultaneously.

These two styles of product development may indicate students conceptions of the process of development. By comparing these two styles it is apparent that students think about production differently at different times in the production. It seems that at times students are building strings of information while at other times they are building collages.

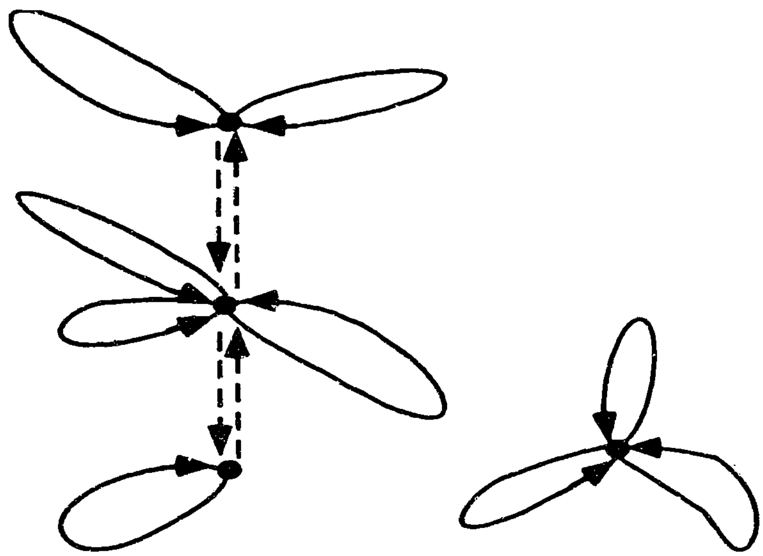


Figure 4-13. A parallel processing model of development

Both students were involved in the activities and made use of the IS in the process of concept development. Using concepts developed, students could act upon them in the activities and further develop concepts. The skills and concepts developed enabled students to generate interactive products while in an interactive mode. Figure 4-14 is a model of this development. In its center are the patterns and processes that Susan and Logan used in developing products and thereby developing concepts. Students using the IS and completing the activities actively developed concepts, moving from IS to activity to IS to activity.

When these students completed these activities, and sometimes before this completion, they began working on larger productions based upon their own ideas, upon activities and programs they had created, on teacher or other students suggestions, upon images and sounds available in public domain file, etc. Each began a process of using AP to create a final project. The ideas for the stories and the impetus for the developments came through students actions. Topics seemed to be drawn from the personal interests of students and from images and sounds available from public domain files on the hard drive. They created stories of fiction, each devising a program based upon their ideas of what the final project should be: whether a teaching program, a story, a comic strip, an animated short story, a children's' book, a video game, a programming exploration, etc. ... (see Chapter 3). These productions were created when students began to have the skills and concepts to express themselves in the computer context and based upon their ideas. It appeared that students were enabled to develop these multimedia products.

Students' efforts in the creation of the living essay were interactive with the medium. Other students, the teacher, the researcher, sounds and graphics on the file servers and the computer itself all provided students with interaction possibilities. The student interacted with the medium based upon prior experiences (history) and their conceptions of the medium (structure). The students and medium codetermined students' actions. The medium occasioned a sphere of possibilities. Students acted within this sphere based upon their history and structure. Students activities and efforts in this medium were productive. Students generated products through their actions and interactions. They also generated concepts and skills through their activities and interactions. These 'products' were evident to the observer.

In viewing students' interactions with the software the files produced were executed. Results expected by students and the actual execution were not always the same.

The student interacted with the software to determine why they were not the same and to make the execution meet their expectations. The students, by their actions, attempted to create understanding of these situations. In the words of Maturana and Varela (1987):

> The business of living keeps no records concerning origins. All we can do is generate explanations, through language, that reveal the mechanism of bringing forth a world. By existing, we generate cognitive "blind spots" that can be cleared only through generating new blind spots in another domain. We do not see what we do not see, and what we do not see does not exist. Only when some interaction dislodges us -- such as being suddenly relocated to a different cultural environment -- and we reflect upon it, do we bring forth new constellations of relation that we explain by saying that we were not aware of them, or that we took them for granted. (p. 241-242)

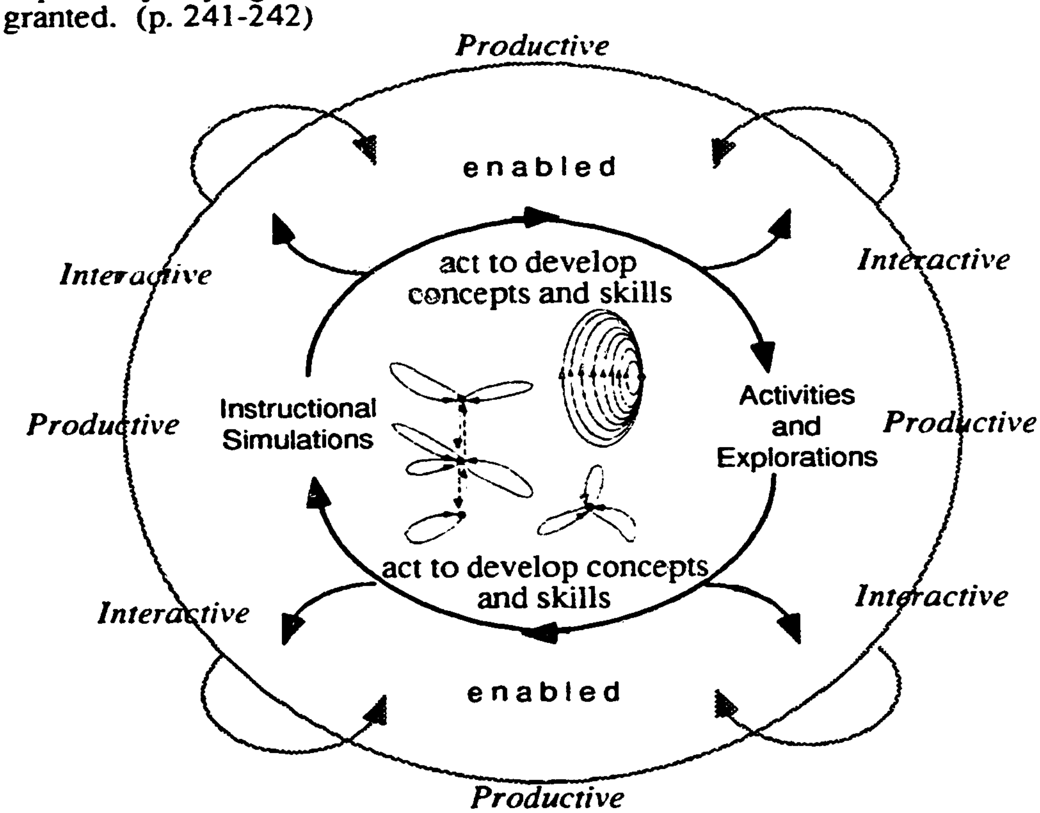


Figure 4-14. Patterns of conceptual development and enabling of productive and interactive actions.

Evidence of students construction of new concepts and invoking of historically based concepts is hard to find using traditional methodologies and representation techniques for data. By exploring student's constructions in a computerized environment, through in-depth video recordings in an intensive case study approach, the computer and the context of its use can be viewed from a new standpoint: as an expressive environment where students create multimedia products. But more importantly they construct a view of computing based upon personal experiences and personal meanings. These findings have broader implications than the computer environment mediated by AP and are applicable to other computer contexts in which the computer can be used for expression of ideas and creation of personal meaning. Further study is needed to uncover these meanings and to explore these notions at a deeper level. This study simply uncovers an opportunity which until now has remained untapped.

Implications

The findings of this study are twofold. They are methodological in that methods have been developed to 'capture' the essence of students processes in the enaction of concepts and skills. They are also substantive. These findings support the idea that enactive approaches to knowledge building seem to operate in this interactive, dynamic, and visual computer environment. Students have constructed concepts that are productive through their iteractions.

The study addresses a need for understanding and knowledge of the developmental nature of student-computer interactions. It focuses upon students' interactive and productive actions and the ways icon-based programming (authoring) languages can provide students with a sphere of behavioral possibilities (environments) through which students enact skills and concepts which are productive and interactive. The data suggest that computer research needs to focus not only on the products of student computer efforts but on the processes they use in creating these products and the student concepts enacted through these processes. By explorations of this kind enactive ways of viewing the interactive and productive processes of students in a computer environment emerge. The articulation and development of an enactive stance toward computer-education research may become the future task of this field.

Commentary

Chapter 4 concludes the section of the dissertation devoted to explorations in the domain of AP. Chapters 2, 3, and 4 have provided insight into 1) students' understandings of concepts within this environment, 2) students' interpretations of their actions within this environment, and 3) students' activities and iteractions within this environment. The environment (the context in which the study played out) was important to students' activities and interactions. It occasioned students' actions in a particular direction. The role of the environment within these human-computer interactions has been described.

To provide this dissertation with a broadening of view it is important to now view another series of activities and interactions in a different environment. The environment was constructed through the researcher's interactions with a computer, readings, and mathematical learners/researchers. This environment is intended for the exploration of mathematical concepts and thus occasions activity to this end. This environment is very different from that of AP. The two environments taken together provide the impetus for viewing human-computer interaction from an enactive stance which will be further described in Chapter 7.

Chapters 5 and 6 provide insight into a mathematical exploration environment. Chapter 5 overviews its features and functions and provides the researcher's view of some of its usefulness and limitations. Chapter 6 describes and interprets students' explorations of this environment.

Chapter 5

Simultaneous Graphics Presentations in the Study of Non-Linear Dynamics

Interaction with computers enables students, teachers and mathematicians to explore complex ideas of mathematics, in particular that of dynamic systems and chaos theory. The purpose of this chapter is first to describe some mathematical concepts in the study of dynamic systems and chaos theory. Secondly, two computer programs used for exploration of these concepts are described. Finally, the chapter discusses the types of graphics inherent in these computer programs and their values and limitations in regard to developing understanding of the mathematical concepts. The chapter specifically describes the graphics used by the computer programs and considers the features of each graphic which are most salient. These graphics are then considered together and the value of simultaneous presentations for exploration of mathematical concepts is discussed. Finally the limitations of use of these computer graphics is discussed.

Explanation of Mathematical Concepts

Iteration. To iterate means (in broad terms) to repeat a process over and over again. In the study of mathematical dynamic systems, the process that is iterated (repeated) is the evaluation of a mathematical function. A function is defined here as a rule that converts a set of numbers into another set of numbers. It is an action or process that changes numbers each time it is applied or invoked. An example of a function that is common in mathematics is the x^2 function. To experiment with iteration of x^2 use a calculator, enter 2 and then press the x^2 key. The result is 4. If the x^2 key is pressed again, the result is 16. When the function x^2 is again invoked by the keypress, the result is 256. This is an example of iteration of the x^2 function in which we entered 2 as the initial or seed value. The answer or output was shown to be 4 after invoking the function. The output of the previous action (key press) is the input for the next step in the iteration.

The Logistics Equation. A set of functions important to the study of non-linear dynamics and which has been explored extensively by mathematicians is known as the Logistics Equation: $y = \mu x(1 - x)$. By substituting different values of μ into the equation different graphs of the iteration result ,with each graph dependent on the seed value x_0.

Iterate Graph. The type of graphing to be called Iterate Graph is generated by plotting the value of the iterate on the y-axis against the number of the iterate on the x-axis. This graphing technique provides a plot of the values of each iteration. However, these graphs can become vary large as the number of iterates grows. Figure 5-1 is a diagram of an Iterate Graph produced from the iteration of the function $y = x(1-x)$ with the seed value of .5 and iterates 0.25, 0.1875, 0.15234375, 0.12913513. The ordered pairs graphed are (0, .5), (1, 0.25), (2, 0.1875), (3, 0.15234375), and (4, 0.12913513).

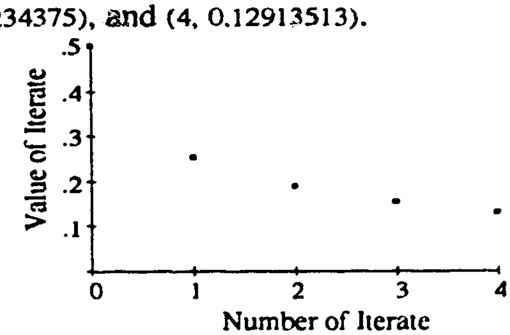


Figure 5-1. Iterate Graph of $y = x(1-x)$ with seed value $x=.5$.

Cobweb Graph. A Cobweb Graph (Devaney, 1990) is a special graph which traces the iteration of a function. This technique involves 1) drawing the function to be iterated and the line y = x, 2) drawing a vertical line from the x-axis to the graph of the function at the specified seed value, 3) drawing a horizontal line from the upper end of the last line to the y = x line, 4) again drawing a vertical line from the end of the last line to the graph of the function, and 5) again drawing a horizontal line to the y = x line. The process of Cobwebbing repeats the drawing of vertical lines to the graph of the function followed by horizontal lines to the y = x line.

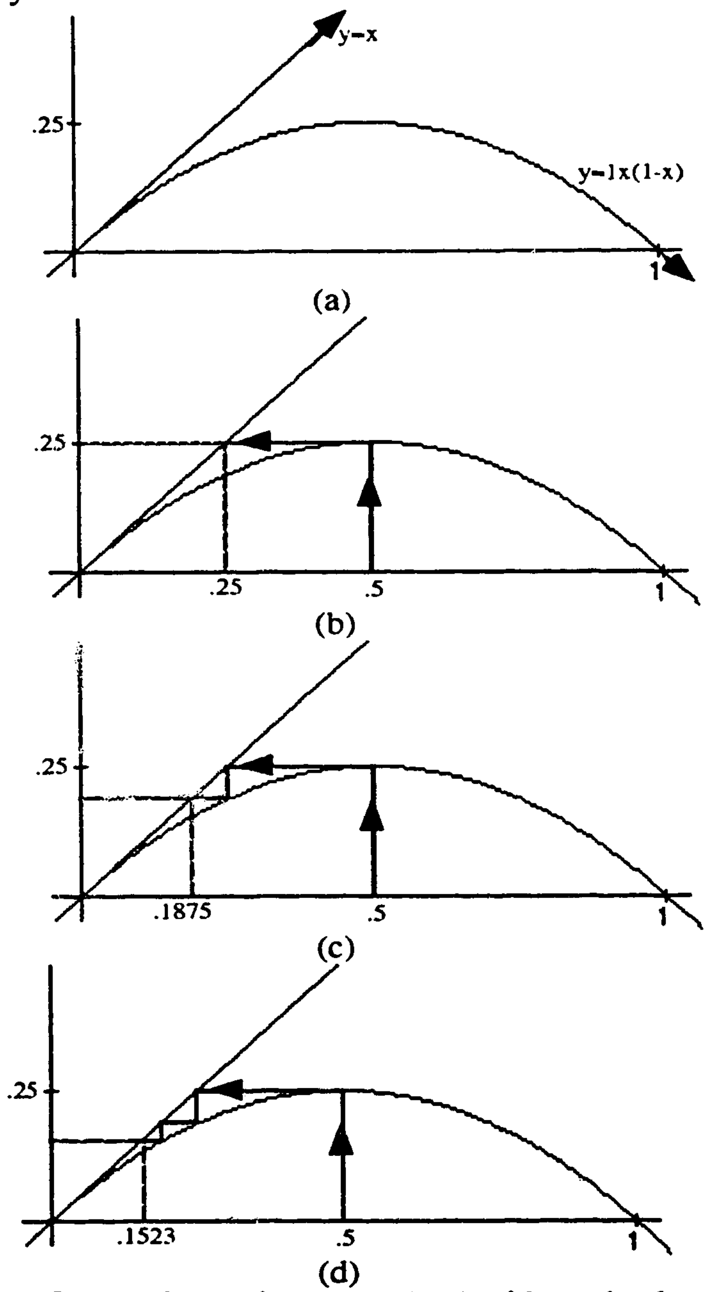


Figure 5-2. Cobweb Graph of Function y = 1x(1-x) with seed value of .5.

To generate a Cobweb Graph of y = 1x(1-x) the line y = x is provided (Figure 5-2a). The Cobweb Graph is based upon the seed value of x=.5 (Figure 5-2b). The iterates are .5, .25, .1875, .1523438, ... from a calculator. This pattern can also be seen from the graph by drawing in a series of lines. 1) The first line is drawn from the x-axis to the function at x = .5 (Figure 5-2b). 2) The second line is a horizontal line from the end of the first line across to the y = x line (Figure 2b). This line is at a height of .25 on the y-axis. 3) The third line is drawn vertically from the point (.25, .25) down to the function (Figure 5-2c). The process from here is repeated. Notice that the Cobweb Graph is really just a staircase which descends, approaching the fixed point (0,0) (Figure 5-2d).

The drawing of a Cobweb Graph usually continues 1) until the lines of the Cobweb approaches a point where the graph of the function crosses the y = x line (this point is known as a fixed point and an attractor because it is approached), 2) until the lines of the Cobweb begins to repeat themselves (a cycle has been reached as in Figures 5-3 and 5-4), or 3) until it is clear that the graph neither approaches a value nor cycles (Figure 5-5).

It is difficult to determine whether the iteration cycles or not but graphing may help to determine if patterns are present and what the nature of these patterns are (see Figures 5-3 and 5-4). Graphing may also help to determine that cycles do not exist. In Figure 5-3 a cycle that takes four iterations to repeat itself is evident in the 'boxes' in the Cobweb. Count the number of horizontal lines produced by these 'boxes' to see that this is a four-cycle.

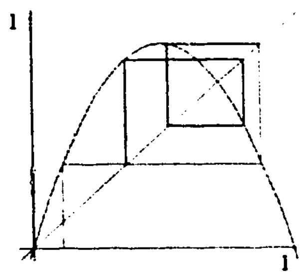


Figure 5-3. Cycle with period four (or four-cycle) for the Function: f(x) = 3.55x(1 - x).

In Figure 5-4, another cycle is apparent from the diagram. If you look at the number of dots (which represent iterates) needed to return to a value you will see that 8 dots are needed. This means that an initial value is followed by seven dots before that initial value is repeated. From both the Cobweb and Iterate Graphs a cycle is evident.

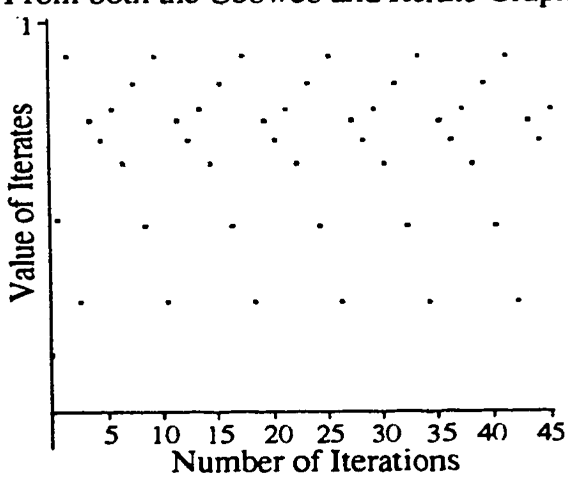


Figure 5-4. Iterate Graph of an 8-cycle for the Function f(x) = 3.66x(1-x). Seed value = .2.

In addition, graphs with values of iterates which do not cycle and do not approach infinity are also possible. These iterates may appear to be random in that the values do not repeat themselves (as this would be cyclic) and do not tend toward any direction (increase without bound or approach a particular fixed point). However, since they are based on the iteration of a function, their values cannot be random but are completely determined by the function. A large number of iterations are necessary to determine whether these functions have cycles of large period or whether they do not cycle. If they are found not to cycle then they are said to be in a state of chaos. Chaos does not imply unpatterned but rather a particular class of patterns. Drawing both the Cobweb and Iterate Graphs, as well as numeric values are helpful in determining these patterns as they are not obvious. However, drawing Cobweb or Iterate Graphs by hand is problematic for detailed explorations as imprecision causes difficulty. Lines are often not truly horizontal or vertical and non-cycles become cycles and cycles become non-cycles. As a result of this limitation two computer programs were developed.

Materials

The Cobwebber. A computer program called the Cobwebber[1] was developed by the researcher to facilitate exploration of concepts of non-linear dynamics and chaos theory. It generates 'Cobwebs' of the Logistics Equation [$y = \mu x(1 - x)$, $0<\mu\leq 4$] on the left side of the computer screen simultaneously with an Iterate Graph on the right side of the computer screen (see Figure 5-5). The purpose of this program was to enable the exploration of graphical images of the Logistics Equation.

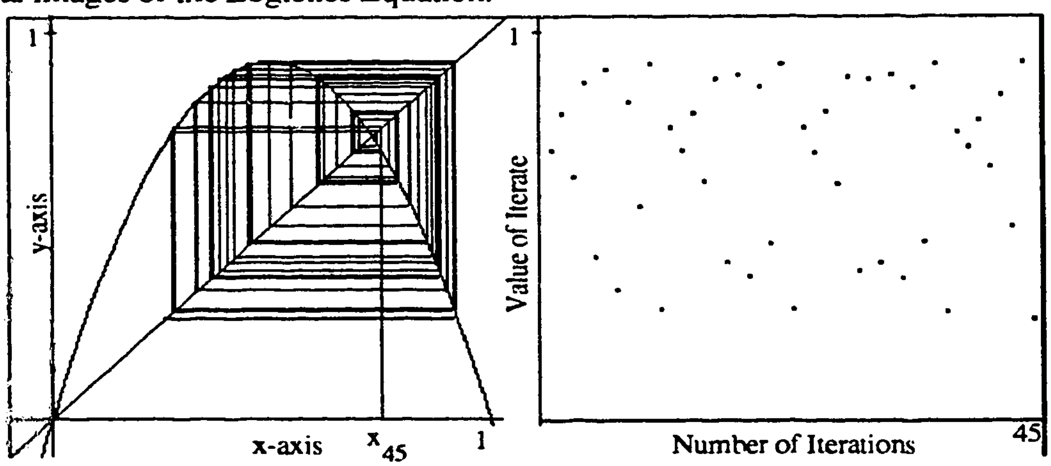


Figure 5-5. Cobweb and Iterate Graphs of the Logistics Equation for the Function $f(x) = 3.7x(1 - x)$ when seed value is .75 and 45 iterations.

When Cobwebbing the restriction of the value of μ to $0 < \mu \leq 4$ provides that all graphs remain in the domain $0 \leq x \leq 1$ and range $0 \leq y \leq 1$. This means that a box drawn with equal sides of length 1 will completely contain the graph of the function. This is particularly helpful for explorations on the computer screen.

To explore the Logistics Equation using the Cobwebber the user of the software must enter the value of μ (between 0 and 4). A seed value (between 0 and 1) must also be entered. This program eliminates the tedious task of drawing the Cobweb Graph by hand and allows for a large number of repetitions to be drawn quickly and effortlessly. Permanent records of explorations can be kept by printing the results to a printer or disk. It is possible to print from the computer while the program is running and then allow the program to continue to run. This allows the user to 'capture' a series of images of the Cobweb Graph as it is drawn.

1 X-commands from the Zap-a-Graph Toolkit (produced and distributed by Brain Waves Software, Inc., Fitsroy Harbour, Ontario, Canada) were used under license in the display of the function graphs incorporated into the Cobwebber.

The Spreadsheet Logistics Equation. Another exploration medium called the Spreadsheet Logistics Equation was developed using a spreadsheet program. The Logistics Equation was entered into a row of 100 cells in such a way that the value of one cell was the input for the cell on its right. Thus the iteration of the Logistics Equation was accomplished through the Spreadsheet Logistics Equation data file. The spreadsheet program provided graphing capabilities in Iterate Graph format with the number of iterations on the x-axis and the value of each iterate on the y-axis. By organizing the views of the values in the cells and the graphs produced it was possible to see both the graph and the values which had produced the graph simultaneously (Figure 5-6). Therefore it is possible to compare the graph with the values exact to 8 decimal places. This comparison made pattern noticing possible.

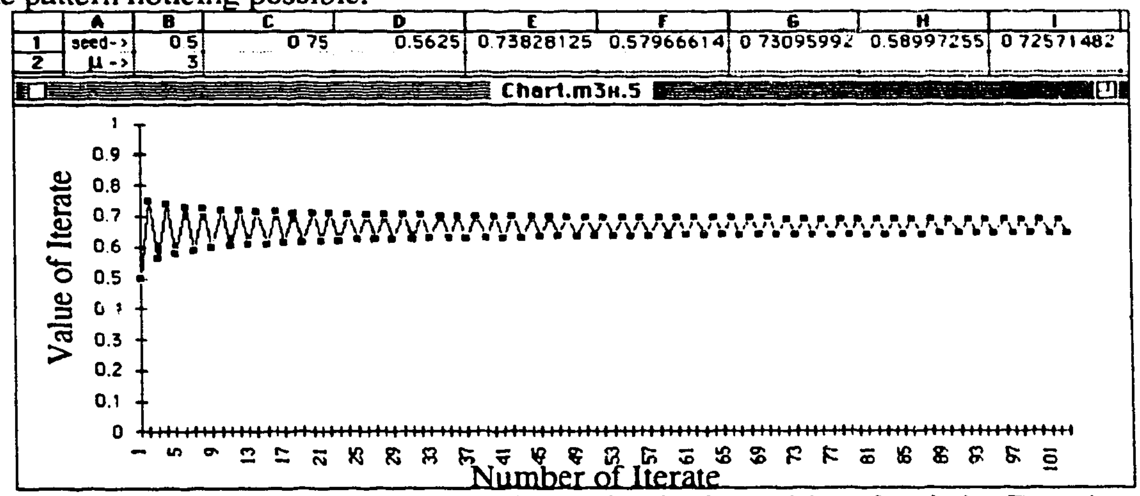


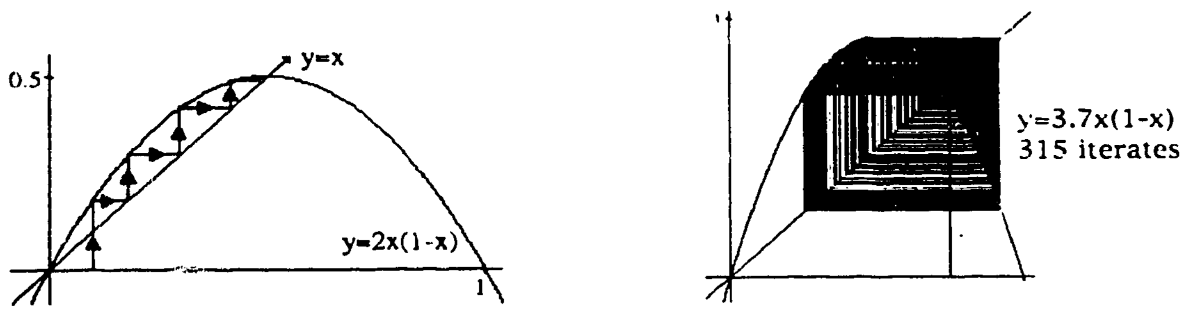
Figure 5-6. Iterate Graph and Iterate Values for the Spreadsheet Logistics Equation for $y = 3x(1 - x)$, $x_0 = .5$.

Types of Presentations and Uses of Each

Three types of presentations of the iterates of the Logistics Equation $y = \mu x(1-x)$ are utilized in the two computer programs. These include 1) the Cobweb Graph, 2) the Iterate Graph, and 3) numeric values. Each of the three makes some features of the iterate values or the patterns of iterate values salient.

The Cobweb Graph enables a view of how one iterate value leads to the next. The Cobweb Graph uses a finite space on the surface on which it is drawn to make the view of the iterates possible unlike the other two methods. Two types of patterns are particularly salient using the Cobweb Graph: 1) the iterates approach to an attractor (Figure 5-7a) is clearly visible from the Cobweb Graph, and 2) after repeated iterations of a function producing chaos, regions become black and others remain white which allows us to see that even in chaos there are patterns (Figure 5-7b). A drawback of the Cobweb Graph method is that in some cases although a cycle has been attained the graph is sufficiently cluttered that it is not possible to clearly see the cycle (Figure 5-b) while in other cases the cycle is visible (Figure 5-8a).

Iterate Graphs are also useful for viewing patterns of iterates. They can easily portray patterns of approaching an attractor (Figure 5-9). Compare this portrayal of $y = 2x(1-x)$ with seed value of $x_0 = .1$ to that in Figure 5-7a which is a Cobweb Graph of the same function and seed value. The advantage of the Iterate Graph is that plotting points on a graph is a skill that is utilized for other graphing situations and thus these skills may be more readily transferred to the understanding of iteration than to begin interpreting Cobweb Graphs.



(a) (b)

Figure 5-7. Patterns Clearly Depicted by the Cobweb Graph: (a) Patterns Approaching an Attractor and (b) Patterns within the Chaos.

(a) (b)

Figure 5-8. Cycles of Iterates (a) May or (b) May Not be Visible From the Cobweb Graph.

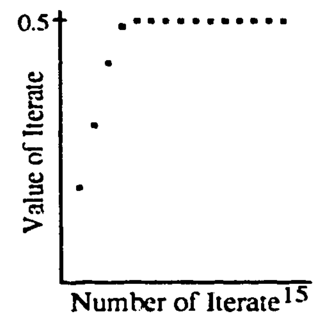


Figure 5-9. Iterates Approaching an Attractor using an Iterate Graph for $y = 2x(1-x)$, $x_0 = .1$.

In Figure 5-10 the Cobweb and Iterate Graphs can be compared. Both graphs show the first 45 iterates of $y = 3.66x(1 - x)$ for a seed value of .25. However, the two graphs show the same information in different ways. The iteration 8-cycle is obvious from the Iterate Graph. This feature may be less noticeable from the Cobweb Graph. However when the program is running the cycle is more evident because the graph no longer produces new lines when the function begins to cycle.

Even when the number of iterations becomes large the Cobweb Graph is small in area and can effectively represent this information in a confined area. The Iterate Graph, however, would become very large continually adding to the right as the number of iterations became larger. The patterns may be difficult to assess over such a large space. An advantage of the Iterate Graph is the display of clearly distinct points which are related but not connected. This feature is not obvious from the Cobweb Graph.

The display of both types of graphs allows the comparison of the two graphs and the association of them. While the period of the cycle may be clear from the Iterate Graph, the interconnectedness of the points forming the cycle and the pattern of its progression may be more clear from the Cobweb Graph. Viewing the two graphs simultaneously enhances the understanding of the function and its iteration by allowing the user to see the information from two points of view each making some features salient.

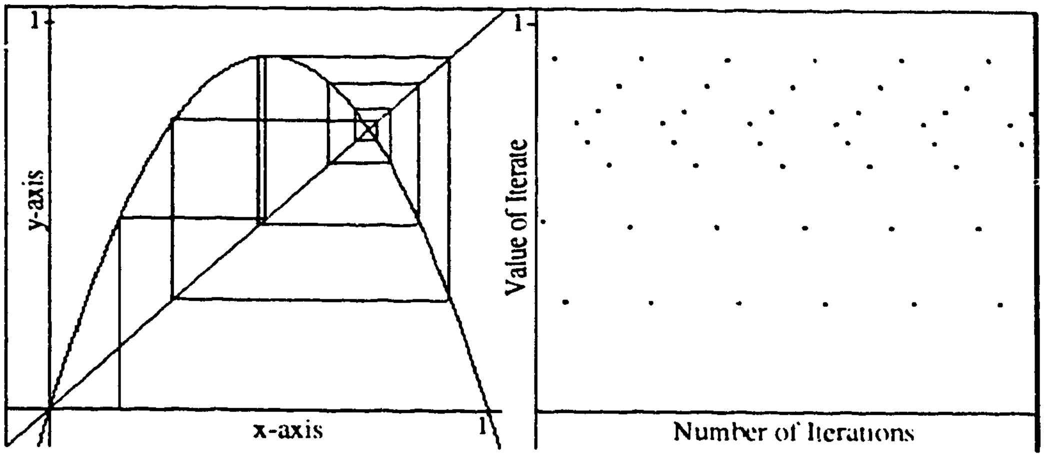


Figure 5-10. A Cobweb Graph (left) and an Iterate Graph (right) for the function $f(x) = 3.66x(1 - x)$ in which the 8-Cycle is visible from the Iterate Graph.

The numeric values and the Iterate Graph provided together in the Spreadsheet Logistics Equation have several important reasons for usage. The notion of iteration can initially be explored using the function keys on a calculator. This is a numeric beginning. The numeric values used in the Spreadsheet Logistics Equation can be tied to this early understanding. The Spreadsheet Logistics Equation instantaneously displays all iterates in an Iterate Graph format. This graph format has advantages as discussed earlier and the instantaneous nature of the Spreadsheet Logistics Equation means that exploration results can be achieved quickly. The presentation of the numeric values with the Iterate Graph allows for the exploration of patterns that are too fine to see only from the graph. In some cases it may be clear that a cycle is present from the Iterate Graph but too difficult to determine if it is an 8-cycle or a 16-cycle (Figure 5-11). Viewing the numeric values can assist in making the distinction (Table 5-1). Every 16th value in Table 5-1 is a very similar value while every 8th value is related but not as similar in value. This can be seen by examining the rows in Table 5-1. From such an examination it can be determined that this is not an 8-cycle but a 16-cycle.

Limitations of Using a Computer For Explorations

The Cobweb Graph of a function in chaos becomes increasingly blackened as iteration proceeds. Compare Figures 5-5 and 5-7b: does the Cobweb Graph area become filled? To answer this question we must consider the resolution of the screen on which the graph is generated as well as the number of decimal places to which the calculations are taken. Because we are drawing on a screen with the capacity to show only a certain number of vertical lines and a certain number of horizontal lines, the capacity of the display is limited. So when we calculate a value and draw it on the screen, the screen displays the line closest to this value, rather than the exact value because of the limitations of the screen. If we were to 'zoom in' on any region of the Cobweb Graph which initially appeared black, thereby increasing the screens ability to show this region, we would find that it wasn't black at all but actually had white regions between the black lines but the screen was simply incapable of showing us this detail. But if we allowed the program to continue to Cobweb Graph we could eventually fill this space too if we draw long enough. But then we could zoom in again and white regions would again appear. So when considering the question of screen filling, we can say that at one level the screen may appear black but on another level we know that there is still distance between the lines and thus white space between them.

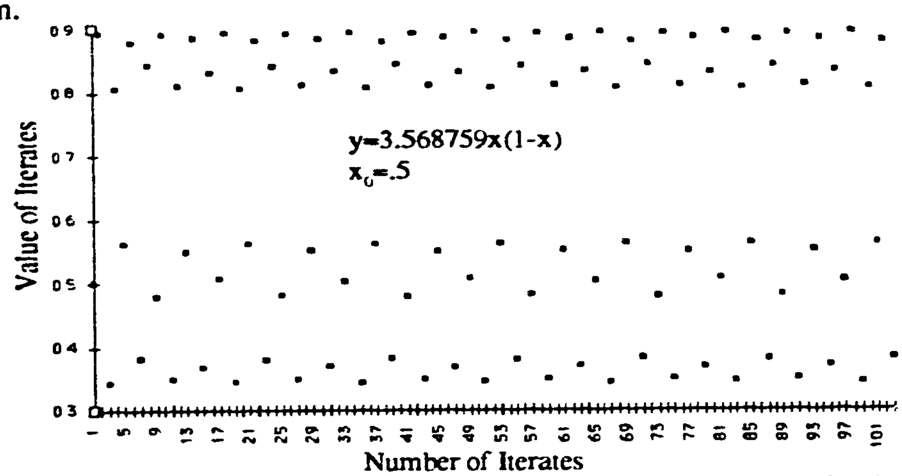


Figure 5-11. Iterate Graph of a 16-Cycle for the Function y = 3.568759x(1-x) with seed value .5.

Table 5-1. List of Values for the iterates of y = 3.568759x(1-x) with seed value of .5.

0.892189975	0.343326894	0.80452452	0.56124023	0.8788056	0.38009539	0.84088129	0.47749977
0.890388303	0.348831477	0.81007829	0.54905877	0.88360059	0.36704895	0.82910843	0.50564904
0.892207586	0.343358768	0.80488073	0.56046554	0.87914208	0.37918523	0.8400994	0.47939995
0.8906753	0.34750008	0.80919388	0.55101333	0.88290256	0.36895841	0.83090738	0.50141173
0.892218264	0.343328885	0.80454679	0.56119183	0.87882675	0.3800382	0.84083233	0.47761887
0.8904021	0.348826162	0.81002073	0.54918615	0.88355593	0.36717122	0.8292244	0.50537657
0.89208659	0.343355768	0.80484723	0.56053843	0.8791106	0.37927041	0.84017283	0.4792217
0.890648898	0.34757348	0.80927376	0.55083704	0.88296663	0.36878327	0.83074346	0.50179878
0.8921782	0.343330126	0.80456067	0.561116164	0.87883993	0.38000257	0.84080182	0.4776931
0.89041394	0.348822863	0.809985	0.54926521	0.88352816	0.36724726	0.82929648	0.50520719
0.89209298	0.343353978	0.80482724	0.56058193	0.8790918	0.37932129	0.84021666	0.47911526
0.89063316	0.3476176	0.80932175	0.55073108	0.88300504	0.36867828	0.83064509	0.50203097
0.89217503	0.343331014	0.80457061	0.56114005	0.87884935	0.37997708		

Another way to approach this problem is to consider reducing the number of decimal places so the computer can represent exactly the values on the screen. Each of the values iterated may be rounded to two decimal places. The resulting iterates from such a limiting always cycled for the values tested. The iterates never produced non-cyclic patterns. This begs the question of whether limiting to 8 or 16 decimal places is reason why the iterates act the way they do. This is a challenge which is posed by the technology itself.

The limitations of the computer screen resolution and calculation completion ability might lead one to the conclusion that explorations must occur in environments without resolution or decimal-place limits. This implies calculations with numbers of infinite length and infinite screen resolution which is not possible. However, calculations are possible to a very large number of decimal places and screens with very high resolution are available (one can also 'zoom in' on regions to reduce the effect of limited resolution). However, high powered computers are necessary for these investigations.

Indeed the computer used in this investigation produced diagrams with far greater precision than possible by hand drawing thus enabling more extensive explorations than would be possible without the technology. Despite their limitations the computer programs offer the user an opportunity to visualize dynamic change based on simple mathematical functions. The control of the exploration may be placed in the hands of users who can ask questions. generate hypotheses and seek understanding in the realm of mathematics. Growth of understanding of mathematics is possible through explorations within a computerized context. Questions for further study that emerge from this discussion are: How does the computer help one generate a mathematical solution? What types of image making are involved in using these programs? What mathematical properties may be noticed through exploring this computer medium? Can the use of these software packages lead to formalizing and structuring of mathematical ideas?

Commentary

Chapter 5 has discussed computer programs for use in explorations of the mathematics of dynamic systems and chaos theory. This chapter described and discussed an exploration environment for mathematical meaning making. Some of the meanings that the researcher made from these materials were provided.

Chapter 6 extends the discussion of meaning making through this mathematical exploration environment by providing these materials to high school students. A discussion of students' explorations of this mathematical exploration environment is provided. Insight into student mathematical findings emerges through the discussion. Through this visual and experimental style of mathematical inquiry students engage in mathematics through their activities with the computer. Students involved in human-computer interaction take action in this medium through mathematics. The interplay of mathematics and computer interaction through students activities is the focus.

Chapter 6

Using Computers To Look At Non-Linear Dynamics: What Do Students Do? What Do Students Think?

Goldenberg (1989) describes a view of the mathematics educational experience that encourages knowledge generation by learners. From this view mathematics

> *can* be the most freeing of subjects, a game in which the player is free to invent any set of playing pieces, rules, and constraints, and then reason out or observe the consequences of these choices. It is a game whose players frequently use the words *elegant* and *beauty*, and whose beauties are both visual and intellectual. Yet we show little or none of this to our students" (p. 170-171)

How is the game of mathematics to become a part of students experiences? Computers and calculators can be used by students to explore the visual and intellectual beauty of mathematics. Students can use computer technology to investigate functions, generate hypotheses and explore the usefulness of these hypotheses. One field that can be explored by students and which represents a frontier in mathematical research is dynamic systems and chaos theory. Using computer-based exploration tools students can explore and generate mathematical concepts and visualize the mathematics.

Described in this chapter is a set of activities involving four mathematical exploration environments (a calculator and three computer-assisted spaces) for students' explorations of underlying concepts of chaos theory and non-linear dynamics. The structure and interactivity of these mathematical environments propels students to engage in inquiry and generate ideas and interest in mathematics. The nature of students computer interaction within these mathematical exploration environments is addressed.

In regard to these activities it is useful to ask: what non-linear dynamics and chaos theory mathematics concepts can students explore and generate when provided a computer environment for exploration and activities through which to explore? What materials and procedures can enable exploration of mathematical concepts? What is the nature of the mathematical findings that students generate through their activities and actions? How do students explore through these computer-based environments? What is the nature of human-computer interaction in a mathematical exploration environment?

Students Introduction to the Study

At the first meeting a group of six high school students (aged 16 or 17 years) assembled after school in the Macintosh computer laboratory. (At later meetings a seventh student joined the group.) They had come on this early closure day[1] to be involved in an extension activity to their regular mathematics class. They were to remain one hour on this first day and for each of the following five weeks. A microphone sat next to each of the six computers these students were to use for their explorations. These microphones were attached to two audio mixers attached to two audio-tape recorders. Near the computers two video cameras stood on tripods ready to record students activities. The mathematics teacher asked the students to sit down in front of the designated computers and then introduced the researcher. The researcher described the format of the after-school mathematics sessions.

Students, one to each Macintosh computer, were to explore individually on the computer or together with other students by viewing each others computer screens and discussing the explorations. Students were to become involved in activities and record computer entries and observations made during explorations on record keeping forms provided by the researcher. Video and audio recordings of students activities would be

1 The school in which this study was conducted had one day per week which ended one-hour earlier than usual. This is referred to as the early closure day.

made. At the end of each session students would be asked to write about what they had explored by responding to questions and statements posed by the researcher. They were to hand in their writings to the researcher.

Following students introduction to the study the researcher collected the consent forms from the students (which were distributed by the teacher earlier). The researcher asked the students to wear the microphones around their necks and turned on the audiotape and videotape cameras. She then introduced students to the concept of iteration.

What is the Meaning of Iteration?

Exploration of the concept of iteration of a function began with an explanation of terms: to iterate means (in broad terms) to repeat a process over and over again. In the study of mathematical dynamic systems, the process that is iterated (repeated) is the evaluation of a mathematical function. A function is defined as a rule that converts a set of numbers into another set of numbers. It is an action or process that changes numbers each time it is applied or invoked. Students participated in exploring iteration and seeing function as action by using a calculator. Students entered 2 and then pressed the x^2 key. The result was 4. When the x^2 key was pressed again, the result was 16. When the function x^2 was again invoked by the keypress, the result was 256. This example of iteration of the x^2 function used 2 as the initial or seed value. The answer or output was 4 after invoking the function. The output of the previous action (key press) was the input for the next step in the iteration. Thus 4 became the input when x^2 was again pressed and the output was 16.

In this first medium of the calculator (a computer with a mathematical purpose) students were introduced to iteration as an action upon a function. Students action of pressing a key invoked action on the part of the calculator. The calculators' numeric calculation provided feedback to students action and occasioned students to further action. Students continued this procedure of key press, viewing results, another key press and further results until the calculator provided a non-numeric result; either an infinity ∞ symbol or an error message indicating that the numeric value was beyond the capacity of the calculators digital display. The calculator could no longer take action in response to the student's key-press. At this point students began the process again with another input value. As a group, students were asked what the calculator had provided as the final result. They were asked to describe the change in value as a result of their input. Students determined that the values "kept getting bigger" until it reached the limits of the calculators display capacity. From this simple example students interaction with a computer (mathematical calculator) was one of mathematical activity. The key-press in this medium enacted a calculation. This activity was a mathematical activity to students.

To further explore the iteration of a function students were provided an **Iteration Program** on the computer. This computer assisted instruction program overviewed the square root function drawing an Iterate Graph[2] of each of the values generated by the repeated calling of this function. The type of graphing plotted the value of the iterate (y-axis) against the number of the iterate (x-axis). This graphing technique provided a plot of the values of each iteration. Students had the opportunity to input initial values and see the results of repeated iteration upon these values by the square root function. A summary of the patterns which emerged from the graphs was dynamically drawn (Figure 6-1) indicating that when the input (x_0) occurs in particular ranges a pattern of results emerges [$x_0 > 1$ approaches one from above, $0 < x_0 < 1$ approaches 1 from below, $x_0 = 1$ remains 1].

In this computer medium the students' actions of inputing a value invoked action on the part of the computer software which drew an Iterate Graph of each value of the iteration. Students could view the results of iteration from both the iterate values which

2 Refer to Chapter 5 and Figure 5-1 for an explanation of the process of generating an Iterate Graph.

appears in a column of numbers and the graph of these values. The computer software ceased to calculate and draw when it was very close to 1. Students viewed the results of the sequence of calculations which they had initiated and then input another value. The specific patterns of behavior of the square root function were summarized by the software for students in Figure 6-1. In this computer medium as with the calculator example students activity of interacting with a computer (mathematical software) was one of mathematical activity.

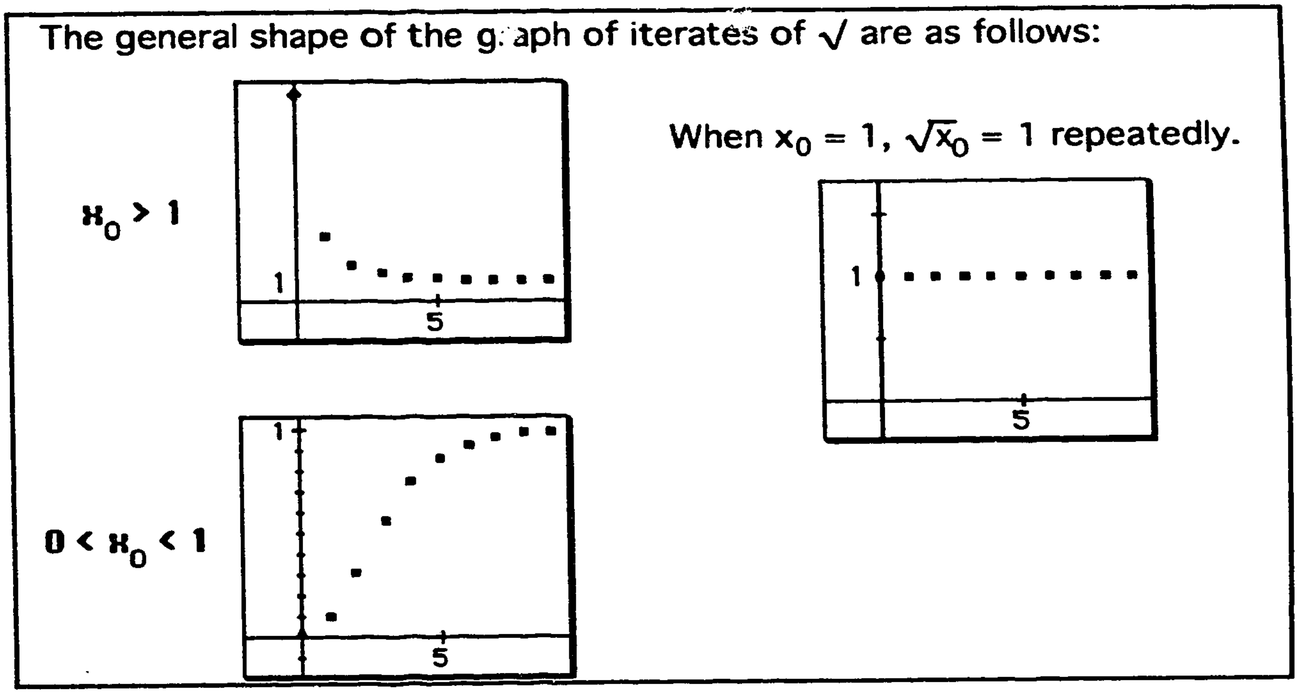


Figure 6-1. Images of the Square Root Function Under Iteration from the Iteration Program

At the end of class students were asked to write about iteration. They handed in their writings to the researcher. Rather than describing a general definition of iteration, Anne described the patterns produced when $\sqrt{x}$ was iterated (as in Figure 6-1). Cindy wrote that "Iteration is a process of repeating a function and finding the changing results." Tony wrote: "An iteration is to repeat a function many times." Cheryl wrote: "Uses the answer from the previous calculation as a starting point for the next calculation." Mike wrote that "Iteration - repeating over and over, using the result of the previous computation as the input for the next." John wrote: "Repeating a function, using input form previous function and using it in the following function."

Anne's ideas of iteration are specific to the function explored. The repeating of a function as a component of iteration is evident in the responses of Cindy, Tony, Mike and John. However, the idea that the output of one iteration is the input to the next is only evident in written responses provided by Cheryl, Mike and John. Students writings indicate that some features of iteration are more salient than others to them. Three levels of noticing may be specified. 1) The concept of iteration is specific to the function. 2) Iteration is a more general concept including the idea of repetition. 3) Iteration includes the concept of repetition and the idea that output of one stage of iteration is the input to the next stage. Iteration is an important concept for understanding these computer programs that were used for explorations.

Students' Initial Explorations Using the Cobwebber

A **Cobwebber** computer program was used to explore iteration of the Logistics Equation [$y = \mu x(1 - x)$, $0 < \mu \leq 4$]. This program generated 'Cobwebs' of the Logistics Equation on the left side of the computer screen simultaneously with an Iterate Graph on the right side of the screen[3]. The purpose of this program was to enable the exploration of graphical images of the Logistics Equation.

A Cobweb Graph is a series of lines the first of which is drawn vertically from the x-axis to the function at horizontal position of the input value[4]. The second line is a horizontal line from the end of the first line across to the $y = x$ line. The third line is drawn vertically from the point on the $y = x$ line to the function. The process from here is repeated drawing a horizontal line to the $y = x$ line and then a vertical line to the function. From the Cobweb of the iteration a pattern can often be determined.

Each student entered values for μ and x into the Cobwebber and watched the iteration of the Logistics Equation unfold. Students were to record their explorations and observations on a record keeping forms provided by the researcher. These record keeping forms had the Logistics Equation $y = \mu x(1 - x)$ at the top and five columns: 1) Value of μ, 2) value of x-input, 3) total # of iterates, 4) Why choose these values?, and 5) Interesting/Important things about this function (patterns noticed, value approached, chaotic, cyclic, ...). They were asked to notice patterns and describe them in a space on the recording sheet.

Mike's approach was to enter whole number values for μ and carefully record the result by drawing the Cobweb portion of the graph for each x-value tried. (Software generated Cobwebs of Mike's input values are provided in Figure 6-2). He used x-values of .1, .5, and .9 for each of the three values of μ investigated. His approach was systematic and his focus was clearly the Cobweb portion of the graph. He stated that the reason for choice of values was that he "want[ed the] value of x-input [to] become bigger and bigger" thus explaining the .1, .5, .9 progression of x-values tried.

Based upon these explorations Mike stated that "μ effect[s] the height of the graph [parabola]". Looking at Figure 6-2a, d and g notice the increasing height of the parabolic curve associated with the increasing values of μ from 1 to 3. Mike wrote that "the graph on the left [Cobweb] shows what x-value look[s] like on the graph." To Mike it appears what is salient from the Cobweb is the x-input. Mike may have meant that the entry of an x-value determined the beginning point for the Cobweb. For example when $x = .5$ (Figure 6-2b, e and h) the Cobweb begins in the center of the screen. Another concept that Mike may have developed looking at the graphs systematically was that given the same μ-value the Cobweb (and Iterate Graphs) approached the same value despite differing x-values. For example when $\mu = 1$ (Figure 6-2a, b, c) the Cobweb approached 0, when $\mu = 2$ (Figure 6-2d, e, f) the Cobweb approached .5 and when $\mu = 3$ (Figure 6-2g, h, i) the Cobweb circled around getting ever closer to the intersection of the parabola and the $y = x$ line (diagonal). Mike may have developed this understanding more through the Iterate Graph since he stated that "the graph on the right [Iterate Graph] show[s] how iteration go[es]" which may indicate that what was salient from the Iterate Graph was the pattern of iteration. Perhaps the two graphs are seen to have different salient features and are not associated with each other. To associate the Cobweb and Iterate Graphs compare Figures 6-2d and 6-3a and Figures 6-2g and 6-3b. In particular notice the height of the points in the Iterate Graph compared to the height of the horizontal lines of the Cobweb Graph.

---

3 Refer to Chapter 5 and Figure 5-5 for an explanation of the Cobwebber and the graphs it produced.

4 See Chapter 5 Figure 5-2 for an image of cobwebbing and further explanation of the process involved in creating a cobweb.

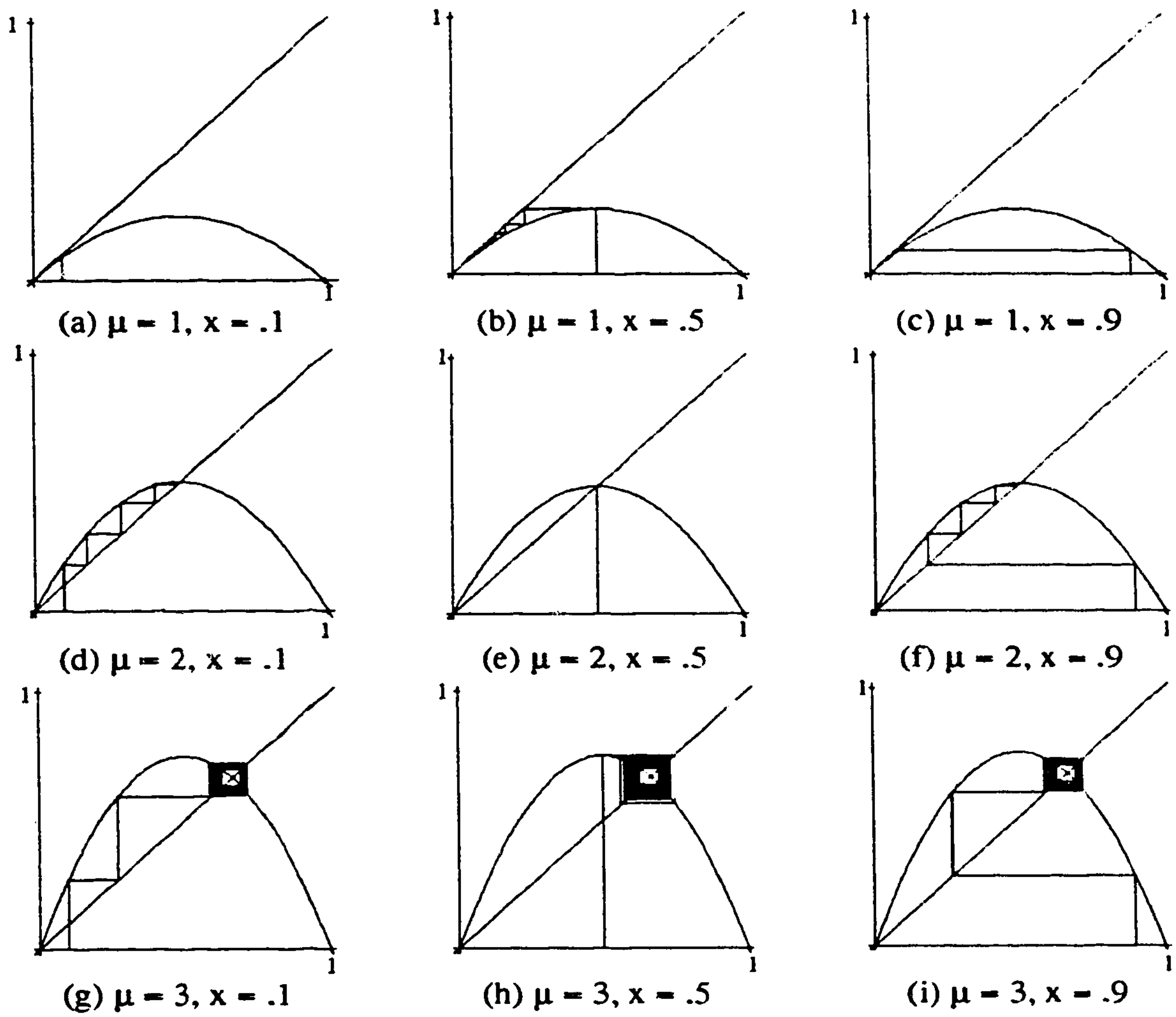


Figure 6-2. Cobweb Graphs of the Logistics Equation $y = \mu x\,(1 - x)$ with μ and x-input values indicated.

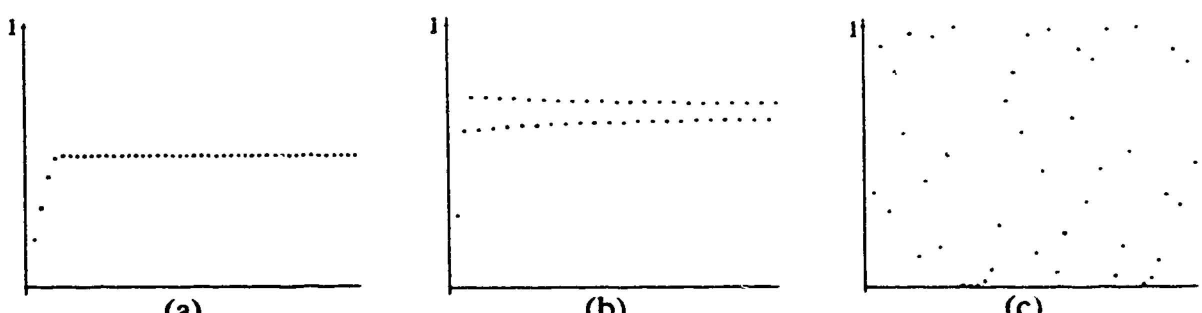


Figure 6-3. Iterate Graphs of the Logistics Equation with μ and x-input values defined as follows: a) $\mu = 2$, $x = .1$, approaches a fixed point, b) $\mu = 3$, $x = .1$, also approaches a fixed point, and c) $\mu = 4$, $x = .1$ which is chaotic.

Anne, like Mike, chose to use whole number values for μ beginning with 2. However Anne chose to use the Iterate Graph as her focus. Images of what Anne would have seen on the computer screen are provided in Figure 6-3. For $\mu = 2$ she described a "straight line" (Figure 6-3a). For $\mu = 3$ (Figure 6-3b) she stated "on the right $\frac{1}{2}$ of the screen [the Iterate Graph] a funnel pattern appears" for which she drew a graph (Figure 6-4). But for the chaotic $\mu = 4$ she stated its "an unknown pattern" (Figure 6-3c). Anne has noted some particular behaviors of the iterates of the Logistics Equation. She has discovered that iterates approach a particular value (fixed point) either from below (Figure 6-3a) or from two sides (Figures 6-3b and 6-4). She has also noted chaos (Figure 6-3c) although not by that name.

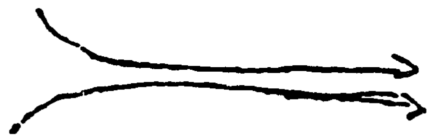


Figure 6-4. Anne's "funnel pattern" drawn to represent the Iterate Graph which appears when $\mu = 3$ and $x = .98$ and the gradual approach of a fixed point.

Cindy , like Anne, seemed to be primarily viewing the Iterate Graph in her explorations. For $\mu = 2$ she states "the pattern went up and then stayed constant" but for $\mu = 4$ she stated "a chaotic pattern, dots appeared all over but in a definite unexplainable pattern". For $\mu = 2.5$ and 1.6 she described "straight line" patterns noting that for these values of μ a fixed point was approached by the graph.

Anne wrote that "when μ value is increased the graph (parabola) narrows" but the graphics she had seen on the computer screen actually did not change in width but were all constantly portrayed in a box of width one unit. Her statements do not seem related to her explorations of the Logistics Equation through the graphics provided by the computer. Perhaps this statement was related to the content of the mathematics course in which she was enrolled where she learned that the width of a parabola changes when the coefficient of the x^2 term was varied. She also stated that "x determines the pattern."

Cindy stated succinctly that the "logistic equations produces a downwards parabola [opening downward] with a linear pattern evolving from different values of x". She added that the "μ function effects the length or width of the parabola". In regard to the effect of x she stated "x contrasts the shapes or forms of the linear pattern." Cindy wrote "the graph on the right [Iterate Graph] and left [Cobweb] both show the pattern if it repeats or contrasts itself" which indicates that for Cindy the two graphs are related to each other. It is possible she noted that the height of lines in the Cobweb are at the same height as points in the Iterate Graph.

Cheryl began her explorations with $\mu = 3.9$ which she explained by the statement "39 is my favorite number so I used the decimal". Her later choices of μ values were 2.3, 1.1, and 0.1 chosen for "no reason. Just picked 2 numbers". She later tried $\mu = 3.9$ again with $x = .99$ which she chose because she wanted to "try to use the 2 highest numbers to see what would happen". She followed this with $\mu = 2$, $x = .5$ the "middle of each" range. In all of her observations she described the "dots" of the Iterate Graph. For $\mu = 3.9$, $x = .99$ she described "dots scattered across the screen. Bunches of dots and then blank spaces. Whenever there seems to be a pattern, something screws up". Cheryl has noticed that true repetitive patterns do not occur here. She has noticed chaos.

What does the Logistics Equation produce? Cheryl wrote "the graph on the right side of the screen with many different patterns." What is the effect of μ on the Logistics Equation? "The larger you make this value, the larger the graph's parabola will be. If you make it small, the graph will be very small and wide. The bigger the # [for μ], the thinner the graph." Cheryl seems also to have associated the changing value of μ with the width of the parabola although she also has associated it with the parabola's height. When asked to describe the relationship between the two graphs on the screen Cheryl stated that she had "absolutely no idea". Perhaps an understanding of the Iterate Graph is emerging but as yet is unassociated with the Cobweb.

Tony appeared to use both the Cobweb and Iterate Graphs. For $\mu = 3.3$, $x = .55$ he stated "there were two straight lines at .82 and .48" (Figure 6-5). He appears to have noted the 2-cycle in this range from the Iterate Graph. Tony then entered $\mu = 3.3$ again but with $x = .9$. This time he noted "It made a line of boxes gradually enlarging and moving down and to the left". This observation is of a Cobweb. For $\mu = 2.54$ Tony noted "it formed a stair-like pattern" and he drew a Cobweb approaching a fixed point (Figure 6-6).

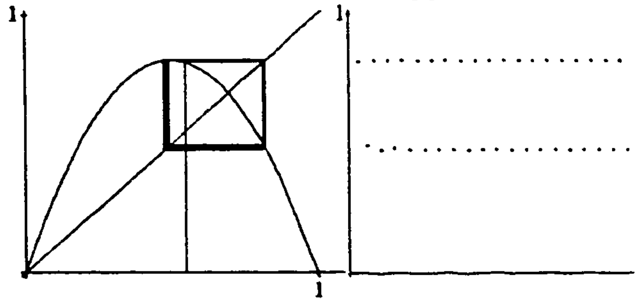


Figure 6-5. Cobweb and Iterate Graphs of the Logistics Equation with $\mu = 3.3$ and $x = .55$ which demonstrates a 2-cycle or a pattern in which two values are simultaneously approached by the graph.

Figure 6-6. Tony's Cobweb Graph for $\mu = 2.54$, $x = .12$ which approaches a fixed point in a "stair-like pattern".

For Tony the Logistics Equation "produces a parabola with it's size depending on the amount you put in for μ" and added that μ "determines the size of the parabola". Tony stated "the left graph [Cobweb] shows the parabola and the effect x has and the right graph [Iterate Graph] shows the product of y." Again the two graphs are seen to make different features of the iteration salient.

John began his explorations with $\mu = 3.9$, a chaotic function (Figure 6-7a). He noted from the Iterate Graph that there was "a high concentration[of points] in 2 wide bands [on] top and bottom [of range]." He proceeded to explore $\mu = 3.5$ from which he noted "two horizontal narrow V's" (Figure 6-7b) for which he described the lower band as "wide" and the upper band as "narrow".

John stated that the Logistics Equation "Produces a graph between 0 & 1. ... Produces series of lines on original parabola graph forming squares". The first statements indicates that all values of the iteration are contained in the range of 0 to 1. John's second statement emphasizes the feature of the Cobweb with it's vertical and horizontal lines. In regard to the "effect of μ - I think it affects location of dot clumping" making clear reference to the Iterate Graph and the fact that changing the μ value effects the value approached or the region of the range which is utilized by the iterates. He stated that the "effect of x [is to] waver size at beginning." Perhaps this indicates that the x-input determines the height of the initial point on the Iterate Graph or the position of the first line drawn on the Cobweb. John also stated that the "location of squares [Cobweb] on first graph direct relation to location of points on second [Iterate Graph]." From this statement it is clear that the graphs represent the same patterns but perhaps in different ways. For John the two graphs are related.

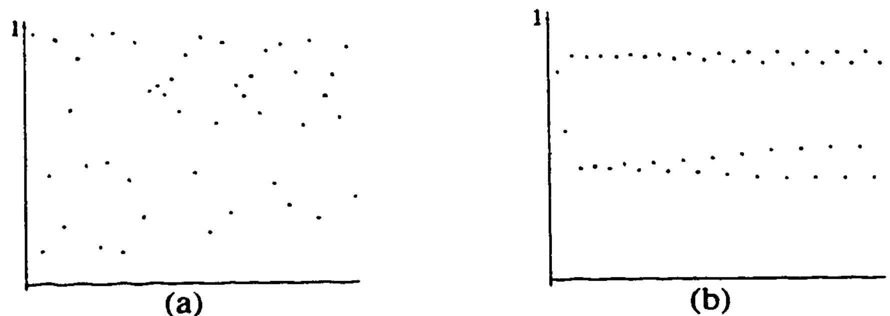


Figure 6-7. Iterate Graphs for a) $\mu = 3.9$, $x = .46$ which generates a chaotic pattern, and b) $\mu = 3.5$, $x = .35$ which approaches four values simultaneously so is a 4-cycle.

Students were asked to describe questions they could now answer based upon the days activities. Cheryl wrote "What happens after continuous pressing of the $\sqrt{}$ or x^2 key/button on your computer/calculator?" She appears to have confidence in her knowledge of the patterns produced by these functions. Anne suggested she could answer "What are iterations?" Mike described "how μ and x change the graph" despite the fact he left the question about the effect of x unanswered when asked. Tony did not provide any questions he felt he could now answer. Cindy, like Anne, suggested "what is iteration?" and also suggested "what's a Logistics Equation?" John stated he could answer "very basic definition type question" but did not specify terms he could now define.

From the experiences of the first meeting it was clear that students understood that the Logistics Equation produced a parabola. The effect of μ which increased the height of the parabola was not yet clear to students but the x input was understood to establish the starting position of the graph. Only John saw the relationship between the two types of graphs while the other students were as yet unclear. Students had begun to notice patterns produced by the iteration of the Logistics Equation including approaching a fixed point, a two-cycle and chaos. What other patterns could students consider and explore? What perceptions of the Cobwebber would students develop?

Students had interacted within these computerized environments; their actions were mathematical in nature. The patterns they noticed, the questions they asked, and the general statements they made were about the mathematical content of this setting. The context had occasioned their actions to be mathematical ones despite the fact that they were immersed within a computer environment. One might have expected students activities to be computer focused because they were in a computer laboratory, using computer software and hardware for their investigations. However the actions taken by students were mathematical ones.

What is a Cobweb?

During the second meeting students discussed what they had learned the previous day and were encouraged to execute the Iteration Program (Figure 6-1) if they wished to review. They then were reintroduced to the Logistics Equation and were invited to explore the Cobwebber. Students were provided record keeping forms as during the first meeting and were again asked to make records of their explorations. Anne and Cheryl were absent but Joy, a student who had been absent for the first meeting, arrived to participate. Tony was asked to describe to Joy what had been done last day. However, he was unable to tell her any more than what values were acceptable when μ and x were to be entered when using the Cobwebber. Joy was directed to use the Iteration Program to learn about iteration. The other students explored values of μ and x. Students were asked to record the values they entered, observations and patterns they noticed and to provide a list of examples or questions as they proceeded.

After viewing and exploring with the Iteration Program Joy used the Cobwebber. Joy explored $\mu = 2$, $x = .3$ for which she drew a partial Cobweb looking like an ascending set of stairs with three steps and wrote "curve was tall" making reference to the parabola. She continued to explore values of $\mu = .2, .7$ and $.4$. For $\mu = 0.1$ she wrote "practically straight line, curve was flat" and she observed the effect of her inputs "μ changes height and width of curve [parabola], curve opens downward". She explored $\mu = .2$ and $.4$ noting the Cobweb "rectangle" shapes. For $\mu = .1$, $x = .1$ she noted "nothing happened". The flat curve for $\mu = .1$ was displayed but no Cobweb was drawn as the x-value was too close to zero to appear.

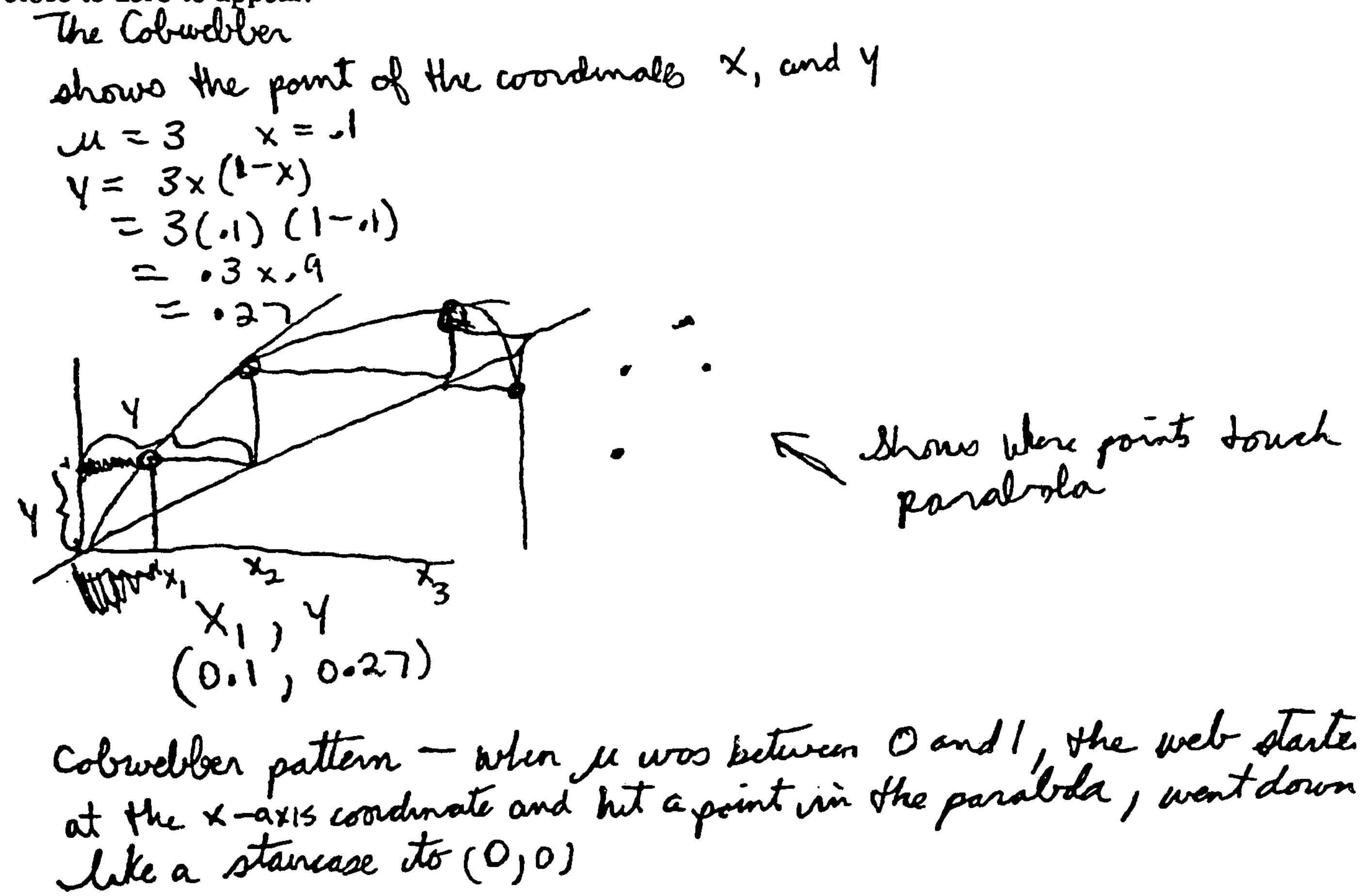


Figure 6-8. Joy's notes about the Cobweb and Iterate Graphs of the Logistics Equation for the Function $y = 3x(1 - x)$, $x_0 = .1$

During a class discussion of Cobwebbing Joy made notes and recorded information about how the Cobweb was drawn (Figure 6-8). Joy noted $\mu = 3$, $x = .1$ and wrote "all points on Cobweb form a 90° angle either touch the parabola or the $y = x$ line". From her notes notice that she first created the function $y = 3x(1 - x)$ and then substituted $x = .1$ into it. The result is .27. She noted below her drawing of the Cobweb that the point (x_1, y) was (0.1, 0.27). Notice on her drawing of the Cobweb that the point (x_1, y) is indicated as the leftmost darkened point on the parabola. She also displayed three other darkened points on the parabola that the Cobweb 'touched'. To the right of the parabola these four points are repeated in an Iterate Graph which Joy indicated "shows where points touch parabola." This is an explanation of the Iterate Graph as it relates to the Cobweb. Notice also from the Cobweb drawing that two values of y are labeled and indicated by parentheses, one vertically along the y-axis and the other horizontally. This indicates that the value of x_1 is converted into a y-value by drawing a horizontal line from the point (x_1,y) to the $y = x$ line. This action results in the iteration by which the result of the first iterate of $x_1 = 0.1$ is $y = .27$. This y-value of .27 becomes the next x-value or x_2 which is then entered into the function. This process of iteration can be done algebraically (using substitution) as Joy has begun to do above the Cobweb drawing or it can be done graphically using the Cobweb method.

Below the drawing of the Cobweb Joy wrote "Cobwebber pattern - when μ was between 0 and 1, the web started at the x-axis coordinate and hit a point in the parabola, went down like a staircase to (0,0)." She has made a generalized statement of the results of iteration when $0 < \mu < 1$. The graph approaches a fixed point of (0,0) and does so in a "Cobweb drawn from right to left like a descending stair case". Later she noted that for $\mu = .5$, $x = .5$ that the "first point crosses midpoint of parabola". By this she indicates that the input of $x = .5$ creates a line on the Cobweb drawn vertically half-way across the width of the parabola. She indicates her knowledge of how the x-input affects the drawing of the Cobweb in this instance. On her record sheet she noted her favorite Cobweb: "when the μ value was 0.1 and the x-value was 0.1, the web seemed to be in a straight line with a very flat parabola." Since the parabola is so flat with a μ-value of .1 a web cannot be drawn as the screen-resolution cannot demonstrate this detail. Joy has noticed a number of important patterns through her activities. These include 1) a pattern of behavior for the range $0 < \mu < 1$, 2) the effect of the x-input on the behavior of the Cobweb, and 3) the limitation of the capacity of this exploration tool to draw when values are small. She is learning both about the Logistics Equation (the subject of study) and the Cobwebber (the medium through which she studies). Limitations of the medium must limit and shape her learning. However her learning and understanding may also shape her exploration of the medium. Joy demonstrates an interplay between learner and medium.

Mike began by exploring $\mu = 4$ with $x = .5, .9, .2$, and .1 and drew carefully the Cobweb for each (Figure 6-9). Notice for each that the initial value of x is noted below the initial vertical line of the Cobweb for Figures 6-9a, c, and d. Notice also that the Cobwebs seem to proceed from these initial lines. Through these drawings Mike is attempting to trace the pattern of events of the Cobwebs drawing. Also notice that each of these drawings proceeds for only 4 iterations.

For $\mu = 1$, $x = .3$ and .5 Mike drew the Cobwebs and noted that "when $\mu = 1$ the line [Cobweb] is going down [and approaching the fixed point] to (0,0)." Mike explored Cobwebs in the region of chaos and stated that his favorite iterate was "when the $\mu = 4$ the line just reflect[ed] by the $y = x$, then reflect by the curve. I wonder ... why the line [is] reflect[ed] on the $y = x$ and the curve." Mike has noted a pattern of behavior of the drawing of the Cobweb by viewing a chaotic Cobweb. From this he questions why the Cobweb is drawn as it is. Perhaps Mike's choice of a chaotic Cobweb meant that he did not see a particular finite pattern to watch. Thus he found himself noticing the drawing of the Cobweb rather than the behavior of the function.

Tony explored $\mu = 2.8$, $x = .34$ and .5 but made no record of results. It seems this exploration was important enough to note but not important enough to record findings. He then explored $\mu = .5$ and .3 with $x = .5$ for which he wrote "the web has one line that continuously approaches 0" and "the web constantly approaches zero". The approach of the fixed point is evident. Tony then systematically explored $\mu = 3.5, 3.0, 2.5, 2.0$ and 1.5 all with $x = 0.05$. He stated for $\mu = 3$ (Figure 6-2h) "the curve [parabola] is tall and the web [Cobweb] doesn't exit the curve until the curve goes down." From his comment Tony may have meant that the lines of the Cobweb stayed close to the parabola until the slope became negative at which point it changed and approached a fixed point. Tony's systematic approach may have lead him to these conclusions about the behavior of the function and the drawing of the Cobweb.

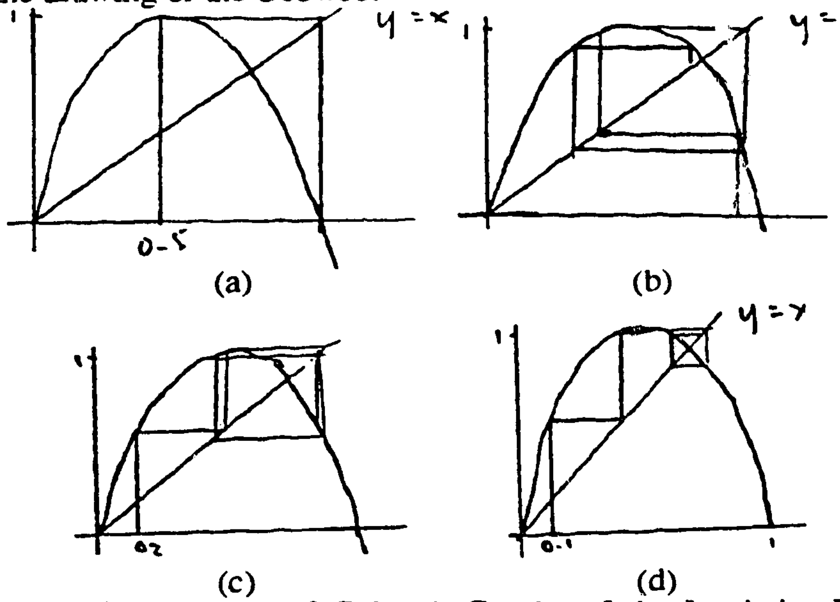


Figure 6-9. Mike's drawings of Cobweb Graphs of the Logistics Equation for the Function $y = 4x(1 - x)$ and x_0 values of a) 0.5, b) 0.9, c) 0.2, and d) 0.1.

Tony observed for $\mu = 2.5$ that "the curve [parabola] is smaller than before [$\mu = 3$] and the web looks almost the same". For $\mu = 1.5$ Tony wrote "the curve gets smaller and it [the Cobweb] has the same reaction". It appears that for each value of $\mu = 3, 2.5, 2$, and 1.5 that the pattern he noticed was the same but the nature of the pattern was unclear from his description. Perhaps what he meant by "has the same reaction" was that each approached a fixed point. Tony expressed some frustration during the session because he did not seem to feel focused nor understand what was "going on". As a result the researcher explained Cobwebbing to the students as demonstrated by Joy's notes (Figure 6-8).

When Tony explored $\mu = .9$, $x = .5$ he described "the web goes from the top of the curve [parabola] to the $x = y$ line and down to the curve again. The web eventually stops at (0,0)." Tony noted the action of the Cobweb and the pattern it demonstrated. Tony stated when prompted that "the iterate of my choice is 3.9 and 0.99 because it never seems to fall into a pattern that is indistinguishable between the lines of the web."

Cindy explored on this second day primarily focusing on the Iterate Graph. She noted for $\mu = 3.2$ "the dots went up then continued to make a pattern going down then up" which seemed to be a description of a 2-cycle. For $\mu = 3.9$ she stated "a chaotic pattern -> undefined pattern." For $\mu = 2.1$, $x = .77$ she stated "two dots went up then it stayed at a constant line." In her first 3 observations she noted a 2-cycle, chaos, and approaching a fixed point (attractor). Cindy then explored $\mu = .3, .7, 1.3, 1.9$, and 1.65 with $x = .5, .9$ and .1 for each. For $\mu = .3$ and .7 she noted that the Iterate Graph "sloped down and stayed constant" but the constant was not described (although it was 0). For $\mu = 1.3$ she noted that it "sloped upwards and over" to the right. For $\mu = 1.9$ the approaching of fixed point was more clear in her statement "sloped up and stayed constant across." Cindy noted for $\mu = 1.65$, $x = .5$ "slightly sloped down and stayed constant" but for $x = .1$ and .9 "sloped up and remained constant." She seemed interested here in how the constant was approached rather than the value of the constant. However, the value of the constant was implied in her note that "between [$\mu =$] 1 + [and] 2 the value of x when equal to 0.5 makes the graph slope downwards otherwise the graph slopes upwards." She also noted that "at

$\mu = 1.9$ and $x = 0.5$ the graph just goes straight across." From this statement it appeared that $\mu = 1.9$ approached the constant 0.5. Actually it was slightly less than this but the screen resolution could not demonstrate the difference.

Cindy made notes about Cobwebbing based on class discussion. She stated that "Cobwebbing is the idea of drawing out the logistic equation. Instead of calculating you can follow the pattern. It maps out the value of 'y'. The $y = x$ line is a guide along with the parabola. You start at the initial value until you hit the parabola then over until you hit '$y = x$'. This allows you to read off the values of 'y'."

John explored values of $\mu = 1.5, 0.5, .75, 2.0$ and 2.3 noting the length of the "tail" (list of points on the Iterate Graph) and the observation that it "levels at" or "descends to" .33, 0, 0, .5 and .57 for the respective μ values. From this point on John simply wrote a number (the value of the fixed point) in the observations column. He noted for $\mu = 1.6, 1.7, 1.8, 1.9, 1.1, 1.2, 1.3, 1.4$ the respective fixed points 0.38, 0.41, 0.49, 0.57, 0.1, 0.17, 0.23, and 0.29. John had a clear image of the behavior of the Logistics Equation which approach a fixed point. When asked about his favorite iterate he noted "between [$\mu =$] 1 & 2 are interesting. Final # between 1.5 & 1.9". Not only has John noted the approach of a fixed point but he stated the range in which the fixed point fell. John had not been encouraged to explore this way but from his own activities these generalizations emerged.

Students at this point had begun to discover patterns of events within individual Cobwebs or Iterate Graphs. All students had begun to explore and discover mathematical patterns, in particular those of fixed point, 2-cycle and chaos. From the class discussion of Cobwebbing it seemed that the functioning of the Cobwebber was becoming clear. These findings then were twofold. Students were increasing their understanding of the mathematics through the context. And they were beginning to make sense of the medium through which these explorations were taking place. In essence students were bringing forth a world of mathematics through the computer medium. Students' actions in this computerized medium were mainly mathematical.

Continued explorations with the Cobwebber might facilitate further noticing of patterns evident from specific functions and further noticing of patterns generalized across a number of Cobwebs. To facilitate such noticing, explorations by systematically viewing the results of varying values of μ (like those undertaken by John) were planned for group participation at the next meeting. This systematic exploration could be directed to maximize the likelihood that students would make comparisons between values of μ and the effect of this value.

What is the effect of μ?

At the third meeting each student was assigned a value of μ to explore beginning with Mike with $\mu = .1$, John $\mu = .2$, Cheryl $\mu = .3$, Joy $\mu = .4$, Cindy $\mu = .5$ and Anne $\mu = .6$. Students were asked to try different x-values to determine their effects. Students were also asked to look at each others computer screens and compare the effects of changing the μ-value. What was noted from these explorations was that all values of μ thus far led to iterates which approached 0 despite the value of x. Students continued to explore increasing values of μ by .1 until $\mu = 1$. Cheryl wrote the observation "Everything to the μ value of 1 ... approaches zero." Cindy noted for $\mu = 1.1$, $x = 0.5$ that the Iterate Graph "sloped down slowly $\neq 0$, [but] .09." She then tried $\mu = 1.1$, $x = 0.1$ and noted "straight across then down one point [pixel on the screen] to .09." She then tried $\mu = 1.1$, $x = 0.09$ and noted "straighter across at .09". The value approached was not 0 but .09 which Cindy confirmed for herself by her systematic explorations. Cheryl noted that "after 1.2 it [fixed point] starts to climb." John noted for $\mu = 1.4$, $x = .2$ that the Iterate Graph forms an "ascending tail [which] levels at point at ($x = y$) & parabola" or the intersection point of the line $y = x$ and the parabolic curve. Joy noted for $\mu = 1.6$, $x = 0.1$ "staircase is climbing and contained in area between $y = x$ line and parabola last meet. Stops are .37".

The researcher recorded values students reported on the white board at the front of the room. A chart of values listing the value of μ and the x-value approached was created (Figure 6-10).

μ	x	μ	x	μ	x	μ	x
.1	0	.6	0	1.1	.09	1.6	.37
					} .06		} .04
.2	0	.7	0	1.2	.17	1.7	.41
					} .06		} .03
.3	0	.8	0	1.3	.23	1.8	.44
					} .06		
.4	0	.9	0	1.4	.29	1.9	
					} .06		
.5	0	1	0	1.5	.33	2	

Figure 6-10. Chart of Values for μ with the value approached for each and the difference between subsequent values approached.

After completing the chart with values of the fixed point in the range $.1 < \mu < 1.8$ the researcher asked students to predict the value for $\mu = 1.9$. Cindy noted that for the fixed point x there was a change of .06 for 3 values of μ, then a change of .04 for 4 values of μ and then a change of .03 for the next values of μ. From this pattern she predicted a fixed point of .47 which was a change of .03. Most students saw the pattern and predicted .47 for $\mu = 1.9$. The last 2 values in the chart $\mu = 1.9$ and 2 were completed with .47 and .5 respectively.

Students noted that these values approached the point of intersection of the $y = x$ line and the parabola. Since the equation for the parabola being discussed was $y = 2x(1 - x)$ the researcher suggested that students could solve the system of equations $y = x$ and $y = 2x(1 - x)$. Students had solved systems of linear equations in their Mathematics classes. Joy's solution to the system proceeds in Figure 6-11.

Students solved the system of equations to the step where $0 = x(-2x + 1)$ but were unsure as to how to continue. They had previously only solved systems of equations using two linear equations. To solve a system with a linear and quadratic equation was new to them. The researcher suggested that they set each factor equal to zero and solve. Students solved the system finding two solutions. The zero solution had to be explained as one of the points where the two graphs intersected. The second solution was $x = \frac{1}{2}$ which students had previously realized was the fixed point when $y = 2x(1-x)$ was explored.

Now it was suggested that students solve the system of equations in the same way using $y = x$ and $y = \mu x(1 - x)$ to gain a general solution. Joy's solution to this system is provided in Figure 6-11b. Students were encouraged to test these general statements to see if it produced appropriate results for some μ values already established. It was also suggested that students try this general solution to predict the outcome of an unknown μ value and to then verify the result using the Cobwebber. At the bottom of Figure 6-11b you can see Joy's solutions and her explorations for $\mu = 1.1$, 1.12, and 2.1.

At the end of this third meeting very little time remained for students to respond to researcher questions so students were asked to simply record what they had learned during this day.

John began with the title "Stuff learned" and listed the following 4 points:

1) x-value causes tail [list of points in Iterate Graph] to ascend from 0 - 0.5
   [drew a curve rising and leveling off as it moves to the right]
   causes tail to descend from 0.5 - 0.99
   [drew a curve falling and leveling off as it moves to the right]

2) general statement for prediction $\frac{\mu - 1}{\mu} = x$
3) 0 - 0.4 μ value takes longer to approach zero
4) leveling (approach) takes place at point of intersection of $x = y$ line & parabola

Statements two and four are direct generalizations from the days class discussions. The fixed point that is approached by the Logistics Equation with a given value can be predicted by using the value in the prediction formula. Statement four indicates that this value that is approached (the fixed point) is determined by the intersection of the $x = y$ line and the parabola.

Statements one and three were not part of class discussion but are statements made solely on John's explorations. Statement one indicates that when the x-input is below the midpoint of the parabola it causes the values of the iterates in the Iterate Graph to ascend toward the resulting value. But when the x-value is above the midpoint of the parabola it causes the iterates to descend toward the fixed point.

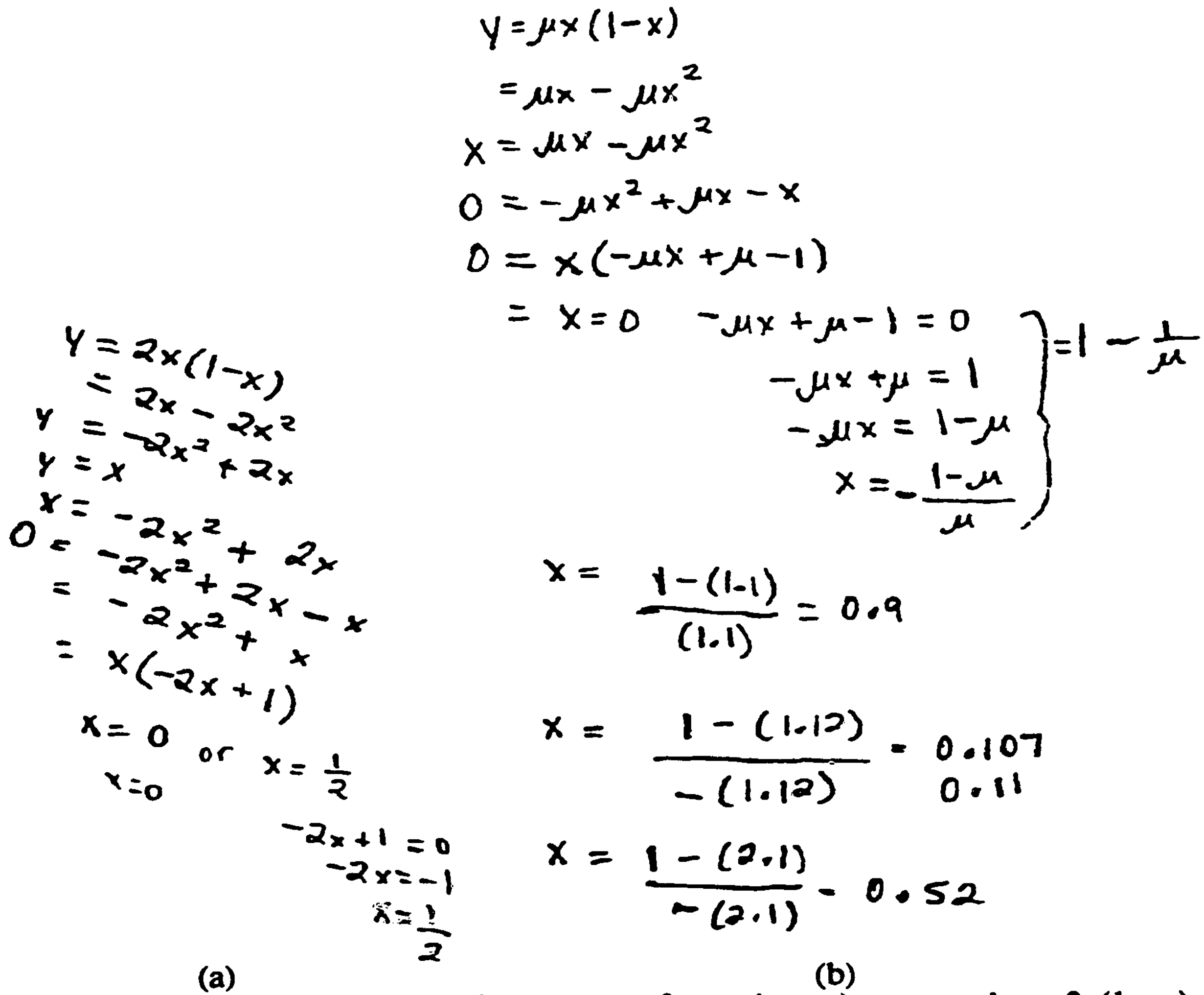


Figure 6-11. Joy's solutions to the systems of equations a) $y = x$ and $y = 2x(1 - x)$ and b) $y = x$ and $y = \mu x(1 - x)$.

Statement three seems to indicate that very small values of μ approach the fixed point of zero in a large number of iterates thus taking a long time. This statement is perhaps contrary to expectations because the larger the μ value the more points it will take to reach the fixed point of zero if given an x-value near the maximum point of the parabola.

However if the parabolic curve is very flat the initial point will be very near zero so one might expect it to approach very quickly. However since the points begin very close to zero they approach it in very small increments. It may take many iterates to notice that an approach of the fixed point 0 is actually occurring. If however the μ value is larger the fixed point is approached in larger 'steps' making the pattern of the approach obvious in a relatively small number of iterates.

Cheryl wrote statements perhaps derived or generalized from the class discussion.

"We discovered the μ formula or the formula where you can figure x out by simply doing $1 - \frac{1}{\mu}$."

"Every μ value up to 1 approaches zero on the graph. At line, it bordered on 0.9. After 1.2, the graph starts to climb."

"You were sort of able to predict the next x-value for μ form the other ones already done. There was a pattern."

"The 2nd graph [Iterate Graph] always ends [approaches] where [at the height] the $y = x$ line crosses on the parabola." For this one she drew a parabola and $y = x$ line intersecting and then drew an Iterate Graph next to it leveling off at the same height as the intersection.

Joy wrote

"Patterns discovered - if μ is 1 or less the Cobweb pattern is a descending staircase. If μ is larger than 1, the staircase starts to climb and is contained in the area between $y = x$ and the parabola. The pattern ends at the point where $y = x$ and the parabola meet. As μ is increased, the parabola becomes taller. The beginning of the pattern starts to shift to the right as μ is increased. At 1.2, the pattern starts before the middle of the x-axis. At 1.6 the pattern starts at the middle of the x-axis."

She drew two graphs explaining this point. The first graph showed a parabola through which the $y = x$ line was below the maximum of the graph. The Cobweb looked like a staircase ascending to the intersection of the $y = x$ line and the parabola. The second graph showed a parabola and $y = x$ line which intersected to the right of the maximum of the parabola. The Cobweb approached this intersection by circling around it.

Joy finally wrote

"$x = 1 - \frac{1}{\mu}$ this is the final value that the pattern approaches."

Anne wrote

"When the value of μ is below 1.2 the graph slopes down toward 0."

"When the value of μ is greater or equal to 1.2, the graph slopes upward towards a given number which can be determined through the formula $1 - \frac{1}{\mu} = x$"

Cindy wrote:

-"for 1 value of μ the graph always slopes either up or down to certain value. This is consistent."

-"this value can be calculated through the formula $x = \frac{\mu - 1}{\mu}$"

-"while testing it appears that a μ value between 1 - 2 the graph slopes up (unless its near the calculated x value). A μ value between 0 - 1 shows the graph slopes down and always to zero."

-"now we can calculate the value the graph reaches and remains constant at by using the formula $x = \frac{\mu - 1}{\mu}$."

Mike wrote

"When μ value is between 0 - 1 then .7 approach 0 and is going down." He added as an aside "0 means the numbers are rounded off to 0."

"When μ value is > 1 then the no. [number] that approach is higher. That means it approach 1. The equation $y = \mu x(1 - x)$ when let μ = a number then we can know where $y = x$ touch parabola."

"The equation $y = \mu x(1 - x)$ after solve it to $x = 1 - \frac{1}{\mu}$, then we can predict the x value." Mike also drew a parabola intersected by the $y = x$ line and darkened the intersection points at 0 and $1 - \frac{1}{\mu}$.

A few general statements seem clear through students statements. It seems clear to these students that functions with μ values less than one approach zero when repeatedly iterated. Functions with μ values greater than one approach a value predicted by $1 - \frac{1}{\mu}$ or $\frac{\mu - 1}{\mu}$ and this value is the intersection of the parabola with the $y = x$ line. However in some cases there seemed to be some confusion. Mike for example suggested that values of μ greater than 1 all approach 1 but then stated the fixed point is where the two "touch" which variable based upon μ.

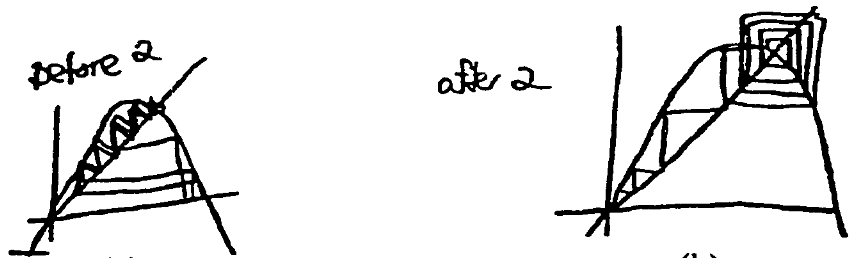
Each student also added their own personal meanings to this list. Cheryl noted that prediction of the x-value approached for particular μ values was possible because there "was a pattern". Mike emphasized that when a statement was made that a function went to zero it actually was "rounded off" to this value seeming to emphasize the fact that this was an approach not a reaching of this value. John noted that some values of μ ascended to reach the fixed point while others descended. It seems there was concern with not only the final value approached but also how this value was approached. Joy, as did John, noted how the approach of a fixed point occurred. She noted the change in approach due to increased μ values. Anne and Cindy both used the terms "sloped down" and "sloped up" in describing the approach of a fixed point. They also seemed to be concerned with the process of approach rather than only the product.

What is the effect of μ as it increases from 2?

The fourth gathering of students to explore mathematics began with a review of the previous day and the regeneration of the μ and fixed-point chart for $\mu = 0.1$ to 2 (Figure 6-11). The equation for prediction of x-values approached based upon μ, or $x = 1 - \frac{1}{\mu}$, was stated by students when it was asked of them. It was suggested by the researcher that students continue exploring values and complete the chart for $\mu > 2$. As on the third meeting the group worked together, each student explored a μ value increased by .1 from the last. Students suggested the x-value approached and the researcher recorded their responses in the chart which quickly included values for $2.1 \leq \mu \leq 3$.

Tony simply listed on his record sheet for $\mu = 2.2$ and 2.8 that "it ends on 0.54" and "it approaches 0.64". John explored $\mu = 2.3$ for x = .1, .2, .4, .6, .5, .8, .95 and .55 and found that it approached 0.57 for each but he noted that the type of approach was different depending on the x-value. He described a "tall 4-point tail" for x = .1 and "no perceptible tail" for x = .4. For x = .95 there was a "really tall 4 point tail" and for x = .55 "no tail at all". The term tail appeared to refer to the Iterate Graph and the number of iterates required to approach the fixed point. No tail seemed to imply that the fixed point was reached immediately while a 4 point tail seemed to imply 4-iterates before the fixed point appeared to be reached.

Cheryl drew Iterate Graphs for $\mu = 2.4$ showing patterns of points produced by the iteration of various x-values. She then summarized her findings of the Cobweb graphs by drawing two Cobweb graphs which typified the types of Cobwebs she had seen below a μ-value of 2 (Figure 6-12a) and above 2 (Figure 6-12b). Note in these that below 2 the fixed point was approached directly in her diagram but above $\mu = 2$ the approach was to circle around the fixed point ever approaching it. It appears that below $\mu = 2$ the Cobweb was caught in the region between the $y = x$ line and the parabola while above $\mu = 2$ the Cobweb approached the fixed point first from above and then below circling around it as it slowly approached.



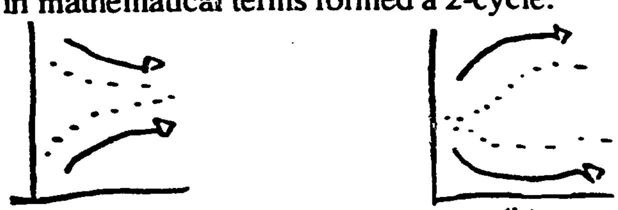
(a) (b)

Figure 6-12. Cheryl's Summary Cobwebs of the Process of Approaching a Fixed Point when a) $1 < \mu \leq 2$ and b) $2 < \mu \leq 3$.

Joy explored $\mu = 2.5$ with $x = .1, .4, .7, .9\ .3,$ and $.6$. She described the appearance of these Cobwebs as a "staircase going up, approached 0.6". She drew some of the Cobwebs to exemplify the process of approaching the fixed point. She used the term "spirals" when describing the appearance of these Cobwebs as they approached the fixed point. Cindy explored $\mu = 2.6$ with $x = .5, .2, .9$ and $.99$ noting an "up and down pattern leveling off at 0.62". Mike explored $\mu = 2.7$ with $x = .1$ and wrote that it approached .63. From each of these students statements it was clear that the Logistics Equation still approached a fixed point when $1 < \mu < 3$.

Students began to explore $\mu \geq 3$ and noted a change which occurred. Cindy noted for $\mu = 3$ and $x = .5$ that the iterates seemed to "alternate up and down" at the values .65 and .68. Joy explored $\mu = 3.1$, $x = .1$ and noted "x alternates between two values .56 and .76. Spiral ascends and forms a square spiral around $y = x$ and parabola crossing point." She then used $x = 1 - \frac{1}{\mu}$ and solved for $\mu = 3.1$ for "$x = 0.6774193$". Joy then entered $\mu = 3.1$, $x = .66$ and drew the resulting Iterate Graph, appearing like two almost horizontal lines of points one with an increasing slope and one with a decreasing slope. She also drew the lines of the Cobweb appearing to "spiral" around a value.

Cheryl had explored $\mu = 3.2$ and compared this to $\mu = 3.0$. She drew the Iterate Graph and added arrows, which emphasize the convergence of $\mu = 3$ to a single value (Figure 6-13a) and the divergence (convergence to 2 values) of $\mu = 3.2$ (Figure 6-13b). For $\mu = 3.0$ the iterates still approached a fixed point but for $\mu = 3.2$ the iterates approached 2-values or in mathematical terms formed a 2-cycle.



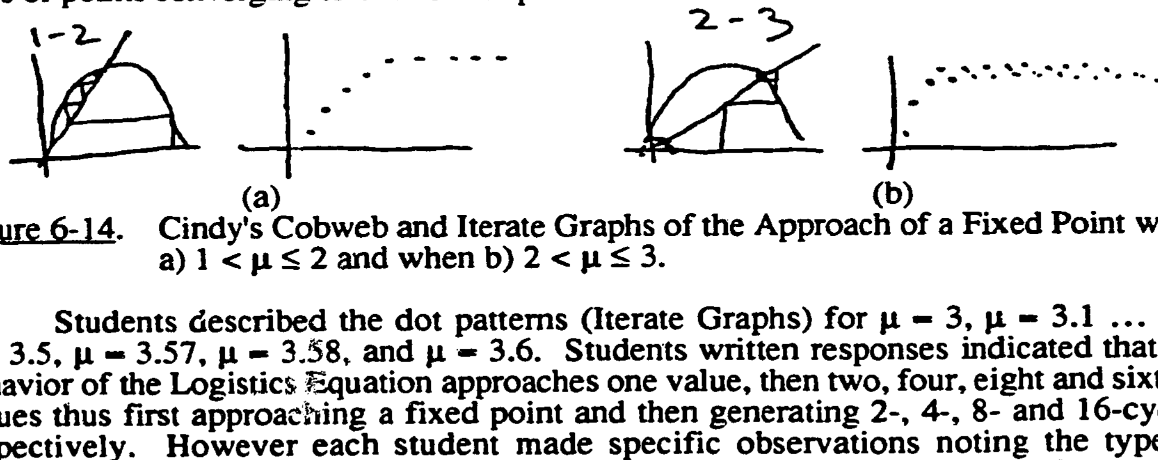
(a) (b)

Figure 6-13. Cheryl's Cobweb and Iterate Graphs of the Approach of a Fixed Point when a) $\mu = 3$ and the Process of Approach of a 2-cycle when b) $\mu = 3.2$.

Tony simply noted that for $\mu = 3.3$ "it goes out and it approaches 0.48 and 0.82". He then explored $\mu = 3.0$ and 3.1 drawing Iterate Graphs noting (incorrectly) that $\mu = 3.0$ "approaches 0.64 and 0.76." He needed to watch more iterations to note that this function actually approached one value midway between these two value. His Iterate Graph showed that for $\mu = 3.1$ the graphs approaches 2 values but for $\mu = 3.0$ his graphs looks like 2 parallel line of points, not obviously approaching 2-values. It may be that Tony was lead to conclude that this was a 2-cycle because others around him were exploring higher values which were indeed 2-cycles. He may have expected to see a 2-cycle so he saw one.

John did not list all of his explorations but instead drew the Iterate Graphs for $\mu = 3.5$ and 3.4 with iteration points connected and noted "4-set values" (a 4-cycle) and "2-set values" (a 2-cycle) respectively. He also noted that $\mu = 3.7$ produced an "erratic" (chaotic) result.

What changes did students notice about the Cobweb when $\mu = 2$ and $\mu = 2.1$, 2.2, ... 3 were entered? Joy noted that "When μ was above 2 but below 3 the pattern ... seemed ready to spiral around the intended final x value. The pattern started to move out of the confines of [the space between] y = x and the parabola". Cheryl wrote that "In between 2 and 3 the [Cobweb] made a sort of spiral, but not quite a full one." She added incorrectly that "3 & up it made a spiral around the approached value." In fact above 3 it did not spiral around a single approached value but approached more than one value. Cindy clarified in written remark that "When entering [μ =] 2 - 3 the Cobweb changed from working it[s] way up to going up then down alternately." (Figure 6-14) In fact between 2 and 3 the Logistics Equation continued to approach a fixed point. Initially it approached it directly from either above or below the fixed point but later it approached from above and below at the same time thus forming an "up then down" pattern. John generalized the pattern of approach of the fixed point for "$\mu = 2$ - all lines in region between x = y & parabola. Stopped at intersect point." He added that for "$\mu = 2.1$, 2.2 ... some lines got above intersect point." In other words spiraling or up and down patterns occurred. Mike drew diagrams to illustrate typically what he thought was happening in each range of value. When $\mu = 2$ the Cobweb was an ascending staircase and the Iterate Graph an ascending curve of points. For $\mu = 2.1$ to 2.9 the Iterate Graph was the same but the Cobweb approached from both sides. For $\mu = 3$ the Cobweb spiraled around the point of intersection of y = x and the parabola and the Iterate Graph was composed initially of two rows of points converging to one row of points.



(a) (b)

Figure 6-14. Cindy's Cobweb and Iterate Graphs of the Approach of a Fixed Point when a) $1 < \mu \leq 2$ and when b) $2 < \mu \leq 3$.

Students described the dot patterns (Iterate Graphs) for $\mu = 3$, $\mu = 3.1$ ... 3.4, $\mu = 3.5$, $\mu = 3.57$, $\mu = 3.58$, and $\mu = 3.6$. Students written responses indicated that the behavior of the Logistics Equation approaches one value, then two, four, eight and sixteen values thus first approaching a fixed point and then generating 2-, 4-, 8- and 16-cycles respectively. However each student made specific observations noting the type of approach in generating these cycles. John noted that the "Dot pattern for $\mu = 3$ [generated] 2 lines, one ascending, one descending." He indicated that for "$\mu = 3.1$–3.4 [the pattern generated] 2 lines separating [into] 2 values & leveling off." However he noted that for $\mu = 3.5$ the pattern was different beginning as "2 lines zig zag instead of leveling" and then approaching "4 fixed values." John noted that for "$\mu = 3.57$ [there were] 8 values [created by] Big zigzagging." Finally John stated that $\mu = 3.58$ marked the "beginning of chaos."

Mike provided images of the general patterns or behavior for $\mu = 3$, $\mu = 3.1 \ldots 3.4$, and $\mu = 3.5$ (Figure 6-15) indicating the convergence to one value at $\mu = 3$ and the divergence to two values when $3.1 \leq \mu \leq 3.4$. For $\mu = 3.5$ his diagram seems to indicate more than just two values if one looks closely but 4 values is not definite from his diagram. For $\mu = 3.57$ and $\mu = 3.58$ Mike reports 8 and 16 values respectively noting chaos for $\mu = 3.6$.

(a) (b)

Figure 6-15. Mike's graph of convergence to one value when $\mu = 3$ and divergence to two values when $3.1 \leq \mu \leq 3.4$.

Cindy described the results for "$\mu = 3$ [as] the dot pattern goes up and down but eventually levels off at .5." She noted that the approach of the 2 cycle for $\mu = 3.1$-3.4 "starts together and eventually spreads apart." She also noted that "the dots go up and down at 4 values" for $\mu = 3.5$. Cheryl described this same pattern as "dots were scrambled for 4 pts then repeated themselves." Joy reflected a similar notion in describing the patterned approach as "the dots go up and down and approach 4 values." All students noted that at $\mu = 3.6$ the behavior was chaotic and Cindy and Joy used the term erratic to further describe this behavior.

A generalized pattern of behavior of the Logistics Equation has been noted by students beginning with the approach of a fixed point when $0 < \mu \leq 3$. The fixed point approached was 0 when $0 < \mu \leq 1$ and approached the value $\frac{\mu - 1}{\mu}$ when $1 < \mu \leq 3$. Above this value students noted 2-, 4-, 8-, and 16-cycles as well as chaos in the range $\mu > 3.6$. Students were asked to provide a graph of the patterns and behavior based upon their generalized findings possibly referring to the charts of fixed point values they had generated. Figures 6-15 to 6-20 present these graphs.

Figure 6-16 shows Joy's graph of the patterns of behavior of the Logistics Equation using the μ values as the independent variable and the values approached as the dependent variable. However her graph shows a 'typical' Cobweb in each range from $0 < \mu < 1$, $1 < \mu < 2$, $2 < \mu < 3$, and $3 < \mu < 4$ and actually does not indicate the actual values approached but seems rather to indicate the pattern of approach. Also notice that the height of the parabola decreases as μ increases. This is actually reverse of the events students witnessed in their use of the Cobwebber. Looking at Joy's first Cobweb ($0 < \mu < 1$) it is unclear where this Cobweb begins. If it begins near the maximum height of the parabola then it approaches zero in stair-like steps which is true for this range. However it can also be read beginning at the lower left side increasing in steps which would not be correct. It is unclear which is intended here.

Joy's second Cobweb showed a Cobweb beginning at the x-axis, climbing to the parabola and horizontally to the $y = x$ line. However she then continued this horizontal line to the parabola instead of continuing with a vertical line to the parabola. However she continued to draw step-like stairs increasing toward the intersection of the $y = x$ line and the parabola with the lines of the Cobweb contained between the $y = x$ line and the parabola. This was true for all values in this range. Joy then drew a Cobweb ($2 < \mu < 3$) which approached the intersection point by spiraling around it. Again this was correct for this range. Notice that the $y = x$ line no longer appeared to be at a 45 degree angle but was at about 30 degrees. This shift was necessary in order to accomplish an intersection point between the parabola and the $y = x$ line. Had she continued to draw the $y = x$ line at 45 degrees no intersection would have occurred. This would have meant that any Cobweb would have approached zero. This graph would have been appropriate for $0 < \mu \leq 1$. In the range $3 < \mu < 4$ Joy provided a Cobweb showing a stepwise increase followed by a spiral seeming to approach the intersection of the y = x line and the parabola. However it may indeed simply be showing that the values spiral and eventually approach 2 values. In either case the drawing was appropriate for either $\mu = 3$ (approaching the intersection) or $3 < \mu < 3.4$ (approaching two values). This Cobweb could not indicate the approach of cycles larger than two which occurred when $3.4 < \mu < 3.6$ or $\mu \geq 3.6$ this in the chaotic range.

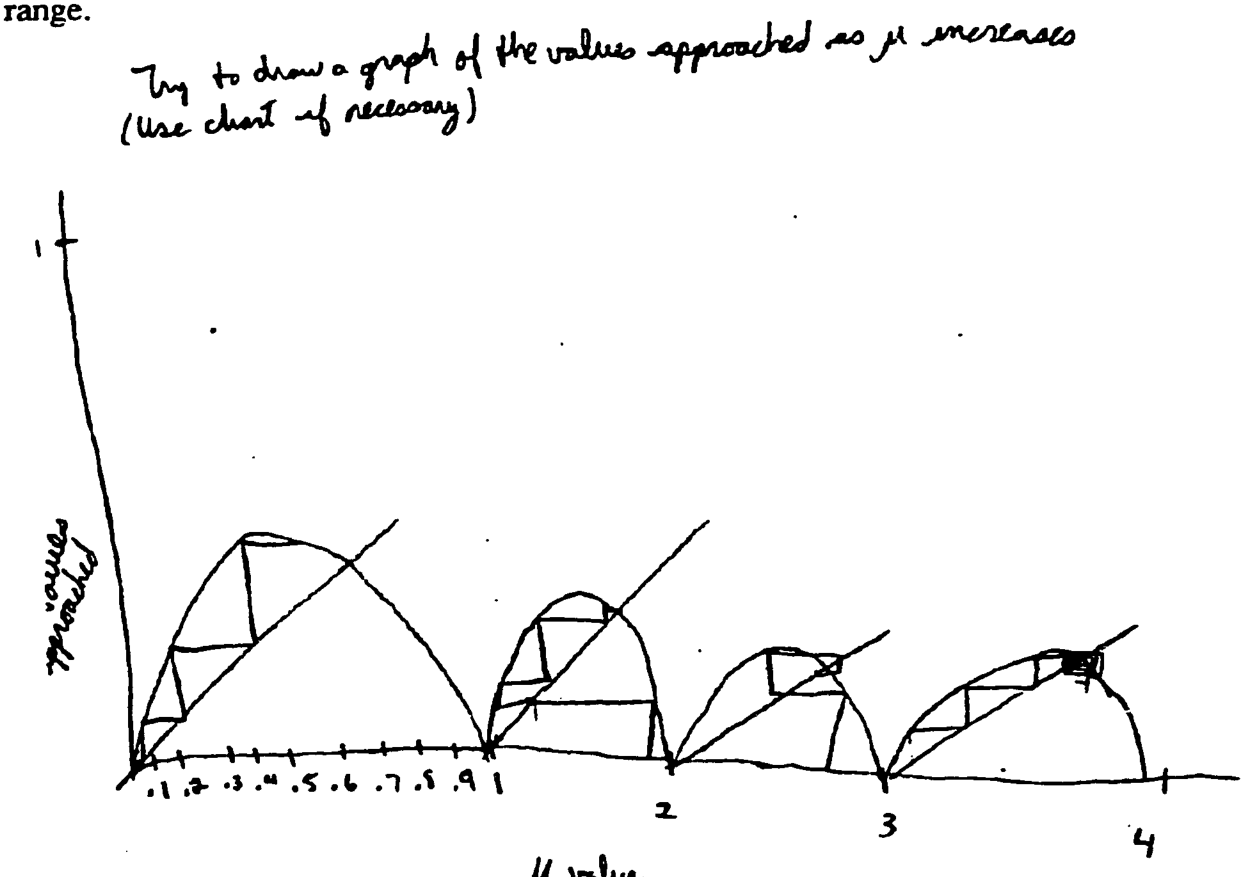


Figure 6-16. Joy's graph of the values approached as μ increased.

In viewing this graph the question emerges: what is Joy's general notion of graphing because this is not a graph with the usual relationship between independent and dependent variables. This non-standard approach managed to convey a considerable amount of information and provided some insight into her understandings. How did the notion occur that the parabola decreases in height as μ increases? In regard to this point notice that the tallest parabola is only about 1/3 units in height if the tick at the top of the graph is to be assumed to be 1 unit in height . Does Joy really wish to indicate that graphs only achieve a height of 1/3? How has Joy come to these conclusions? Perhaps Joy's missing the first day in which students explored Iterate Graphs of the square root function and meaning of those graphs has somehow affected her later perceptions. Perhaps she did not make as much sense of graphs of the Logistics Equation given less early experience. Or perhaps she simply interpreted the task of drawing a graph of the values approached as one of drawing the typical patterns occurring in each range. Whatever the reason for Joy's graph she has indicated that particular patterns of approach of a fixed point occur in particular ranges. This knowledge will be useful to her further explorations.

Cindy drew the graph presented in Figure 6-17. In it she indicated that all values of μ between 0 and 1 approach a fixed point of 0. Beyond this value she indicates the specific values approached as μ increases to $\mu = 3$. However Cindy is unable to draw what occurs in the range $3 < \mu < 4$ which she indicated with a question mark. The approach of a fixed point either of zero or a value greater than zero was clear in her drawing. However Cindy did not draw a graph of 2-, 4-, 8- and 16- cycles as well as chaos.

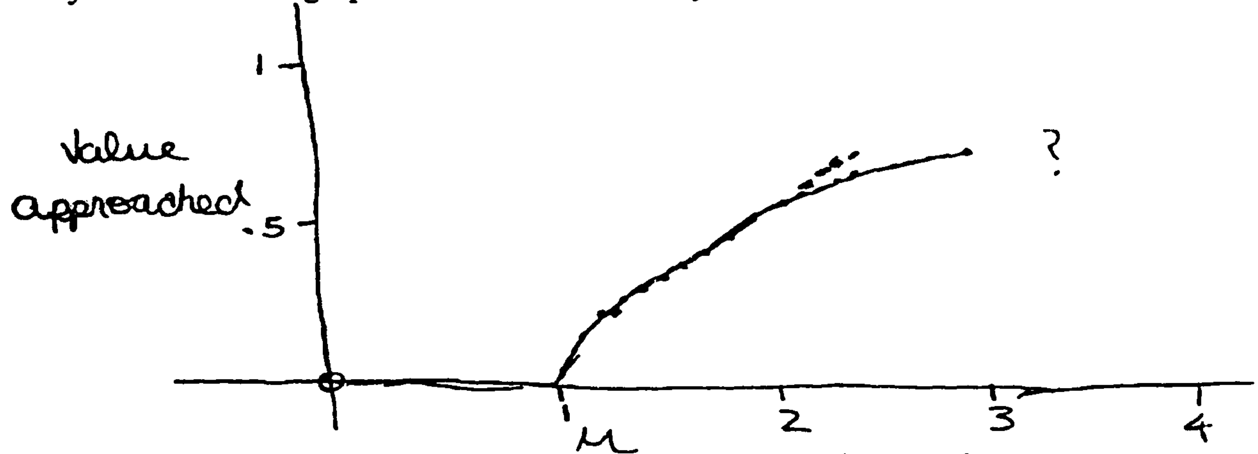


Figure 6-17. Cindy's graph of the values approached as μ increased.

Mike provided a graph (Figure 6-18) indicating specific points that would have been displayed in an Iterate Graph. But which Iterate Graph is this? To answer this question choose two points listed, for example .87 and .38. Substitute these into the logistics equation so that $.38 = \mu(.87)(1 - .87)$. The result is $\mu \approx 3.4$. It appears that Mike has drawn the Iterate Graph of $y = 3.4x(1 - x)$ indicating its particular iterate values. It appears that Mike was unable to generalize from the patterns he had seen in drawing this graph.

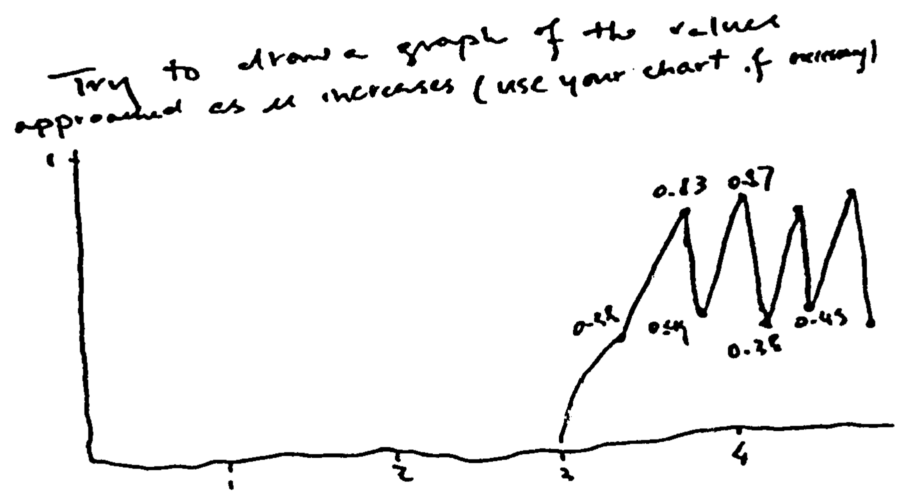


Figure 6-18. Mike's Iterate Graph of the values for $y = .3.4x(1 - x)$.

Tony's graph of the values approached for μ values (Figure 6-19) indicates discrete points approached in the range $0 < \mu < 3.4$ showing only two types of patterns: 1) the approach of a fixed point and 2) a 2-cycle. This graph is interesting because it incorporates discrete points rather than smooth curves. Tony does not appear to have extrapolated that the points between those tried will also fall into the pattern. He also seems unable to draw 4, 8 and 16-cycles nor is chaos indicated. Cheryl's graph (Figure 6-20), like Tony's, uses discrete points to indicate the patterns of approach of a fixed point and a 2-cycle. She however also indicates a 4-cycle. Her graph does not indicate approaching a fixed point of zero for $0 < \mu < 1$.

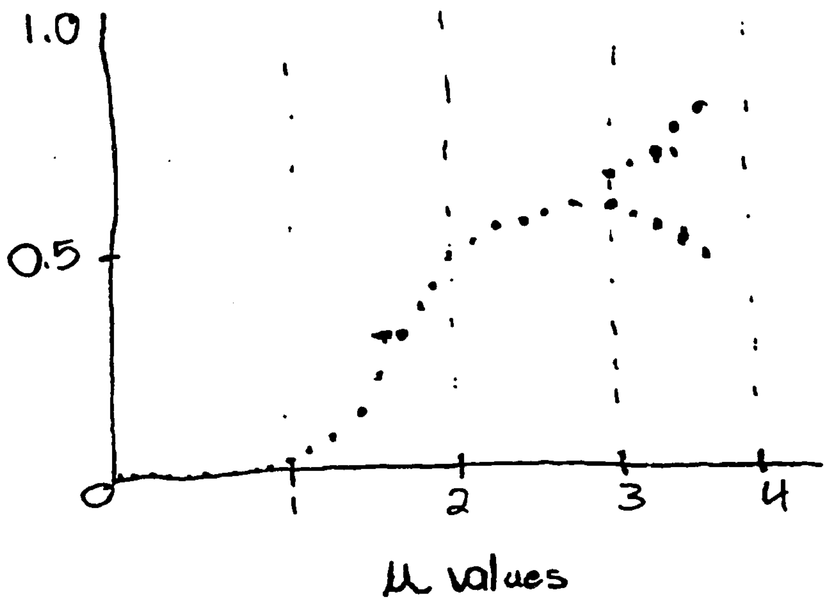


Figure 6-19. Tony's graph of the values approached as μ increased.

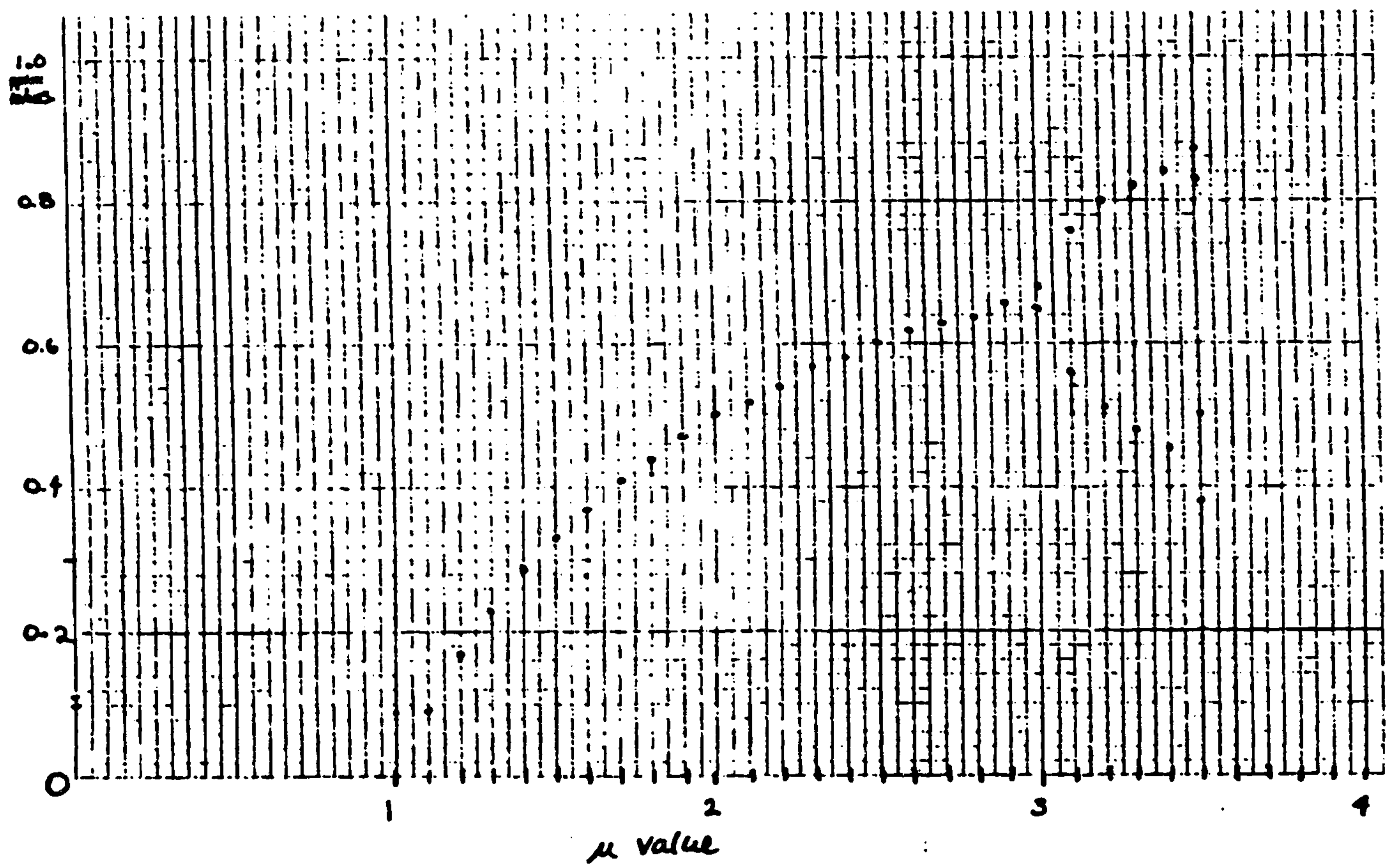


Figure 6-20. Cheryl's graph of the values approached as μ increased.

John's graph (Figure 6-21), like Cheryl's, did not indicate approaching a fixed point of zero for $0 < \mu < 1$. However it indicated the approach of a fixed point for $1 \le \mu < 3$ and the approach of a two cycle up to $\mu = 3.5$ which was a 4-cycle. At $\mu = 3.57$ there is an 8 cycle and then the word chaos was written to the right indicating its range. John was observed while drawing this graph and he did not refer to his table of fixed point values but simply drew this graph (commonly referred to as the bifurcation diagram). John's graph was the most complete including 4- and 8- cycles and indicating the range for chaos. Perhaps John's own discovery of cycles prior to in-session instruction facilitated understanding which enabled his drawing a more complete graph. Generally bifurcation diagrams also include images of the chaos range and within this range some stable cycles that emerge.

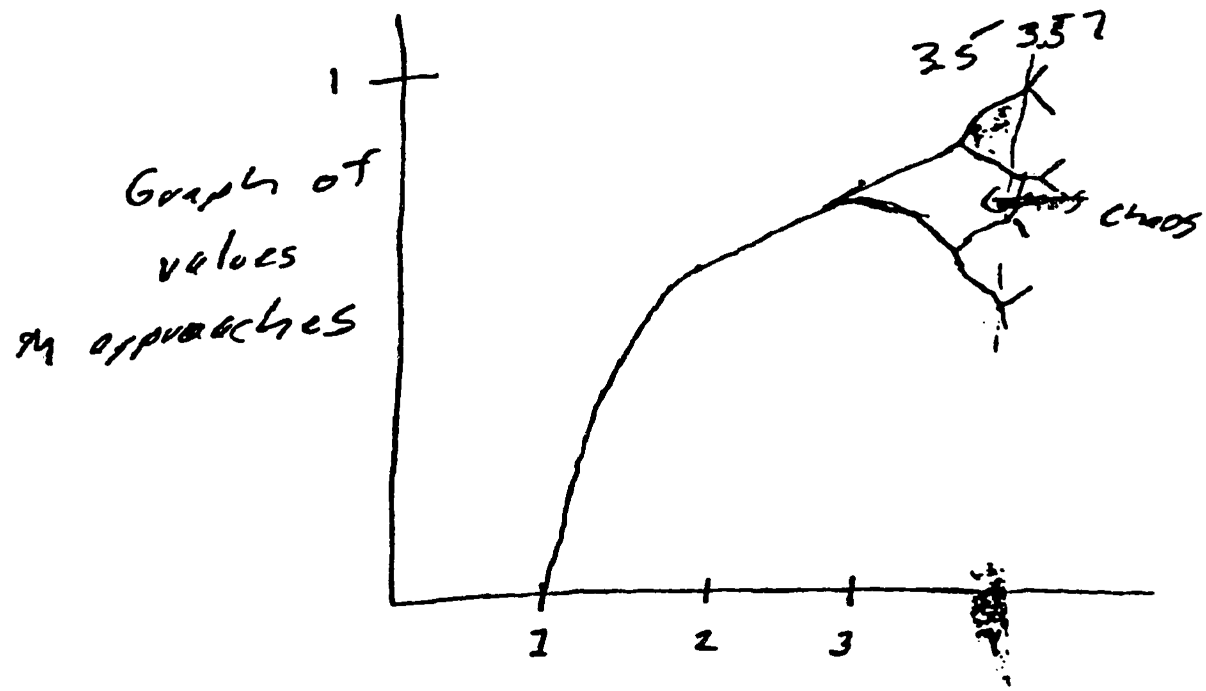


Figure 6-21. John's graph of the values approached as μ increased.

During this meeting students systematically explored the region where $2 \le \mu \le 3.6$. Students discovered that at $\mu = 3.6$ the iteration no longer formed a stable pattern but became chaotic. The task for the next meeting became the exploration of the region of chaos.

Exploring Chaos

Students used the Cobwebber until this fifth meeting at which time the students were exploring the region of chaos produced by the Logistics Equation. In order to notice patterns in this region a larger number of iterates were required. Students grew restless waiting for the Cobwebber to produce a sufficient number. At this point students were introduced to a spreadsheet computer program which calculated values for the Logistics Equation. They were shown how to 1) enter different values of μ and x, 2) see all the values that had been calculated (only 10 appeared on the screen at one time) and 3) generate graphs of the equation. In this program the Logistics Equation was entered into each of 100 cells in a row of cells. The spreadsheet produced an Iterate Graph like that in the Cobwebber but connected the points and provided access to the numeric calculations.

By organizing the views of the values in the cells and the graphs produced it was possible to simultaneously see both the graph and the values (Figure 6-22).

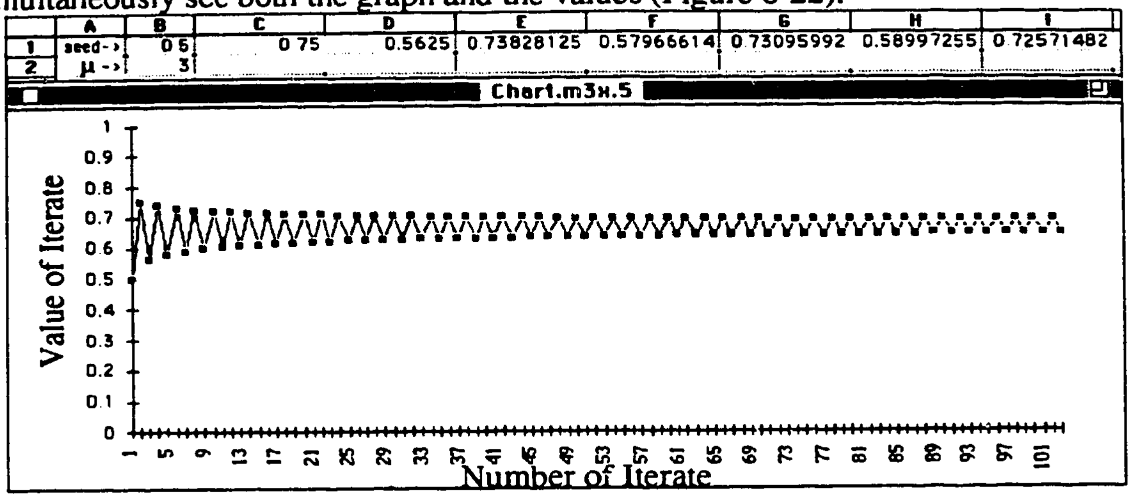


Figure 6-22. Iterate Graph and Iterate Values for the Spreadsheet Logistics Equation for $y = 3x(1 - x)$, $x_0 = .5$.

Students made very few records of their activities on this day. Joy did attempt to make some records as she explored within the region of chaos. She began by exploring $\mu = 3.9$, $x = .6$ for which she wrote "various x values approached, chaos". For $\mu = 3.7$, $x = .6$ she drew a line graph showing the iterates and described it as "patterns almost similar". For $\mu = 3.8$, $x = .6$ she simply wrote "chaos". She then explored $\mu = 4$ with $x = .6, .3, .9$, and .8 and listed the highest and lowest values visible as 1 and 0 respectively. She then explored $\mu = 3.9$ with $x = .8$ and .1 but made no records of her findings. Tony's only record for the days work was for $\mu = 3.8$, $x = .6$ for which he simply wrote "chaos". John explored $\mu = 3.825$, 3.82 and 3.9835 for $x = .5$ and listed the first pattern as "none". For the second function he wrote "close [to a pattern] but not same" and listed the last values he found writing "no repetitions". Cheryl explored $\mu = 3.4$, 3.9 3.8, 3.7, 3.6, 3.5 and 4.0 for $x = .6$. For $\mu = 3.4$ she wrote "There are two points that the x-values approach." For $\mu = 3.9$ she recorded that "There are too many to count". She then made no notes on here explorations except for $\mu = 4.0$ for which she wrote "totally chaotic". Mike, Anne, and Cindy, although present, made no records of their explorations but answered the questions asked by the researcher.

During the meeting a book about fractal geometry was displayed and discussed. The relationship of the beautiful colorful fractal images were described as being generated by repeated iterations of a mathematical expression. If through repeated iteration the resulting pattern approached a particular value it was assigned a color. If it approached infinity at a particular rate it was assigned a color. Various approach rates were assigned various colors. Using these methods colorful images emerge. The fractal known as the Mandelbrot set was shown to students and described as using a function that incorporates complex numbers (which students had been introduced to in their mathematics class). Students looked at the fractal images and admired their colorful patterns.

Students were also shown an image of a bifurcation diagram like those they had generated in the previous session. This image included points drawn in the chaos range which appeared to form a 'grey area' (area of many black points on the white background) and areas of stable patterns (a few points forming a pattern) within this range. These were identified for students inspection. Students were surprised and pleased to find an image like their own drawings found in such a book. The numbers on the bottom of this diagram were different than those discovered by the group of students. It was explained that the bifurcation diagram drawn in this book was for cosine and not the Logistics Equation. The patterns of the bifurcation diagram were the same for both functions although the values were different. Students seemed interested with the generalizability of their drawings.

John asked about the use of fractals and chaos. This seems to be the familiar "what is the use of mathematics?" question but asked out of interest rather than disinterest. Chaos theory was described as useful for as diverse a range of things as a model for psychoanalysis in the social sciences to modeling particle movement in the sciences.

Students made reference to fractals in their discussions of interesting things about chaos. Cindy reported that "the colored patterns [fractals] are interesting as to how the patterns are formed." Cheryl wrote "you can make different sorts of images flowers, pschecelic [sic] wallpaper patterns?" Joy added that "you can draw flowers and actual pictures that represent things in life." Is it important that three female members of this group commented on the fractal images while the male members did not? Are these artistic and beautiful fractal images of greater interest to these female students than to the males? Is the addition of mathematically represented "things in life" of value in relating to the mathematics? There is not enough use of fractal images in this study to answer these questions but they are posed as potential research questions for future study.

What was interesting about chaos? Cindy wrote "Chaos is interesting in the way it's not really chaotic at all. Inside chaos there are many patterns in it." Anne noted that "we were able to discover patterns within patterns. It was also interesting to find out that chaos patterns become stable and chaotic within its own pattern." Perhaps Anne is referring to the bifurcation diagram and the area in the midst of the chaos which stabilizes into another bifurcation sequence before becoming chaotic again. John noted that there were "some bands of non-chaos in chaos." Tony wrote that "it was interesting to see the graph [bifurcation diagram] with the pattern deep within it." Joy reported that "you can make images that show a distinct pattern." Mike also discussed the making of "different kinds of patterns." John recorded that chaos produced "some approximate patterns but [one] never get[s] 2 identical values or the same pattern exactly." Cheryl wrote "I found it interesting that the chaos only lasts for a few decimal points" perhaps in reference to the bifurcation diagram and its images. Anne noted that "The use of these patterns is amazing. ... They can be used in so many different areas of study and technology."

In reviewing these student comments it seems apparent that the term pattern is used in a number of different ways. Cindy stated that inside chaos there were many patterns. Was she referring to the bifurcation diagram with its continual branching into 2, 4, 8, 16 forks? Or did she mean the area within the chaos range where there is an area of many points (grey area) which then becomes stabilized again into this bifurcation branching into 2, 4, 8, 16 forks? Or perhaps she was referring to the patterns formed when she used the Cobwebber. When using the Cobwebber the points from the Iterate Graph seem to come together toward the intersection of the $y = x$ line and in a 'pattern' like those typical when approaching a fixed point. However, in the chaos range this approach is for a short time and then when the points get close to this intersection they are repelled by this intersection, appearing to move away again from this intersection. This repeatedly happens in the chaos range. At one level the term pattern is used to refer to the resulting iteration of a specific function like $y = 3.6x(1 - x)$. On another level pattern refers to the result of generalizing from specific functions to the Logistics Equation $y = \mu x(1 - x)$ across a range of values $0 < \mu < 4$. Or perhaps the term pattern is used to refer to a specific subset of this range. The bifurcation 'pattern' is the sequence of 2, 4, 8, 16 forks (Figure 6-23). This pattern is repeated in the stable region within the chaos range.

Students were asked for a written response to "What exploring did you do and what patterns did you notice?" More information about the use of the term pattern can be gleaned here. Cindy noted that "between [$\mu =$] 3.6 and 4 the pattern is very chaotic. Couldn't find any patterns between 3.6 and 4." The behavior of the iterates which had been formerly called erratic are now referred to as pattern. Below this range where the bifurcation occurs there were no 'patterns'. Anne uses pattern differently when she states that "between roughly [$\mu =$] 3.6 and 4 the patterns seem to be chaotic while below that the patterns seem regular and stable." It seems that this use of pattern is merely a replacement for the term behavior. Mike describes that "when I look at $\mu = 4$, although different points show on the graphic seem to follow a pattern" (Figure 6-24). Pattern here refers to the behavior of a specific function which seems to approach a fixed point and then is repelled by it. Joy described this "I explored values for [μ =] 3.7 to 4 and noticed that the patterns kept repeating in the same general region. The patterns weren't the same but they were very similar. I noticed that at a μ value of 4, the highest x value approached was 1 and the lowest was 0. Both the highest and lowest were repeated." John states that he "Explored around [μ =] 3.82 and discovered some 3-cycles and 6-cycles. Never with exactly identical values though." If they did not possess identical patterns were they cycles? Were they patterns? Cheryl wrote that she "just tried the rest of the #'s between [μ =] 3 & 4. The patterns were the same we saw last night." To which patterns is she referring? Tony adds another meaning to the term pattern when he notes that he "looked at chaos and some of the patterns that nature makes like the [fractal] flowers and the spectrum that was in one of the books." Pattern is the fractal image produced by repeated iteration.

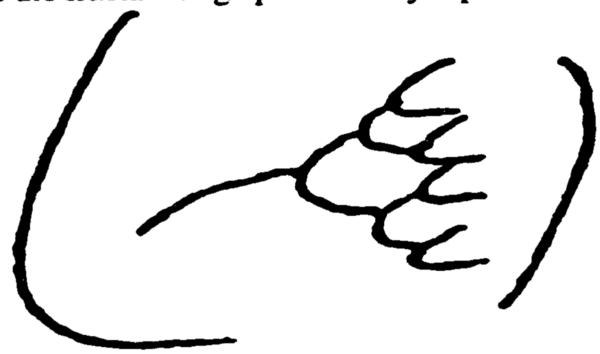


Figure 6-23. Anne's Bifurcation Diagram of the illustrating the bifurcation "pattern".

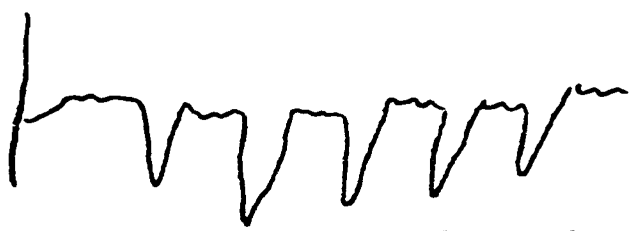


Figure 6-24. Mike's Iterate Graphs of the patterns of chaos when $\mu = 4$.

The simple term pattern is used in different ways. What is the source of this varied usage? Perhaps the language is not particularly effective in describing the nature of the images. In writing this discussion, many images in figures have had to be provided in order to describe the results. Perhaps the understanding of this mathematics does not occur through the language of words and text but rather it occurs through images and visualization and thus the words get confusing because the language is insufficient to describe the complexity of these mathematical concepts. Perhaps this is why the field of mathematics has developed a language which may seem foreign to non-mathematicians. The language is used to express the ideas and experiences of mathematics. It must be developed within a mathematical context or in this case in a computer-based context which is brought forth through mathematics. It is likely that the researcher and the students used terms like pattern without clarifying the meaning of these terms. The term was simply used in the context of discussions of the mathematics which were enacted through the interactions. The term pattern was likely also enacted through the expressions and students' actions. The result was a different and changing use of the term pattern and possibly other terms as well. Certainly on the first day students made different statements about iteration. If the meaning of iteration had been further pursued in the study its meaning would likely have changed and shifted as well. It seems that language used in this way is context codependent and enacted through students' actions and expressions. The language is dependent upon the context of its use and the user in the context. It is part of the mathematical meaning making in which students are engaged. Perhaps the mathematical language developed in a context with students is an issue for further study.

Students were asked to describe the 'grey area' of the bifurcation diagram and its creation. Mike simply wrote that "The grey area is formed by chaos". Cheryl noted that "the grey area was when the pattern started getting chaotic." Several methods of its production were suggested. Cindy noted that "it might be created by it being the various dots of the equation plotted and presented as being chaotic." Anne suggested its "produced by millions of the first part of the pattern [shows bifurcation diagram here, Figure 6-23], one joined to the other." John described that the "results are not in a line anymore but randomly scattered close together, producing grey color/shade." Tony noted that "it was created by the continual separation of the lines" (the period doubling route to chaos). Joy described "The "grey area" of the bifurcation diagram is just dots, not distinctive lines. It is chaos created by many, scattered x values that are approached. It is not part of the tree. After approaching 16 values there are numerous values that are approached in no particular pattern."

Based upon their experiences with the Cobwebber students generated hypotheses of the production of the 'grey area' of the bifurcation diagram. The experiences they had to bring to this question were based upon interactions with Cobwebs and Iterate Graphs of individual functions. Just as students were able to see that the iteration of individual functions produced the 'tree structure' of the bifurcation diagram, they also were able to suggest methods that might account for the 'grey area' in the bifurcation diagram. However their interactions with the chaos range and the 'grey area' were limited. They might have been better able to make sense of this range if they had taken the results of an individual Iterate Graph in the spreadsheet file and compressed the 100 iterates into a horizontal space of an inch. Through such an action individual iterates and patterns within the iteration are obscured. However the areas in which iterates fall are visible. Ranges without iterates are also visible from the results of such an action. This compression method might have helped students experience the 'grey area' as "various dots of the equation plotted" "producing grey color/shade." Students might then have been able to imagine each function, with all its x-values, compressed into a single vertical line. From this line there would be ranges with many points and ranges without points. The individual iterates in some cases would fall on top of each other or very close to one another so these ranges would look grey. For each function a vertical line of points could be generated and these vertical lines could be laid next to each other in order then the 'grey area' could be generated. This is one of the researcher's hypotheses of how the 'grey area' of the bifurcation diagram is generated. The compression approach above has been explored and tried with other students. It seemed to validate such an hypothesis.

Students were asked to write down questions they had about iteration, chaos, period, bifurcation, Cobwebbing, etc. Cheryl asked "Why can you only go up to [μ =] 4?" She had earlier noted that she had "tried [μ =] #5 & [μ =] #9 to see what happened but either they're not programmed in or they're just straight lines so not done". She is correct that they are not possible in the Cobwebber although they could have been done in the spreadsheet. Cheryl was not able to explore these regions because they were not available to her. However she was aware that they existed and that they perhaps could be explored. She has brought forth an understanding that this mathematics was not restricted to the set of μ-values between zero and four. These limits were imposed upon the explorations by the nature of the computer environments provided. However these limits are artificial in regard to the mathematics which Cheryl envisions to move beyond these limits. Her activities have allowed her to extrapolate beyond her explorations and to question what she might find through such activity. Tony asked "Is there continual chaos after μ is larger than 4?"

Joy wrote "Is there anything or pattern beyond chaos?" Where does it go from here seems to be the nature of these questions.

Tony also wondered about moving beyond the limits of the Cobwebber when he wrote, "What would happen if your x-value was a negative?" John asked, "In 3 & 6 cycles do 2 values ever exactly repeat? At what value do cycles stop?" Several other questions asked about why the iterates acted this way. Joy asked, "Does the one pattern [stable region around $\mu = 3.83$] found in chaos also lead to a second chaos?" Cheryl asked, "Why does it go chaos for only a few points?" She added, "Is this what is going to be Math class in the future?"

These students have enacted a view of this mathematics which expands beyond their interactions with the computer environment. They have seen the artificial limits imposed by the software but have realized they are not the limits of this mathematics. These students have not yet begun to determine what the mathematics might be beyond the confines of the software but have realized that they can think about this mathematics and perhaps even generate it.

The computer software in this context became a background to the mathematical activities of these students. In fact students began to invision a mathematical world beyond the confines of the computer software. Indeed even considering the bifurcation diagram was going beyond their direct experience in the software and beyond their direct interactions with the computer environments. These images were not generated by the computer but by the students generalizing from their explorations. Other generalized findings were made using the software as a basis for exploration. The use of the computer in this case supported the mathematics being developed.

Period Doubling and Feigenbaum's Number

Students had explored the five previous days discovering many patterns and making generalizations. Students had seemed impressed by bifurcation diagrams in the books that were similar to the ones they had drawn. They also seemed impressed by the idea that the bifurcation diagram was not only for the Logistics Equation but also could be generated by different functions. Student's exploring was specific to the Logistics Equation but could be generalized through the bifurcation diagram. Students were asked to explore the region of period doubling and determine the values of μ at which the bifurcations occurred. This would allow students to compare the distance between the bifurcations and determine the approximate value of Feigenbaum's number. As a group task students began using the Spreadsheet Logistics Equation with a μ-value between 3 and 4 and determined whether this function was a fixed point, 2-cycle, 4-cycle, 8-cycle or 16-cycle. Each student entered the μ-value and an x-value and determined the period of the cycle. Students already knew that $\mu = 3$ was a fixed point but values above 3 became a 2-cycle. It was determined that $\mu = 3.5$ was a 4-cycle. Based upon this finding the group set out to narrow in on the values at which the 2-cycle became a 4-cycle. Students tried $\mu = 3.4$ and determined this to be a 2-cycle. Students then tried values $3.4 < \mu < 3.5$. Note from Joy's written records (Figure 6-25) that when $\mu > 3$ it is a 2-cycle but at 3.4494 to 3.4495 there is a 4-cycle. Students then explored the change from a 4-cycle to an 8-cycle. This was a time consuming task and involved all students working as an exploration team vocalizing their results to assist others in exploring closer and closer to the values. One difficulty was that the tools were rather crude for such particular exploration. The screen resolution could not possibly make visible the explorations to 4 decimal places. Despite this restriction students explored and closely approximated the values for the first two bifurcations.

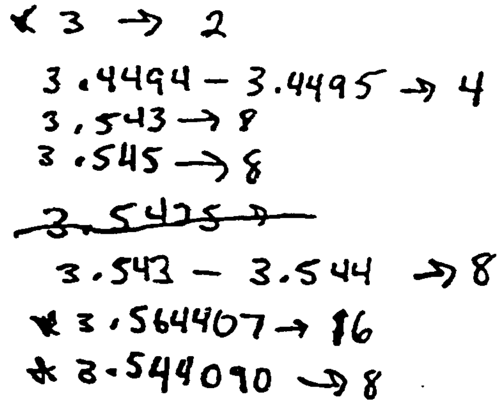


Figure 6-25. Joy's records of the finding of μ-values for the bifurcations.

Students were provided more precise values for the first two bifurcations and the later bifurcations from Lauwerier (1991, p. 119) as follows: 3.0, 3.449499, 3.544090, and 3.564407. Tony described how to calculate Feigenbaum's Number: "To get Feigenbaum's number you take the distance where the graph splits in two and where it splits in four and divide it by the distance between the distance the graph splits in four by the distance that it splits in eight. Feigenbaum's number is 4.69." Joy made records of these calculations (Figure 6-26). Using the values provided above students determined the approximate values were 4.7520 and 4.6557. However they were told that when these values were calculated for each bifurcation in each of the ranges they approached the value 4.69. Joy provided a summary of the calculation of Feigenbaum's number (Figure 6-27).

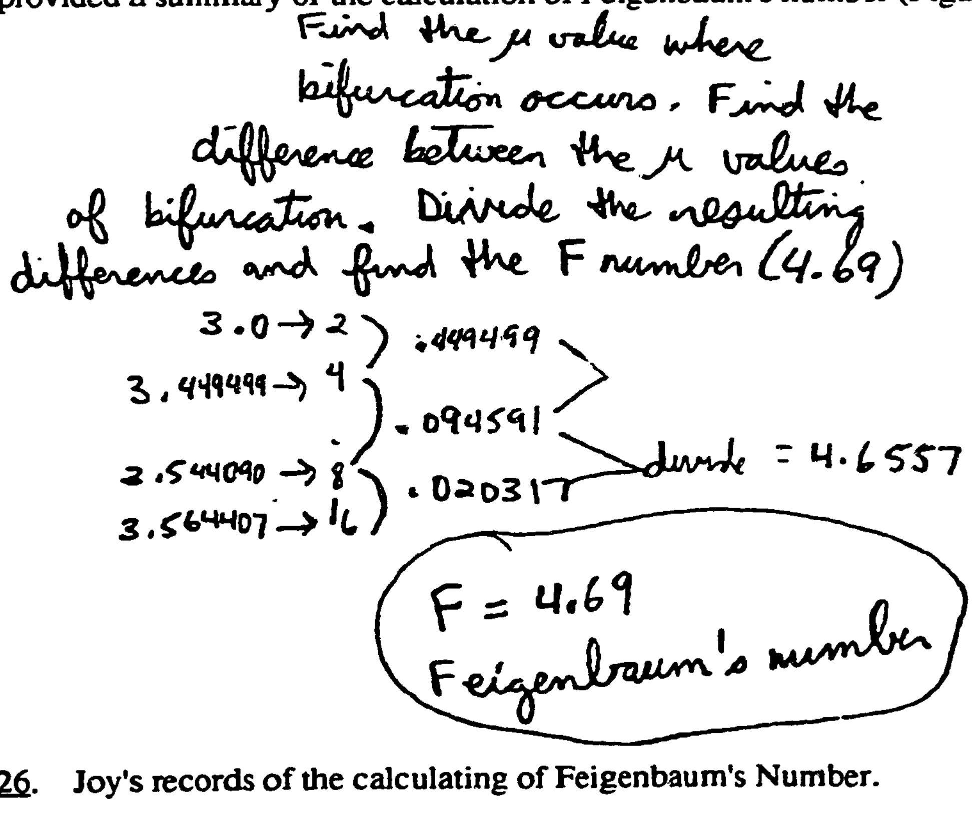


Figure 6-26. Joy's records of the calculating of Feigenbaum's Number.

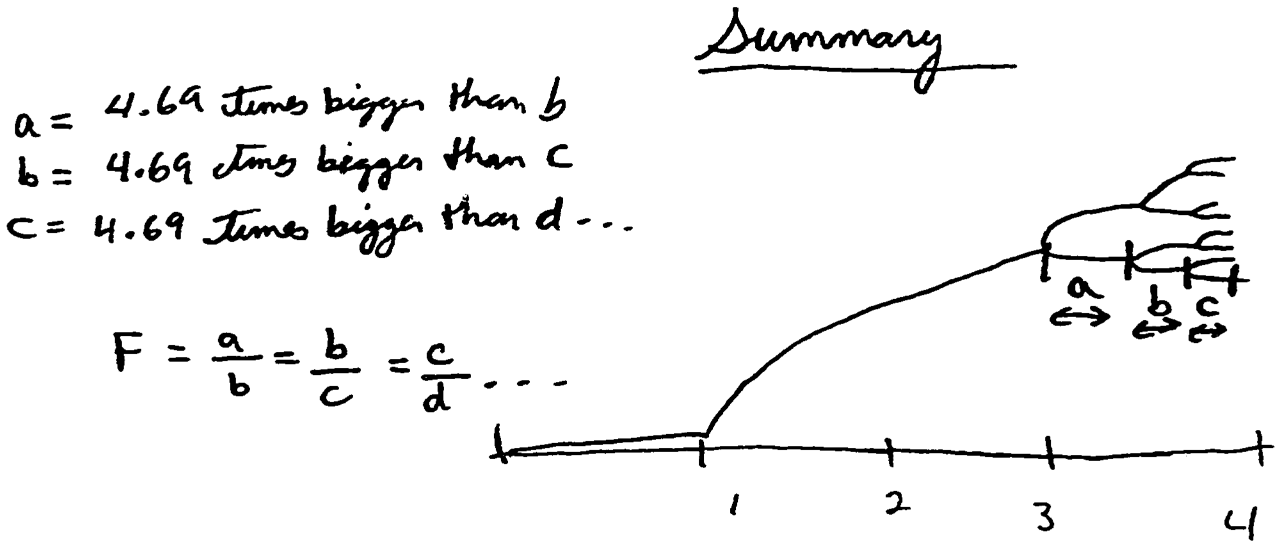


Figure 6-27. Joy's records summarizing the meaning of calculating Feigenbaum's Number.

The Curriculum, It's Development and Rationale

The events through which students explored formed a curriculum in an objective sense of the word. The selection of activities, the rational for this selection, and students' actions and developing understanding through this curriculum provided further insight into the study.

Iteration is an important concept for an understanding of non-linear dynamics and chaos theory. To facilitate its understanding an activity was designed with a calculator. Students could enter a value and use a function key available on their calculators. What student at some time or another had not played this game with a calculator? Now this game had the name iteration and it was considered a mathematical activity worthy of interest and study. From these activities we could make generalized mathematical findings: 1) all values greater than one approached infinity, 2) all values between zero and one approached zero, and 3) zero and one were constant.

This same game could be played with other functions. Another function with which students were likely familiar was the square root function. Similar generalized findings could be made for this function. It could have been done with a calculator but the introduction of a graphing technique was desirable as it was to be used later. So a computer program was developed which would generate Iterate Graphs of the function and generalize the findings. Students could have control of the entry of values and were instructed to notice patterns of behavior for the square root function as they had for the squaring function.

Students could be active in thinking about the mathematics and noticing patterns from the graphics displays as they watched what was being drawn. They were not entangled in the process of drawing but rather were able to explore the mathematical significance of the drawing once it was understood. Students needed to learn to interpret from these graphics which required some time to develop. However once understood students could explore and begin to determine patterns.

Other functions are historically important to the study of chaos theory and non-linear dynamics. These functions are formed when a value $0 < \mu < 4$ is selected for the Logistics Equation $y = \mu x(1 - x)$. This simple function exhibits complex behavior which is interesting to study. By what means should this function be studied? Two graphing techniques are useful to its exploration: Cobwebs and Iterate Graphs.

What is salient from the Cobweb? The x-value which determines the beginning point of the Cobweb. The descent of a Cobweb to the fixed point of zero or the approach of a fixed point at the intersection of the y = x line and the parabola. What is salient from the Iterate Graph? The pattern of iteration is visible for 2-, 4-, and 8-cycles which appear as 2, 4, and 8 rows of points respectively on the Iterate Graph. What is the relationship between the two graphs? The height of the points of the Iterate Graph are comparable to the height of the horizontal lines of the Cobweb. What student explorations and actions might have led them to develop these understandings? These are summarized in Table 6-1.

Table 6-1. Student actions and that which could be occasioned by the environment through these actions.

Action (exploration)	What could be occasioned by the environment?
- repeatedly square a number using the last square as the next to be squared	- iteration - patterns emerge with all numbers greater than 1 go to infinity, all numbers between 0 and 1 go to zero, 1 remains 1 and zero remains zero.
- repeatedly iterate the square root function	- iteration - patterns emerge: all numbers greater than 1 approach 1 from above, all numbers between 0 and 1 approach 1 from below, 1 remains 1 and zero is undefined.
- choose a μ-value for the Cobwebber	- a function is chosen from the Logistics Equation $y = \mu x(1 - x)$
- choose an x-value for the Cobwebber	- a seed value for the iteration is selected
- input a μ-value for the Cobwebber	- a particular parabola is drawn with height varied by the μ-value
- input an x-value for the Cobwebber	- the seed value determines the beginning height of the Iterate Graph and the position along the horizontal axis where the Cobweb begins to be drawn
- watching the Cobweb or Iterate Graph alone	- notice the approach of a fixed point, entering into a cycle, or chaos.
- watching the Cobweb or Iterate Graph together	- notice that the height of the iterates in the Iterate Graph are the same as the height of the horizontal lines in the Cobwebber.
- enter x-values increasing the value for each entry but keeping μ constant $0 < m \leq 3$	- the initial position of the Cobweb or height of the Iterate Graph changes but the fixed point approached is the same.
- enter μ-values increasing the value for each entry $0 < m \leq 3$	- height of the parabola increases - value approached remains zero for $0 < m \leq 1$ - value approached increases for $1 < m \leq 3$
- solve the system of equations $y = x$, $y = \mu x(1 - x)$	- the fixed point approached can be predicted by $x = 1 - \frac{1}{\mu}$
- enter μ-values increasing the value for each entry $3 < m < 3.6$	- approaching 2 values then 4, 8, and 16

- enter μ-values increasing the value for each entry 3.6 ≤ m < 4	- a repeating pattern is not evident (chaos) - the range of values touched by the chaos increases filling the range from 0 to 1 by the time it reaches μ = 4 except for a band which is not touched in the midst of the range.
- enter values of μ continually closer to the change from a cycle to its double	- the value of the change points between 2-, 4-, 8- and 16-cycles are approximated - fixed point to 2-cycle 3.0 - 2-cycle to 4-cycle 3.449499 - 4-cycle to 8-cycle 3.544090 - 8-cycle to 16-cycle 3.564407
- calculate the distance between the change points	- the value of the distances are .449499, .094591, .020313.
- calculate the ratio of consecutive distances thus approximating Feigenbaum's number	- the values are 4.75 and 4.66 which approximate Feigenbaum's number which is 4.69

Mathematical Findings

Students have explored and experienced mathematics as a dynamic and generative entity within computerized environments. If one were to assess this study by the amount of mathematical language students used, the number of calculations they determined, the number of graphs they themselves drew, or the number of mathematical definitions they uttered or wrote then it might be said that these students had not achieved very much. But of course students explored, made mathematical statements, and had mathematical experiences. They enacted mathematical significance from their computer experiences. The question is what is the nature of these explorations, what is the meaning of these statements and how can these experiences be interpreted within the computer environments?

Students interacted with a calculator and the iterative computer program through which they enacted varying ideas of <u>iteration</u> including 1) ideas specific to the function (x^2) studied, 2) a more generalized statement of iteration as repetition, and 3) a more complete general statement including repetition and the idea that the output of one step in the iteration is the input to the next. They saw the software iteration of the square root function generate a graph called an <u>Iterate Graph</u> which is a graph of the numeric values produced when a function is iterated. This graph has the number of iterations as the x-axis and the value of the iterate as the y-axis. It involves the graphing of ordered pairs which is familiar to most students. Following the explorations with these first two functions (x^2, $\sqrt{x}$) in these two computer-based environments students began to explore other functions.

The <u>Logistics Equation</u> generates many <u>functions</u> based upon the μ-value entered. These functions when graphed produce a <u>parabola</u> which opens downward. Students stated that the μ-value determined the width or height of the graph. That the μ-value determines the width seems a misconception since the parabola drawn always had x-intercepts of 0 and 1. If however one considers that when an object is less tall it can appear more wide then this statement would be justified. From observations of the Cobweb the μ-value determined the maximum value of the parabola. Students referred to this as the height of the graph. Is this problematic for students who must learn in the standard curriculum that the width of the graph is determined by the coefficient of a squared function? Or has this procedure provided students with experience with parabolas and the effect of coefficients which will enable them to better understand the algebraic procedures used to determine the value of the maximum of a parabola? The effect of this small study upon students performance in their mathematics class was not determined, if indeed their was an effect. Further study into the effect of experiences with mathematical exploration environments such as these on students ideas of mathematics may be of benefit to mathematics education.

A Cobweb is a graphical procedure for determining iterates of a function and the pattern of iteration which emerges. It is directly related to the numeric calculations of iterations possible using the equation for the Logistics Equation, the μ-value and the x-value. The Cobweb begins with the drawing of a vertical line from the x-axis at the value of the x-input to the parabola. The Iterate Graph places a point at this same height. A horizontal line is then drawn on the Cobweb to the $y = x$ line. This converts the x-value to a y-value thus the output of the first iteration becomes the input to the next iteration. Another vertical line is then drawn to the parabola. The Iterate Graph adds another point at this height. Another horizontal line is drawn to the $y = x$ line converting the iterate. This process continues 1) until a fixed point is reached, 2) until a cycle is entered, or 3) eternally for a chaotic function.

The Cobweb is related to the Iterate Graph in that the height of the horizontal lines of the Cobweb are the same as the height of the iterates on the Iterate Graph. Fixed points are evident from the Cobweb when the lines of the Cobweb seem to step toward or converge upon a particular value or seem to form boxes which continually grow smaller encircling a value. On the Iterate Graph a fixed point is evident when points level off to a particular value and remain constant at that value. Cycles are evident from both the Cobweb and Iterate Graphs when no more lines appear to be drawn on the Cobweb or when the Iterate Graph forms a number of stable rows of points. Chaos is evident when a repeating pattern cannot be determined from either the Cobweb or the Iterate Graph or a specific point is not reached. In other words, the Cobwebbing continues to add new lines and the Iterate Graph continues to add points which appear to be randomly placed.

Students noted the approach of a fixed point varied. In some cases it approached slowly and in others quickly. If the steps toward the fixed point were of large size then the approach was evident in only a few iterations. But if the steps toward the fixed point were small in size (as is the case when beginning very close to the fixed point) then it took many iterations to notice these small changes. Therefore when close to the fixed point it took more iterations to notice the approach than when far from the fixed point. In addition some fixed points were approached from above in a stair-like manner. Others were approached from below in a similar manner. Still others were approached from both sides with the Cobweb appearing to encircle the point approached. From the Iterate Graph the approach of a fixed point appeared to be a set of points falling and then leveling off at one value, rising and leveling off, or a series of high and low points continually getting closer together until they converged upon one value midway between them.

Ranges of μ-values had similar patterns of behavior. For $0 < \mu < 1$ all approached a fixed point of zero. For $1 < \mu \leq 3$ the fixed point was predicted by $x = \frac{\mu - 1}{\mu}$ or $x = 1 - \frac{1}{\mu}$. For $\mu = 3.1$, 3.2, 3.3 and 3.4 a 2-cycle was evident. For $\mu = 3.5$ a 4-cycle was evident, $\mu = 3.57$ was an 8-cycle and $\mu = 3.58$ was a 16 cycle. For $3.6 < \mu \leq 4$ chaos was evident.

A bifurcation diagram generalized the results with μ-values on the x-axis and value(s) approached on the y-axis. Students made a variety of graphs with varying degrees of generalization. 1) In one case a student drew an Iterate Graph and the values it approached. This was not a generalization of results. 2) In another case a diagram was drawn of a typical Cobweb in each range. This was not a bifurcation diagram. 3) Several others plotted points (ordered pairs) of the μ-value and the fixed point approached. The points were not connected. 4) In another case the ordered pairs were connected showing the general pattern of fixed points being zero initially, then rising, followed by the bifurcations into 2, 4 and 8 forks. 5) Chaos was noted on only one of the diagrams. None of these students were able to draw the chaos range. Limited exposure to this range may have been a limiting factor.

The chaos range was explored by students by entering various values of μ and x. Students found that in some cases there appeared to be a pattern (approach of a fixed point) but then the iterates moved away (were repelled) by the intersection of the $y = x$ line and the parabola. Students noted the lack of repetition in this range as denoting chaos. However this movement toward the intersection and then away occurred repeatedly although never with exactly the same values. This is the notion of self-similarity. This notion also was evident when students discussed the area of stability within the chaos range. This area was created by another bifurcation pattern just like the first one but on a smaller scale.

Fractal images were described as "colored patterns" which were of interest because of the patterns they formed. Those students who mentioned the fractals noted their use to create images representing real life.

Students hypothesized about the generation of the 'grey area' of the bifurcation diagram. They knew it was generated by chaos. They hypothesized that it was generated by many repeated bifurcation patterns, by points plotted from the chaotic equation, by many "scattered x values that are approached"

Feigenbaum's Number is a ratio present in the period doubling of the bifurcation diagram. The μ-value at which the transition from one cycle to another occurs (from fixed point to 2-, 4-, 8- and 16-cycles) are called bifurcation points. The transitions between the points are called bifurcations. The increase in the μ-value between each period-doubling is in a ratio to the next increase. The ratios of consecutive increases approach a limit known as Feigenbaum's number denoted as $F = 4.6692016$ (Lauwerier, 1991, p. 120)

How Did Students Explore?

At times students did not systematically explore but chose values for "no reason. Just picked 2 numbers." as Cheryl described. Some students explored systematically varying one input value at a time and varying this value by ever increasing it at more or less constant increments. Mike, for example, chose a particular μ-value and then systematically varied the x-value choosing low, medium and high values. He did this for $\mu = 1, 2,$ and 3. John explored $\mu = 1.5, 0.5, 0.75, 2.0,$ and 2.3 he noted that the iterates "level[ed] at" a particular value. It seems that once John began to expect a pattern he then very systematically explored $\mu = 1.6, 1.7, 1.8, 1.9, 1.1, 1.2, 1.3,$ and 1.4 noting the fixed point. John was then able to make a generalized statement of his findings stating that when the μ-value was between 1 and 2 the value approached was between 1.5 and 1.9. Later the term fixed point was used to describe these findings.

Other examples of students systematic explorations are evident. Mike explored the constant value $\mu = 4$ varying $x = .5, .9, .2$ and .1 drawing each iteration. Beginning the Cobweb at the x-input value is evident. If he had systematically tried this with a function that cycled he may have noted that despite his input the approach was always to the same values. What he seems to have explored was the process of Cobwebbing noting the way each line related to the last. Tony also explored systematically varying $\mu = 3.5, 3.0, 2.5, 2.0,$ and 1.5 with x constant at 0.05. He noted changes in the height of the parabola and the way the Cobweb is drawn. Cindy had made the observation that $0 < \mu < 1$ approached a fixed value of zero. She explored $\mu = 1.1$ with $x = 0.5$ and noted that it did not approach 0 but .09. To confirm this she tried $\mu = 1.1$ with $x = 0.1$ which created a level row of iterates before stepping down one pixel to .09. She then tried $\mu = 1.1$, $x = .09$ which made a level row of iterates. With this approach Cindy confirmed that the fixed point was indeed .09.

Not all explorations allowed students to make generalized statements or to see their explorations as meaningful. The Cobwebber is complicated with two images being simultaneously generated each making some patterns salient while a comparison of the two taken together producing further complexity. Tony expressed frustration with the Cobwebber and requested an explanation of the Cobwebber's functioning. A discussion with all students of the Cobweb method seemed to alleviate Tony's frustration.

The introduction of a systematic exploration of the $0 < \mu < 3$ range where fixed points were evident was in response to John's generalized findings. He had made these statements based upon systematic exploration. Perhaps this same approach would be helpful for other students. The systematic group approach explored the values of fixed points for $1 < \mu \leq 3$ by each student trying values and reporting to the group. The values tried were incremented by .1 generating a chart of fixed point values. Students were able to predict values in this chart based upon the previous values within it. Clearly a pattern was being generated. It was noted that this value was the intersection of the $y = x$ line and the parabola. Students solved this system of equations using $y = x$ and $y = \mu x(1 - x)$ for $x = \frac{\mu - 1}{\mu}$ or $x = 1 - \frac{1}{\mu}$. Students were able to predict the fixed points for $2 \leq \mu \leq 3$ using this formula. However, students also used the Cobwebber to confirm these predictions. They would calculate a fixed point value by substituting into the formula but they would also determine the fixed point by entering the values into the Cobwebber. Four presentations of the iteration to a fixed point were used: 1) numeric prediction using the pattern of previous fixed points, 2) numeric calculation based upon the algebraic formula, 3) graphic images using the Cobweb, and 4) graphic images using the Iterate Graph.

The way that the fixed point was approached varied with a direct approach when $\mu < 2$ and the Cobweb circling around when $\mu > 2$. The Iterate Graph had two lines of points one descending and one ascending when $2 < \mu \leq 3$ converging on a value midway between the two lines of points. But when $\mu > 3$ the two lines of points did not converge to a single value but remained at two lines of points. To determine if the behavior of a function was going to converge to a fixed point at the intersection of the $y = x$ line and parabola or enter a 2-cycle the Cobwebber could be run again. This time the x-input value could be chosen so that it was very close to the intersection. If the iterates were to converge then they would continue to get closer together and closer to the intersection. However if the behavior of the function was a 2-cycle then the iterates would diverge in two lines of points leveling off at the 2-cycle.

A similar procedure was used to determine if a function was a 2-cycle or a 4-cycle. Students noted the four values at which the function seemed to have leveled off. They then entered an x-input between the four values tried and noted whether the 4 values seemed to converge to two values or diverge to 4 values more spread apart. In this way students were able to determine a 4-cycle. However as students continued this type of exploration it meant waiting for the Cobwebber to draw many iterations. It was tedious to wait so long for results. Switching to the spreadsheet allowed for 100 iterates to be drawn almost instantly. This made it possible to test behavior by entering an x-value and then watching what happened. Students could then take the value of the one-hundredth iterate and enter it as the x-input. They would then see the next 100 iterates. If further iterates were required they could repeat the process. In this way students could see patterns much more quickly.

Human-Computer Interactions

Students were actively involved in computer assisted explorations throughout the study. They entered values and watched as the software responded in predefined ways to their input. Student input initiated the software activity. Students attempted to understand the software, its graphic images and numeric values. They began to understand these presentations as they explored and they began to direct their explorations based upon their growing understanding. For example, John noted that iterates "level[ed] at" a particular value. Based upon this realization he began to explore systematically noting the fixed points and making the generalized statement of range of μ-input and range of fixed point.

In this case the computer-context shaped the students' learning by generating graphics in predefined ways and representing the iterations for the student's exploration. However the student shaped the context by his input. Had he input different values he would not have seen the graphs he saw and may not have made the conclusions he made. So the student shaped the context by his actions.

If students explored systematically (as did John) then they may have noticed the approach of a fixed point. Students' systematic approaches brought forth a particular aspect of the medium. If students confined themselves to the chaos range (as did Mike) they may have noticed no patterns of behavior other than the effect of the x-input on the initial location of the Cobweb. The way the Cobweb was constructed may have been the focus rather than the behavior of the function. The graphics enabled Mike as the learner to explore Cobweb construction rather than the behavior of the function. John noted the approach of a fixed point from his experiences with the software. The software, through its predefined action, thus brought forth a particular kind of student action dependent upon students' experiences and the choices they made within the interactions. The students brought forth a particular view determined by their experiences and their interactions with the medium. The medium occasioned the student's view by its structure and the students each brought forth a view of the medium through their interactions. The medium and the student can be said to coemerge in this context. Each is codetermined by the other through their interactions.

As another example consider when the μ-value as input is very small and the parabola drawn is very flat either appearing to be a straight line or not appearing at all. The Cobweb cannot be drawn for such an input and thus students might get the impression that it is not possible to iterate when μ is small. This software limitation affected students' experiences but it was the students' choices of input which brought this limitation into the foreground. Students wondered what happened beyond the range $0 < \mu \leq 4$ and if other types of behavior occurred beyond chaos. The Cobwebber was not programmed to provide exploration opportunities beyond this range because such Cobwebs are not bounded within unit x and y axes and thus are difficult to demonstrate using graphing techniques. However students were left wondering about the behavior beyond this range. The software and student codetermine each other in this example.

Students were asked to hypothesize about the origin of the 'grey area' in the bifurcation diagram. Students used their knowledge of chaos and its behavior to make suggestions of how and why this area occurred. Chaos was generally accepted as the cause of this area. However students suggested it may have been caused by many bifurcations repeated on a small scale, by all the iterates of the equation being plotted, the iterates approaching many values, or a random scattering of iterates. Students generated these hypotheses based upon their experiences with the software medium. The interaction with the software brought forth the students' understandings.

Perhaps students' watching a computer is not an inactive phase if it is directed watching. If students can interpret what they see they can make sense and understand. In addition, students provision of input places them in a position to interact with the medium and provides them the opportunity to 'bring forth' this medium. Perhaps the point is not that students interact with a medium but that they enact meaning within the medium. Students enacted mathematical experiences within the computer mediums. The structure of this medium and its constraints become evident and enacted a sphere of behavioral possibilities through which students acted. The programs' structures determined students'action in a particular direction. Students input to the programs selected from the possibilities and initiated the actions of the programs. Students could then bring forth an understanding of what it was about their inputs that produced these results. The interplay between the student and the computer software is significant. It is not merely interaction but enaction. The context shaped the learner and the learner shaped the context.

Chapter 7

Human-Computer Enaction: Implications for Education and Research

What is Human-computer interaction (HCI)? As the name implies it is the interaction between a human and a computer. But what is meant by interaction? The importance of interaction within the realm of HCI seems almost unquestioned. The discussion in this chapter carries the notion from the sphere of 'interaction' to that of embodied action or "enaction" (Varela, Thompson, & Rosch, 1991) shifting the view of HCI from a representationist stance to an enactive one.

> The HCI label appears to be self-explanatory – that is, anything to do with people interacting with computers – yet it has been interpreted more narrowly as simply the study of user interfaces, which seems an extremely limited view. (Bannon & Bødker, 1991, p. 231)

We can broaden this view of HCI by considering the notion of interface from an enactivist stance. The interface is often considered the boundary between the user and the computer software. It may be considered to be the place where the two parties communicate or it may be considered a barrier that stands between the two, separating them. But from an enactivist stance it is the place where the user and software coemerge. The software becomes evident to the user through the interface. The interface is the means of occasioning the users actions. The user takes action upon/within the software through the interface. The interface is thus the place where the software and the user codetermine each other. From the enactivist viewpoint the medium and the user coemerge.

As an example of the coemergence of user and medium consider Susan's creation of a living essay in Authorware Professional (AP) (described in detail in Chapter 4). She wanted to create an image of the main character in her story (Cheeser) holding a piece of cheese. She drew an image of a triangle of cheese in Superpaint and then attempted to transport it to AP. However in the action of copying and pasting the mouse and cheese, the cheese always remained uncopied despite repeated attempts. Susan finally resorted to drawing a round pizza in AP and changed the text of her story to be about pizza to accommodate her difficulty with the cheese image. The medium in which Susan worked (a computer environment containing AP and Superpaint) constrained Susan's story and thus Susan's productive efforts coemerged through/with the medium and Susan.

The enactivist viewpoint is a stance toward cognition which addresses the question "what is it for a human to know a world?" Many scholars have pondered this question and many schools of thought have emerged which attempt to address this question. Descartes determined that the mind and body were distinct. A consequence of Cartesian dualism is the representationist view for which the world is pregiven and represented in a mental model by the mind of the subject in this world. Cartesian dualism results in a realist stance in which the world is an external foundation. Another point of view also based upon representation is the idealist stance which sees the world as dependent upon representations by a subject since all aspects of the world can only be accessed by the senses of a subject. The world then is an internal ground. Both approaches, realist and idealist, look for a groundedness externally or internally.

The conceptions of studies in this dissertation were originally enframed within the representationist stance. The world (composed partially of computer programs) was pregiven and the students of the mediums were to determine the features and functions by constructing mental models. The predefined features of the mediums were to be represented by the student and acted upon based upon these representations.

The AP context, discussed in Chapter 2, was viewed as a preconstructed context. Students were expected to read about and come to understand the medium through the instructional simulations (IS). The expectations were that features of AP must be understood in order to function within this environment. This understanding would be based upon how well the organization of the students' mental models matched the world. These features to be understood included AP's use of icons, flowchart, pull-down menus, and toolboxes to provide a user interface. It was expected that users of this software must become familiar with the features of this interface in order to function within this environment. From the representationist stance analysis would interpret students' views to determine if they matched the predefined features of the medium. Students views would be assessed for this purpose.

The chaos theory context was described (Chapter 5) as a pregiven world. This world needed to be explained and its features needed to be understood in order to then function within it. The features of this world described were: 1) iteration, 2) the Logistics Equation, 3) Iterate Graph, and 4) Cobweb. The medium was defined through the computer software provided: the Cobwebber and the Spreadsheet Logistics Equation and later (in Chapter 6) through the Iteration Program. Students generated concepts which were specific to the environment through which they explored as well as more generalized concepts. An example of the concept of iteration was provided in Chapter 6. Students generated qualitatively different conceptions of iteration. Some students understood iteration as specifically related to the square root function. Others saw it as a more generalized process which included the idea of repetition. Still others had a more complete generalized concept which included the idea of repetition but also included the idea that output of one stage of iteration is the input to the next stage. This view of the context was representationist in that the students were to explore but were to generate a set of concepts which could be predefined and which existed in the independent world of mathematics.

In both studies an initial representationist stance provided a framework for viewing the computer programs as external entities to be understood and represented by mental models. In both cases the computer was viewed as an intermediary between the user and the task, thus separating them. It was the student's task to overcome this obstacle posed by the software and hardware or at the very least the student needed to interpret and model what the hardware and software presented. The computer was also viewed as the focus of student attention and the object of study. Students were to learn about the software and hardware and make this initially a study in itself. Students were to learn to enter values into the mathematics programs correctly in order for it to graph appropriately. They were to learn to select appropriate options in AP to achieve results. Students were to have a technical understanding of this external phenomenon which stood between the student and task.

This idea of separation of mind and body, student and task is a very representationist one but prevalent in studies of computer-based learning. It relies on the notion of an external and an internal. This idea is at the core of numerous research programs about computer assisted instruction and cognition. But what if we were to see these computer programs differently? What if we were to envision a way without this separation?

Understanding the medium from an enactive stance

The way proposed by Varela, Thompson, and Rosch (1991, p. 149) is termed embodied action or enaction. "Knowledge depends on being in a world that is inseparable from our bodies, our language, and our social history–in short, from our *embodiment*.. ... We must see the organism and environment as bound together in reciprocal specification and selection" (p. 174). The organism and world mutually specify each other, each enacting the other. They are said to codetermine each other. The system's (organism's) behavior is determined by its structure and history of interaction in the environment. A path of action is laid down by the mutual specification of an organism acting on and in the world.

The AP Context. The purpose of Chapter 2 was to explore student's concepts through interviews providing examples of students' discussions in order to characterize concepts formed while using AP. Students developed concepts specific to the AP interface as well as more generalized concepts through their activities and the IS. One example of concepts discussed in Chapter 2 was that of icon. The concept of icon was described by students in three ways: 1) it represented a command, 2) it was used to write a part of a program, or 3) it was used to execute a function. Discussion of display and erase icons showed that students had various conceptions of these icons but these conceptions were specific to the AP context. An extension of this discussion lead to the concept of object. Other concepts discussed included the flow of activity through the flowchart via the flowline and the mediating effects of pull-down menus and toolboxes as well as the concept of program. The analysis of the concepts from the initial AP study (Chapter 2) was conducted to characterize students' concepts. The emphasis was upon concepts students generated through their interactions with AP and the IS. The task was to understand students' views of the AP environment. Such an approach is not representationist but enactive.

An enactive approach is described by Varela, Thompson, and Rosch (1991, p. 174) in these terms: "the organism both initiates and is shaped by the environment." A quote from Merleau-Ponty (1963) provided by Varela, Thompson, and Rosch (1991, p. 174) describes this reciprocal relationship for movements of an organism:

> Since all the movements of the organism are always conditioned by external influences, one can, if one wishes, readily treat behavior as an effect of the milieu. But in the same way, since all the stimulations which the organism receives have in turn been possible only by its preceding movements which have culminated in exposing the receptor organs to external influences, one could say that *behavior is the first cause of all the stimulations*. Thus the form of the excitant is *created by* the organism itself, by its proper manner of offering itself to actions from the outside.

How can we interpret the work of students in the AP environment from an enactivist perspective? Students brought forth the AP context describing the icons, flowchart and toolboxes (in Chapter 2) as they had experienced them. The interpretation put forth sought to understand students' perceptions of this context and the world students brought forth which contained the AP software. For example students were asked about the erase icon in AP. They described its function as an eraser. Students perceptions of what the erase icon erased were variable. Some students thought it erased the first screen displayed, others thought it erased the last screen displayed, while others thought it erased the screen just prior to the placement of the erase icon. A final student indicated that the erase icon erased the contents of the icon that had been selected to erase. This analysis (and the remainder of the analysis in Chapter 2) was limited because it showed only pieces of students' views (it did not connect these views with those expressed about other icons) and did not describe the students' activities in attaining these views nor students' perceptions of these views.

The later studies attempted to rectify these shortcomings, and better assess the enactive features of the AP environment, by tracing students' actions and discussing students' views. In the later studies with high-school students the same instructional simulations were used but activities had been added. These activities were developed based upon interactions between the researcher and adult students interviewed in the study for Chapter 2. These activities provided students an opportunity to experience AP through open ended creations of products within the confines of the assignments but with content and structure based upon students' ideas. Students' ideas were given a fuller place in the context of the completion of these assignments. Students produced many different kinds of 'click/touch' type interactions using many methods, some similar to those they had been shown. For example, students watched an IS about how to use an interaction icon and attached icons to create a multiple choice question. Students activities produced an interaction which allowed users to click on a predefined region of the screen and the users were provided feedback to their choice. No specification of the format or content of the interaction was provided. Anne (discussed in Chapter 3) used the multiple choice format to provide a series of questions embedded inside questions. Logan (Chapter 4) created the question **Would you like to see my show?** responding to two answers yes and no with the respective feedback 1) **On With The Show !!!!!!!!!!** and 2) **Oh, well you are going to see my show whether you like it or not.** Each student in this context specified the question text, answers to be tested and feedback to be given to the user. Each student set parameters suitable for the type of interaction each had created.

The later approach to students' learning about AP was not merely a drill and practice approach because students did not learn exact procedures to follow and steps to perform. Their actions were not intended to imitate an external model (as is often the case in computer-studies activities). The structure of the files produced were to be determined by students. These activities provided students the opportunity to explore the medium and generate approaches assuming students would vary these procedures and structure the products as they saw fit. These activities were enactive in that their design acknowledged that students must generate products according to their perceptions of the assignment and based upon their experiences with AP and their personal history.

Through the actions of creating a product in AP we are transformed as well as our relations to everything else in the world because creating in AP, like any experience, is woven into our experience of living. By writing a story about Cheeser the Mouse Susan created a story, a piece of her world, which had not existed before. She existed anew in the experience of writing this story. She was not the same as she was before she wrote this story for now she had written in a multimedia format. She may have begun to see herself as an author, director, programmer, designer, or teacher. This new view of self was enacted in activities in relation to the computer and through the products students generated.

It may be suggested that the products of students' actions remain static. In the sense of their structure and the words and graphics within them this is true. However the act of watching this file is different each time because the watcher is different and the context is different. We can never watch the file the same way twice. This is true whether we actually alter the contents of the file or not. Our meanings for our activities and their products are ever changing even if the content of such products is unchanged. Likewise when we produce a product in AP the process is never the same twice. The actions of placing icons and filling them with text and graphics or specifying their options may be the same but the meanings of these actions are altered. In productions in AP we are enacted and we enact.

> The overall concern of an enactive approach to perception is not to determine how some perceiver-independent world is to be recovered; it is, rather, to determine the common principles or lawful linkages between sensory motor systems that explain how action can be perceptually guided in a perceiver-dependent world. (Varela, Thompson, & Rosch, 1991, p. 173)

By recording students' actions and interactions speculation about some of the connections between students and their world may be made in terms of how actions can be perceptually guided in the student-dependent world. However by watching the actions of students on the tape we are reconstructing a world that has already ceased to be because the world now exists in a new moment. The tapes do not have the same meaning that they did when they were recorded because the parties watching them are not the same as they were when they were recorded; actions in the world have changed the parties. The interactions with the world are not the same because the parties are different and so the world brought forth by each party is different. The world is ever-changing. So interactions with this world are not the same. The parties are changed by these interactions. The parties and the world are codetermined.

Consider the students in Chapter 3 who used the AP software to bring forth a world they created through their living essays. The nature of that world was determined by seemed to have enacted coherence of their actions on the basis of their embodied history of explorations and activities.

As suggested by Ihde (1979) the computer both enframes and enables students' actions. The computer software limited pathways open to students (for example the interaction types possible) and enhanced the possibilities of other pathways (by making loop structures possible, for example). However it was the learner's structure and history which codetermined the actual pathways possible and chosen. AP has icons which can be combined in a vast number of varieties. However, students frequently used display icons followed by wait and erase icons. These structures were made possible by AP but ultimately chosen by students by their activity in AP.

The learner created a path occasioned (in part) by the teacher's specifications. Experience with AP also codetermined the path selected. The teacher specified particular structures in AP which students were to incorporate into AP. These included multiple choice and drag object interactions, graphics, story and sound. However students did not incorporate simple multiple choice questions with 4 answers, namely a, b, c, and d, but created branching and interactive sequences through the multiple choice structure. Where they incorporated these, and the structure of these interactions were codetermined by students' history of interactions with AP.

The creations of products by these students was codetermined by the computer software, the student, the teacher, the classroom encounters with other students, and other factors in the lived history of the student. The student, based upon this history, codetermined the path. The computer was both a limiting and enabling factor, as well as a space to enact the path chosen. In so doing it was the vehicle to relate to the living essay in and of the world. It was also the environment which formed the background, active through student activities.

The Chaos Theory Context. Through their actions students enacted a world of mathematics described by students in their record-making during the study. Students did not necessarily develop views consistent with 'the independent world' of mathematics (from a representationist viewpoint). For example the bifurcation diagrams students produced should have been replicas of one another all representing the same information in the same way and matching the bifurcation diagrams apparent in the various books about chaos theory. Students did not create carbon copies of these diagrams. They experienced mathematics through their explorations making observations, generating diagrams, and devising generalized statements. They devised procedures to facilitate finding cycles. When confronted with the realization that the functions they explored through this computer medium approached the intersection of the parabola and the $y = x$ line they solved a system of equations and factored the resulting expression. Their history of algebraic explorations and experiences enabled them to perform this task and enact these procedures. They brought forth a world of mathematics through their experiences of interaction within the computerized environments which was dependent upon their history.

The world each student described was different just as their descriptions were different. Given that each had a different mathematical history and a different set of experiences within this study the fact that the features of the world were described differently is not surprising. But the important thing to note is that students did not see this world in a static unchanging way. They interacted with the computer programs provided and each day the patterns they saw were different than the last day. For example, early exposure to a chaotic function led students to draw the Cobweb and try to figure out the pattern of Cobwebbing. Later exposure to a chaotic function led students to wonder about the patterns within this chaos noting the fact that a pattern was almost reached but then a change occurred. Early in the study students saw the graphics produced and wondered what it was all about. Later they 'made sense' of these graphics and focused upon the patterns produced by the graphics. The same graphics were seen but the meaning of the graphics had changed. Through interactions with the programs students had come to new meanings which resided not in the graphics and not in the students but was ever-changing and within the interplay between the students and the graphics.

The graphics, as with the software and computer, are not neutral in this context. What is generated is codetermined by the student and the graphics. The graphics occasion certain kinds of paths and limit others. Each graphic, given its structure and features, occasions its own paths. But it is the student who generates the path through embodied action codetermined by the students' history (including interacting with the software) and the software itself. A software designer, through the building of the software structure, may intend certain actions and pathways be taken but it is the user of this software who codetermines (with the software) the paths taken. In this sense computer software cannot teach. It can only codetermine with the user.

The computer was a part of this experience, codetermining the path laid down through students' actions. The cycle of change was ever active through the codetermination of subject and environment. In Chapter 6 students attempted to understand the softwares' graphic images and numeric values. As students explored they began to understand these depictions and began to direct their explorations based upon their growing understanding. By systematic exploration of the Logistics Equation $y = \mu x(1 - x)$ in the range $0 < \mu < 1$ students saw the iterates of these functions approach a fixed points of 0. But exploration in the range $1 < \mu < 3$ students saw a fixed point but its value changed, increasing as the μ-value increased. Explorations in the range $3 < \mu < 4$ led students to see not fixed points but 2-, 4-, 8-, and 16-cycles (bifurcations) and then chaos. The computer medium, by responding in predefined ways (through its structure), shaped students' understandings. But it was the students' choices of values which shaped the interaction and thus brought forth the medium. The computer software and students' actions were codetermined.

Implications for Education and Research

The meaning of instruction is codetermined by students in context. The structure and meaning of the 'information' is codetermined by students in the setting. Students structure their activities according to their perceptions of the medium and based upon their experiences and personal history.

> There is no "transmittal information" in communication. Communication takes place each time there is behavioral coordination in a realm of structural coupling. ... Each person says what he says or hears what he hears according to his own

of amplification which Ihde might have expected in his discussion of "ordinary usage." By interacting with students it became necessary to question "ordinary usage" expectations. Educators exploring computer environments with their students need to examine what types of activities are amplified by interaction with the environment and what types of activities are reduced. This determination can only be done by interacting with students as they explore the medium. This will allow educators to balance the activities, hopefully providing a more inclusive view of computing and broader applications and usage of skills and concepts than tradition narrow definitions of computer applications.

Chapter 4 traced two students interactions with the AP environment. Ways students interacted with this environment were evident through the description of students' activities. The way that students interacted in this environment was through/with the computer and with/in the assignment and context. Students enacted the computer environment in this context as an expressive environment through which they could develop stories. The stories involved images, text, animations and sounds. The nature of the stories was codetermined by the confines of the environment and yet were enhanced by the provision of facilities for students to develop stories in this context. Students were both empowered by the environment's functions and enframed by its limitations. Students in classrooms where multimedia production is the task have varying experiences with the technology. The computer environment, including hardware and software, facilitates particular types of explorations and activities by its structure like the inclusion of simple interactive sequences into students' stories. Thus it enframes students' activities to a particular range of activities delimiting the types of interactive sequences to a particular range of interactions (for example not allowing full sentence text responses to questions). However within that range the computer environment frees students to try new and exciting things they may not have experienced previously like the inclusion of animated graphic images into their text, or the inclusion of sounds to highlight activities of characters in their stories. Thus it both empowers and enframes students' activities.

Teachers need to be aware of this duality of function of the computer environment and examine how students' experience the context including its empowering and enframing elements. If particular elements are reduced by the particular computer applications and activities in which students are involved then the teacher might generate other activities, with or without the computer, to compensate for this enframement. If students perceive the computer environment to be unduly restrictive, either because of software or hardware limitations, then the environment likely does not occasion students to take action and produce experiences for themselves that will expand their possibilities. Teachers aware of this situation can take action to encourage exploration and learning by proposing alternative activities or allowing students to propose them. If the computer environment is too expansive and unlimited students may perceive it to be too complicated to explore and generate experiences through which patterns of coherence are possible. In such cases students may not wish to take action. Teachers can provide structure and pattern by delimiting the environment, for example only introducing students to a limited number of icons to begin with and then building up students' experiences with more icons over time. Computer environments, therefore, need to be carefully selected or generated so they encourage students to engage these environments and bring forth from the sphere of behavioral possibilities a world with consistencies where effective action can be taken.

Chapter 5 provides an overview of several computer programs for the exploration of mathematical concepts in the field of non-linear dynamics and chaos theory. The environment provided by these software packages was brought forth by the researcher through her interactions with it. Each package provided particular features which made aspects of the mathematics salient. The computer programs, taken together, generated a space for exploring and generating mathematical concepts consistent with the field of non-linear dynamics. This chapter provided insight into how the researcher brought forth the world of non-linear dynamics within this computer environment. For teachers who wish to use or develop software it may be useful for them to overview the features which are salient within a software medium and the limitations of this medium. For example in the case of the Cobwebber described in Chapter 5 only the Logistics Equation $y = \mu x(1 - x)$ was provided. A more general Cobwebber, capable of exploring other functions would be useful in understanding more fully the activity of Cobwebbing. The patterns of mathematical activity (approach of a fixed point, cycles, and chaos) would be generalizable to other functions if they could also be explored. In addition the Cobwebber only provided the opportunity to explore functions from the Logistics Equation for $0 < \mu \leq 4$. This limited the graphing of the Cobweb to a 1 x 1 square and thus was functional from a programming point of view. However, Cobwebbing can be done for values of $\mu > 4$ and the capacity to encourage these explorations may be beneficial. So an environment complicated enough to encourage exploration is necessary but it must be limited in some aspects to allow patterns of behavior to be evident. However it would be best if the environment could then be expanded to allow greater exploration. For example, the Cobwebber only dealt with a 'different equation'. What types of patterns of behavior occur for differential equations? This cannot be answered by the Cobwebber program as provided. As an alternative it might be best to create an equivalent of the Cobwebber in a mathematical system like Mathematica[1]. This would allow the delimiting of the activities using a Mathematica notebook with the appropriate commands. Once exploration of the existing parameters was sufficient that coherent patterns of activity were evident then the environment could be expanded to enable other possibilities and further explorations, for example other functions, differential equations and expanded ranges of the Logistics Equation.

Chapter 6 demonstrated students taking action within computer environments building mathematical meanings. By interacting with these students, by watching what they did, how they explored, and by having them record their activities, findings and perceptions it was possible to get some sense of their enacted meanings in the context. Their writings revealed an experience of this computer medium through mathematical actions. The medium seemed to be of mathematical significance to them. They noticed patterns and the lack of patterns, developed understandings of the context, and enacted this environment through explorations and interactions. They noticed the approach of a fixed point, cycles of activity when iterating, and chaos. However these students wondered what happened after chaos following $\mu = 4$. Was another type of behavior evident? They also wondered about values of $\mu < 0$. What effect would this have? They wondered what effect input values of x, other than $0 \leq x \leq 1$, might have. In particular they wondered about negative values. Students were introduced to fractal geometry in the study. They were interested in the connection of their non-linear dynamics mathematical explorations to the study of fractal geometry. Many fractal packages exist which use iterative functions to generate fractals. These could also be explored by students to extend their mathematical activities.

If the patterns revealed through each of the studies characterizes human-computer interaction in these two contexts then the role of the teacher, the student, the environment, the curriculum and the computer changes. The effect of each is codetermined by the context in which activity takes place. The teacher's role changes from provider of information to facilitator of exploration becoming a learner and explorer with the students. The students' activities take on new importance as these are the ways that students experience and generate new meanings. The environment mediates the activities of students so it must be considered in educational decision making and educational research. The curriculum becomes that which is enacted by students immersed in a context. The notion of curriculum as transmitted from a provincial, state, or federal government to the classroom is no longer reasonable because what the teacher and students bring forth is what the curriculum becomes. Commonly sited goals of computer literacy become

1 Mathematica is a mathematical system created by Wolfram Research, Inc., Champaign, Illinois.

unreasonable. New pedagogic concerns will undoubtedly emerge from student and teacher involvement in computer-enhanced contexts. These become places for further research.

When one enters into research from an enactive stance, this stance is embodied in one's actions. The environment takes its place as the background to the participants in the study who are in the foreground. The environment and participants codetermine each other. The participants enacting from the sphere of behavioral possibilities a world of coherence which becomes the focus. The question emerges: what is the nature of human embodied action in the foreground against a background of the environment? Thus the participants to be studied and the context of the study must be specified. The participants, as always, will be determined by availability and researcher access. However the backgrounds which are of interest to educators engaged in computer research can be specified, constructed and explored.

Mathematical computer environments can be developed within mathematical systems like Mathematica where students can engage in algebraic/mathematical exploration and imaging. Students can enact 1) geometric exploration through geometry software, 2) mathematical imaging through graphing packages, and 3) numeric manipulation and graphics generation through spreadsheet files. Research can focus upon students' exploration potential, activities and findings within these environments. Specific research questions might focus upon the kinds of environments which are expansive enough to encourage exploration but limited enough not to overwhelm. The environments might then be expandable to encourage students' continued exploration. The interplay between students' explorations and the expansion of environments is worthy of study.

As an example of such an exploration consider the spreadsheet environment. Students could be encouraged to create formula placed into cells and then graph these cells (as described in the spreadsheet Logistics Equation in Chapter 5). Students could then explore functions, numeric relationships, and equation generation using this technique. The students could initially be introduced to generating equations using simple mathematical expressions involving addition and multiplication of values housed in other cells. When students are capable of manipulating these expressions and capable of interpreting or even predicting results then students could expand their activities incorporating calculations using more than one cell as a source of values, including other types of calculations like division and subtraction, and using iterative approaches to writing expressions. Activities would need to be open ended enough for students to explore the spreadsheet medium but with some initial limitations of complexity to enable exploration and coherence. Later activities could be expanded by incorporating a larger number of functions guided by students' activities and questions.

Many other research contexts could be explored within the computer domain. Research could ask how drawing and painting packages enable students to generate graphics and images through these spaces. How do word processors provide opportunities for students to engage in language exploration? How do desktop publishing packages, which integrate language and image presentation spaces, provide students possibilities to see the interplay between text and image? How do students integrate images, text and dynamic video through desktop video? In what types of activities do students engage when students explore multimedia development packages which integrate sound, animation, text, and images? Networks and electronic mail provide students spaces for information sharing and exchange. How do students exchange, distribute, use and explore information within these computer-enhanced environments? These questions all point to issues of students' activities, actions, perceptions and meaning making in computer environments. This may be the focus that research in computer education takes in coming years.

Conclusion

An enactivist stance is an alternative to a representationist stance. "The autonomy of the living being [must] be given its full place" (Maturana & Varela, 1987, p. 253). This stance is a view of knowledge which is quite different from that usually found in the domain of cognitive science. This stance to cognition has broad implications. If we cast out our representationist views then what is research? What is instruction? What is educational experience? If this non-representationist view takes the sense-making capacity of autonomous living systems as its focus then research agendas and educational endeavors must take new forms. This stance implies the need to look deeply at our own practices and change them to encompass this broader view. This focus toward change will engage us in action, bringing forth a world of research and education from an enactive stance.

Epilogue

The *knowledge of knowledge compels*. It compels us to adopt an attitude of permanent vigilance against the temptation of certainty. It compels us to recognize that certainty is not a proof of truth. It compels us to realize that the world everyone sees is not *the* world but *a* world which we bring forth with others. It compels us to see that the world will be different only if we live differently. It compels us because, when we know that we know, we cannot deny (to ourselves or to others) that we know. (Maturana & Varela, 1987, p. 245)

References

Alessi, S. M. (1988). Fidelity in the design of instructional simulations. Journal of Computer-based Instruction, 15(2), 40-47.

Bannon, L. J. & Bødker, S. (1991). Beyond the interface: Encountering artifacts in use. Designing Interaction: Psychology at the Human-Computer Interface. Cambridge, MA: Cambridge University Press.

Burgess, C. G. (1991). A graphical, database-querying interface for casual, naive computer users. International Journal of Man-Machine Studies, 34(1), p. 23-47.

Devaney, R. (1990). Chaos, Fractals and Dynamics: Computer Experiments in Mathematics. Menlo Park, CA: Addison-Wesley.

Eaton, S., & Olson, J. (1986). Doing computers? The micro in the elementary curriculum. Journal of Curriculum Studies, 18(3), 342-344.

Falconer, K. (1990). Fractal Geometry: Mathematical Foundations and Applications. Chichester, England: John Wiley and Sons.

Gittens, D. (1986). Icon-based human-computer interaction. International Journal of Man-Machine Studies, 24, p. 519-543.

Goktepe, M., Ozguc, B., & Baray, M. (1989). Design and implementation of a tool for teaching programming. Computers and Education: An International Journal, 13(2), 167-178.

Goktepe, M., Ozguc, B., & Baray, M. (1989). Design and implementation of a tool for teaching programming. Computers and Education: An International Journal, 13(2), 167-178.

Goldenberg, P. (1989). Seeing beauty in mathematics: Using fractal geometry to build a spirit of mathematical inquiry. Journal of Mathematical Behavior, 8, 169-204.

Hazen, M. (1987). Criteria for choosing among instructional software authoring tools. Journal of Research on Computing in Education, 20(2), p. 156-164.

Ihde, D. (1979). Technics and Praxis. Boston: D. Reidel Publishing.

Jones, T. (1990). Children and animated computer icons. Journal of Research on Computing in Education, 22(3), 300-309.

Jones, T. (1990). Children and animated computer icons. Journal of Research on Computing in Education, 22(3), 300-309.

Kember, D. (1991). Instructional design for meaningful learning. Instructional Science, 20(4), p. 289-310.

Kitajima, M. (1989). A formal representation system for the human-computer interaction process. International Journal of Man-Machine Studies, 30(6), p. 669-696.

Lauwerier, H. (1991). Fractals: Images of Chaos. New York: Penguin Books.

Lefevbre, J. A. and Dixon, P. (1986). Do written instructions need examples? Cognition and Instruction, 3, p. 1-30.

Lodding, K. N. (1983). Iconic interfaces. IEEE CG&A, 28(2), 11-20.

Lodding, K. N. (1983). Iconic interfaces. IEEE CG&A, 28(2), 11-20.

Mandelbrot, B. (1982). The Fractal Geometry of Nature. San Francisco: W. H. Freeman.

Marton, F. (1981). Phenomenography - Describing conceptions of the world around us, Instructional Science, 10, p. 177-200.

Maturana, H. R. & Varela, F. J. (1987). The Tree of Knowledge. Boston: New Science Library of Shambhala Publications.

Mayer, R. E. (1981). The psychology of how novices learn computer programming. ACM Computing Surveys, 13, p. 121-141.

Mayes, J. T., Draper, S. W., McGregor, M. A., and Oatley, K. (1988). Information flow in a user interface: The effect of experience and context on the recall of MacWrite screens. In D. M. Jones & R. Winder (Eds.), People and Computers IV. Cambridge, UK: Cambridge University Press.

McLean, R. S. (1983). Ontario Ministry of Education specifies its microcomputer. In Preceding of the Fourth Canadian Symposium on Instructional Technology (pp. 436-441). Winnipeg, Manitoba: National Research Council of Canada.

Merleau-Ponty, M. (1963). The Structure of Behavior. Trans. Alden Fisher, Boston: Beacon Press.

Milnor, J. (1978). Analytic Proofs of the "Hairy Ball Theorem" and the Brouwer Fixed Point Theorem. American Mathematics Monthly, 85, p. 521-524.

Murphy, E. D. and Mitchell, C. M. (1986). Cognitive attributes: implications for display design in supervisory control systems. International Journal of Man-Machine Studies, 25, p. 411-438.

Norman, D. A. (1980). Twelve issues for cognitive science. Cognitive Science, 4, 1-32.

Norman, D. A. (1983). Some observations on mental models (p. 7-14). In D. A. Norman, Mental Models. Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D.A. (1990). Why interfaces don't work. In Laurel, B. (ed.) The Art of Human Computer Interface Design (p. 209-219). Reading, Massachusetts: Addison-Wesley.

Nussbaum, J. and Novick, S. (1982). Alternative frameworks, conceptual conflict and accommodation: Toward a principled teaching strategy. Instructional Science, 11, p. 183-200.

Palmiter, S. and Elkerton, J. (1991). Animated demonstrations vs written instructions for learning procedural tasks: A preliminary investigation. International Journal of Man-Machine Studies, 34(5), p. 687-701.

Putnam, H. (1989). Representation and Reality. Cambridge, MA: MIT Press.

Reigeluth, C. M. & Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. Journal of Computer-Based Instruction, 16(1), 1-10.

Rheingold, H. (1990). An interview with Don Norman. In Laurel, B. (ed.) The Art of Human Computer Interface Design (p. 5-10). Reading, Massachusetts: Addison-Wesley.

Rode, M. and Poirot, J. (1989). Authoring systems – Are they used? Journal of Research on Computing in Education, 22(2), p. 191-198.

Searle, J. (1992). The Rediscovery of the Mind. Cambridge, MA: MIT Press.

Stead, R. (1990). Problems with learning from computer-based simulations: a case study in economics. British Journal of Educational Technology, 21(2), 106-117.

Strike, K. and Posner, G. J. (1985). A conceptual change view of learning and understanding. In L.H. West and A.L. Pines (Eds.), Cognitive Structures and Conceptual Change. New York: Academic Press.

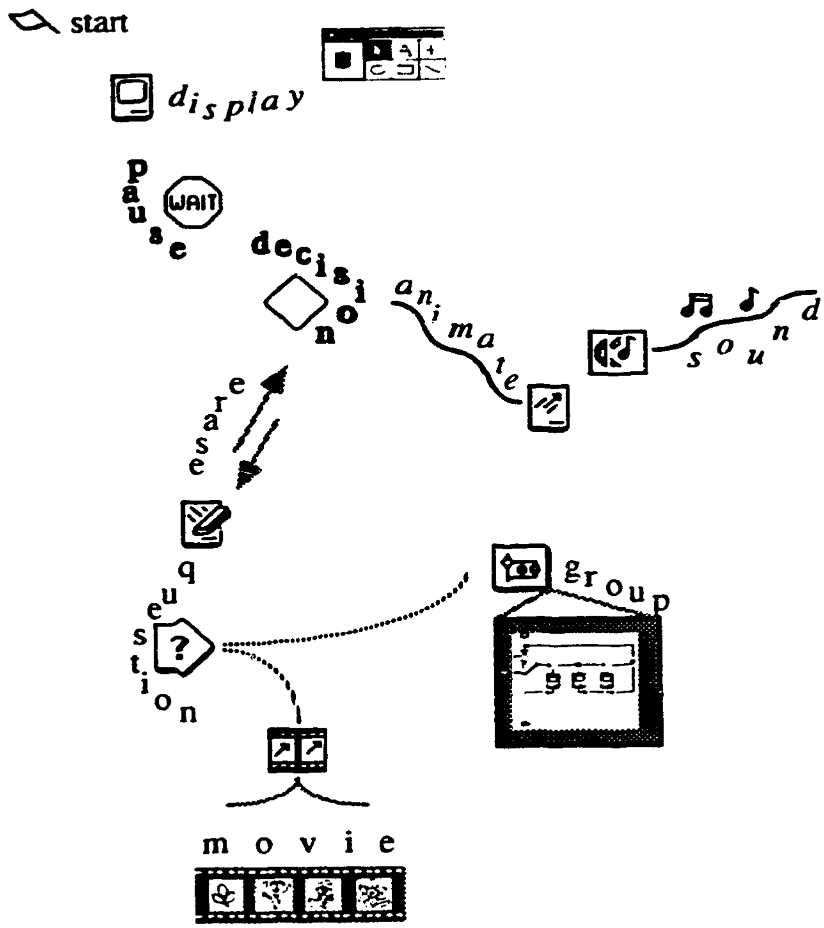
Townsend, B. M. (1986). Iconic Commands and Command Systems. Unpublished manuscript. Toronto, ON: Sheridan College, Software Evaluation Unit.

Varela, F., Thompson, E., and Rosch, E. (1991). The Embodied Mind: Cognitive Science and Human Experience. Cambridge, MA: MIT Press.

West, L. (1988). Implications of recent research for improving secondary science learning. In P. Ramsden (Ed.), Improving Learning: New Perspectives. London: Kogan Page.

White, B. and Horwitz, P. (1988). Computer microworlds and conceptual change: A new approach to science education. In P. Ramsden (Ed.), Improving Learning: New Perspectives. London: Kogan Page.

Appendix



Learning to Use Authorware Professional™
A Module for Computer Processing 20

By Judy M. Barnes©
University of Alberta

stop

Outline of the Study

1) a) After completion of the first CAI lesson **1. Introduction** which introduces the concepts of icon and flowchart and introduces the first three types of icons (display, erase, pause) in Authorware Professional the student could be asked questions (listed under the heading Interview One) in regard to their understanding of these concepts.

   b) After completion of the interview the students will be given an opportunity to create a lesson (Activity 1) with at least two displays and incorporate erases and pauses. The task will involve exploration of effects, lines, fills and modes. This task is described under "Activity One"

2) a) Students will now use the second CAI lesson **2. Introduction to the Icons**. This lesson introduces each of the icons with special emphasis on the decision icon.

   b) Students will answer questions (Activity Two). This will assist in determining any difficulties with the loop structures before proceeding.

   c) The student will then complete Activity Three to further explore understanding of loop structures in Authorware Professional. By interacting with students during this creation it will be possible to understand students ideas of creation of a flowchart and to better understand student conceptions of flowcharts.

3) a) After completion of the CAI lesson **3. Multiple Choice** each student will create a multiple choice question (Activity Four) of his/her choice.

   b) By interacting with students during this creation it will be possible to understand students ideas of creation of a interactions in Authorware Professional.

4) a) After completion of the CAI lesson **4. Drag Object** each student will be asked to create a drag object question (Activity Five) of his/her choice. By answering student questions and observing student progress difficulties in conceptualizing the process can be identified and intervention to correct misconceptions is possible.

5) a) Activity Six relies on student ability to construct flowcharts and requires the creation of a text answer question. This question is provided so that students explore various aspects of question creation including test response type, group icons as feedback, and order of feedback. Students ability to create and analyze flowcharts will be assessed.

   b) Students will then create a text answer qeustion (Activity Seven) of their own choosing.

6) During Activity Eight students will return to the five programs they have created: i) displays, erases and pauses, ii) loop structure, iii) multiple choice question, iv) drag object question, v) text answer question. They will 'group' each one of these five mini-lessons into individual icons. These five group icons will then be copied and pasted into one Authorware File called **Student Name6**. Students will have created a complete and complicated program integrating small interconnected pieces into a complete program.

7) Following these explorations students will again be interviewed about conceptual understanding of icons, flowcharts and Authorware (Interview One, Questions 1 a-g). This will assist in understanding the mediating effect of experience in understanding Authorware Professional. The question of what is Authorware Professional and what it creates can once again be asked. This may indicate a richer understanding than previously indicated.

8) Following these activities students will create another lesson of their own choosing (Final Project) which uses their new skills in working with Authorware Professional.

Interview One

This interview is to follow the completion of the first CAI lesson "1. Introduction". It's intent is to establish the conceptual nature of students understandings of icons, flowcharts and Authorware Professional and to establish the experience level of students

1) Questions pertaining to **Conceptual Understanding of Icons, Flowcharts and Authorware**
   a) What is an icon in Authorware Professional? Compare these icons to those in the Macintosh environment.
   b) What do display icons display? 
   c) What do erase icons erase? 
   d) What is a flowchart and what is its function? Compare these flowcharts to those used by programmers.
   e) What is the function of a toolbox?
   f) What do pull-down menus do in Authorware Professional? Compare these menus to those in the Macintosh environment.
   g) What is Authorware Professional?

2) Questions related to **Demographics and Experience**
   a) Age? Gender?
   b) How many years have you used a computer?
   c) How many years have you used a Macintosh computer?
   d) Do you consider yourself comfortable with the use of a computer?
   e) Was learning from the CAI package helpful?
   f) Have you ever used Authorware Professional before this course? Explain.
   g) Do you own your own computer?

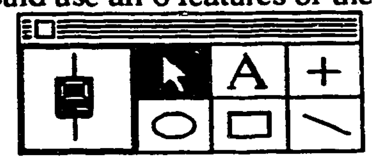
Activity One

If you do not already have one, create a new folder on your diskette or hard drive and give it your name.

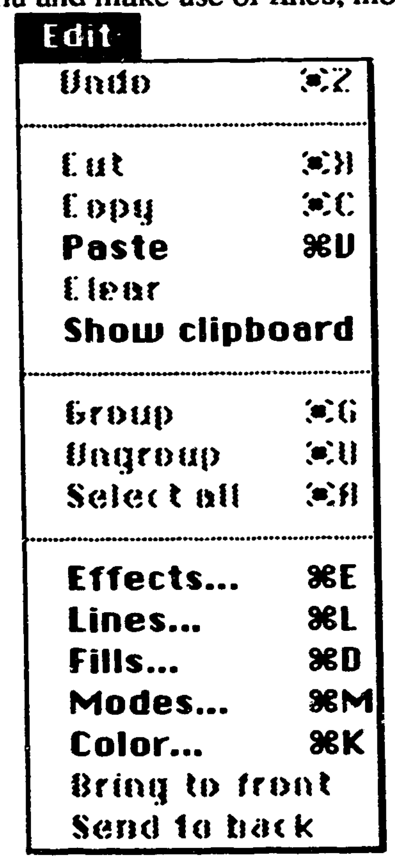
Double click on the file in your folder named **1. Introduction**. You will see a simulation of the use of Authorware Professional. Pay attention to the motion on the screen and the text descriptions you see. When a blank screen at the end of the lesson appears you should quit the lesson.

Double click on the Authorware Working Model. When prompted to do so create a new file called **Your name1**. In this file you will place display icons, erase icons, and wait icons.

Within the display icon you should use all 6 features of the toolbox shown below.



Explore the Edit pull-down menu and make use of lines, modes, fills and effects.



In addition you should make use of the Font and Style pull-down menus to create text of different styles, sizes and types. When you have created this file save it in your folder on the hard drive. Now quit the Authorware Working Model and copy the file to your diskette. This way you will have a backup of the ile for later use.

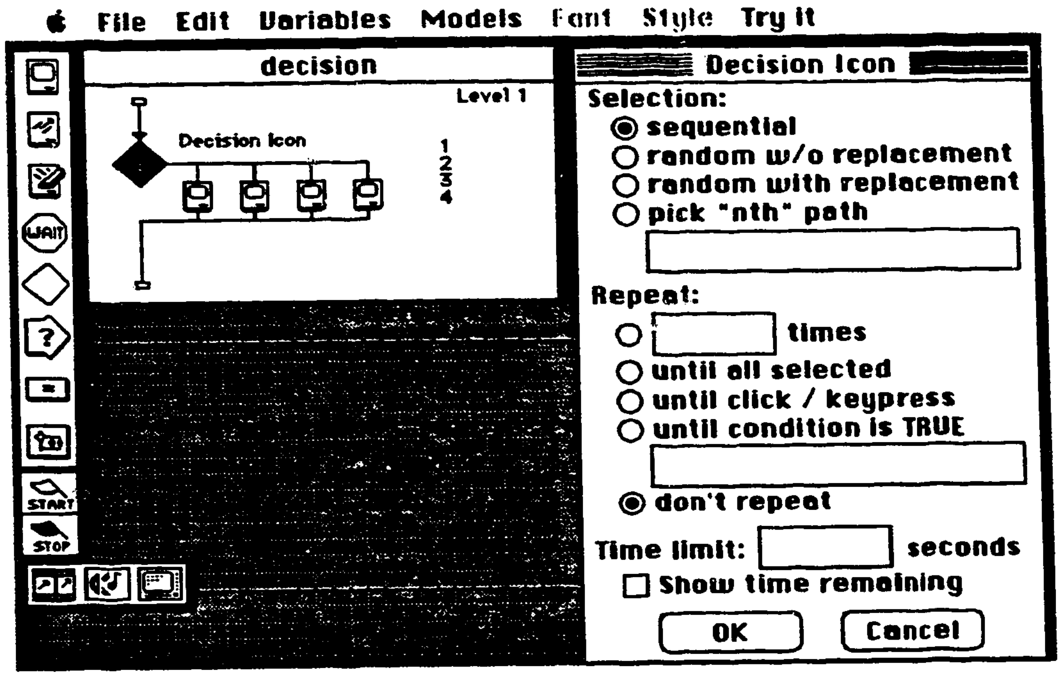
Activity Two

Double click on the file in your folder named **2. Icons Explained**. You will see another simulation. From the simulation you will see the names of each of the icons and some of what these icons do.

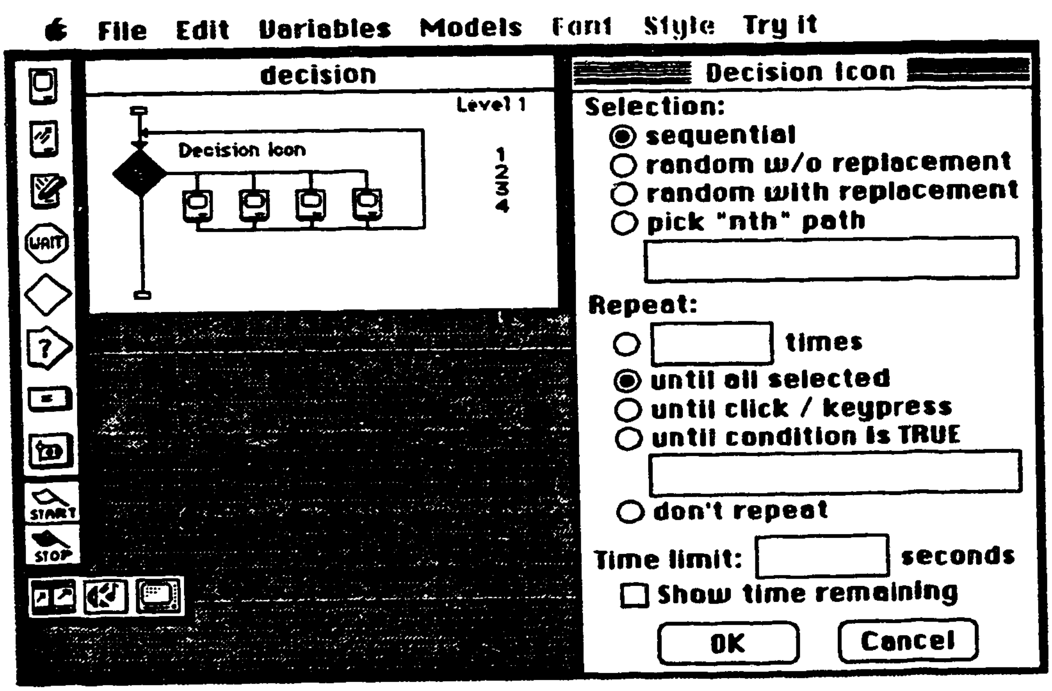
State the name of the icon and describe what each of the icons does. Pay particular attention to the last icon in this list as part 2 of this activity involves this icon extensively.

2) a) Describe what the following flowcharts do? Where possible, draw the path the flow indicator would take in this flowchart. If you cannot draw the path flow then describe the path flow.

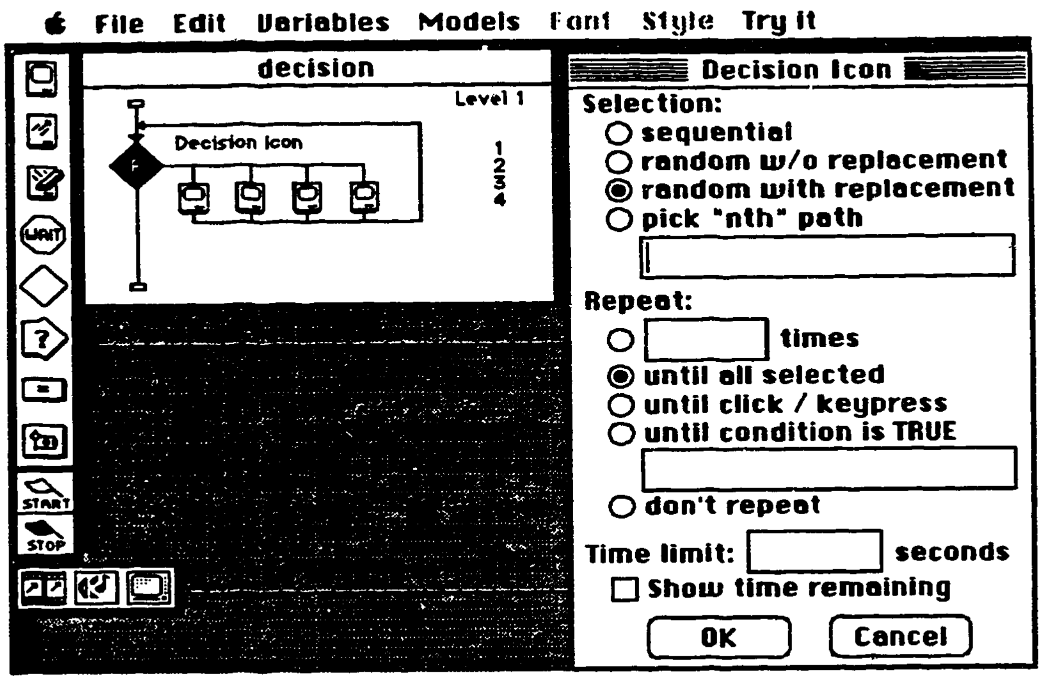
i)



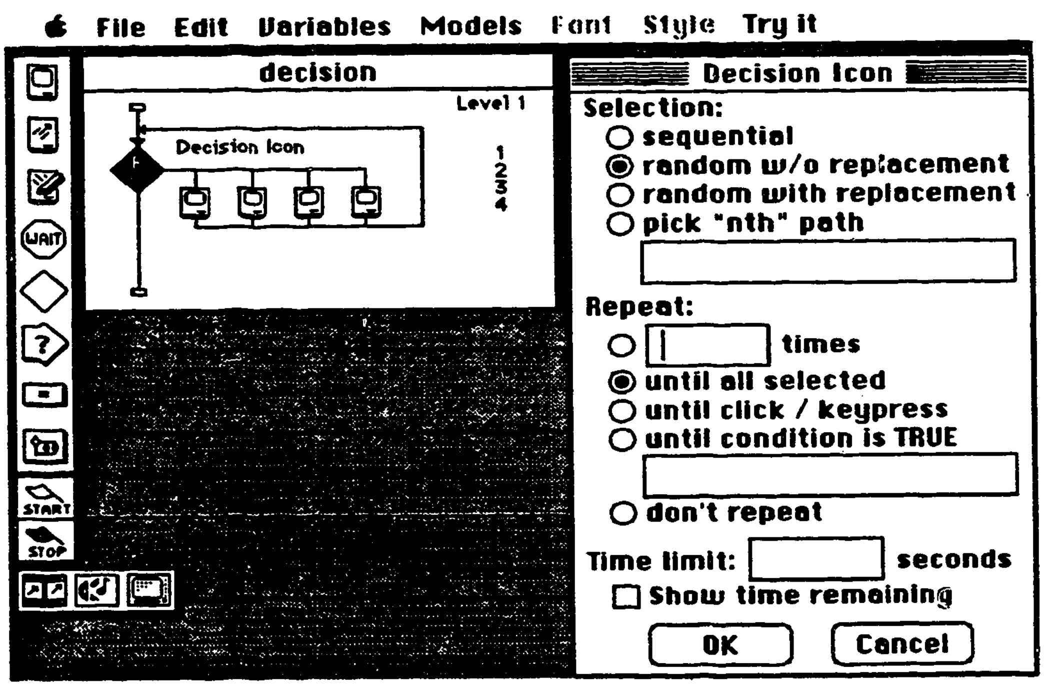
ii)



iii)

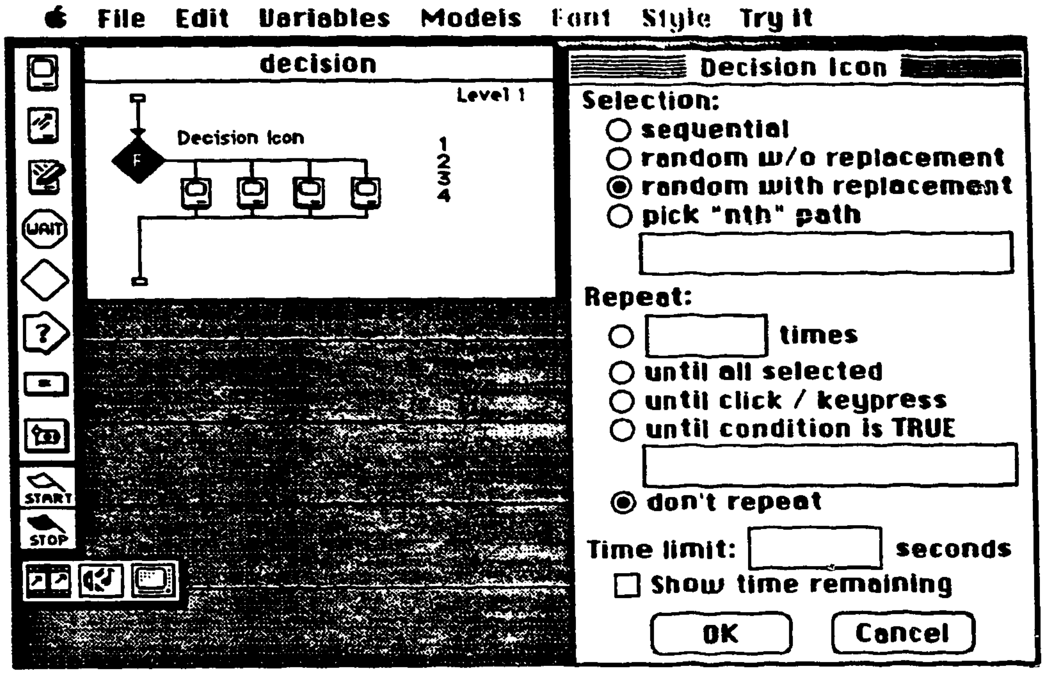

File Edit Variables Models Font Style Try it
decision
Level 1
Decision Icon
1
2
3
4
START
STOP
Decision Icon
Selection:
sequential
random w/o replacement
random with replacement
pick "nth" path
Repeat:
times
until all selected
until click / keypress
until condition is TRUE
don't repeat
Time limit:
seconds
Show time remaining
OK
Cancel

iv)

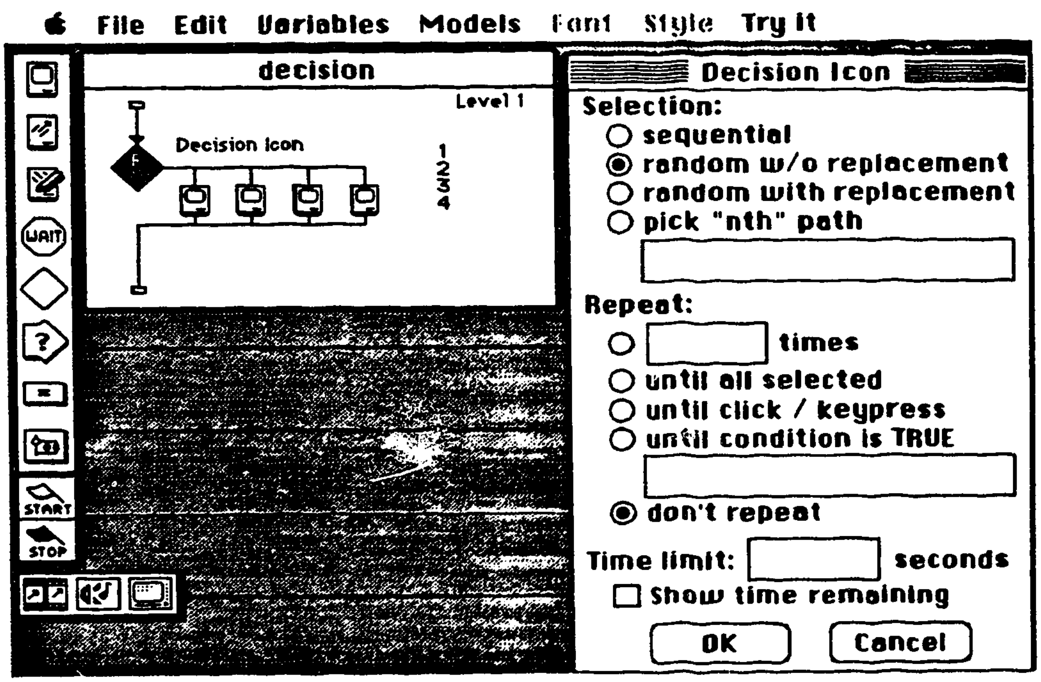

File Edit Variables Models Font Style Try it
decision
Level 1
Decision Icon
1
2
3
4
START
STOP
Decision Icon
Selection:
sequential
random w/o replacement
random with replacement
pick "nth" path
Repeat:
times
until all selected
until click / keypress
until condition is TRUE
don't repeat
Time limit:
seconds
Show time remaining
OK
Cancel

b) Describe the functioning of the following flowcharts (labelled i and ii) and compare and contrast the function of the two flowcharts.

i)

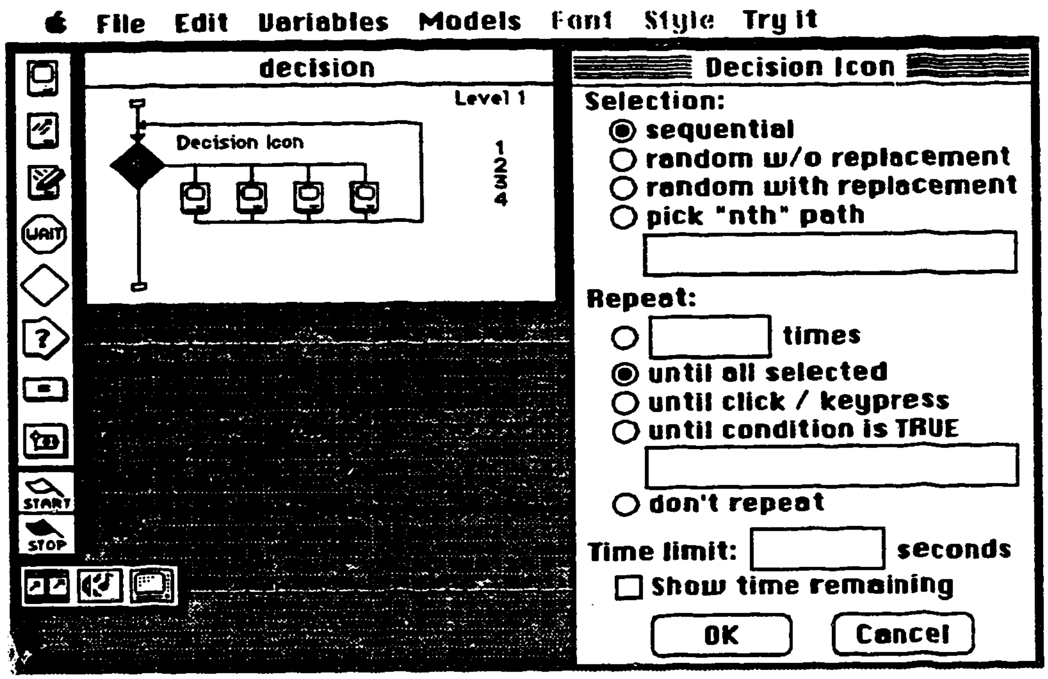


ii)

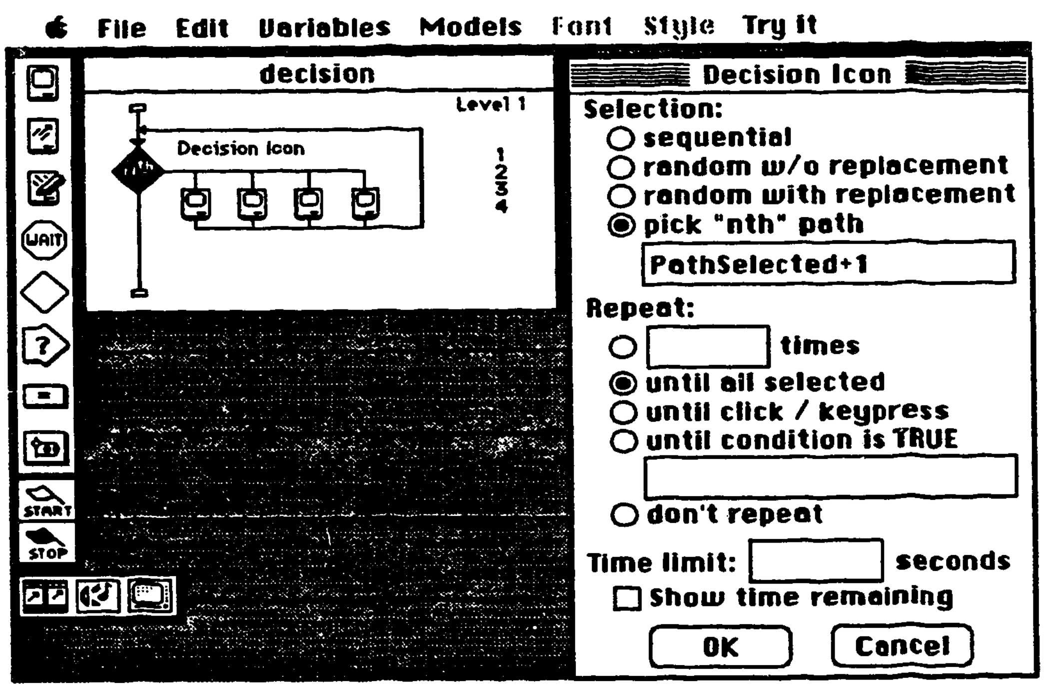


c) Describe the functioning of the following flowcharts (labelled i and ii) and compare and contrast the function of the two flowcharts.

i)

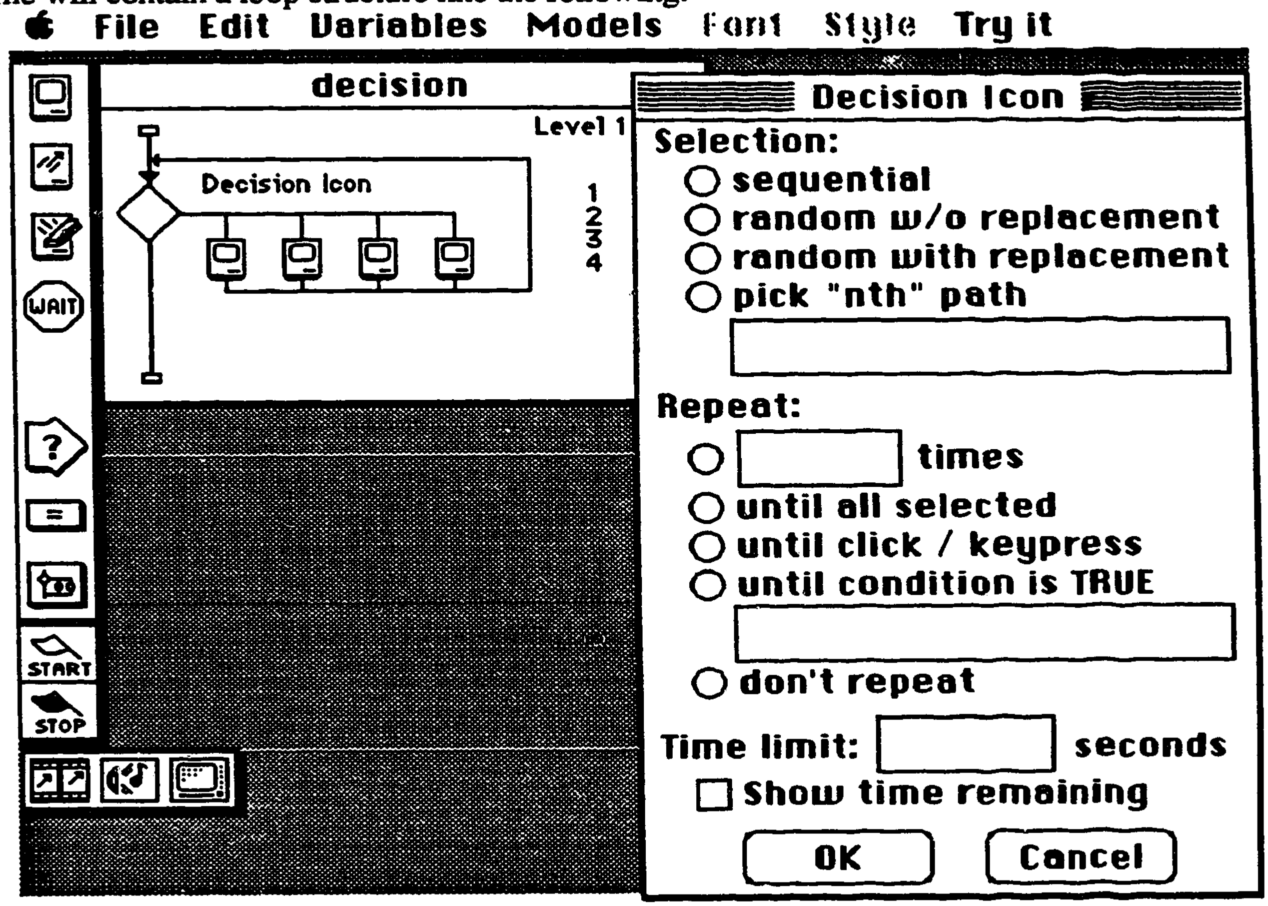


ii)



Activity Three

Open the Authorware Working Model and create a new file called **Loop Structure.** This file will contain a loop structure like the following.



Inside the first display icon you are to place the number 1. In the second display type the number 2. In the third display type the number 3. In the fourth display icon type the number 4. Set each of the display choices to **Pause before continuing** by placing an x in the box for this choice when the menu appears. The decision icon can initially be set to **sequential** and **don't repeat.** Once you have this file created you are to run the file and record the numbers that appear in the order that they appear. You will then open the decision icon and change the options to **sequential** and **until all selected.** You will again run the file to determine the order in which the numbers appear. You will continue to change the options to those listed below. You will run the file after each change. Record your answers in chart like the one below.

1) Sequential, don't repeat ____________________
2) Sequential, until all selected ____________________
3) Sequential, 3 times ____________________
4) Random w/o, don't repeat ____________________
5) Random w/o, until all selected ____________________
6) Random w/o, 3 times ____________________

7) Random with, don't repeat ______________________

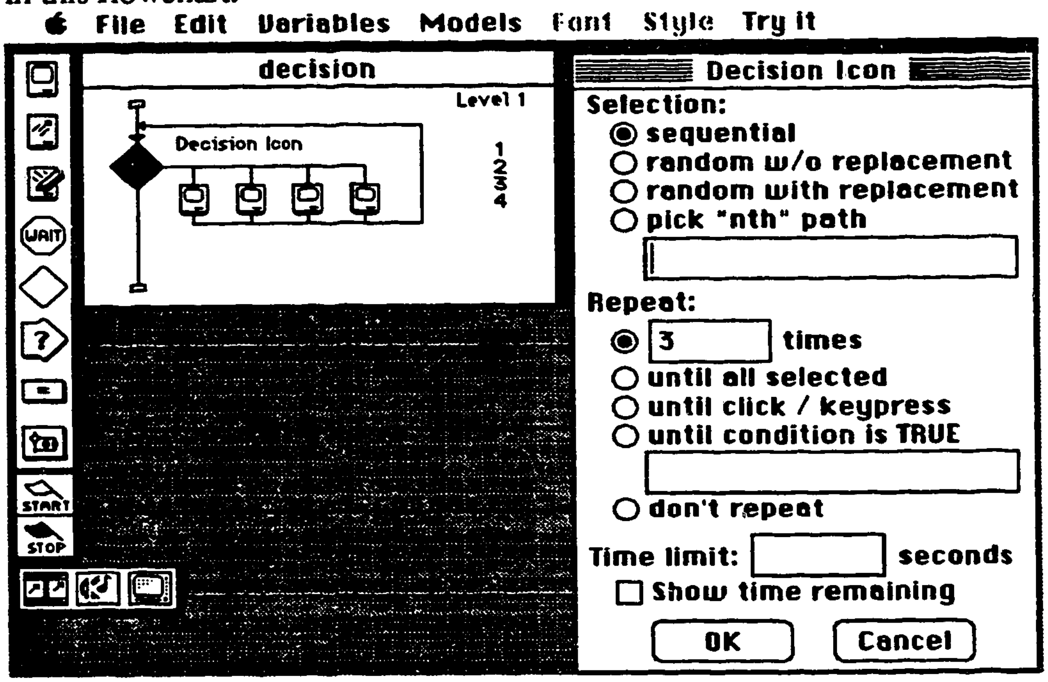
8) Random with, until all selected ______________________

9) Random with, 3 times ______________________

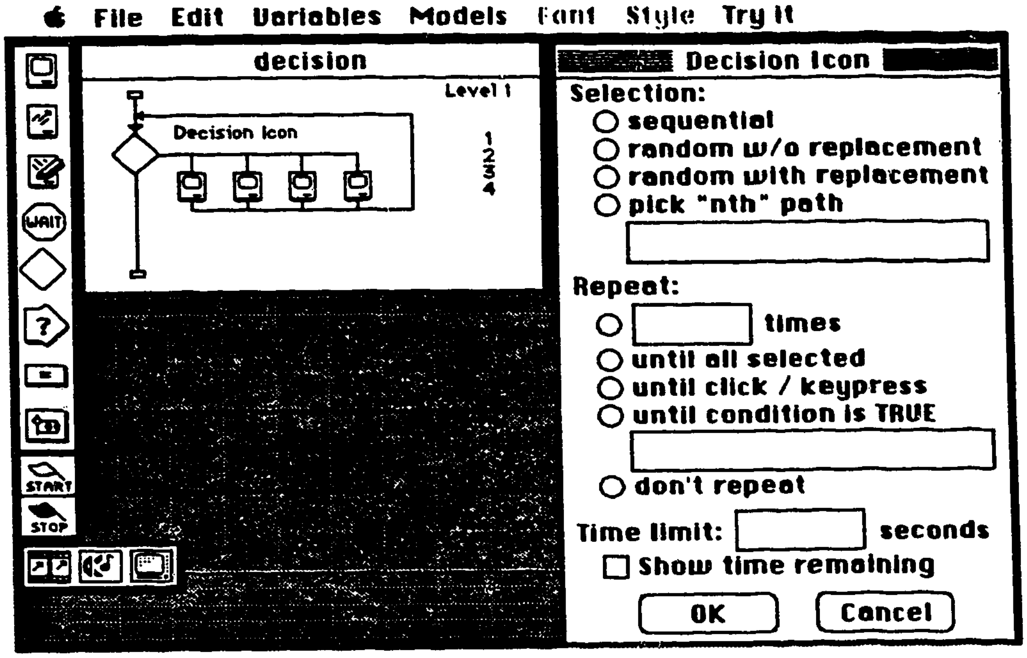
You may also want to try setting other options to see what they do. Your program can now be used to explore other option combinations too. Two explorations to use with this program are listed below. Think about what flows these options might create and then see if you can solve the problems.

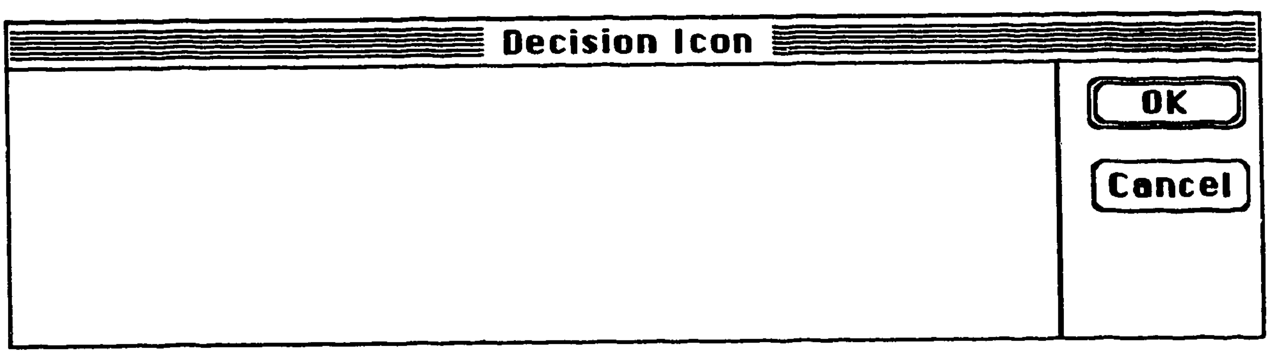
Bonus:

a) Describe what the following flowchart does? Describe the path the flow indicator would take in this flowchart.



b) Suggest another way to produce a sequential path that does not use the sequential option of the decision icon menu or PathSelected+1 in the pick "nth" path option. Use the following flowchart and decision icon option box to indicate how this might be done.





When you have finished your exploration you should save your file with sequential, don't repeat.

Activity Four

Complete the CAI lesson called 3. Multiple Choice. This simulation shows the creation of a multiple choice question using Authorware Professional.

Double click on the Authorware Working Model. When prompted to do so create a new file called **Your name2**. This activity requires that you create a multiple choice question of your own choosing using Authorware Professional. You may want to refer back to the CAI lesson 3. Multiple Choice if you have difficulty.

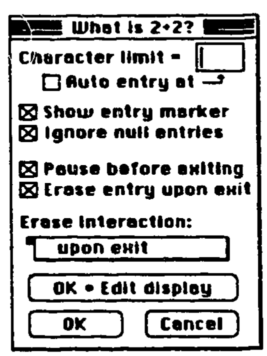
Activity Five

Complete the CAI lesson called 4. Drag Object. This simulation creates a drag object question using Authorware Professional. Make sure to try several wrong answers to the first question which asks you to drag a compact disk on to a compact disk player. This way you will see all the options available.

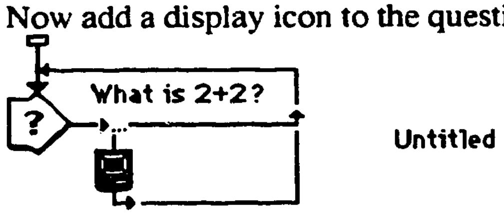
Double click on the Authorware Working Model. When prompted to do so create a new file called **Your name3**. This activity requires that you create a drag object question of your own choosing using Authorware Professional. You may want to refer back to the CAI lesson 4. Drag Object if you have difficulty.

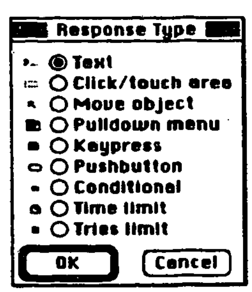
Activity Six

1) a) In this exercise you are to create a flowchart. Begin with a question icon named **What is 2+2?** Use the information provided to select options for the option box.

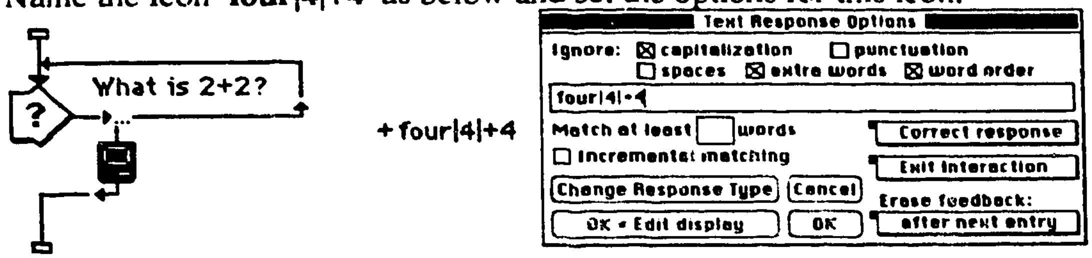


Now add a display icon to the question.

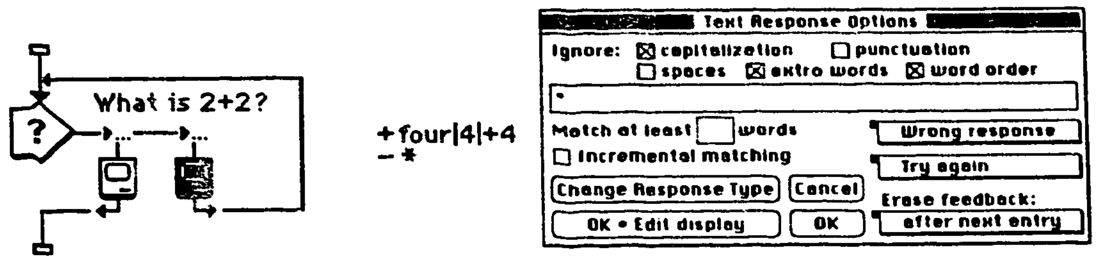




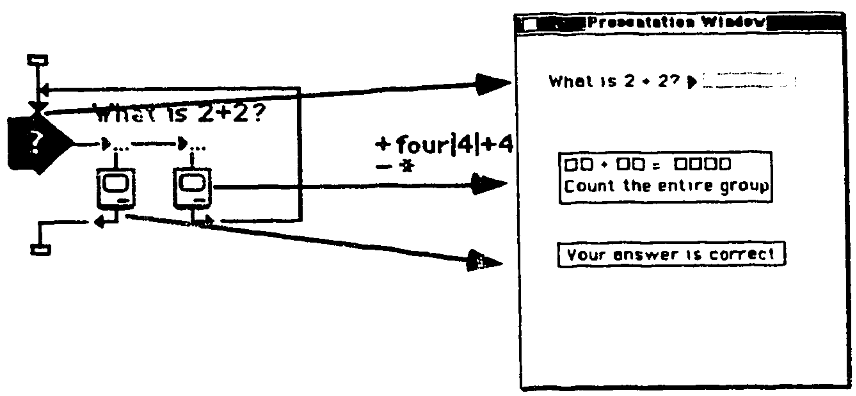
Name the icon **four|4|+4** as below and set the options for this icon.



Now add another display icon to the question and name it as below.



Now add the following information to the display icons and the question.



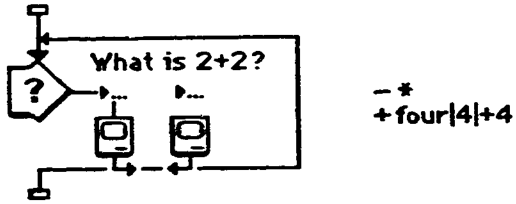
i) What does the question icon ask in your program?

ii) What is the correct answer that is tested for?

iii) What answers (besides the correct one) are tested by the question?

iv) What response type (ex. drag object, click/touch, etc.) is the correct answer? the incorrect answer?

b) You are now to modify the flowchart you have made by dragging the first display icon to the right of the second display icon. The result will look like the flowchart below.



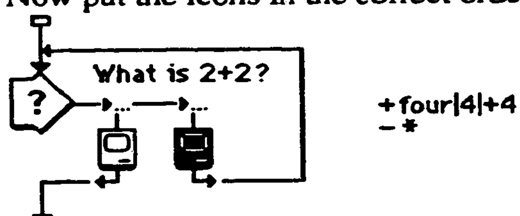
Now run the program and enter a wrong answer.

i) Describe what is displayed when the wrong answer is entered.

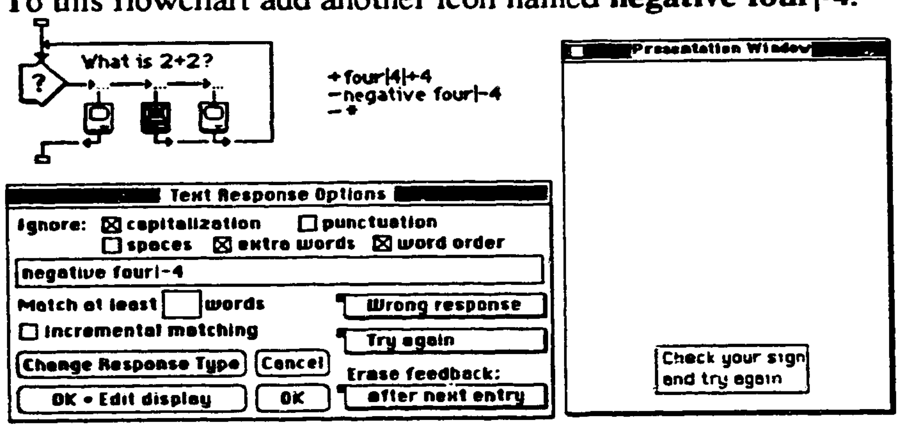
ii) Describe what is displayed when the correct answer is entered.

iii) Compare the answers for i) and ii) above and describe why they occur.

c) Now put the icons in the correct order as below.



To this flowchart add another icon named **negative four|-4**.

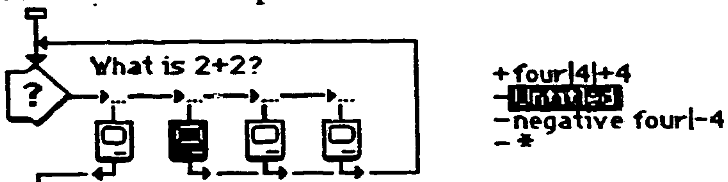


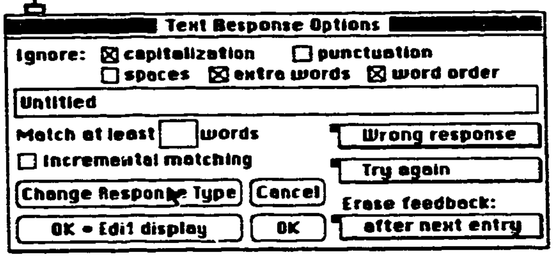
i) Describe the running of the program when -4 is entered. Why does this occur?

ii) Describe the running of the program when 3, 7, -4 are entered. Why does this occur?

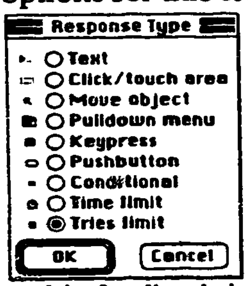
iii) Why have we placed the **negative four|-4** answer before the * answer? What would happen if they were placed in the program in reverse order?

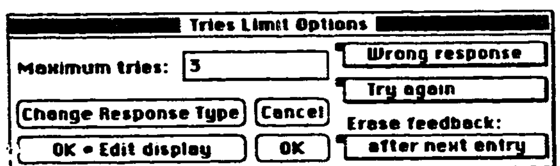
d) To your flowchart an icon is added. You will need to change the response type as the arrow on the option box shows.



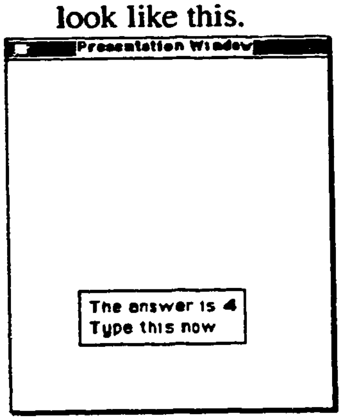


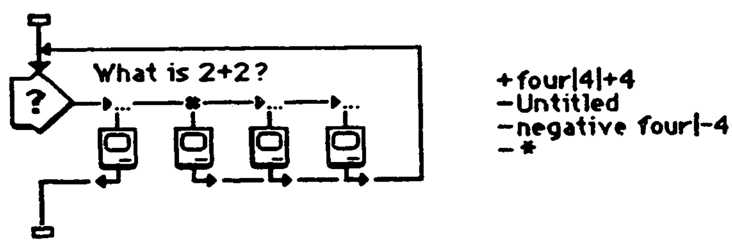
Change the response type to tries limit as shown below and set the tries limit options for this icon.



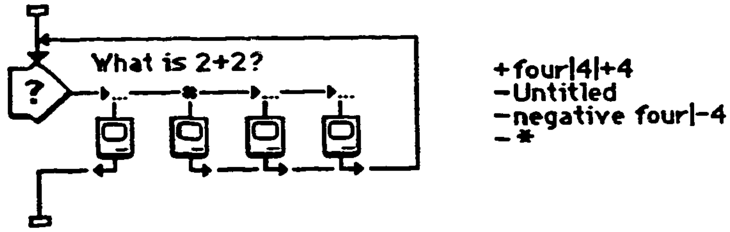


Enter this feedback in the display icon for the tries limit. Your flowchart should now look like this.

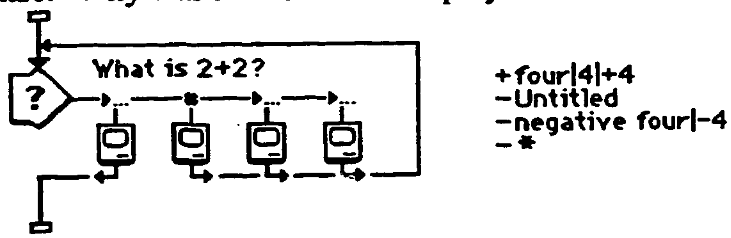




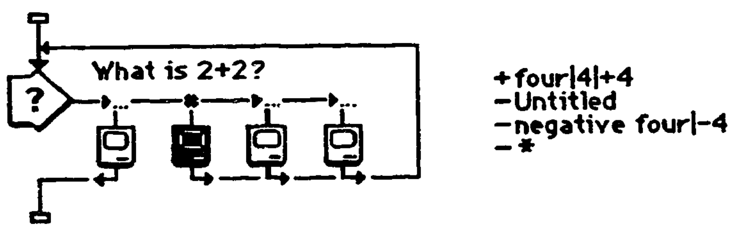
i) Describe the feedback given when the sequence 3, 5, 7 of wrong answers is entered and trace the path that was followed (when 3, 5, 7 was entered) on this flowchart.

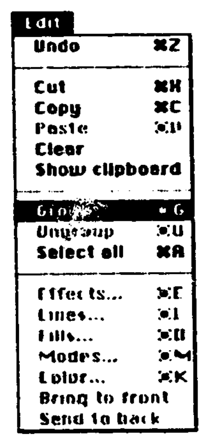


ii) Describe the feedback given when the sequence -4, -4, -4 of wrong answers is entered and trace the path that was followed (when -4, -4, -4 was entered) on this flowchart? Why was this feedback displayed?

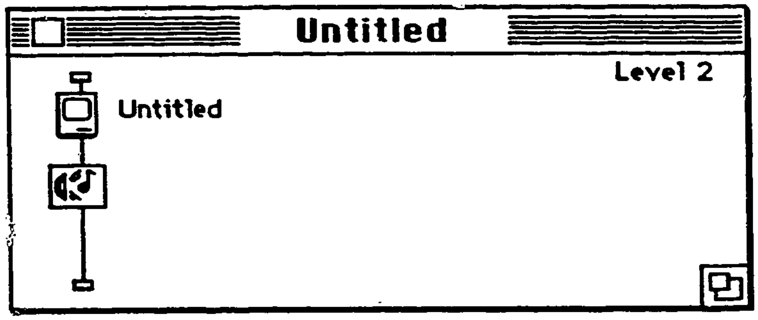


e) This section is optional as it involves the use of sounds from the Authorware folder.
Select (make black) the tries limit answer which is called **Untitled**. From the **edit pull-down menu** and select **group**.





Open the group icon and place a sound icon on the line and then select a sound from those in the Authorware folder on the hard drive.



i) Run the program and describe the feedback for the third wrong answer.

ii) Describe how a group icon can be used as feedback to a question icon.

Activity Seven

Create a new file called **Your name4**. This activity requires that you create a text answer question of your own choosing using Authorware Professional. This question will be like that created in activity six. Try setting these options and experimenting with the settings to see what they do. Save the text answer question to your file on the hard drive. You may want to refer back to Activity Six if you have difficulty.

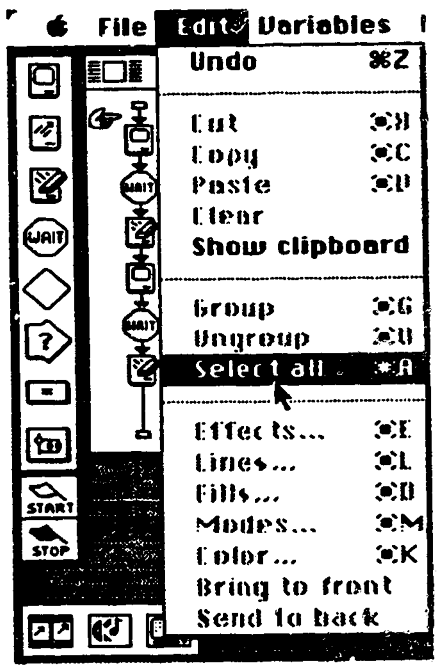
Activity Eight

You have now created five programs which are in your folder on the hard drive. Four of these icons which we will now use include: i) displays, erases and pauses **Your name1**, ii) multiple choice question**Your name2**, iii) drag object question**Your name3**, iv) text answer question**Your name4**.

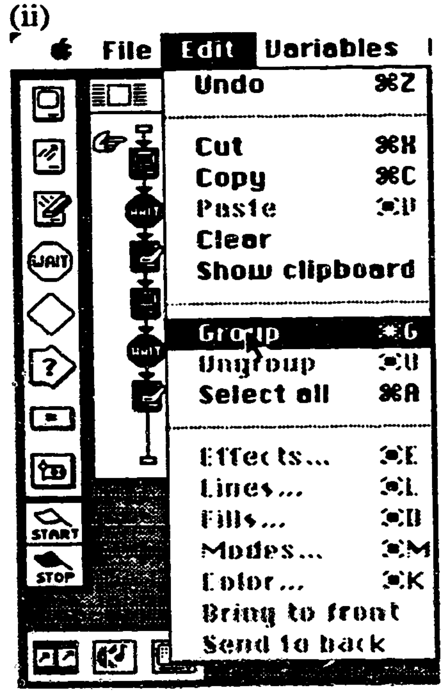
Warning: You can only paste 50 icons into the flowline. If these four lessons used many icons you may not be able to complete this activity as listed here because your file gets too big. The solution to this problem is to only copy a portion of each of the four programs created so a total number of icons is less than 50.

1) Open the first lesson you created. (i) Using the Edit pull-down menu **select all.** (ii) Now choose **group** from this menu. Name this group icon lesson one. (iii) **Copy** this group icon. (iv) Now **close** this first file.

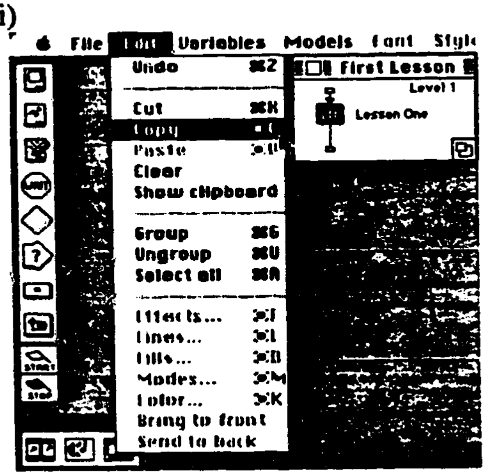
(i)



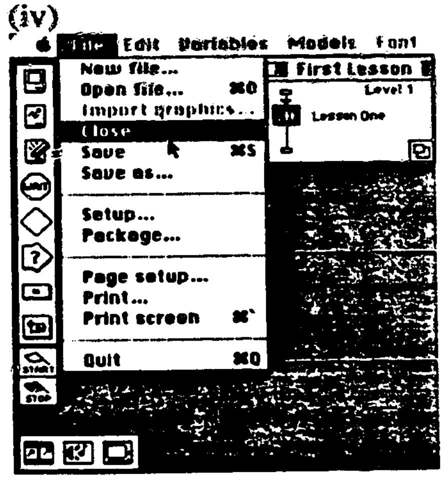
(ii)



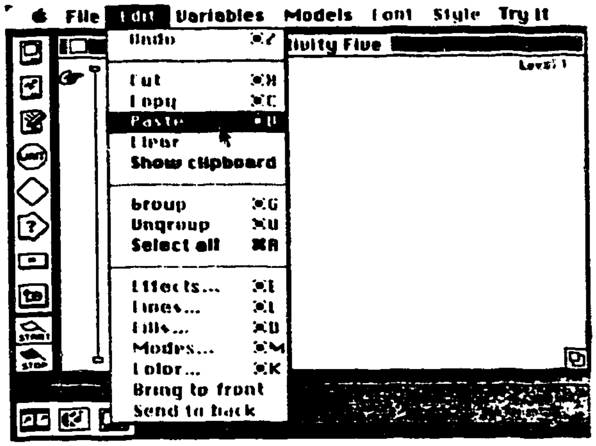
(iii)

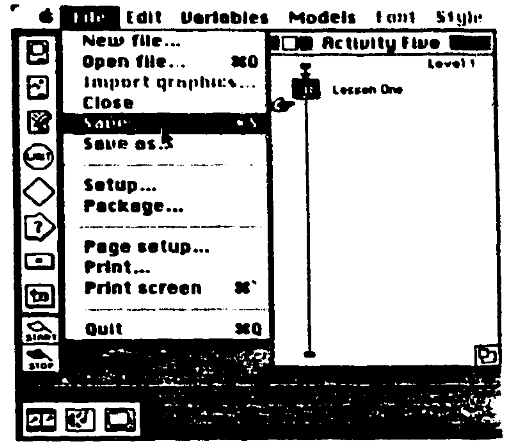


(iv)

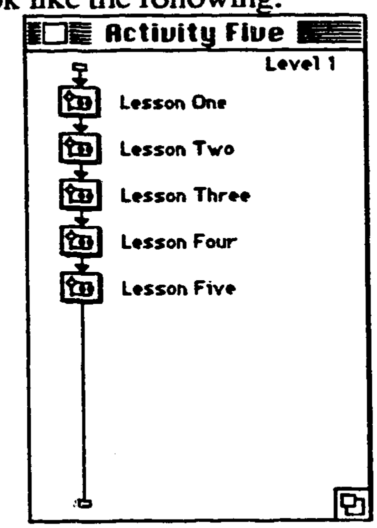


2) Create a new file and name it **Your Name5**. Click the mouse on the flowline and **paste**. Then **save** the new file.

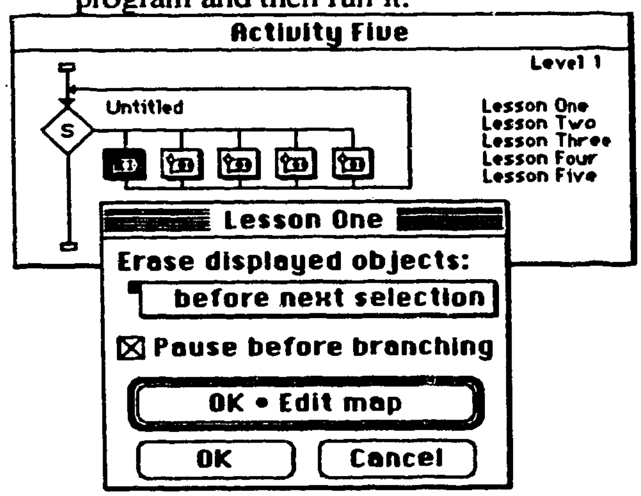




3) You will repeat the above process for the other four lessons you created. When you finish all five your file will look like the following.



4) Now you have five group icons on your flowline. You will now place a decision icon onto your flowline and place each of the group icons next to the decision icon so they stretch out to the right of the decision icon. Set the options as shown. Save the complete program and then run it.



5) You have now created a complete program composed of small pieces. Congratulations.