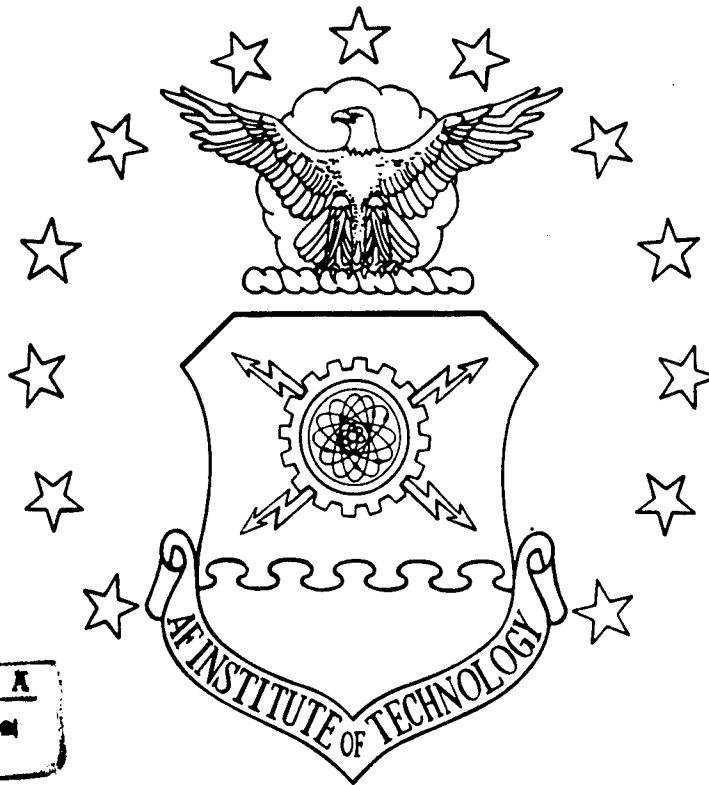


**SDTIC**  
**ELECTE**  
DEC 28 1994  
**C D**



**DISTRIBUTION STATEMENT A**  
Approved for public release  
Distribution Unlimited

ADVANCED EDUCATIONAL METHODS IN THE  
DEPARTMENT OF DEFENSE: APPLICATION OF CASE  
THEORY AND COMPUTER-ASSISTED INSTRUCTION TO  
BUSINESS PROCESS REENGINEERING

THESIS

Brian A. Brown, Captain, USAF  
Susan M. Brown, Captain, USAF  
Karen L. Cook, Captain, USAF

AFIT/GIR/LAR/94D-2

19941221 126

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY  
**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

AFIT/GIR/LAR/94D-2

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

ADVANCED EDUCATIONAL METHODS IN THE  
DEPARTMENT OF DEFENSE: APPLICATION OF CASE  
THEORY AND COMPUTER-ASSISTED INSTRUCTION TO  
BUSINESS PROCESS REENGINEERING

THESIS

Brian A. Brown, Captain, USAF  
Susan M. Brown, Captain, USAF  
Karen L. Cook, Captain, USAF

AFIT/GIR/LAR/94D-2

Approved for public release; distribution unlimited

The opinions and conclusions in this paper are those of the authors and are not intended to represent the official position of the DoD, USAF, or any other government agency.

AFIT/GIR/LAR/94D-2

ADVANCED EDUCATIONAL METHODS IN THE DEPARTMENT OF DEFENSE:  
APPLICATION OF CASE THEORY AND COMPUTER-ASSISTED INSTRUCTION  
TO BUSINESS PROCESS REENGINEERING

THESIS

Presented to the Faculty of the School of Logistics and Acquisition Management  
of the Air Force Institute of Technology  
Air University  
In Partial Fulfillment of the Requirements  
for the Degree of  
Master of Science in Information Resources Management

Brian A. Brown  
Captain, USAF

Susan M. Brown  
Captain, USAF

Karen L. Cook  
Captain, USAF

December 1994

Approved for public release; distribution unlimited

## Preface

The purpose of this research was to create a computer-assisted instruction (CAI) prototype tutorial to teach the fundamental principles of IDEF0 activity modeling. Our project was sponsored by the Defense Information Systems Agency (DISA), office of Corporate Information Management (CIM) as a possible aid in training Department of Defense personnel in this method of activity modeling within the DoD.

By far, the most challenging task was programming the tutorial using Authorware Professional™ authoring software. Although the tutorial is not included with this text, development of our prototype is documented by means of a model for creating CAI programs. This model incorporates elements of both instructional design and software development methodologies, addressing most of the educational and technical issues involved. Perhaps the most valuable aspect of this model is a series of iterations during each phase of development which virtually eliminates retroactive programming in successive phases. Of course, the model has the added benefit of adapting to every authoring software application.

We would like to thank our research advisors, Major Steven L. Teal and Lt Colonel William L. Schneider for their guidance in scoping our research and selecting an authoring software. We especially appreciate the effort Lt Colonel Schneider put into making resources available. As well, we would like to thank our respective families for their support in completing the project. It has been a long journey for us all.

Brian A. Brown  
Susan M. Brown  
Karen L. Cook

## Table of Contents

	Page
Preface . . . . .	ii
List of Figures . . . . .	v
List of Tables . . . . .	vi
Abstract . . . . .	vii
I. Introduction . . . . .	1
General Issue . . . . .	1
Background . . . . .	2
Specific Research Goal . . . . .	3
Research Objectives . . . . .	3
End Product . . . . .	3
Limitations . . . . .	4
Contributions of the Research . . . . .	5
II. Literature Review . . . . .	6
Overview . . . . .	6
Business Process Reengineering . . . . .	6
Computer-Assisted Instruction . . . . .	9
Roblyer/Hall Model for Instructional Design . . . . .	14
Prototyping . . . . .	15
Conclusion . . . . .	18
III. Methodology . . . . .	19
Overview . . . . .	19
Justification for Combining Models . . . . .	19
The New Model . . . . .	20
Phase 1: Requirements Definition . . . . .	21
Phase 2: Instructional Development . . . . .	22
Phase 3: Programming . . . . .	23
Phase 4: Testing and Modification . . . . .	24

	Page
IV. Findings and Discussion .....	26
Overview .....	26
Research Objective One Results .....	26
Evaluation Results .....	28
Conclusions .....	33
V. Summary, Conclusions and Recommendations .....	34
Summary .....	34
Conclusions .....	35
Recommendations .....	36
Appendix A: Initial Course Requirements .....	38
Appendix B: IDEF0 Tutorial Storyboards .....	41
Appendix C: Flow Chart of Authorware Program .....	58
Appendix D: IDEF0 Workbook and Case Study for use with the IDEF0 Computerized Tutorial .....	65
Appendix E: Summary of Initial Prototype Assessment .....	77
Appendix F: Physical Design Evaluation Questionnaire .....	79
Bibliography .....	83
Vita .....	87

List of Figures

Figure	Page
1. Roblyer/Hall Model for Instructional Design. . . . .	15
2. Typical Prototype Development Model. . . . .	16
3. Combined Model. . . . .	20

List of Tables

Table	Page
1. Evaluation of Prototype Flexibility and Control. ....	29
2. Evaluation of Visual Clarity. ....	30
3. Assessment of Feedback Received. ....	31
4. Evaluation of Instructional Content. ....	32
5. Comparison of CAI to Traditional Instructional Methods. ....	32

Abstract

The goal of this research was to create a computer-assisted instruction (CAI) prototype tutorial to teach the fundamental principles of integrated, computer-aided manufacturing definition activity modeling (IDEF0), a component of business process reengineering within the Department of Defense. To accomplish this, the authors evaluated current instructional material related to IDEF0, then adapted the material to a computerized learning environment using the Authorware Professional™ authoring software and a hybrid methodology which incorporated elements both of instructional design and software development. One of the most notable characteristics of this hybrid model was a series of iterations at each phase of development, which eliminated the need for retroactive modifications at successive development phases.

The objectives for this study were accomplished by completing a literature review, identifying learning objectives and testing strategies, creating a prototype tutorial using authoring software, incorporating a case study, and then operationally testing the prototype. Analysis of this research is limited to a discussion of the results of a qualitative survey provided to evaluators of the prototype. In general, the results of the survey indicated acceptance of CAI as a method of instruction and provided recommendations for improving the tutorial.

ADVANCED EDUCATIONAL METHODS IN THE DEPARTMENT OF DEFENSE:  
APPLICATION OF CASE THEORY AND COMPUTER-ASSISTED INSTRUCTION  
TO BUSINESS PROCESS REENGINEERING

I. Introduction

General Issue

Following the massive build up in national defense during the Reagan presidency, and the subsequent dissolution of the Soviet Union, the Department of Defense and its mission have been the targets of considerable scrutiny. As a result of changes in the national defense strategy, Congress dictated a \$10.5 billion defense budget cut for fiscal year 1993 and a 500,000 personnel force reduction before 1996 (CQ, 1992:3184). With equal taskings and fewer resources with which to perform them, the DoD has therefore been forced to reengineer the way in which it operates (Appleton, 1993:5). By no means a revolutionary idea, this restructuring of DoD operations mirrors the Business Process Reengineering (BPR) initiative which is rapidly expanding within the corporate sector.

The BPR philosophy is based on the principle that our increasingly technical and automated workplace is often hampered by antiquated policies and procedures established to control business practices during the industrial age. Thus, in order to reduce operating costs and improve efficiency, businesses must identify and eliminate unnecessary business practices. BPR stresses that this elimination is especially important with respect to data automation and information management; businesses should process and store only that information which is essential to supporting their business objectives. The Air Force has recognized the potential gains to be made through process reengineering, and has taken steps to enforce its implementation. According to Air Force's Director of Information

Management, BPR studies are now "essential to acquiring funding for future program administration" (Info. Manager, 1993:3).

### Background

In 1989, as a result of the Goldwater-Nichols Act which dictated a shift in defense strategy towards a composite fighting structure, the Deputy Secretary of Defense at that time set a team of consultants to the task of developing guidelines to integrate the information infrastructures of all branches of the military (Strassman, 1992). The Office of the Assistant Secretary of Defense delegated overall responsibility for the project to the Director of Defense Information (DDI). In turn, the DDI chartered the Defense Information Services Agency (DISA), and as an off-shoot of that, the Corporate Information Management (CIM) initiative (Appleton, 1993:4). CIM has two guiding missions: to develop long term, fully-interoperable information management practices, and to use information technology to assist in cost reduction tasks (Appleton, 1993:5). As stated in the previous section, the DDI believes neither goal can be accomplished without first questioning the military's current business practices and administering BPR (Strassman: 1992).

Once the decision had been made to incorporate BPR theory, the DoD moved quickly to adopt a standard methodology. It was decided to incorporate several BPR tools already in use throughout various organizations within DoD, such as Activity Based Costing (ABC), Functional Economic Analysis (FEA), and IDEF modeling (Appleton, 1993:6). The two former applications are essential elements of BPI efforts within CIM (Appleton, 1993:11). However, the focus of this thesis to explore the third element, IDEF modeling.

Integrated Computer Aided Manufacturing Definition Language, or IDEF, is a modeling technique first developed by the Air Force in the 1970s to maximize productivity

in manufacturing (Appleton, 1993:10) and now is the mandatory standard to use in modeling business processes as directed by the DoD CIM Information Technology Policy Board (Appleton, 1993:62). Its basic philosophy employs an in-depth study, or model, and analysis of a given process to identify and eliminate unnecessary and resource-intensive practices. IDEF provided the common language and symbolism for making these models (Appleton, 1993:10). Although there are IDEF models available for every aspect of a BPR project, the remainder of this study will focus on one particular IDEF strategy known as IDEF0, or process modeling.

### Specific Research Goal

The purpose of this study is to develop an IDEF0 training prototype using Computer-Assisted Instruction (CAI) technology and current IDEF0 instructional materials, and then to evaluate the tutorial to determine the effectiveness of using this non-traditional teaching method.

### Research Objectives

To accomplish our specific research goals, the following objectives must be met:

1. Evaluate current IDEF0 instructional material and adapt the material to a computer-based instructional format.
2. Demonstrate the advantages/disadvantages of teaching activity modeling using CAI versus traditional teaching methods.

### End Product

Once the prototype has been completed, it will be tested to evaluate its effectiveness as a medium for teaching IDEF0. The resulting product will in small part indicate some of the advantages and disadvantages to using computer-based instruction versus more traditional teaching methods.

When completed, the main product will be in the form of a PC-based software program consisting of two basic components, an instructional program and an industry case. The instructional section will present the underlying framework and strategies of BPR and a tutorial in the modeling symbolism. As well, it will provide some measure of the individual student's comprehension of these concepts. The industry case will provide the student with a real-life scenario which illustrates process modeling techniques and to which the newly-learned concepts may be applied.

### Limitations

There are three primary factors which act to limit the goals of this study. First, our research is limited by the uncertainty of our user population. At the completion of this research, we will forward our prototype to DISA for evaluation. They will distribute the program at their discretion. Accordingly, it is difficult to anticipate the level of education, prior experience, and current environment of the program users, and to develop the prototype to meet their needs.

Technological limitations present a second limitation to our research. Although we are developing our prototype with software designed for technologically advanced computers, we realize the end users will likely have less-advanced systems which are unable to support many of the functions the software is capable of performing. Thus, we must target our applications to the capabilities of a very basic computer set-up in lieu of an elaborate, technologically-advanced program.

Finally, our prototype is restricted to the use of a case study to re-enforce the concepts the program has presented. While in theory this may be an effective tool for aiding retention, our case study is designed for easy modeling. We can in no way predict or prepare the user for the possible complications of process modeling in a real BPR study.

### Contributions of the Research

In addition to creating an instructional prototype, it is hoped our research will yield other benefits. A better understanding of BPR principles and a critical examination of the IDEF0 process modeling course material may reveal an alternate, more effective method for teaching this concept. The physical prototype we develop will certainly provide valuable information on the use of computers in instructional programming, and the data we collect during validation may in some small way bridge the gap between educational theory and computer programming in CAI. As well, it is hoped our research will raise additional questions, thereby encouraging spin-off research efforts.

## II. Literature Review

### Overview

The goal of this research is to develop and evaluate a computer-assisted instruction prototype to teach the fundamental principles of IDEF0 activity modeling, a part of DoD's Business Process Reengineering methodology. Accordingly, this chapter provides a research base from which to design and develop the CAI prototype. We begin with a summary of the information currently available regarding BPR, its history, an explanation of several important BPR principles, and the advantages and disadvantages associated with reengineering efforts. This summary fulfills our first research objective by providing a context in which IDEF can be understood. The second section describes the background of CAI, the elements by which CAI programs are evaluated, and a discussion of the advantages and disadvantages associated with implementing a CAI system. The third section contains a brief overview of the current educational theories of learning--behaviorism, cognitivism, and constructivism--and issues regarding their use as a framework for CAI programs. The final section of this chapter discusses prototyping and systems development, as well as the benefits and risks associated with prototyping.

### Business Process Reengineering

Business Process Reengineering Defined. Reengineering is the "analysis and design of work flows and processes within and between organizations" (Davenport, 1990:11). Due to the rapidly changing business environment of the 1990s; increased foreign competition, changes in business technologies, and massive changes in consumer expectations; an understanding of reengineering, its applications, and its potential is essential for today's business managers. (Grover, 1993:433). The evaluation and improvement of business processes is generally termed reengineering; however, it has also been called business process redesign and business process improvement (Grover, 1993:433). Regardless of

the terminology, reengineering is a fundamental change in the way businesses look at improving their effectiveness and efficiency. In addition to eliminating inefficient practices, reengineering emphasizes "radically redesign[ing] our business processes in order to achieve dramatic improvements in performance" (Hammer, 1993:4).

History of Reengineering. In 1990, Michael Hammer introduced the idea of business process reengineering in his article, "Reengineering Work: Don't Automate, Obliterate" (Hammer, 1990:104). In this article, Hammer emphasized the need to improve entire processes and to use emerging information technologies to compliment the redesigned processes. According to Hammer, the design of today's businesses can be traced back to Adam Smith's book, *The Wealth of Nations*, which was published in 1776 (Hammer, 1993:11). Smith was the first to put forth the idea of decomposing industrial work into simple, basic tasks which could be completed by one individual (Hammer, 1993:2). In the early 1900s, Frederick Taylor further applied the idea of decomposition of tasks to redesign basic business organizational structure. Taylor's method was to reduce the number of tasks a worker would be required to perform, and then structure the organization into separate work areas for persons doing similar tasks (Davenport, 1990:11). The result of Smith's and Taylor's work was the traditional stovepipe organization. In this type of organization, workers completing a specific task may be isolated from those completing another task. Thus, in initiating an reengineering effort, there is a tendency to focus only on the tasks performed within a particular work area, and to ignore the overall process in the context of the organization.

Principles of Reengineering. Reengineering entails questioning traditional assumptions, destroying the fiefdoms which are typical of traditional organizational structures, and reinventing processes (Corbin,1993:26). The underlying idea behind reengineering is that traditional process structures are "the products of accretion--that is, work methods designed, added to, tweaked, and reconfigured over dozens, sometimes hundreds of years"

(Gulden, 1992:10). Addressing this problem, Hammer states that reengineering requires asking the question: "If I were re-creating this company today, given what I know, and given current technology, what would it look like?" (Hammer, 1993:31).

Advantages and Disadvantages of Reengineering. The potential gains achieved from successful reengineering can be extraordinary in terms of speed, productivity, and profitability (Stewart, 1993:41). Many businesses have undertaken reengineering projects, with varying amounts of success. For instance, Ford Motor Company reengineered its accounts payable department in an effort to cut overhead costs. The result was a 75 percent reduction in personnel while at the same time achieving a dramatic reduction in administrative errors (Hammer, 1993:39-42).

However, not every business which implements a reengineering effort achieves such success. In fact, by one estimate, between 50 and 70 percent of all reengineering projects fail to achieve their goals (Stewart, 1993:41). Difficult enough in the private sector, reengineering can be even more difficult applied in the public sector. Jerry Mechling, director of the program on public-sector computing and telecommunications at Harvard University's Kennedy School of Government, states that tenfold improvements in productivity, as opposed to small changes, are difficult to accomplish in the government. John Randolph, former chief information officer and executive director of the Canadian ministry's information technology division, also argued that many agencies in the public sector are unmotivated to make drastic changes in their work processes for fear of change (Corbin, 1993:32). Reengineering efforts, whether in the public or private sector, may not always be needed and are usually accompanied by stress. Reengineering often includes "downsizing". Indeed, "unions view reengineering as a euphemism for layoffs, and although downsizing is not the goal in every case, reengineering can result in the elimination of positions" (Corbin, 1993:32). Reengineering efforts can also be expensive and a source of problems within the affected organization (Stewart, 1993:42). Hammer

states that "the strain of implementing a reengineering plan cannot be overestimated. But, by the same token, it is hard to overestimate the opportunities" (Hammer, 1990:112).

With this background on the importance of BPR and IDEF, we will now discuss various aspects of computer-assisted instruction.

### Computer-Assisted Instruction

The Use of Computers As Teaching Tools. The use of computers as teaching devices is hardly a new contrivance. They have been used as teaching machines since 1958, when IBM and the University of Illinois began experimenting with the presentation of educational material via a computer medium versus the traditional lecture format (Ralston, 1992:264). Now, with the advent of the affordably-priced personal computer (PC), the number of computers present in classrooms is increasing dramatically (AERA, 1991:2). A 1991 survey of American elementary and secondary schools revealed a student to computer ratio of 20:1, nearly six times the number of computers available five years earlier (Ely, 1992: 21). With the increased accessibility to computers, it is no wonder that educators feel compelled to incorporate the use of computers in their curriculums. These teachers have found additional support from President Clinton, who recently named the development of educational technology, software, and computerized teaching systems as one of six initiatives essential to America's economic growth (Clinton, 1993:35). Thus, computer-assisted instruction has become an increasingly popular alternative method of instruction. In this discussion, computer-assisted instruction will be defined generically as the use of an author-generated computer medium to present educational subject material and evaluate the user's comprehension of that material (Ralston, 1992:264).

Educators' Reluctance To Use CAI. For all its ubiquity and seeming importance, educators are apprehensive about using computers as a teaching tool. Many educators are dissatisfied with the quality of the available courseware (Roblyer, 1988:9). At first, the

computerized educational programs were designed to present material in a strictly linear manner. This format required minimal student interaction and thus helped spawn negative impressions about CAI while leading to its reputation as nothing more than an "automated page turner" (Golub, 1983). Still others contend that a lack of advanced technology was responsible for the poor structure of early CAI programs (Cooper, 1993:14). It has also been speculated that inferior CAI programs were in part the result of poor collaboration between educational theorists and computer scientists, resulting in both a slow and costly development process (Woolf, 1992:50). Commercially-produced CAI programs fared no better, with teachers citing the absence of evaluation and testing elements in some packages, generally poor articulation of course objectives, and a disparity between learner ability and content difficulty as problems with the individual programs (Roblyer, 1988:10). Unfortunately, these poor quality -- and therefore little used -- educational programs only served to fuel educators' reluctance to incorporate CAI (Maddux, 1992:7). However, the recent introduction of authoring software makes possible the relatively rapid development of instructional programs by educators themselves, without the outside consultation of programmers (Maddux, 1992:8). Still other teachers feel threatened that this technology is meant to replace, not assist, them (Ralston, 1992:268). Researchers have attributed this attitude to the introduction of technology without providing the educators with adequate preparatory information and training (AERA, 1991:6). Thus, it is hypothesized that many educators have rejected the use of computers out of ignorance of the basic tenets of computer-assisted instruction and its potential benefits (Ralston, 1992:268).

CAI Terms. Interestingly, different applications of computerized teaching tools are in part reflected in a number of synonyms for the more generic term, computer-assisted instruction. For instance, the terms computer-aided learning (CAL) and computer-based training (CBT) both refer to educational programs geared toward the individual student, but they are differentiated by the emphasis CAL places on learner-initiated and -structured

program flow, whereas CBT is more a learner-independent, sequentially-structured program (Ralston, 1992:264). As mentioned previously, the term, computer-assisted instruction, though used generically to represent all forms of computerized instruction, technically refers to those programs developed and structured by a teacher-author. Thus, these programs emphasize the organization and thoroughness teachers tend to value in computerized instruction (Ralston, 1992:265). Finally, the term, computer-assisted education (CAE), refers to computer systems designed to help educators in administrating and managing educational activities. Although not specifically a teaching tool, CAE is often erroneously associated as a synonym for CAI (Ralston, 1992:264).

“Tutorial” is a term applied to the specific method of computerized instruction in which learning is stimulated by combining the presentation of material with appropriate practice items, and then providing individualized feedback in response to the student’s performance on the practice items (Maddux, 1992: 9). By this definition, computerized tutorials are necessarily interactive; that is, the program elicits student input, assesses that input, and adapts its response accordingly (Roblyer, 1988:11). Because tutorials allow for student interaction of varying degrees, it may be said that tutorials may exist as CAL, CBT, and CAI programs.

Interface. While there is undoubtedly a need to consider theories of cognition when deciding what instructional material to include within a CAI program, it is equally important to consider how the information will be presented. The term, interface, refers to the junction between two or more devices; in this case, the computer’s audio and visual interaction with the user (AECT, 1979:221). The interface is the sole means by which a user and computer interact. In fact, one researcher has observed that "interaction... is at times the most important feature of instructional computing software" (Hazen, 1985:18). In a study conducted by Ravden and Johnson, and reported by Ravden, the importance of visual clarity, consistency, flexibility and control, user control, and error prevention and

correction considerations when developing computerized instruction programs are emphasized (Ravden, 1989:30). Other researchers have attempted to determine other factors such as optimal placement of text on a screen, amount of text presented on a screen, and use of colors in highlighting concepts (Aspillaga, 1991: 89). Currently, however, there is no generally-accepted methodology for ensuring an effective human-computer interface.

Elements of Effective CAI Systems. There are a number of criteria by which the effectiveness of a CAI system is judged, depending on the perspective of the rater (Ralston, 1992:264). For the student, the utility of CAI is gauged by the characteristics of the particular program with which he or she is interacting. General concerns seem to center on how well the computer instructor emulates its human counterpart--whether the system is perceived to be non-judgmental in its feedback, if the feedback appears to adapt to the student's strengths and weaknesses, and how accurately the program assesses the student's responses (Ralston, 1992:264). On the other hand, the value a teacher places on a particular CAI system is dependent upon the organization and thoroughness with which the material is presented (Ralston, 1992:265). Other concerns are how well the program adapts to the different learning styles of individual students and how actively the program involves the student in the learning process. Instructors evaluate CAI systems on how well they do what they are supposed to--their impact in improving academic performance (Apple, 1991:15). In addition to assessing the effectiveness of CAI in general, administrators must also consider the economic costs of developing and maintaining CAI systems (Ralston, 1992:265).

Advantages and Disadvantages of CAI. In spite of what they perceive as a threat to their positions (AERA, 1991:6), teachers generally acknowledge several advantages unique to CAI. First, computer technology permits the student to conduct simulations by accelerating the speed of processes (Ralston, 1992:265). As well, the flexibility of input

devices, user-control over the pace of instruction, and adaptability of screen display are well suited to individuals with handicaps (AERA, 1991:8). Additionally, CAI has been shown to enhance learning in some situations. A study sponsored by the Air Force Office of Scientific Research provided evidence that group activity conducted in a CAI environment had a more positive impact on learning than similar tasks undertaken by students in a traditional classroom setting (Stephenson, 1992:22). Further, the ability to control instruction both in terms of content and pacing was shown to be a motivating factor for students (Milheim, 1991:104).

A survey conducted by the American Educational Research Association (AERA) indicates a shortage of high quality software as a major barrier to CAI use within the classroom (AERA, 1991:6). Criticisms of a similar nature claim that current technology is inadequate for designing truly effective programs (Ralston, 1992:266). Conversely, others blame ineffective CAI not on the technology, but on an undeveloped theory of learning. Proponents of this view contend that in attempting to facilitate learning, CAI is basing its design on a process educators do not yet understand (Woolf, 1992:50). Still others discredit CAI by citing the importance of instructor guidance and the essential role of socializing during instruction, claiming the computer cannot simulate this environment (Stephenson, 1992:25).

Factors Which Contribute To Poor Quality CAI Programs. The poor quality of professionally developed CAI is attributable to a number of factors. First, there is a healthy demand for computerized learning programs of any type. CAI developers exploit the educational community by offering programs with all the 'bells and whistles' state-of-the-art technology can provide, without benefit of a systematic development of the course material (Maddux, 1992:8). Donald P. Ely, an educational researcher, sums up the situation with his observation, "Educational technology continues to be perceived as a field concerned more with hardware and software than with its applications for teaching

and learning (Ely, 1992:43). However, CAI developers are not solely responsible for the proliferation of poor quality instructional programs. Educators have yet to develop reliable evaluation measures for comparing the relative quality of competing courseware, and often rely on subjective measures when choosing which programs to purchase (Ely, 1992:27).

Yet another interpretation of what causes poorly-developed CAI is a lack of coordination between the educators who develop the curriculum and the programmers who translate the curriculum into computer code. In professionally developed programs, the failure of educators and CAI programmers to achieve a common vision of the goals, presentation, and future application of proposed courseware leads to dissatisfaction with the end product (Maddux,1992:8). Of course, CAI which does not meet the needs or expectations of the educator stands little chance of being utilized, resulting in a waste of resources in terms of time, money, and instruction potential (Roblyer, 1988:9).

One final explanation for the inferior quality of available CAI is an apparent lack of consensus among educational psychologists concerning how humans learn (Woolf, 1992:44). Because most current CAI applications are incapable of intelligent thinking, it is essential that educators be able to anticipate the full range of student inputs and identify appropriate responses to be included in the programming. The development of a single theory of learning would go far in assisting educators to predict student responses, and would provide the additional benefit of helping to determine which testing and learning strategies are best suited for material which is presented via the computer medium (Woolf, 1992: 62).

#### Roblyer/Hall Model For Instructional Design

M.D. Roblyer and K.A. Hall, two professors at Florida A&M University, have noted problems with CAI programs similar to those just mentioned. To combat what they

cite as an incomplete or hasty design process, Roblyer and Hall (1985) created a three phase model for designing CAI programs:

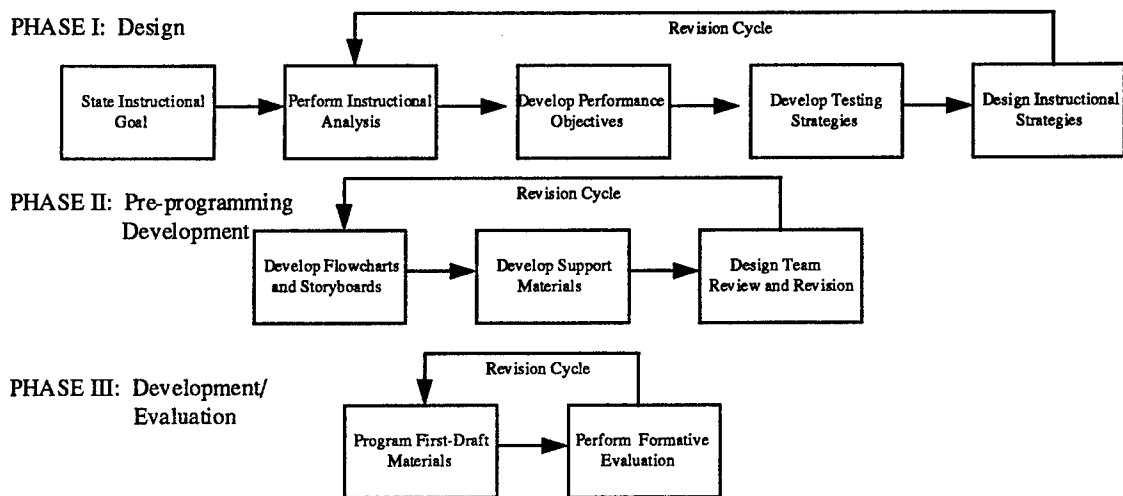


Figure 1. Roblyer/Hall Model for Instructional Design

The design phase includes stating instructional goals, performing instructional analysis, and developing performance objectives, testing and instructional strategies. The second phase incorporates flowcharting, the development of support materials, and review/revision by a design team prior to programming. The final phase includes the actual programming, plus a formative revision cycle before implementation (Roblyer, 1988). This model, along with the prototyping model presented in the next section, provide the elements for the combined model employed in chapter three of this thesis.

### Prototyping

Basic Prototyping Model. A prototype is generally recognized as a quickly developed, preliminary working version of a specific computer application (Zwass, 1992:720).

Prototypes are designed to permit testing and modification of the application throughout the development process.

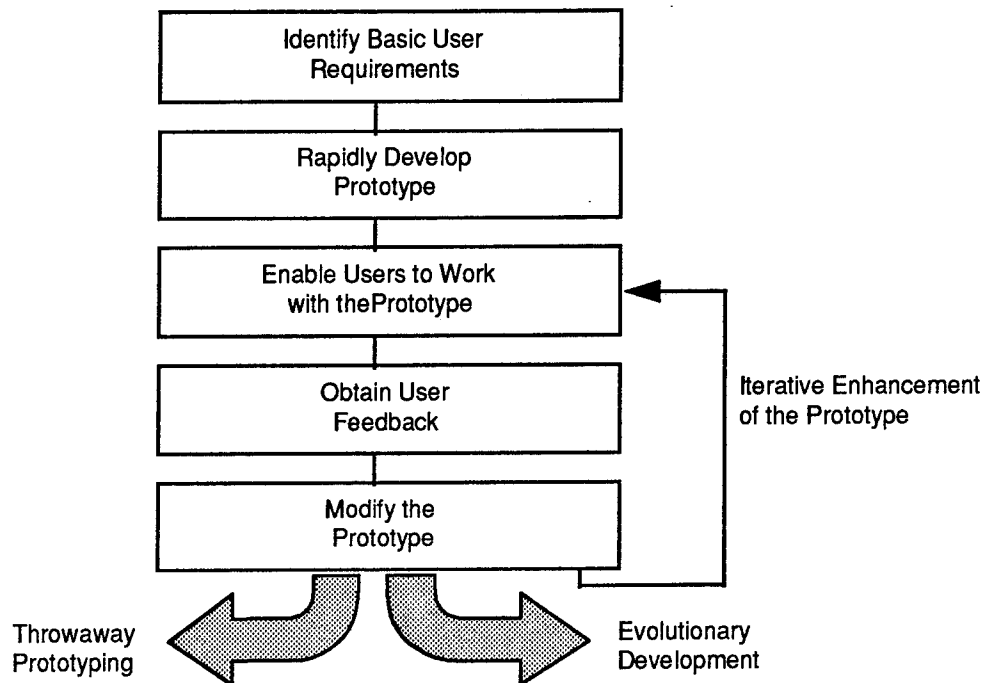


Figure 2. Typical Prototype Development Model

There is no single model that can be used to depict the rapid prototyping approach to software development. Numerous models abound and sometimes even conflict (Klingler, 1986:131). A typical model, however, includes the five phases shown in Figure 2.

Following this model, there are two possible outputs from this development: throwaway prototypes and evolutionary development. Throwaway prototypes are discarded after testing, although the conceptual models and designs formulated by the prototype can be adapted for use in the final system (Ralston, 1993:1241). Evolutionary development, on the other hand, incorporates the actual prototype into its final product (Klingler, 1986:131).

Prototyping Versus Systems Development Life Cycle. Most software systems are developed according to the traditional software development life cycle, which is distinguished by the following, clearly-defined steps (Zwass, 1992: 720):

- 1) Feasibility Study
- 2) Requirements Analysis
- 3) Logical Design
- 4) Physical Design
- 5) Coding and Testing
- 6) Conversion
- 7) Post-implementation Audit

Whereas traditional software development tends to be an inflexible, lengthy process, rapid prototyping combines steps 2 through 4 quickly and provides flexibility and speed which many organizations depend on to get their systems on-line (Zwass, 1992:742). Prototypes are not subject to extensive requirements analysis because the users are involved in defining and refining the requirements at all stages of prototype development (Luqi, 1989:13).

Advantages and Disadvantages of Rapid Prototyping. There are several benefits to rapid prototyping. One major advantage to prototyping is that these scaled-down models are considerably less expensive to build (Ralston, 1992: 1241). As well, prototyping reduces the risk of developing an ineffective system in that it is continually being tested and critiqued by its users (Harker, 1988: 420). On the other hand, according to Phillip Kaufman, traditional development systems,

*can't be adequately verified....with simulation alone. A product must be fully exercised and perform all its intended functions....before we can say the development task is done.*

The benefits associated with user participation during prototyping are offset by the risk involved in allowing the end user to determine requirements (Tillman, 1989:42). Especially in bigger organizations, users lack a global view of the system and its intended

use. As a result, prototypes get designed around individual preferences and self-determined goals (Tillman, 1989:42). The traditional systems development process overcomes this problem in the requirements analysis stage. Yet another risk associated with prototyping, or any systems implementation, is a resistance to change on the part of the users (Tillman, 1989:43). No prototyping can be successful without the help of its users. Similarly, no system can be effective if it is not used. Finally, because prototypes are recognized as both quick and easy to develop, systems are sometimes designed and built with little forethought. As a result, systems lack documentation, which complicates development of an end system, and managers are hesitant to lend necessary support (Guimaraes, 1987:102).

### Conclusion

This chapter provides a research base from which to design and develop a CAI prototype. We first looked at the history and principles of BPR, providing a context in which to place IDEF. Next, we reviewed the background and principles of CAI, and some common causes of poor quality CAI programs. Finally, we looked at prototyping and systems development, which provides a structure in which to develop our CAI program.

### III. Methodology

#### Overview

The research areas of computerized instruction and software prototype design have developed independently and have established one or more models for producing a computerized instruction program or a prototype. However, this research effort required a model which incorporated the specific requirements of both instructional design theory and software development to ensure the resulting product was both instructionally and technologically sound. The following model, a combination of the instructional design and rapid prototype development models discussed in the previous chapter, effectively incorporates the critical aspects of each individual model while satisfying the requirements for both of this research's objectives, namely:

1. Evaluate current IDEF0 instructional material and adapt the material to a computer-assisted instructional format.
2. Demonstrate the advantages/disadvantages of teaching activity modeling using CAI versus traditional teaching methods.

#### Justification For Combining Models

The combined model is a blending of the two individual models. And while each is comprehensive and effective in its own area, neither model addresses the unique concerns of instructional courseware. For example, the courseware design model created by Robyler and Hall (1985) takes into account defining course objectives, designing test and learning strategies, and the presentation order of the instructional material, but fails to consider issues such as technological limitations. Similarly, the software prototype development model provides for continual testing and modification of the prototype, but does not address the programming structure required for interactive applications. Additionally, neither model is adapted to the unique environment of CAI programs.

## The New Model

The model below consists of four main development phases: Requirements Definition, Instructional Development, Programming, and Testing and Modification.

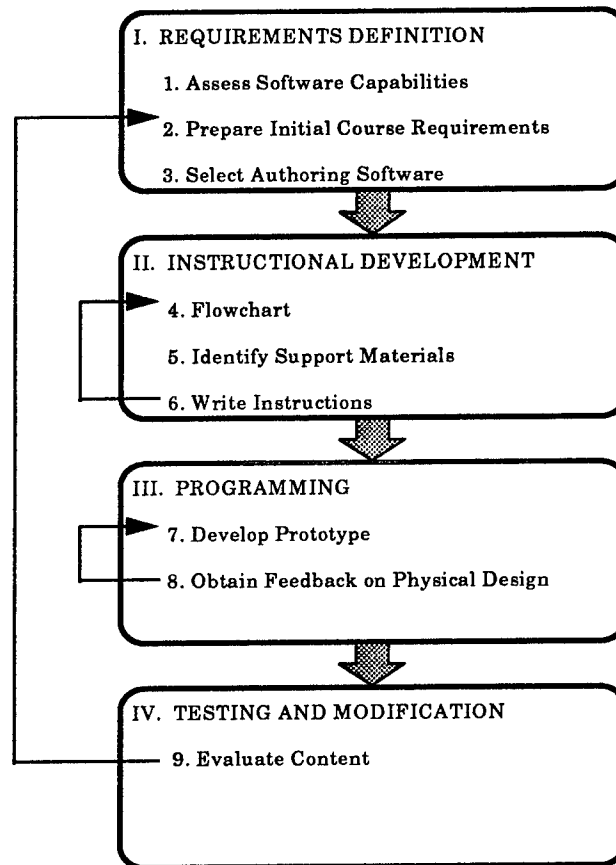


Figure 3. Combined Model

What differentiates this model from similar courseware development models is its consideration of technological factors concurrent with the instructional design of the program. As well, the prototyping nature of the development forces an iterative pattern of testing and modification which aids in validating both the instructional interface and content completeness. This ensures the resulting courseware is both technologically and educationally sound. The specific phases of development are explained more fully in the sections that follow.

### Phase 1: Requirements Definition

The first phase in computerized instruction development is defining instructional goals and selecting a complimentary software application in which to program. This ensures a clear understanding of exactly what learning strategies can and cannot be supported with the available software. This was the primary criteria employed in selecting the authoring software application for the IDEF0 prototype. Course objectives and software requirements were defined in this way:

Assess Software Capabilities. To complete this step, the available literature on software capability was surveyed for text support; required memory size; animation, video or audio capabilities; platform (PC or Macintosh); icon-based or code-generated programming; and importability from other applications, as well as the repertoire of testing and learning strategies which were supported. As well, it was desired to use a software application that was capable of recording student performance in the various testing measures.

Design Initial Course Requirements. We followed the steps specified in the Robyler/Hall model to design the course requirements. Specifically, the steps followed were:

1. State the instructional goals in terms of observable student behaviors.
2. Perform instructional analysis of the proposed course material.
3. Develop performance objectives for what the instruction will enable to student to do and under what conditions.
4. Develop testing strategies which correspond to the performance objectives.
5. Design instructional strategies that reinforce performance objectives.

The course requirements for the proposed IDEF0 tutorial are found in Appendix A.

Select Authoring Software. The potential software packages' capabilities were mapped to the proposed instructional environment and testing/instructional strategies. In this case, Authorware Professional™ was selected based on its ability to fulfill the initial course requirements.

### Phase 2: Instructional Development

Once the performance objectives, testing strategies and learning strategies for the tutorial had been clarified, it was necessary to fill in the framework by determining a sequence of instruction. The decision needed to be made whether or not to include support materials with the tutorial and, if so, what they would contain. Unlike traditional instructional development, our computer-based tutorial demanded consideration for the interactive aspects of instruction flow. Similarly, because the tutorial was designed to operate independent of teacher facilitation, it necessitated the use of written instructions to elicit students' inputs and help them navigate through the program. The instructional sequence and development of support materials were developed in the following manner:

Flowcharts. Storyboards were developed to exhibit the overall organization of the tutorial. Though it was premature actually to design the individual screens at this time, it was helpful to have a general idea of what information would be presented in each segment. The icon-based nature of our authoring software was in itself a flowchart which would flesh out the storyboards and detail the program's structure. These flowcharts helped identify the looping structures to be followed if a student response caused a particular segment to be repeated. Appendix B contains our storyboards, while Appendix C contains a copy of our proposed flowchart along with explanations of the symbols and logic we employed.

Identify Support Materials. Support materials were developed in order to reinforce concepts within the program. In this case, the incorporation of a case study necessitated the inclusion of a workbook, which is contained in Appendix D.

Instructions. Instructions were written to support the program flow. This included prompts for student input, instructions on when to incorporate support materials, and guidance on how to proceed. Where possible, options were provided to empower the student to determine small sequences of instruction. As well, it was decided to provide a persistent “Quit” option which would permit the student to exit the program at any time.

### Phase 3: Programming

At this point in the CAI development, there was a well-developed understanding of precisely how the program would be structured, as well as a general idea of what information would be presented in each segment. The next logical step was to begin building screens and connecting them in accordance with the flowchart. Support materials were developed simultaneously with the courseware to ensure continuity of instruction between the two mediums. Specifically, the programming was accomplished using these steps from Zwass’ model for software prototyping:

Develop Prototype. The programming was completed using both the software application determined in Phase 1 of this model and the storyboards developed in Phase 2. The goal was to produce quickly a program which could support each of the learning objectives identified in Phase 1. Superficial changes in content and screen layout would be accomplished at a later time, based on user evaluation and feedback.

Obtain Feedback on User Friendliness. This task was accomplished by exposing students to the courseware prototype. In this case, 17 students in the Graduate Information Resource program were asked to evaluate the program. Specifically, we solicited criticism of the user interface, clarity of the written instructions, design and

incorporation of support materials, and other non-content related items. The feedback received allowed us to modify the courseware to the expectations and preferences of the intended user, but was not intended to assess the effectiveness of the courseware as an instructional instrument. Copies of the survey instrument and feedback received are included in Appendix F. Development of the survey instrument is discussed in the next chapter, as well as an analysis of the feedback received.

#### Phase 4: Testing and Modification

Although the previous stage of development corrected any flaws in the superficial structure of the courseware, it did not address issues of content effectiveness. Formative testing of the actual instructional content was best conducted after design modification had been completed; our rationale was that student error could then be attributed more to poor content presentation than to faulty physical design. This is naturally the most critical and time-consuming phase of the prototype development, as it reflects the thoroughness in completing each of the preceding phases and indicates the potential educational impact of the finished courseware. It consisted of a single step:

Evaluate Content. Content effectiveness would be assessed through learning measures. It was our intention to study test results for signs of a lapse in instructional flow, incomplete or unclear information, or lack of practice measures. However, due to an exceptionally small sample population and the limitations of time, we deferred completion of this step to future research efforts.

As in the previous two phases, testing and evaluation in this phase should be an iterative process. However, in evaluating instructional content, poor design must be traced back to Phase 1 as it necessarily reflects the performance of instructional analysis. If subsequent changes are made at this initial stage in development, those changes will have a trickle down effect to the subsequent development phases. Thus, in essence the

entire development cycle is repeated until the prototype is instructionally and technologically sound.

## IV. Findings and Discussion

### Overview

This research study is driven by two main objectives: to develop a computer-assisted instruction program to teach IDEF0 modeling, and to evaluate the CAI program as a means of teaching this subject, respectively. This chapter addresses how the first of these objectives was accomplished and presents research findings in the form of comments received by users of the program. A discussion of the second objective is deferred until the next chapter.

### Research Objective One Results

To accomplish this objective, the authors followed a methodology which combined both instructional design and prototyping methodologies. The integration of these two disciplines was necessary to adapt existing IDEF0 course material to the computerized learning environment and to program the actual product, a CAI prototype. Four distinct phases comprised this methodology.

Phase 1: Requirements Definition. In this phase of the research, the overall learning objectives for the prototype were defined and a suitable software application was selected based on these objectives. A number of learning objectives had previously been defined for the IDEF0 course material, and required little more than review and modification for use in the CAI environment. Next, learning and testing strategies were identified to serve as a means for selecting an appropriate authoring software. However, the primary requirement for selecting among software packages was cross-platform programming, or the ability to create a program on either a Macintosh or a PC platform but execute interchangeably. This requirement led us to select an authoring software called Authorware Professional™. Not only did this application support cross-platform

execution, it also provided icon-based programming, which reduced tremendously the learning curve associated with this new software.

The requirements definition phase of the methodology provided an additional benefit; namely, a means of reducing the impact of a major research limitation. As mentioned in the first chapter, this research was severely limited by an ambiguously-defined user population. Without information on the level of education, prior experience, or current environment of our target users, it was difficult to structure the instructional material to their needs. As well, we had no knowledge of what hardware was available on individual bases that could support our program. However, by structuring our learning objectives, learning strategies, and testing strategies at a very basic knowledge level, and by providing cross-platform operation, we were able to target the Air Force community in general, thus mitigating much of the negative impact of this limitation.

Phase 2: Instructional Development. The second phase of this methodology provided an outline of the instruction through use of a flow chart and development of support materials, as well as completion of written instructions which clearly guide the user between the two. At first, our flowchart took the form of storyboards, which aided in developing the three-module structure of the course and identifying organizational clues aimed at reinforcing key concepts while simultaneously providing a means of navigating the program. However, as we began the actual programming process, we discovered that the icon-based programming convention used by Authorware Professional™ was in itself a flowcharting tool. Additionally, the icon-based structure allowed us to visualize the looping structures of the program more effectively than the linear, page by page format of storyboarding. Thus, to improve efficiency, we utilized storyboarding to guide the overall structure of the program, but relied on the logic patterns represented by icons to guide the smaller segments of the program. The original storyboards and icon-based flowcharts are contained in Appendix B and Appendix C, respectively.

Phase 3: Programming. The output of this phase is a fully-developed prototype. However, in accordance with the guidelines for prototype development, a program is not complete until it has been tested and modified to meet user expectations. As mentioned previously, user expectations for the IDEF0 prototype were somewhat ambiguous due to the poorly-defined user population. But, by restricting the questions on our survey to only the most general characteristics of the program, we were able to elicit valuable feedback in spite of an undefined user population. The results of the user evaluation are discussed below. The actual survey instrument and summary statistics are provided in Appendix F.

### Evaluation Results

Seventeen graduate students in the Information Resources Management program were asked to evaluate the first module of the prototype in terms of five main areas: flexibility and control, visual clarity, informative feedback, content, and comparison to traditional teaching methods. Each student was given a copy of the prototype on a disk and a copy of the workbook. With the exception of instructions for accessing the program, no other guidance was given. Responses were opinion-oriented, with ratings provided on a sliding scale. The following paragraphs provide a summary of their evaluations.

The demographics of our sample population were homogeneous in terms of education level, and the students generally assessed themselves as knowledgeable in the operation of computers. Further, over half of the students reported having previous experience with computer-assisted instruction programs. Thus, we suspect these students' assessments to carry slightly higher expectations than students with no previous exposure to computerized instruction. As well, it should be noted that one student's response sheet had to be eliminated due to a numbering error. Therefore, percentages are based on a sample size of 16.

Table 1. Evaluation of Prototype Flexibility and Control

Flexibility and Control				
The tutorial allowed me to control the speed at which information was presented.				
1	2	3 (13 %)	4 (31%)	5 (56%)
I could easily navigate through different sections of the tutorial.				
1	2 (13%)	3 (6 %)	4 (50%)	5 (31%)
I could easily enter and exit the tutorial.				
1	2	3 (18%)	4 (44%)	5 (38%)
On re-entering the tutorial, it was easy for me to find where I had left off.				
1	2 (13%)	3 (44%)	4 (25 %)	5 (18%)
I found it awkward to switch back and forth between the tutorial and the workbook.				
1	2 (50 %)	3 (19%)	4 (31%)	5
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

The results of survey questions addressing flexibility and control are provided above. These questions were aimed at assessing how easily students could access the prototype and, once in the program, how readily they could navigate through the various screens, transition to the workbook, and return to the program. With respect to control over the speed at which information was presented, about 85 percent of the students rated their ability to control the speed of instruction on the high end of the scale. The survey yielded similar results with respect to the ease of navigating within the tutorial and entering/exiting the program. However, the respondents indicated frustration at having to transition between the workbook and the prototype, and also at being unable to return immediately to the last screen they viewed before exiting the program. Also, several written comments addressed frustration with overly-sensitive response icons used for checking answers to the workbook exercise.

Table 2. Evaluation of Visual Clarity

Visual Clarity				
The individual instruction screens were cluttered with too much information.				
1 (38 %)	2 (50 %)	3 (6%)	4 (6%)	5
Organizational clues were available to help identify the flow of instruction.				
1	2 (6%)	3 (13%)	4 (6 %)	5 (75%)
I found the use of different colors within the program to be distracting.				
1 (50%)	2 (38%)	3 (13%)	4	5
The contrast between the background color and the text made the screens easy to read.				
1	2	3	4 (56%)	5 (44 %)
I felt the use of colors within the program helped highlight and clarify concept				
1	2	3	4 (56 %)	5 (44 %)
I found it hard to determine the meaning of the icons based on the icon picture				
1 (25 %)	2 (50%)	3 (13 %)	4 (6%)	5 (6%)
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

A summary of responses to questions addressing visual clarity is provided in Table 2. With respect to visual clarity, or the presentation of instructional material via the computer monitor, students appeared to like the use of color to highlight portions of text and key concepts. They generally agreed that individual instruction screens were not cluttered with too much information, but we have no assessment as to whether the students felt space was being wasted by presenting too little information. As well, nearly all of the evaluators felt the organizational clues provided in the program served to identify the flow of instruction. Almost 90 percent of the students were able to identify the functions associated with available hot buttons by the visual representation presented on the button, though one student observed,

*I probably would have had trouble with it if I'd never seen IDEF before.  
The icons look like the models and the reader is expected to know that.*

Questions addressing informative feedback targeted both instructions for operating within the program and feedback given for exercises with computer-assessed answers. A summary of responses is provided in Table 3. Student ratings were slightly less positive

(agreement versus strong agreement) on the clarity and content of written instructions and regarding the correspondence between the reinforcement exercises and the concepts emphasized in the lessons. The feedback for incorrect responses, however, was generally viewed as constructive in explaining the student error. In addition to the ratings and comments in response to the survey, students identified the operational errors of a screen that failed to erase and hot buttons which were not persistent on every screen in the program. The prototype was modified to correct these programming errors.

Table 3. Assessment of Feedback Received

<u>Content</u>				
The information provided by the tutorial adequately covered the subject matter.				
1	2 (13%)	3 (19%)	4(25%)	5 (44 %)
The tutorial provided me with the knowledge I needed to complete the workbook exercises.				
1	2 (6%)	3 (6%)	4 (69 %)	5 (19 %)
The workbook was clearly written.				
1	2 (6%)	3 (6%)	4 (62 %)	5 (25 %)
The workbook scenario helped me apply the concepts presented in the tutorial.				
1	2	3	4 (69%)	5 (31 %)
The information presented on business process reengineering was relevant to my understanding of IDEF-0.				
1	2	3 (6%)	4 (69 %)	5 (25%)
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

Phase 4: Testing and Modification. The final phase of CAI prototype development is an evaluation of the instructional content. A summary of responses addressing instructional content is provided in Table 4. With respect to instructional content, only 69 percent felt the material adequately covered the subject matter. Similarly, 88 percent of the students felt they had enough knowledge to complete the case study exercise, though these ratings were less emphatic. Also, there was general agreement that the case study provided a means of applying concepts presented in the tutorial. All students felt a

discussion of IDEF0 modeling as it relates to the larger goal of business process reengineering was relevant to the understanding of IDEF0.

Table 4. Evaluation of Instructional Content

<u>Content</u>				
The information provided by the tutorial adequately covered the subject matter.				
1	2 (13%)	3 (19%)	4 (25%)	5 (44%)
The tutorial provided me with the knowledge I needed to complete the workbook exercises.				
1	2 (6%)	3 (6%)	4 (69%)	5 (19%)
The workbook was clearly written.				
1	2 (6%)	3 (6%)	4 (62%)	5 (25%)
The workbook scenario helped me apply the concepts presented in the tutorial.				
1	2	3	4 (69%)	5 (31%)
The information presented on business process reengineering was relevant to my understanding of IDEF-0.				
1	2	3 (6%)	4 (69%)	5 (25%)
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

As Table 5 indicates, student reactions to computer-assisted instruction as a form of teaching were very polarized. Seventy percent of the responses indicated interest in taking a course taught solely by means of computerized instruction. One student, however, gave CAI the lowest possible rating and added the comment,

*Personally, I think computer taught information is painful and tedious. It is too easy to forget after you complete the exercise.*

Table 5. Comparison of CAI to Traditional Instructional Methods

<u>Comparison To Traditional Teaching Methods</u>				
I would be interested in taking a course taught solely as a computerized program.				
1 (6%)	2 (25%)	3	4 (38%)	5 (31%)
I would be interested in taking a course taught as a mixture of classroom learning and computerized program.				
1	2	3 (6%)	4 (50%)	5 (44%)
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

## Conclusions

Generally, the ratings and comments received were favorable toward the prototype. Of course, these evaluations must be viewed in the light that the students had previously been exposed to the subject matter. In terms of the user interface, the most constructive data we received were those criticisms contained in the written comments. Based on this feedback, we were able to modify the program to correct most of the items students felt detracted from the program.

One objective not accomplished by the students' evaluations was an analysis of the effectiveness with which the prototype teaches IDEF0 modeling. The following chapter addresses this subject and presents a number of recommendations to improve the existing prototype.

## V. Summary, Conclusions, and Recommendations

### Summary

The underlying purpose of this study was to develop and evaluate a computer-assisted instruction program as an alternative teaching method for IDEF0. The research was clearly defined in the sense that the subject matter and method of instruction -- IDEF0 and computer-assisted instruction, respectively -- were agreed upon very early in the research. However, the authors faced an enormous learning curve in both areas. A thorough literature review and study of the fundamental IDEF0 principles aided in scoping the project further. These efforts resulted in the creation of a development model which integrated elements both of instructional design and software development. Similarly, the development model aided in the identification of learning objectives for the proposed prototype.

A second, necessary step in completing this study was the selection of an appropriate authoring software which would support each of the learning objectives previously identified, yet required little programming knowledge to use. Again, a review of available literature helped us to narrow our search, and a subsequent demonstration of the software capabilities led us to decide upon one particular program. A student evaluation of the resulting tutorial prototype found general acceptance for computerized instruction and provided the authors with many recommendations for how the prototype could be improved. However, due to time limitations, a more extensive evaluation of the prototype effectiveness could not be completed. Follow-on evaluation and modification of the prototype are among several of the recommendations discussed in the last section of this chapter.

## Conclusions

Designing and programming the IDEF0 prototype taught us many things. In some cases, we made decisions and based our actions on the prior research of others. In other cases, we had no choice but to try alternatives until we found a solution. In retrospect, we can see which actions were constructive and which served to frustrate our efforts. In either case, these were perhaps the most valuable lessons of the research effort. Several of the most notable lessons are included below.

Additional Storyboarding Should Be Done Upfront. Originally, we accomplished storyboards to visualize the overall structure of the tutorial. Because of the icon-based nature of the authoring software, and the requirement to arrange icons in a logical sequence during programming, we thought that these icons would serve to flowchart the details of the program. However, in retrospect, the programming would have benefited from more storyboarding in the beginning. That way, we could map priority relationships between key concepts and subsequently arrange the order of instruction to build upon previous knowledge prior to beginning programming. Instead, we found ourselves painted into corners, expecting the student to make connections based on information that had not yet been presented.

Adaptation of Instructional Material. In collecting instructional material for use in the tutorial, we relied heavily on materials that had been developed for an IDEF0 course taught by an instructor in a normal classroom environment. In adapting that material to a computerized environment in which no teacher is available to clarify concepts or answer questions, we found that we had to break the information into simplified concepts and greatly expand the detail involved. We made ample use of visual clues and animation to compensate for the absence of audio clues communicated through an instructor's voice and gestures. Further, in programming exercises to test comprehension, we had to anticipate every possible student response and program appropriate feedback in each case.

This proved especially difficult with our tutorial because of the subjective nature of the material. To overcome this complexity, we often employed testing strategies like matching and true/false, in which a list of possible answers is already provided.

Importance of Software Selection. One aspect of this research we would not change was our selection of authoring software. The decision to program using an icon-based software versus programming code aided immeasurably to the amount of programming we were able to accomplish. Although still requiring extensive use of logic, programming with icons eliminated the petty code conventions which make debugging a lengthy and tedious process. As well, the icons provided a visual representation of the inherent logic of the program flow. Because of this, we found it easier to assimilate strings of logic and, consequently, we found it relatively easier to piece together individual program segments.

### Recommendations

In completing this research, we encountered four issues which would greatly enhance the quality of the IDEF0 prototype, but were too large in scope to be completed as part of this research effort.

Follow-on Evaluation of the Prototype. Our second research objective involved a comparison of computerized instruction and traditional instructional methods; specifically, demonstrating the advantages and disadvantages of utilizing CAI. However, time constraints and a subject-knowledgeable test sample served to confound our efforts. Therefore, we suggest for further study three issues germane to this comparison.

Evaluate the Instructional Effectiveness of the Prototype. As mentioned above, the students selected to evaluate the IDEF0 prototype had previously received instruction in IDEF0 modeling. And while this was convenient in the sense that these students could better assess the completeness of the instructional content, it did present a problem in that

any attempt to evaluate learning would be flawed by previous knowledge. Therefore, we suggest that additional evaluations be conducted using subject-impartial students to validate the instructional content of the prototype.

Demonstrate the Advantages and Disadvantages of CAI. Computerized instruction, as a whole, is a relatively new field of research. Accordingly, there is little definitive research demonstrating the advantages and disadvantages of instituting this method of instruction. In particular, there are the issues of adapting existing instructional material to a computerized format, the problems associated with implementing computerized learning within a classroom environment, and acceptance of this method of instruction by the students. Each of these aspects could be enhanced through additional exploratory research.

Survey of the User Population. Perhaps the severest limitation we encountered in completing this research was not being able to define our user population. Without information on the education level of potential students, or their purpose for learning IDEF0 modeling, it was extremely difficult to define learning objectives, create support materials, or adopt relevant examples to illustrate key concepts. In addition to student information, developing computer-assisted instruction dictates the need for information concerning the technologies available to the user. This consideration drives everything from the use of animation and sound to the individual colors used within the program. Indeed, any large-scale CAI development or implementation would need this information in order to target the correct user population.

## Appendix A - Initial Course Requirements

### Instructional Goals

At the completion of the IDEF-0 tutorial the student will be able to:

1. Recognize IDEF-0 activity modeling as a necessary and useful step in performing business process reengineering within the Air Force.
2. Construct a node tree to the A1.1 level from a given scenario.
3. Construct a context diagram, complete with all associated inputs, controls, outputs, and mechanisms (ICOM)
4. Construct a decomposition diagram to the A1 level, complete with all associated ICOMs
5. Read and interpret properly constructed IDEF-0 models.

### Intermediate Steps/Prerequisite Skills

In order to fulfill the instructional objectives of this tutorial, the student must first demonstrate the ability to:

1. Understand the meaning of a node in the context of IDEF-0
2. Recognize the interdependency among each level of successively more detailed levels of the IDEF-0 model
3. Identify and differentiate between processes and activities in a given situation, and recognize the relationship between these two terms.
4. Label correctly all levels of a node tree.
5. Recognize the various problems associated with too many or too few nodes at any level of the node tree.
6. Limit the scope of an IDEF0 modeling effort.
7. Define and identify inputs, outputs, controls, and mechanisms for a given activity.

8. Effectively represent and label the transformations of inputs and outputs within an activity in both the context and decomposition diagrams.
9. Recognize and apply the technique of tunneling.

### Performance Objectives

1. Construct a node tree according to the information supplied in the case study
2. Label all levels of a node tree
3. Differentiate between and name inputs, outputs, controls and mechanisms according to the information supplied in the case study
4. Construct and label a context diagram according to the information supplied in the case study
5. Transform inputs to outputs and trace these throughout an activity
6. Construct and label a context diagram according to the information supplied in the case study
7. Demonstrate the appropriate context for and technique of tunneling

### Testing Strategies

1. Matching exercises using a “grab and drag” format
2. Fill in the blank
3. Construct and label diagrams, specifically the node tree, context, and decomposition diagrams
4. True/False

## Instructional Strategies

These steps will be followed for each of the four modules, introduction, node tree, context diagram, and decomposition diagram:

- Transition and introduction
- Presentation of basic concepts
- Reinforcement of basic concepts
- Test basic concepts
- Presentation of more advanced concepts
- Reinforcement of advanced concepts
- Test advanced concepts / Application
- Summary / Transition

## Appendix B: IDEF0 Tutorial Storyboards

The second phase of the methodology called for instructional development. In this phase, we adapted the IDEF0 instructional material to a computer-aided instructional program. As explained in Chapter 4, Findings and Discussion, our first efforts included using storyboarding techniques to map the flow of the tutorial. This appendix includes three iterations of storyboarding.

The first three pages (pgs 43-45) show the initial concept of the tutorial. During this phase we chose to develop the tutorial in three modules, which our final program did incorporate. At this point, we were also planning to include a section on analyzing IDEF0 diagrams--how to determine redundant activities, wasteful practices, and other areas for improvement. This area of instruction was eventually scoped out of our program. The idea of hot keys at the top left of the screen also emerged in this storyboarding phase; however, in our original design the hot keys would used to return to previous instruction modules. Because of their intended purpose, we originally designed the keys for each module to be resident only in successive instruction modules.

The next five pages (pgs 46-50) are the second, more detailed attempt at storyboarding. At this time, we increased our projected number of modules to four, including our original three and adding one on the case study. We also determined some of the actual wording that we would use in the tutorial. The bottom slide on page 46 and the top slide on page 47 are presented in the final program almost without rewrite. The "talking heads" on pages 47 and 48 were an initial idea that also survived to be in the final

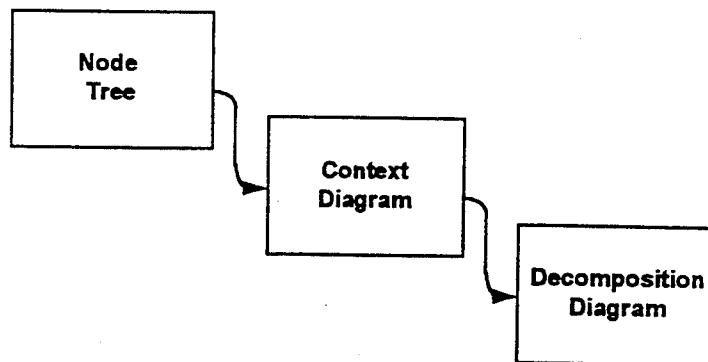
program. The next section of storyboarding shows a more refined view of these heads, while the final, computerized version refined it even further to make use of some of the abilities of the program. The last pages of this storyboard show some very rough ideas for teaching node tree concepts.

The last seven pages of this appendix (pgs 51-57) represent our final storyboards. In this iteration, we introduced the idea of permanent hot buttons that could take the user to the beginning of a module from any point in the tutorial. The number of modules, by this time, had been reduced to the three original, deleting the case study module. In our final program, we added another section to the program called "BPR" (Business Process Reengineering) in order to give the student more background on the use of IDEF0. Also added were permanent exit and continue buttons located on the bottom right of each screen. We later put the exit option on a pull-down menu, and changed the continue button to a "click with the mouse or hit any key" to continue. Our final program eliminated that change as well, and incorporated forward and backward buttons to give the student even more flexibility in navigating the tutorial. The rest of this storyboard is essentially the same as our second phase--we simply refined the ideas and drew them on the computer.

After using these three sets of storyboards to program in Authorware Professional<sup>TM</sup> we discovered that the icon-based programming in Authorware was in itself a flowcharting method, and the extra step of drawing everything on paper did not add to the process of creating a program.

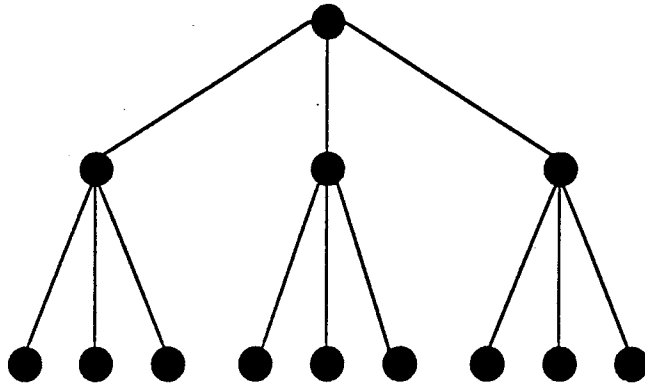
## Overview of IDEF Tutorial

- Intro with bells and whistles—get the student interested in learning by describing BPI and giving a background of IDEF
  - Instructions for tutorial—show slide with overview of program and introduce case study
  - Modules 1, 2, 3—after each module, complete simple exercise and part of the case study
  - Overall case study
  - Analysis of IDEF diagrams
- 



This will be the slide used in the intro to explain to the student what the tutorial consists of and also throughout the program to show the student where he/she is in the tutorial. The boxes will change color as the student completes each module.

MODULE 1

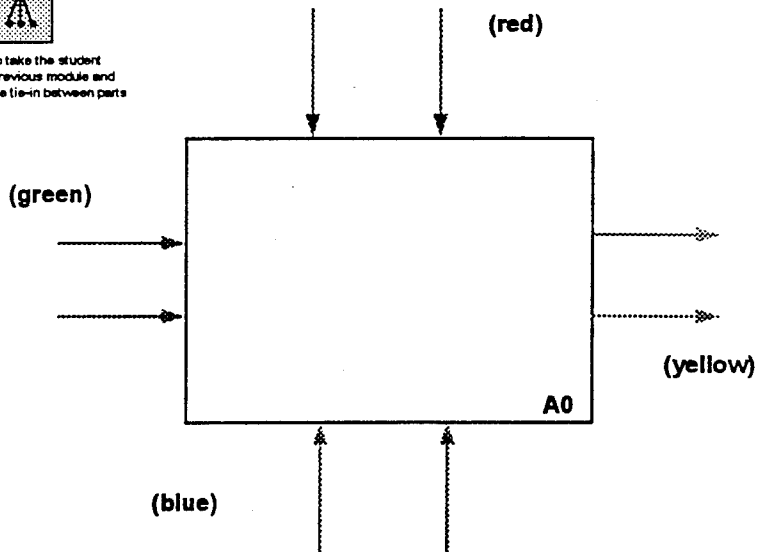


We will use this basic screen to teach about node trees. As each part is discussed, it will be highlighted, and more information put on the screen until it is fully covered.



ICON used to take the student back to the previous module and to assist in the tie-in between parts

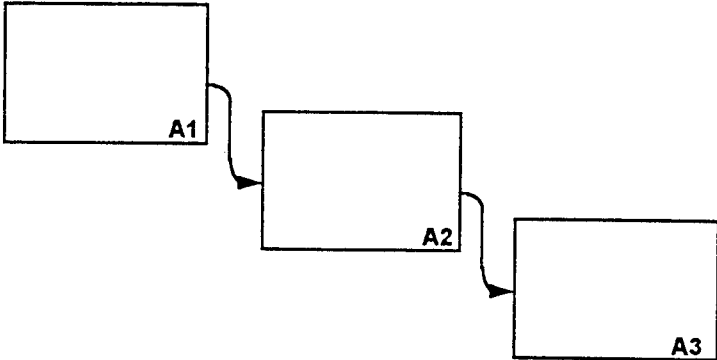
MODULE 2



Basic slide for teaching context diagram. Each area will be highlighted as it is taught. .



ICONS used to take the student back to the previous modules and to assist in the tie-in between parts



Basic slide for teaching decomposition diagrams. Each area will be highlighted as it is taught. .

## OVERVIEW

This tutorial is designed in four sections:

Section 1 - Discussion of ~~process~~ <sup>process modeling</sup> objectives, the use of IDEF as the best approved method of process modeling, and scoping a process through the use of NODE TREES set of standard tools to accomplish

Section 2 - The CONTEXT DIAGRAM

Section 3 - DECOMPOSITION DIAGRAMS

Section 4 - The CASE STUDY  
"Supply Depot"

Welcome to the IDEF tutorial

This program is designed to present a comprehensive instruction in the fundamental concepts of process modeling. Additionally, the program incorporates a case study which illustrates each of the concepts and provides a start-to-finish example of a process model

Continue

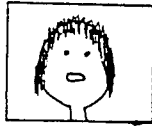
Instruction Time:

Necessary Materials:

Note: This program is designed as a hypertext document. The individual student can control the speed and sequence of instruction by clicking on any button which appears on a particular screen.

Exit button?

②



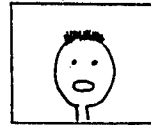
"What is process <sup>①</sup> modeling, and why should we do it?"

### Process Modeling

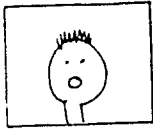
- Fundamental part of Business Process Reengineering (BPR)
- Integrated Computer Aided Manufacturing Definition (ICAD) tools
- Identify non-value added practices
- Achieve more, in less time, with fewer resources

"Process modeling is only a small part of <sup>②</sup> the business process reengineering which helps managers visualize an entire process from start to finish, and across an organization. We use ICAD which stands for <sup>③</sup> Integrated Computer Aided Manufacturing Definition, as a standard set of tools to show the interrelationships among the individual activities within a process. It is important to identify <sup>④</sup> wasteful or unnecessary actions which cost us time, money, and resources -- we call these non-value added practices. By eliminating these non-value added practices, we are "reengineering" ~~the~~ our business processes, and are able to achieve greater results with fewer resources.

Incorporate first part of next slide



④



(process modeling)

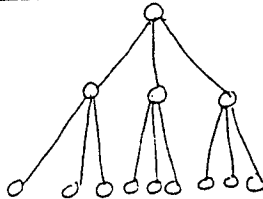
IDEF- $\phi$  methodology is primarily used for understanding the environment -- the functions which are carried out, the relationships between them, and the logical breakdown of those functions into their sub-functions.

There are seven fundamental concepts of IDEF- $\phi$

do we need all seven??  
(what about consistency in definition?)

- 1) You cannot solve a problem if you don't understand it
- 2) Analysis of any problem is Top-Down, Modular, Hierarchical, and Structured
- 3) IDEF- $\phi$  is a functional modeling methodology. It is independent of both organization and time
  - Not an organization chart
  - Not a flow diagram
- 4) IDEF- $\phi$  is a diagramming methodology showing component parts, interrelationships among them, and how they fit into a hierarchical structure
- 5) IDEF- $\phi$  methods support disciplined, coordinated teamwork -- the results reflect the best thinking of a team.
- 6) IDEF- $\phi$  is structured and rigorous -- it follows rules and requires all analysis and design decisions be written.
- 7) IDEF- $\phi$  follows principle of "gradual exposition of detail" ⑥

NS.1



Description  
Purpose

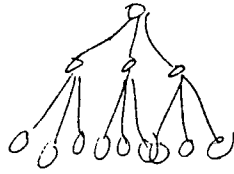
A whole tree is a single diagram that shows the hierarchical breakdown of a ~~process~~ process.

Purpose:

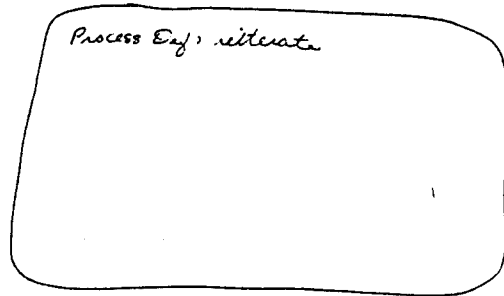
- 1) get ideas down quickly
- 2) represents entire model
- 3) easily show alternative decompositions
- 4) easily communicate a project's scope & depth
- 5) Portray "AS-IS" and "TO-BE" using only 2 pictures

from Teal's slide

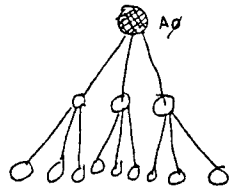
NS. 1  
1.1



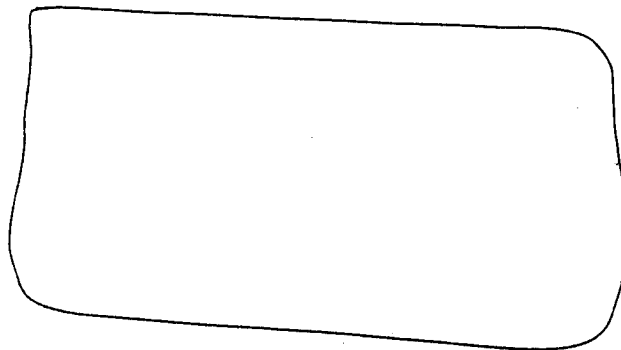
Process  
~~Activity~~ DS  
Process symbol = O = Node



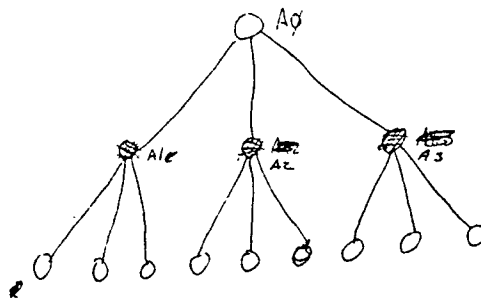
NS. 2



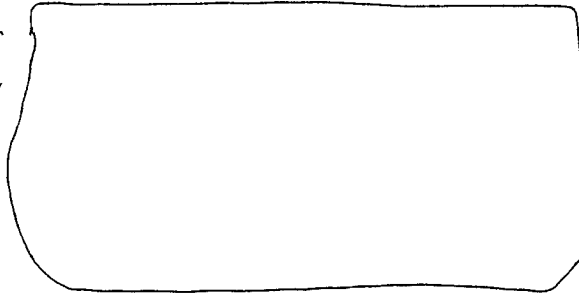
Root Node Activity  
api - mi ~~me~~  
- only me



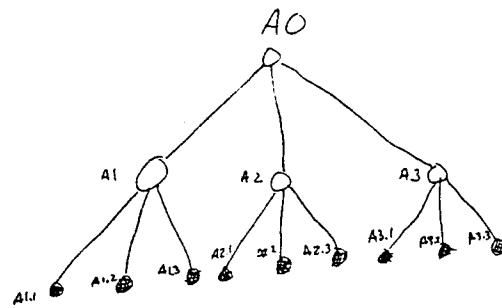
NS.3



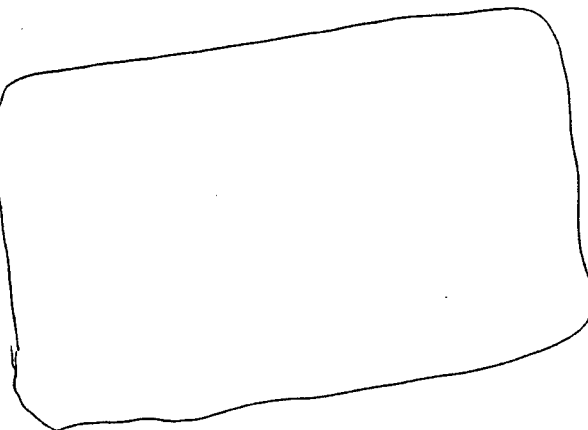
- Subprocesses of A0
- A0 - parent
- A1, A2, A3 - children
- 3 to children of any node
- No naming

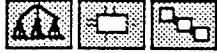


NS.4



- subprocesses of A1 are A1.1, A1.2, A1.3
- children, parents





## Welcome to the IDEF Tutorial

This program is designed to present a comprehensive instruction in the fundamental concepts of process modeling. Additionally, the program incorporates a case study which illustrates each of the concepts and provides a start-to-finish example of a process model.

Exit

Continue

Page #

---



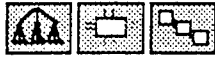
Several "buttons" have been provided for your convenience. The buttons on the upper left of each screen allow you to move directly to the beginning of the different modules from any point in the tutorial.

The buttons on the lower right of the screen allow you to move on to the next screen, or allow you to exit the program at any time.

Exit

Continue

Page #



**Instruction Time:**

**Necessary Materials:**

**Note:** This program is designed as a hypertext document. The individual student can control the speed and sequence of instruction by clicking on any button appears on a particular screen.

**Exit** **Continue**

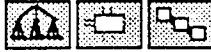
Page #



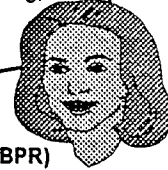
This tutorial consists of three modules, as shown above: the node tree, context diagram, and decomposition diagram modules.

**Exit** **Continue**

Page #



"What is process modeling, and why should we do it?"



### Process Modeling

- 1) fundamental part of Business Process Reengineering (BPR)
- 2) Integrated Computer Aided Manufacturing Definition (IDEF) tools
- 3) Identify non-value added practices
- 4) Achieve more, in less time, with fewer resources



"(1) Process modeling is only a small part of the business reengineering process which helps managers visualize an entire process from start to finish, and across an organization. (2) We use IDEF0, which stands for Integrated Computer Aided Manufacturing Definition, as a standard set of tools to show the inter-relationships among the individual activities within a process. (3) It is important to identify wasteful or unnecessary actions which cost us time, money, and resources—we call these non-value added practices. (4) By eliminating these non-value added practices, we are "reengineering" our business processes and are able to achieve greater results with fewer resources."

Exit

Continue

Page #



### IDEF definition

### fundamentals of IDEF

---

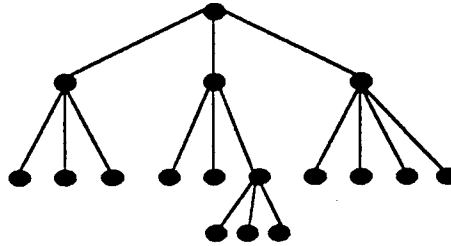
### Overview of IDEF and program

Explain what is/is not a process  
—Give examples of both

Exit

Continue

Page #



A node tree is a single diagram that shows the hierarchical breakdown of a process.

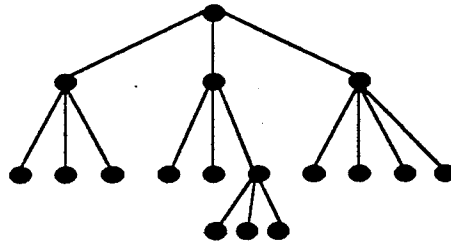
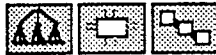
Purpose of a node tree:

- Get ideas down quickly
- Represents the entire model
- Easily show alternative decompositions
- Easily communicate a project's scope and depth
- Portray "AS IS" and "TO BE" using only 2 pictures

Exit

Continue

Page #

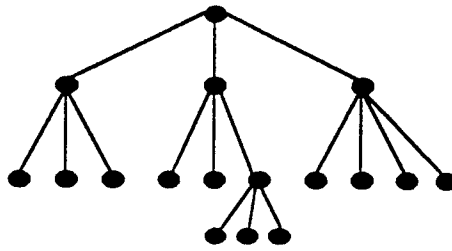
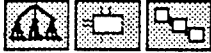


A process is a group of tasks that occur over time with specific results. When developing a node tree of a process, you start with the overall process at the top node, then break it down into its sub-tasks.

Exit

Continue

Page #



For example, the process of xxxxxx would be broken down into xxx and xxx, and then further down into xxx and xxx, and so on.

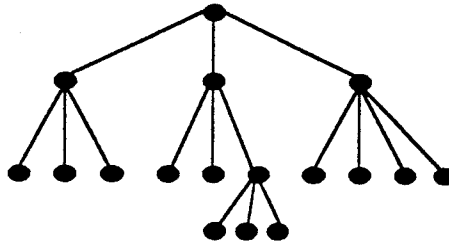
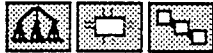
Rule: A node can only be broken down into 3 - 6 subprocesses.

- Any less, and you probably shouldn't break your process down any more than you already have.
- Any more, and you probably have more than one subprocess.

Exit

Continue

Page #

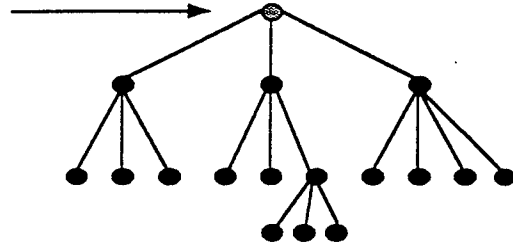
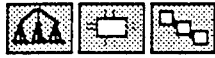


On a node tree, each part of the process is shown as a node "●".

Exit

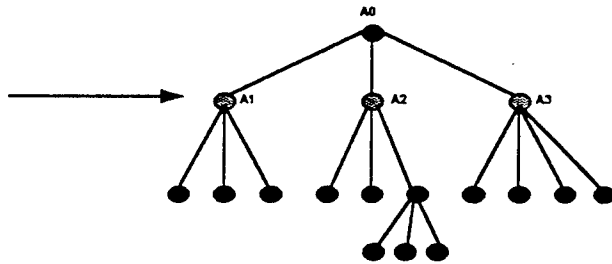
Continue

Page #



The top node is designated as "A0", the main activity.  
There can be only one node at this level. It is called  
the "parent".

Page #

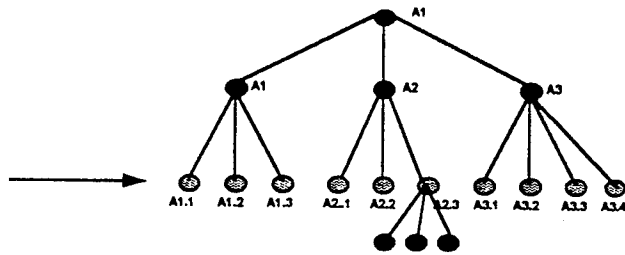
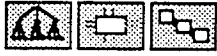


The subprocesses of A0 are called the "children" of A0.

They are designated as "A1, A2, A3, .....An"

They are the "peers" of each other.

Page #



The children of A1, A2,... are designated as "A1.1, A1.2, A2.1, .....An.n". They are considered the "grandchildren" of A0.

The children of A2.3 as shown above would be numbered A2.3.1, A2.3.2, and A2.3.3.

Exit

Continue

Page #

## Appendix C: Flow Chart of Authorware Program

Authorware Professional™ programmers use “object authoring” to create code rather than traditional command lines. This is accomplished by programmers making graphical flow charts displaying the logic of the applications through the use of icons. These icons each represent a different object that can be displayed or moved on screen.

The application is created by dragging icons into a flowline, where they give Authorware instructions to perform during the running of the application. The flowline dictates the order in which to perform the instructions and provides options for users to interact with the program.

Authorware has 11 icons that perform different functions. The different icons allow you to insert text or graphics or choose option settings. Double clicking on the icon, once placed in the flowline, allows the programmer to “open” the icon and edit the contents. The major groupings of icons are:

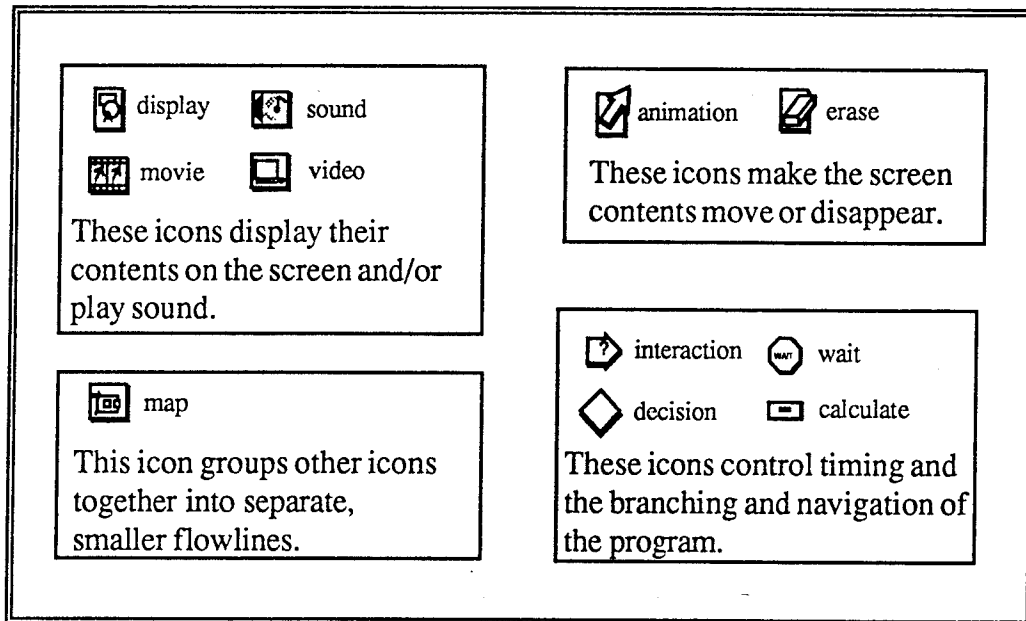


Figure C.1. Authorware Professional™ Icons

The programmer will drag these icons into the flowline, which is presented as a single vertical line in a box marked “Untitled.” Once in the flowline, they can be rearranged,

grouped, cut, pasted, and copied to change the program. No lines of coding are required. This box below represents what the programmer sees on the screen as an new flowline.

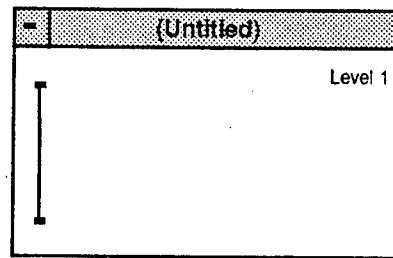


Figure C.2. New Flowline

On the following pages, parts of the flow of the IDEF0 tutorial created for this thesis will be described to document the logic flow of the program. Since many of the sections used the same flow of logic, with only the contents changing, not all of the program will be shown.

The following diagram shows a hierarchy of the program that is described (the grey boxes are the areas detailed later in this appendix).

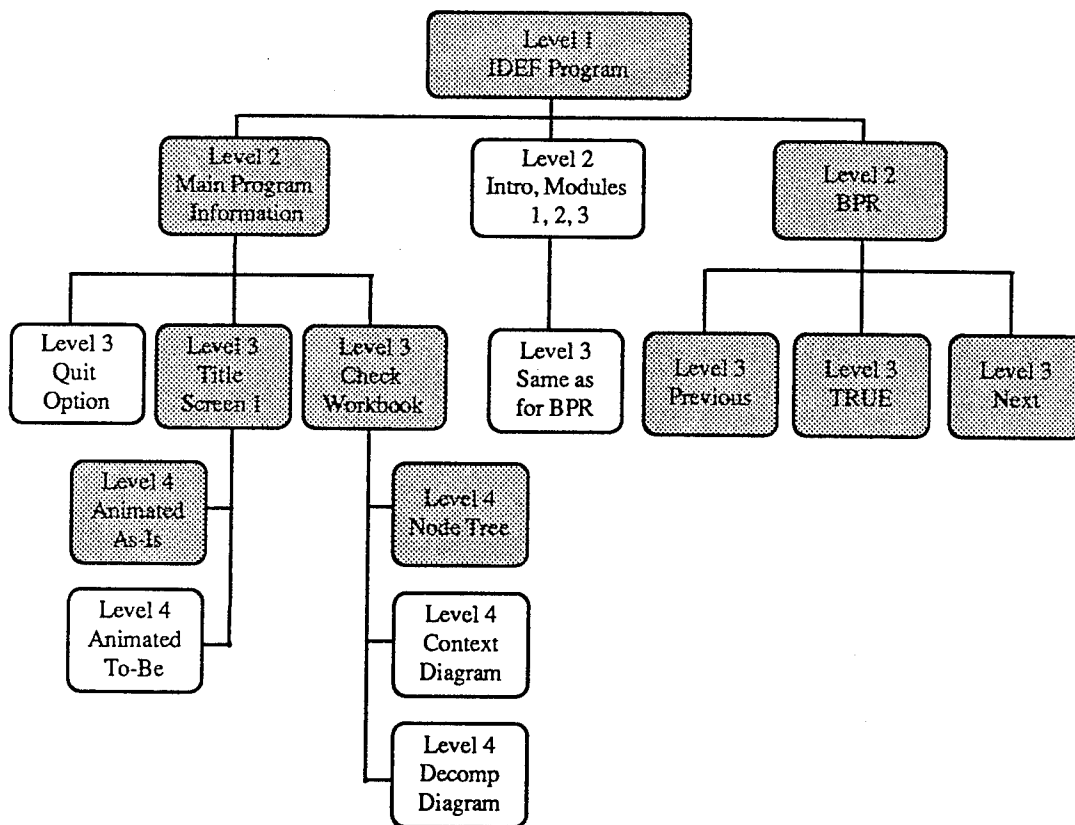


Figure C.3. Hierarchy of IDEF0 Tutorial

## IDEF0 Tutorial Flowline

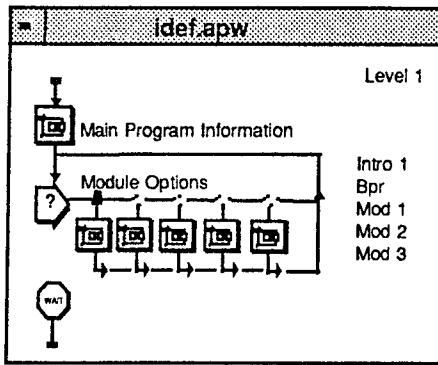


Figure C.4. Level 1 - IDEF0 Tutorial

Figure C.4 shows the flowline for the entire tutorial by using map (grouping) icons to organize the program into distinct areas. The “Main Program Information” map icon contains information about the overall program. The interaction icon below it is titled “Module Options.” The click-touch options shown at the right of this interaction icon allow the user to click on specific areas (the upper-right buttons located on each screen in the tutorial) which will bring them to any module they wish to study. The map icons below the click-touch buttons contain the actual information in the modules.

Figure C.5 is an expansion of the “Main Program Information” map icon shown above in Figure C.4. This flowline generally shows features that appear throughout the entire program. They are: the quit option on the menu bar, the grey background that all the slides build on, the click-touch overlay buttons that allow movement between modules, the “check workbook” option on the menu bar, and the glossary on the menubar. The “Title Screen 1” map icon appears only once in the program, at the beginning when it animates balls moving from chaos to control. The glossary feature only shows five entries here, but the arrows on the right of the entries allow scrolling up and down among the 21 definitions.

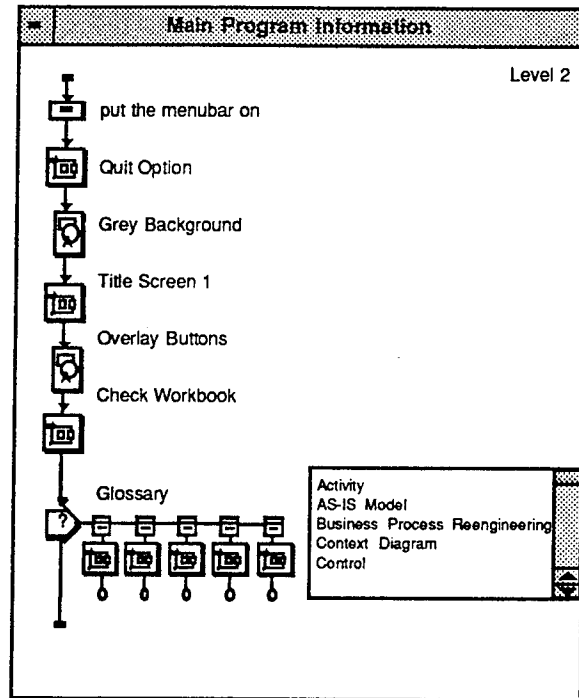


Figure C.5. Level 2 - Main Program Information

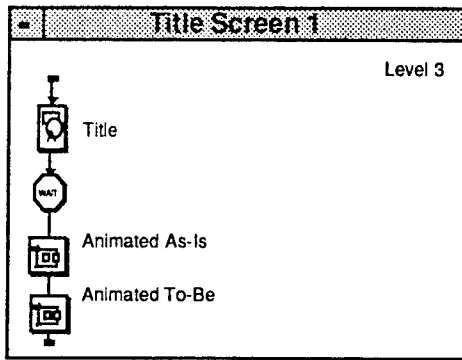


Figure C.6. Level 3 - Title Screen 1

Figure C.6 is an expansion of the “Title Screen 1” map icon shown in Figure C.5. The purpose of this flowline is to animate the moving balls in the introduction to the tutorial, moving from “chaos” to “control.” The “Title” display icon places the words on the screen, the wait icon gives the user a one second pause before the balls begin to move. The “Animated As-Is” map icon flowline, which causes the balls to move in a random fashion, is shown below in Figure C.7. The “Animated To-Be” map icon flowline causes the balls to move in an ordered manner. The programming is similar to the As-Is animation.

Figure C.7 is an expansion of the “Animated As-Is” map icon shown in Figure C.6. This flowline demonstrates the programming necessary to cause the balls to move in a supposed random fashion about the screen. The first display icon draws the box and puts the words on the screen. The first “ball” display icon puts one ball on the screen, which bounces in the pattern established in the “bouncing” animation icon. The other two “ball” display icons appear at the same time on the screen as the first, and move about following the patterns established by their respective “bouncing” animation icons. The wait icon creates a .5 second pause after the bouncing has stopped before everything is erased by the “Erase As-Is” icon. The logic flow of the “Animated To-Be” map icon is essentially the same as Figure C.7.

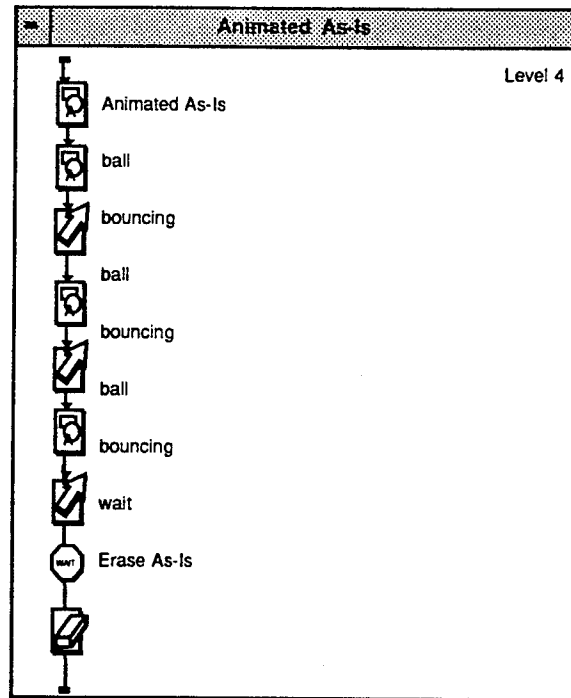


Figure C.7. Level 4 - Animated As-Is

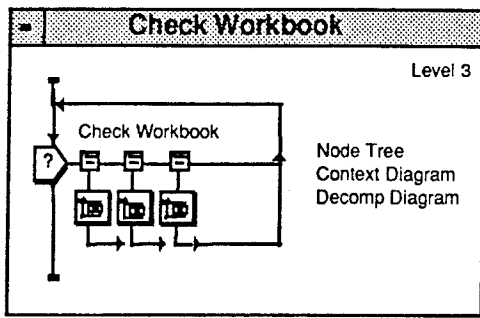


Figure C.8. Level 3 - Check Workbook

Figure C.8 is an expansion of the “Check Workbook” map icon shown in Figure C.5. The purpose of this flowline is to create a pull-down menu item that allows the student to check their workbook answers. The interaction icon creates the pull-down menu titled “Check Workbook” with three options below it: Node Tree, Context Diagram, and Decomp Diagram. Within each of these three options, users of the tutorial are led through a set of exercises to ensure that they have the right answers to the workbook.

Figure C.9 is an expansion of the “Node Tree” map icon shown in Figure C.8. This flowline depicts the steps that the user will go through to check the workbook exercises. First, everything on the screen erases except the grey background and click-touch buttons. This is necessary to get rid of current screen contents since the user may pull down the check workbook menu at any time. Next, the user receives instructions on how to proceed. Then, the node tree is drawn, and the following five map icons creates a list of node tree labels, including some false, “dummy” labels in order to challenge the student. The interaction icon called “Fill in nodes” creates an exercise where the student can click on one of the labels, and drag it to the appropriate spot on the node tree. The “+” map icons are programmed with locations for the right answers and a “You are correct” message, while the “-” map icons will bring up a message with hints of why the answer is wrong if the student chooses an incorrect placement of the label. After the student is finished with the exercise, the exercise will erase, and a message will appear with instructions on how to proceed to the next lesson.

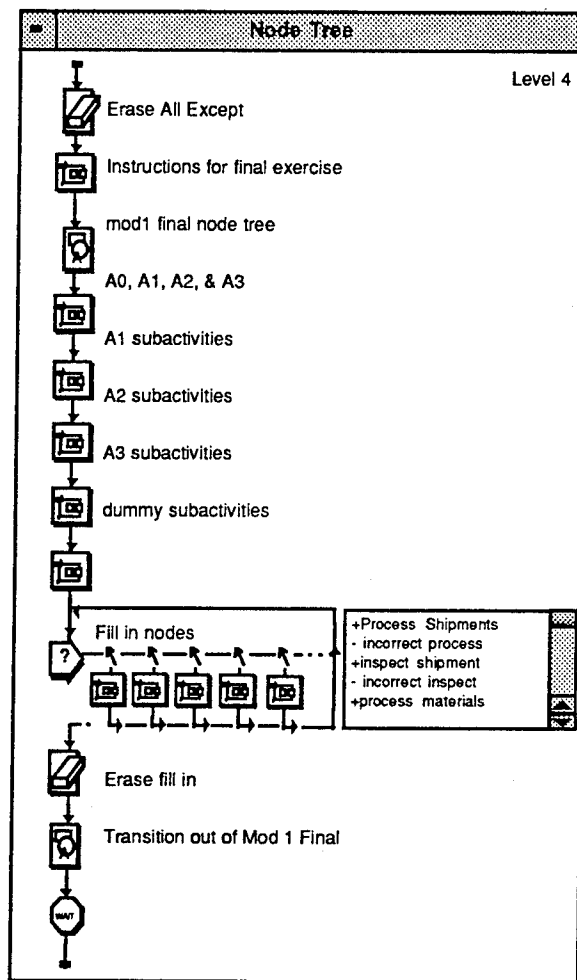


Figure C.9. Level 4 - Node Tree

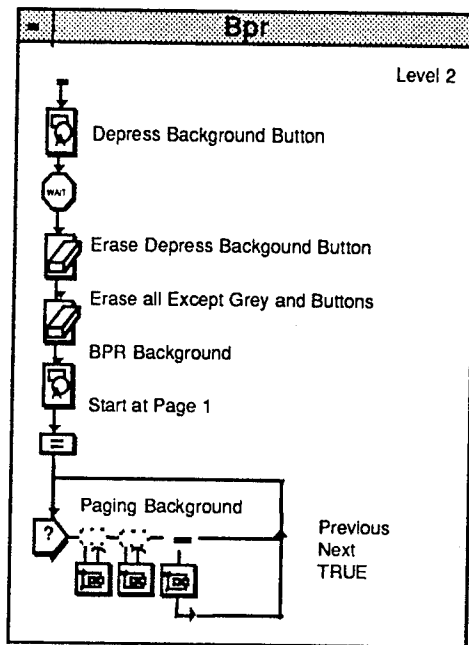


Figure B.10. Level 2 - Bpr Map Icon

Figure C.10 is an expansion of the “Bpr” map icon shown in Figure C.4. The flowline displayed for this icon follows the same logic as the flowlines for the “Mod 1,” “Mod 2,” and “Mod 3” map icons. Since the logic and thus the programming are the same, only this one will be described. In this flowline, first the “Depress Background Button” display icon creates the effect of someone depressing the click-touch BPR button. The wait icon aids in this effect, allowing the button to “depress” for just .5 seconds and then be erased by the next icon. Next, anything else that may be on the screen erases except the grey background and the click-touch buttons. This feature is necessary since the user may click on the BPR button at any time during the tutorial, and the current screen contents may have to be erased. Next, the “BPR Background” display icon puts the words “BPR Information” next to the forward and backward arrows on the bottom of the screen. Finally, the interaction icon contains the actual information about BPR and allows the user to page forward and backward in the program.

Figure C.11 is an expansion of the “Previous” and “Next” map icons shown in Figure C.10. These two flowlines are responsible for the forward and backward buttons and the paging system in the program. This particular set controls the paging for the BPR module. The other modules each have identical “Previous” and “Next” map icons within their flowlines. The display icons simply create the effect of depressing the forward and backward arrows. The calculate icons cause the program to move forward one page for the “Next” icon, and move backward one page for the “Previous” icon.

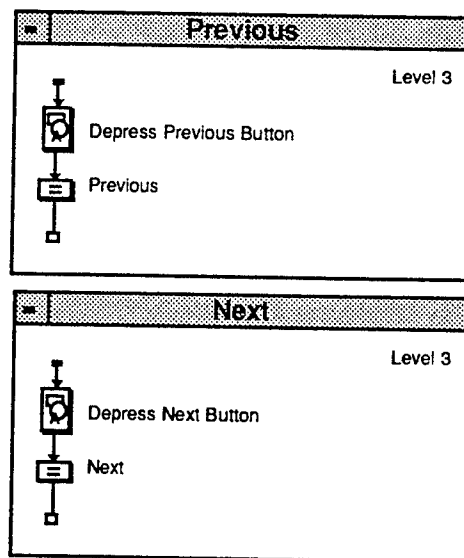


Figure C.11 Level 3 - Previous/Next Buttons

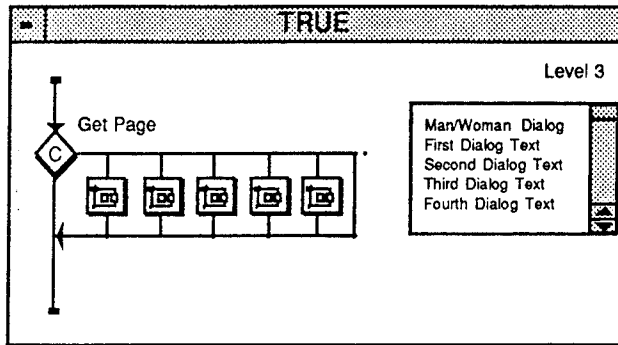


Figure B.12. Level 3 - TRUE Interaction

Figure C.12 is an expansion of the “TRUE” map icon shown in Figure C.10. This is the actual set of screens that students will see for their tutorial. In this instance, it is the BPR information displayed, although the other modules follow the same programming for their instructional material. Although for this example all of the icons to the right of the decision icon are maps, you could just as easily have a display, sound, video, or movie icon here. Inside these map icons, you will find groups of display, erase, and wait icons. As in the glossary shown in Figure C.5, the bar with up and down arrows indicate a scrolling feature that means that there are more than just five pages of information.

Figures C.4 through C.12 represent the majority of the IDEF0 tutorial program. Within the modules, there are other further levels of detail, but these figures give a general idea of the logic flow of this program.

**IDEF0 Workbook and Case Study  
for use with  
The IDEF0 Computerized Tutorial**

This booklet includes:

An Overview of BPR, its Components, and IDEF  
A Case Study for applying Activity Modeling concepts  
Glossary of Key Words and Concepts

# Business Process Reengineering Overview

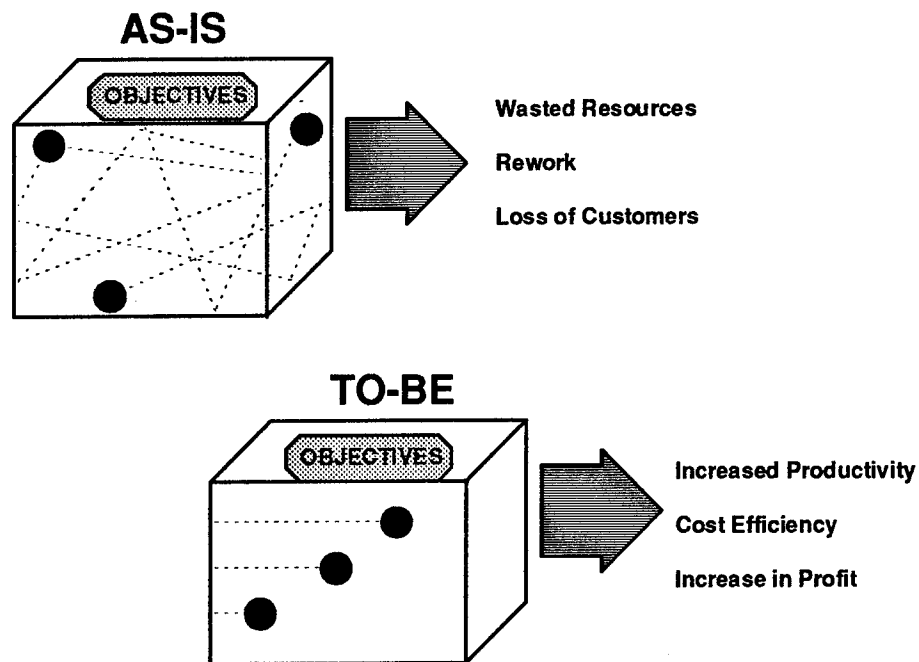
## Business Process Reengineering

Business process reengineering (BPR) is a method of analyzing current business practices, identifying wasteful practices, and redesigning these practices for drastic increases in productivity, reductions in costs, and streamlined procedures.

Because of its emphasis on changing business functions or improving work processes, you may also have heard of BPR being called: Functional Process Improvement, Business Process Improvement, Business Process Innovation, or Business Process Redesign.

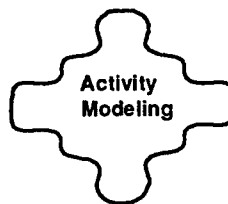
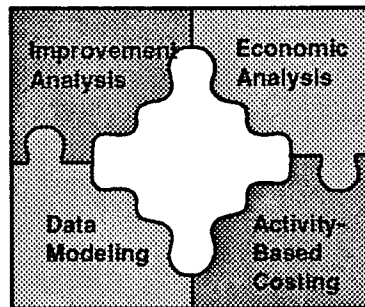
BPR is somewhat similar in philosophy to Total Quality Management. However, there are distinct differences between the two concepts. Total Quality focuses on continuous, small-scale improvements in isolated areas of a business. BPR, on the other hand, is a one-time effort which produces enormous improvements over a much larger portion of the business, often incorporating several individual business areas.

A complete BPR effort can take as long as 1 - 2 years to complete. It involves a series of steps designed to 1) show the complete business as it currently exists 2) produce a picture of how the business should look after making changes. We call these the AS-IS and TO-BE models, respectively. In AS-IS environments, efforts and resources are often misdirected and sub-optimized. After BPR, however, work efforts are aligned with business objectives for more efficient and profitable business practices.



To get a complete picture of the AS-IS environment, we must know how the business is organized, what functions it performs, how information flows through it, and where it spends money. We get this information by performing five tasks: Improvement Analysis, Economic Analysis, Data Modeling, Activity-Based Costing, and Activity Modeling

### Determining The AS-IS Environment



### Activity Modeling or IDEF-0

Thankfully, your job right now is to focus on only the first of these tasks -- Activity Modeling. Activity modeling is nothing more than recording the individual steps it takes to complete a particular process. As you'll learn very soon, the Air Force has designed a standard set of symbols which can be used to model any process. They call this collection of symbols and its use IDEF. There are IDEF methodologies for modeling activities (IDEF-0) and the flow of data through an organization (IDEF-1X). The computer program you are about to run is designed to teach you everything you need to know about activity modeling, or IDEF-0, and how to apply it. In addition, this workbook provides a case study from which you can develop a complete activity model of your own.

**\*\*STOP\*\***

**Return to the tutorial**

## **IDEF0 Process Modeling Exercise: "The Supply Depot"**

**Purpose:** This exercise is designed to allow the student to apply the concepts learned in the IDEF0 tutorial to an activity modeling scenario.

**Overview:** The Supply Depot exercise consists of three main sections which parallel the three modules of the tutorial: the node tree, the context diagram (A0-level), and the decomposition diagram (A1-A2-A3-level). Suggested answers to each exercise are provided in the tutorial, along with explanations as to why these answers are appropriate.

**Objectives:** At the completion of the Supply Depot exercise, the student will be able to:

### Node tree

- Identify major processes of a business from a written description
- Recognize relationships and hierarchies among processes and activities
- Construct a node tree based on identified processes and activities

### Context Diagram

- Select a high-level process suitable for activity modeling
- Identify Inputs, Constraints, Outputs, and Mechanisms (ICOMs) associated with the process
- Construct a context (A0) diagram based on available information and reasonable inferences

### Decomposition Diagram

- Identify activities of a higher-level process
- Identify relationships among activities, and arrange these activities according to the overall process flow
- Construct a decomposition diagram based on available information and reasonable inferences

**Deliverables:** Node Tree developed to the A1.1 level  
Context Diagram with all associated ICOMs  
Decomposition Diagram with all associated ICOMs and relationships among activities

## **The Supply Depot**

### **BACKGROUND**

The Supply Depot is a regional storage and processing facility for a large federal agency. The Supply Depot employs 6,000 persons and consists of five main departments: Receiving, Storage, Shipping, Inventory Management, and Support Operations. Each department has a department manager who reports to the supply depot manager. As well, each department operates as a cost center, charging a sufficient fee to recover any expenses associated with the operation of that department.

You have just been hired as the Assistant Department Manager of the Receiving Department. Your boss, the department manager, is concerned with the problem of excessive backlog of unprocessed shipments in his department. He is also concerned with the growing number of complaints from motor carriers, delivery services, and couriers about the slow unloading and processing times at the receiving department. In fact, the carriers have voiced their complaints in writing to the depot manager, your boss' boss, and have threatened to charge an additional "wait charge" for the excess time the operator and equipment are kept waiting at the unloading docks. Of course, the "wait charge" will be expensed to your department, directly reducing your already slim profit margin.

Your boss, the department manager, is unhappy with the negative publicity his department is receiving. He has considered a number of possible solutions, from instituting overtime (too expensive and unpopular with the workers), to constructing more receiving bays (expensive and reduces much-needed storage space). He has given you the responsibility of finding out where the process is bottlenecked and suggesting alternatives which would fix the receiving department's problems. He has given you 3 months to locate the problems and identify solutions.

Common sense tells you to find out as much as possible about the receiving department and how it operates before making any recommendations to your boss. By interviewing people in your department and reviewing old production reports, you discover the following information:

### **RECEIVING DEPARTMENT**

The receiving department consists of three main sections: Inspection, Material Processing and Administrative Records. The receiving sections occupy approximately 250,000 square feet of covered storage with 8 attached receiving bays. The overall purpose of the receiving department is to take initial receipt of all shipments arriving at the depot.

The Inspection section is responsible for all aspects of inspecting inbound shipments, from meeting the inbound deliveries to directing the unloading of their shipments. An inspector is assigned to each receiving bay, and he/she is responsible for taking the Government Bill of Lading (GBL) and shipping invoices from the delivery driver. The shipping invoice is the supplier's invoice to the depot for the materials being shipped. The shipping invoice is generated by the shipping company based on a purchase order issued by the depot. The

inspector checks the GBL and shipping invoice to ensure that the shipment is destined for this particular depot, then directs the shipment to be unloaded onto 4x4 pallets that will eventually be placed on a conveyer belt for processing. Once the shipment is palletized, the inspector verifies the quality and quantity of the shipment based on the size of the shipment. In cases of bulk storage items which can not be palletized, the inspector directs the driver to take the shipment to the bulk storage area where another inspector performs the necessary quality and quantity checks.

The Administrative Records section controls all of the computer terminals and administrative clerks located near the conveyer belts in the material processing area. In addition to processing the GBL and shipping invoices, this section also prepares discrepancy and exception reports for depot management, and performs data entries into the depot's mainframe computer. There is a single computer terminal in this section dedicated for the use of the inspectors from the bulk storage area. After receiving a bulk shipment, inspectors from the bulk storage area come to the administrative section to enter the necessary GBL and shipping invoice data into the depot's mainframe computer.

The Material Processing section is composed of conveyer belts which move the palletized shipments from the receiving bays to temporary storage. As the shipment moves down the conveyer, an administrative clerk seated at a computer terminal enters the shipment number into the computer. A screen appears with the purchase order information, and the clerk checks this information against the information on the shipping invoice. If all the information agrees, another screen appears with a temporary storage location for the shipment. The clerk annotates the shipping invoice with the storage location for that pallet. Next, the clerk matches the GBL to that specific shipment and sends the pallet down the conveyer to a holding area. Pallets in the holding area are then transported to the storage location annotated on the shipping invoice. If, for some reason, the processing clerk is unable to verify the shipping invoice or GBL and receive a storage location, he/she marks the unverified document with a red flag. This flag is a signal to the processing clerks at the end of the conveyer, who will move the pallet off the conveyer and place it in a holding area with other unverified shipment pallets. A special team is assigned daily to reconcile these problem shipments.

## **\*\*STOP\*\***

You now have enough information to complete a node tree of the Receiving Department and its activities. Use the space below as scratch paper to construct a node tree diagram based on the information you just read. Hint: Use "Process Shipment" as the A0 node.

Some things to remember:

- Don't get bogged down in extraneous details -- focus on tasks that are performed
- Be conscious of using the correct format (i.e. *verb*, *noun*) for naming your activities
- Where possible, use the terminology provided by the people you interview. This ensures that you have a common vocabulary to communicate and clarify ideas, and will also help foster support for your modeling effort.
- Understand that there is no teacher's manual for an activity model. The "correctness" of your model is reflected by how accurately it presents the process at hand, and therefore is completely subjective. Don't be discouraged, then, if your model does not match exactly the answers provided in the tutorial. The goal is to think about the process as you understand it, and to be able to defend your answers based on the information given.

**Now, Return To the Tutorial To Check Your Answers**  
**Select "Node Tree" from the Check Workbook pull-down menu**

## The Supply Depot (continued)

### ADDITIONAL DETAILS

The receiving department classifies inbound shipments as one of three categories: less-than-truck loads, full-truck loads, and bulk shipments. Each of these categories requires special inspection procedures. For less-than-truck loads, the inspector verifies the exact quantity of each shipment. To save time, full-truck loads are inspected by selecting a shipping carton or container, verifying the exact quantity in the container, and then multiplying that number by the total number of containers. This final number is then checked against the shipping invoice. Because the receiving bays lack the equipment necessary to unload bulk shipments, the inspector simply inspects the GBL and shipping invoice before dispatching the driver to the appropriate bulk storage area. In cases of bulk shipment, quantity and quality verification is performed by inspectors at the bulk storage location. After the shipment is unloaded, the bulk storage inspector returns to the administrative records section to input the data to the mainframe.

The receiving department currently works a two-shift schedule. The first shift is from 7:00 AM until 3:30 PM with a half hour lunch. The second shift is from 11:00 AM until 7:30 PM with a half hour lunch. The employees are happy with their schedules as they are, and would prefer hiring more personnel before instituting overtime or changing the current work schedule.

Mondays, Thursdays, and Fridays have the largest number of inbound shipments. Consequently, it is not uncommon to have shipments that arrived on a high volume day still awaiting processing the next morning. As you'll recall from the background information, your boss, the department manager, has considered instituting overtime to solve the processing backlog. Aside from the costs involved and the threat of trouble from the local Inspectors, Clerks, and Forklift Operators' Union, it is company policy that overtime must be approved by the depot manager. Approval is unlikely, though, because it is the depot manager's policy that overtime is to be used to process high priority shipments, not simply to remedy routine processing backlog.

Compounding the issue is the receiving department first-come, first-to-unload policy. Although undoubtedly the fairest way of doing business, this policy often results in drivers having to wait 4 - 6 hours for an empty bay to unload their shipments. Further, the first-come, first-to-unload policy is particularly unpopular with the couriers and other delivery services who have to wait behind full-truck loads to deliver a smaller number of packages.

Space is very limited within the receiving department. Between the build-up of pallets waiting to be loaded onto the conveyer belts, the conveyer belts themselves (there are 8 belts -- one for each receiving bay), and forklifts, empty pallets, and materials needed for palletizing inbound shipments, there is little room to spare. Space is particularly tight at the end of the conveyers, where problem shipments and backlogged pallets often occupy the overflow space and backup onto the conveyers or behind the receiving bays.

**\*\*STOP\*\***

You now have all the information you need to identify ICOMs and add these details to your context diagram for the Receiving Department. If necessary, you can look back at the first exercise information for details. It is also all right to make assumptions about ICOMs, so long as you're certain you can defend your assumptions. Remember, not all the information may be relevant to this level of analysis, and may only be used later in the third module -- Decomposition Diagrams. Hint: We have identified 4 inputs, 3 constraints, 4 outputs, and 2 mechanisms in our solution.

## GLOSSARY

**ACTIVITY** - A named task that occurs over time and consumes resources to produce an output. Activities combine to form business processes. In the context of IDEF activity modeling, activities are found at the A1 level and below.

**AS-IS MODEL** - The complete definition of a business function as it currently exists. Within Business Process Reengineering, AS-IS models include both data and activity models, improvement and economic analyses, and activity-based costing projects. In the context of IDEF0 activity modeling, AS-IS models are the collection of fully-described context and decomposition diagrams.

**BUSINESS PROCESS REENGINEERING (BPR)** - An in-depth study of the current practices of a major section of a business targeted at identifying and eliminating wasteful practices. BPR studies include activity and data modeling, activity-based costing, and improvement and economic analysis to define the current business (AS-IS) environment, then apply benchmarking, functional economic analysis, and activity/data model modification to define the reengineered (TO-BE) environment.

**CONTEXT DIAGRAM** - A visual representation of the scope of a particular activity modeling effort. Depicts the primary process to be examined, as well as the inputs, controls, outputs, and mechanisms associated with the process.

**CONTROL** - Any regulation, resource, or procedure acts to constrain an activity. In IDEF0 activity modeling, constraints enter the activity from the top.

**DECOMPOSITION DIAGRAM** - A visual representation of the collection of activities which comprise a process or higher activity, and the flow of inputs, controls, outputs, and mechanisms associated with each activity. In IDEF0 activity modeling, decomposition diagrams are completed for the A1 level and below.

**FUNCTION** - In the context of Business Process Reengineering (BPR), an extremely high-level business activity or section which is often a candidate for a BPR study.

**ICOM** - Acronym for Input, Control, Output, and Mechanism. Those resources, constraints, products, and energy-providing elements which impact an activity.

**IDEF** - Integrated, Computer-Aided Manufacturing Definition. A methodology created by the Air Force and adopted by the DOD for graphically depicting the processes and activities which occur in a business. The structure of the IDEF methodology forces processes to be broken into their component activities, and these activities to be linked through common inputs and outputs. There are two main IDEF methodologies: IDEF0, or activity modeling; and IDEF1X, or data modeling.

**INPUT** - Any resource or service consumed or transformed by an activity. Inputs to an activity must be used to produce an output. In an IDEF0 activity model, inputs enter an activity from the left side.

**MECHANISM** - Any reusable resource that provides energy to an activity or performs that activity. In IDEF0 activity modeling, mechanisms enter the activity from the bottom.

**NODE** - Name given to the individual junction points of a node tree. Each node represents a specific process or activity to be further examined in the course of an activity modeling effort.

**NODE TREE** - A hierarchy of nodes which depicts the high-level processes and individual activities involved in an activity modeling effort. The purpose of a node tree is to provide a quick overview of the project scope and the associations between the processes and activities involved. The individual nodes, read from the top of the tree, are referred to as root nodes, children, grandchildren, and great-grandchildren, respectively. Nodes occupying at the same hierarchical level are referred to as peers.

**NON VALUE-ADDED** - Term used to describe any activity that provides a negative return on the investment of resources allocated to that activity. An activity whose purpose is to repair mistakes, compensate for lack of quality, or duplicate products of another activity. In general, reengineering seeks to identify and eliminate such activities.

**OUTPUT** - Any service, information, or material produced by or resulting from an activity. In IDEF0 activity modeling, there must always exist at least one output from every activity. Activities exit the activity from the right side.

**PIPELINE** - The combining of two or more closely-related ICOMs into a single, more general ICOM for the purpose of eliminating clutter.

**PROCESS** - A collection of activities that work together to produce a defined set of products or services. Within businesses, processes work to fulfill the mission of the business, and therefore must be aligned in some way with business objectives. In the context of IDEF activity modeling, processes are found at the A0 level.

**PROCESS MODEL** - A visual representation of the major processes and activities of a business, and their interrelationships. IDEF0 is a methodology adopted by the Air Force to produce standardized process models. Within a Business Process Reengineering (BPR) effort, process models will be made for both the AS-IS and TO-BE environments.

**TO-BE MODEL** - The result of business process reengineering or an IDEF0 activity modeling. A representation of a business function or activity model after modification and elimination of non value-added activities.

**TUNNELING** - In IDEF0 activity modeling, the temporary elimination of one or more ICOMs from a higher-level diagram. Although not explicitly modeled at the higher level, these ICOMs are still considered part of the activity, and must reappear within the next lower level diagram. Tunneling is used to facilitate the reading of diagrams by removing clutter.

**VALUE-ADDED** - Term used to describe any activity which contributes to the performance of a business' mission, and which could not be eliminated without impairing the mission. These activities are of importance to the business, and will be preserved in the reengineering process.

## Appendix E: Summary of Initial Prototype Assessment

The comments below were provided as part of the initial prototype assessment. Specifically, this assessment was designed to critique only the user interface, though comments concerning the instructional flow of the tutorial were also included. Below each comment is a short statement reflecting the changes we incorporated in response to the comment.

1. What about backing up to previous screens? I accidentally hit two keys several times and was whisked past information that I could only get once again by completely restarting the module.

Disposition: Incorporating this change required fundamentally altering the programming structure of the tutorial. Once completed, however, this change allows the user to move forward and backward through the tutorial, in addition to jumping between modules of instruction.

2. You never really explain the buttons that you use up in the left hand corner. You mention them real quickly but don't give any explanation that leads the user to understand their use. It can easily be figured out but computer novices may have trouble.

Disposition: Additional detail was added to the initial instruction screens to clarify the meaning of these icons and explain their use.

3. There is no action response when you click on any of your buttons. Normally a button should highlight or something so that a user knows that their action has been accepted. Your buttons are static. Again this can very disconcerting for the novice user since they are guessing that their action is what resulted in the current program action.

Disposition: All user-interface buttons were modified to highlight when activated by a mouse click.

4. Why do you have a quit button if the user is instead consistently told to select "Quit" from the menu bar if they want to quit the program?

Disposition: The persistent "Quit" button was removed from the tutorial screens and that function consolidated into a single "Quit" option accessed through a pull-down menu.

5. Your "Quit" function is quite ungraceful. There should be an intermediate screen or something that smoothes the way for a system transfer. It currently just ungracefully dumps me back into windows without any instructions about restarting or thanks for coming.

Disposition: Two intermediate screens were added to the tutorial; one gives the user the option of exiting or returning to the tutorial, the other screen transitions the user back into windows and acknowledges the intent to exit.

6. When you discuss the input ICOM a new button shows up at the top labeled, "example." Where does this come from? There is no reference to it. Why don't you have the same thing for other ICOMs?

Disposition: We felt the use of an example to illustrate important concepts was a necessary part of the tutorial. However, we added references to the example throughout the text, and were careful to include transition screens before presenting the example.

7. You mention the case study but it doesn't actually seem to be incorporated into the program anywhere.

Disposition: Instruction and transition screens were added to smooth the flow between the tutorial and the workbook. In general, student evaluators felt the incorporation of the workbook aided in reinforcing important concepts from the tutorial.

8. Don't use yellow as a highlight. I couldn't read it at all on the white background you used. If you used a different background on your machine, then retest the program. The user machine that we said we should expect out in the field was a 386 with a VGA.

Disposition: Background colors and colors used within the text were modified to accommodate the palette of a VGA screen.

## Appendix F: Physical Design Evaluation Questionnaire

### Abstract

The survey below was presented to 17 graduates students in the Information Resources Management program at the Air Force Institute of Technology. The questions were designed to illicit qualitative data and subjective opinions about a computer-assisted instruction prototype and supplementary materials developed as part of this thesis. The number of student evaluators who gave the indicated rating is provided in parentheses next to the rating. Additionally, written comments provided by the evaluators are included at the end of the survey portion of this appendix. Note: One respondent had to be eliminated due to an incorrectly completed answer sheet. Therefore, total responses will number 16.

### Survey

The following questions are designed to guide your critique of the IDEF0 tutorial you just completed. Your comments will be used to modify the existing program, so please be both clear and thorough in your responses. We appreciate the time you've invested in helping us complete our research.

Please circle the answer that reflects your agreement with the following statements based on this sliding scale:

- |                              |                   |
|------------------------------|-------------------|
| 1. strongly disagree         | 4. agree          |
| 2. disagree                  | 5. strongly agree |
| 3. neither agree or disagree |                   |

### Demographics

1. I consider myself knowledgeable in the general operation of computers.

1 (1)      2      3 (1)      4 (3)      5 (11)

2. I have previous educational experience with information presented through computer-assisted instruction (the use of computers to present instructional information and evaluate comprehension).

1      2 (4)      3 (3)      4 (5)      5 (4)

### Flexibility and Control

3. The tutorial allowed me to control the speed at which information was presented.

1      2      3 (2)      4 (5)      5 (9)

4. I could easily navigate through different sections of the tutorial.  
 1                      2 (2)                      3 (1)                      4 (8)                      5 (5)
5. I could easily enter and exit the tutorial.  
 1                      2                      3 (3)                      4 (7)                      5 (6)
6. On re-entering the tutorial, it was easy for me to find where I had left off.  
 1                      2 (2)                      3 (7)                      4 (4)                      5 (3)
7. I found it awkward to switch back and forth between the tutorial and the workbook.  
 1                      2 (8)                      3 (3)                      4 (5)                      5

Visual Clarity

8. The individual instruction screens were cluttered with too much information.  
 1 (6)                      2 (8)                      3 (1)                      4 (1)                      5
9. Organizational clues were available to help identify the flow of instruction.  
 1                      2 (1)                      3 (2)                      4 (1)                      5 (12)
10. I found the use of different colors within the program to be distracting.  
 1 (8)                      2 (6)                      3 (2)                      4                      5
11. The contrast between the background color and the text made the screens easy to read.  
 1                      2                      3                      4 (9)                      5 (7)
12. I felt the use of colors within the program helped highlight and clarify concepts.  
 1                      2                      3                      4 (9)                      5 (7)
13. I found it hard to determine the meaning of the icons based on the icon picture.  
 1 (4)                      2 (8)                      3 (2)                      4 (1)                      5 (1)

Informative Feedback

14. The program initiation screens were clear and provided me with the necessary information to access the program.  
 1                      2                      3 (3)                      4 (10)                      5 (3)
15. Directions were clear and concise.  
 1                      2                      3 (3)                      4 (7)                      5 (6)

16. The test exercises addressed important concepts.

1                      2 (1)                      3                      4 (10)                      5 (5)

17. I felt the feedback was constructive in helping to explain incorrect responses.

1                      2                      3 (3)                      4 (7)                      5 (6)

### Content

18. The information provided by the tutorial adequately covered the subject matter.

1                      2 (2)                      3 (3)                      4 (4)                      5 (7)

19. The tutorial provided me with the knowledge I needed to complete the workbook exercises.

1                      2 (1)                      3 (1)                      4 (11)                      5 (3)

20. The workbook was clearly written.

1                      2 (1)                      3 (1)                      4 (10)                      5 (4)

21. The workbook scenario helped me apply the concepts presented in the tutorial.

1                      2                      3                      4 (11)                      5 (5)

22. The information presented on business process reengineering was relevant to my understanding of IDEF0.

1                      2                      3 (1)                      4 (11)                      5 (4)

### Comparison To Traditional Teaching Methods

23. I would be interested in taking a course taught solely as a computerized program.

1 (1)                      2 (4)                      3                      4 (6)                      5 (5)

24. I would be interested in taking a course taught as a mixture of classroom learning and computerized program.

1                      2                      3 (1)                      4 (8)                      5 (7)

### Additional comments:

1. A "bye bye" screen at your departure point would have been nice.
2. The overall node tree test is somewhat challenging -- a good test of knowledge
3. Module 1 appears to end on page 22 or 23 vs 24 of 24.  
When I finished Module 1, it said "23 of 24". What happened to page 24?
4. Overall, this is a great way to teach this stuff!

5. [switching between prototype and workbook] wasn't awkward. I think it interrupts momentum and flow of my concentration. When instructed to return to the workbook, I was tempted to go on or quit without reading the book.
6. In the workbook you state that BPR is similar to TQM and then spend time discussing the differences. I don't think BPR is anything like TQM.
7. I don't like the talking heads -- too juvenile for my tastes, but then it depends on who your audience is.
8. I like knowing how many pages there are in each module.
9. Overall I think this was good -- if I didn't know BPR and IDEF0 already, it would be helpful.
10. I probably would have had trouble [determining the meaning of the icons] if I'd never seen IDEF before. The icons look like the models and the reader is expected to know that.
11. Directions should be clearer as to how to navigate in and out of the program. Also, the reader should have the ability to leave the program and easily return to any point without having to scan through all the pages.
12. Your tutorial had two flaws I saw - It wouldn't let me place a correct answer in a spot. [The program] should only enter [the student response] after the entire click and drag is done. (That may be a software flaw - I don't know)
13. Personally, I think computer-taught information is painful and tedious. It is too easy to forget after you complete the exercise. This is a general comment not specifically aimed at this program.

## Bibliography

- AECT. Educational Technology: A Glossary of Terms. Washington DC: Association for Educational Communications and Technology, 1979.
- AERA. "The Use of Computers in Education Worldwide: Results from a Comparative Survey in 18 Countries," Proceedings of the Annual Conference of the American Educational Research Association. Chicago, 3-8 April 1991.
- Apple Classrooms of Tomorrow: Philosophy and Structure; What's Happening Where. Cupertino CA: Apple Computer, Inc., 1991.
- Appleton, Dan, et.al. Corporate Information Management--Process Improvement Methodology for DoD Functional Managers. D. Appleton Co., 1992.
- Aspillaga, Macarena. "Screen Design: Location of Information and Its Effects on Learning," Journal of Computer-Based Instruction: 89-90 (Summer 1991).
- Chase, P.N. "Designing Courseware: Prompts From Behavioral Instruction," The Behavior Analyst (January 1985).
- Clinton, William J. and Albert Gore, Jr. Technology for America's Economic Growth, A New Direction to Build Economic Strength (February 1993).
- Cooper, Peter A. "Paradigm Shifts in Designed Instruction: From Behaviorism to Cognitivism to Constructivism," Educational Technology: 12-18 (May 1993).
- Corbin, Lisa. "Reengineering: The Next Management Revolution," Government Executive: 26-32 (September 1993).
- CQ. Congressional Quarterly Weekly Report, Vol. 50: 3184-3189 (10 October 1992).
- Davenport, Thomas H. and James E. Short. "The New Industrial Engineering: Information Technology and Business Process Redesign," Sloan Management Review: 11-27 (Summer 1990).
- Ely, Donald P. "Trends in Educational Technology," Clearinghouse on Information Resources: 3-46 (June 1992).
- Golub, L.S. "With the Microcomputer, Behaviorism Returns to Early Childhood Education," Paper presented at the annual meeting of the American Educational Research Association, New Orleans (1983).

- Grover, V., J.T.C. Teng, and K.D. Fiedler. "Information Technology Enabled Business Process Redesign: An Integrated Planning Framework," Omega: 443-447 (July 1993).
- Guimaraes, Tor. "Prototyping: Orchestrating For Success," Datamation: 101-106 (December 1, 1987).
- Gulden, Gary K. and Robert H. Reck. "Combining Quality and Reengineering Efforts for Process Excellence," Information Strategy: The Executive Journal: 10-16 (Spring 1992).
- Hammer, Michael. "Reengineering Work: Don't Automate, Obliterate," Harvard Business Review: 104-112 (July-August 1990).
- Hammer, Michael and James Champy. Reengineering the Corporation: A Manifesto for Business Revolution. New York: Harper Collins Publishers, Inc., 1993.
- Hannafin, Michael J. and Barry W. Carney. "Effects of Elaboration Strategies on Learning and Depth of Processing During Computer-Based Instruction," Journal of Computer-Based Instruction: 77-82 (Summer 1991).
- Harker, Susan. "The Use of Prototyping and Simulation in the Development of Large-Scale Applications," The Computer Journal: 420-425 (October 1988).
- Harrington, H. James. Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness. New York: McGraw-Hill, Inc., (1991).
- Hazen, Margaret. "Instructional Software Design Principles," Educational Technology: 18-23 (November 1985).
- Information Manager, AFRP 37-1. Department of the Air Force: (July 1993).
- Janson, Marius. "Applying a Pilot System and Prototyping Approach to Systems Development and Implementation," Information and Management: 209-215 (April 1986).
- Jonassen, D.H. "Objectivism Versus Constructivism: Do We Need a New Philosophical Paradigm?," Educational Technology Research and Development: 5-14 (March 1991).
- Kaufman, Phillip A. "Prototyping Alternatives," Computer Design: 27-29 (April 1992).
- Klingler, Daniel E. "Rapid Prototyping Revisited," Datamation: 131-332 (October 15, 1986).

- Lamos, J.P. Instructional Development: The State of the Art. Best Books: McAlester OK, 1984.
- Landau, Sidney I., ed.-in-chief. Funk & Wagnalls Standard College Dictionary. New York: Funk & Wagnalls Publishing Company, Inc., 1973.
- Luqi. "Software Evolution Through Rapid Prototyping," Computer: 13-76 (May 1989).
- Maddux, Cleborne D. "User-developed Computer-Assisted Instruction: Alternatives in Authoring Software," Educational Technology, Research and Development: 7-14 (April 1992).
- Milheim, William D. and Barbara L. Martin. "Theoretical Bases for the Use of Learner Control: Three Different Perspectives," Journal of Computer-Based Instruction: 99-105 (Summer 1991).
- National Science Foundation Authorization Hearings, Vol. 1. Washington DC: U.S. Government Printing Office (April 1967).
- Orey, M.A. "Using Intelligent Tutoring Design Principles to Integrate Cognitive Theory into Computer-Based Instruction," Proceedings of Selected Research Presentations at the Annual Convention of the Association for Educational Communications and Technology (1991).
- Ralston, Anthony and Edwin D. Reilly, editors. Encyclopedia of Computer Science (Third Edition). New York: Van Nostrand Reinhold, 1993.
- Ravden, Susannah J. and Graham I. Johnson. Evaluating Usability of Human-Computer Interfaces: A Practical Method. Chichester, England: Ellis Harwood, Ltd, 1989.
- Robyler, M.D. "Fundamental Problems and Principles of Designing Effective Courseware," Instructional Designs for Microcomputer Courseware. Hillsdale NJ: Lawrence Erlbaum Associates, Publishers, 1988.
- Robyler, M.D. and K.A. Hall. "Systematic Instructional Design of Computer Courseware: A Workshop Handbook." Tallahassee FL: Florida A&M University, 1985.
- Saettler, Paul. The Evolution of American Educational Technology. Englewood CA: Libraries Unlimited, 1990.
- Sawyer, W.D.M. "The Virtual Computer: A New Paradigm for Educational Computing," Educational Technology: 7-14 (1992).

Stephenson, Stanley D. "The Effects of Student-Instructor Interaction and Paired/Individual Study on Achievement in Computer-Based Training (CBT)," Journal of Computer-Based Instruction: 22-26 (Winter 1992).

Stewart, Thomas A. "Reengineering: The Hot New Tool," Fortune: 10-18 (23 August 1993).

Tillmann, George. "Prototyping for the Right Results," Datamation: 42-45 (April 1, 1989)

Woolf, Beverly Park. "Toward A Computational Model of Tutoring," Educational Technology Research and Development: (49-63) April 1992.

Zwass, Vladimir. Management Information Systems. Iowa: Wm. C. Brown Publishers, 1992.

### Vita

Captain Brian Brown was born on 9 November 1966 in International Falls, Minnesota, and graduated from Indus High School in Birchdale, Minnesota in 1985. He earned a Bachelor of Science degree in Business Administration and Management from North Dakota State University in 1989 and was commissioned through the Reserve Officers Training Corps program. Captain Brown was assigned as Headquarters Squadron Section Commander for the 3340th Technical Training Group, Chanute AFB, Illinois. He entered the School of Logistics and Acquisition Management, Air Force Institute of Technology, in May of 1993.

Permanent Address: Route 7, Box 395  
International Falls, MN 56649

## Vita

Captain Susan Brown was born on 7 December 1967 in Beaver, Pennsylvania, and graduated from Beaver Area High School in 1986. She earned a Bachelor of Arts degree in Literature and Education from Duke University in 1990 and was commissioned through the Reserve Officers Training Corps program. She taught ninth grade English in Durham, North Carolina before being assigned as Squadron Section Commander for the 60th Component Repair Squadron, Travis AFB, California. Captain Brown entered the School of Logistics and Acquisition Management, Air Force Institute of Technology, in May of 1993.

Permanent Address: 675 Brady's Ridge Road  
Beaver, PA 15009

## Vita

Captain Karen Cook was born on 28 March 1964 at Tachikawa Air Base, Japan, and graduated from the Torrejon American High School of Madrid, Spain in 1982. She earned a Bachelor of Science degree in Business Management from Virginia Polytechnic Institute and State University in 1986 and was commissioned through the Reserve Officers Training Corps program. She was assigned to Mather Air Force Base, California from 1986 to 1989 where she served as the Section Commander for the Civil Engineering Squadron and as the Deputy Program Manager for the Administration Automation Program. From 1989 to 1993, she was assigned to Edwards Air Force Base, California where she was the Executive Officer for the F-15 Combined Test Force, Headquarters Squadron Section Commander for the B-2 Combined Test Force as well as the B-2 Combined Test Force Protocol Officer, and the 412th Test Wing Executive Officer. Capt Cook entered the School of Logistics and Acquisition Management, Air Force Institute of Technology, in May of 1993.

Permanent Address: 315 Truslow Road  
Falmouth, VA 22405

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE ADVANCED EDUCATIONAL METHODS IN THE DEPARTMENT OF DEFENSE: APPLICATION OF CASE THEORY AND COMPUTER-ASSISTED INSTRUCTION TO BUSINESS PROCESS REENGINEERING		5. FUNDING NUMBERS	
6. AUTHOR(S) Brian A. Brown, Captain, USAF Susan M. Brown, Captain, USAF Karen L. Cook, Captain, USAF		8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GIR/LAR/94D-2	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583		10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) DISA CIM Attn: Bob Storm 2100 Washington Boulevard Washington DC 22204		11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION / AVAILABILITY STATEMENT  Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  <p style="text-align: center;">The goal of this research was to create a computer-assisted instruction (CAI) prototype tutorial to teach the fundamental principles of integrated, computer-aided manufacturing definition activity modeling (IDEF0), a component of business process reengineering within the Department of Defense. To accomplish this, the authors evaluated current instructional material related to IDEF0, then adapted the material to a computerized learning environment using the Authorware Professional™ authoring software and a hybrid methodology which incorporated elements both of instructional design and software development. One of the most notable characteristics of this hybrid model was a series of iterations at each phase of development, which eliminated the need for retroactive modifications at successive development phases.</p> <p>The objectives for this study were accomplished by completing a literature review, identifying learning objectives and testing strategies, creating a prototype tutorial using authoring software, incorporating a case study, and then operationally testing the prototype. Analysis of this research is limited to a discussion of the results of a qualitative survey provided to evaluators of the prototype. In general, the results of the survey indicated acceptance of CAI as a method of instruction and provided recommendations for improving the tutorial.</p>			
14. SUBJECT TERMS Education, Instructional Materials, Prototypes Computer-Aided Instruction		15. NUMBER OF PAGES 98	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL